iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** IV **Month of publication:** April 2018

DOI: http://doi.org/10.22214/ijraset.2018.4566

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue IV, April 2018- Available at www.ijraset.com

A Pragmatic Review of Data Cleansing models and using Elastic Search shards for Removing Duplicate data

Subhani Shaik[1], Nallamothu Naga Malleswara Rao[2]
[1]*Research scholar, CSE Department, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India.*
[2]*Professor, Department of IT, RVR & JC College of Engineering Chowdavaram, Guntur, Andhra Pradesh, India*

Abstract: ***The quality of the data is significant issue for the accomplishment of any development. Unrelated and insignificant information leads to an inaccurate analysis, which can be very destructive to the campaigns. Data cleansing plays an essential role when a large amount of data is concerned. Duplication of the information may lead to confusion, accidental deletion of the authentic information, loss of time, etc. Data cleansing is the process of incomplete, incorrect, inaccurate, irrelevant, etc. parts of the data and subject to replacing, modifying, or deleting this outlier data or coarse data from a hefty record set. After data cleansing, the data set will be unswerving for analyzing or any other operations with other similar data sets in the system. The inconsistencies data should be analyzed and detected or removed from the original database. Elastic search is an extremely flexible platform to systematize data along with replication process. This paper will aim to answer queries like how many shards should we have and how large should our shards be. Elastic Search was chosen because it is convenient to set up nodes and clusters. The API makes it easy to use different languages like Java, Ruby, Perl, Python, and more. In runtime, elastic search manages distribution: Adding a node is quite easy and data is redistributed automatically.***

I. INTRODUCTION

Data cleansing is the process of incomplete, incorrect, inaccurate, irrelevant, etc. parts of the data and subject to replacing, modifying, or deleting this outlier data or coarse data from a hefty record set. After data cleansing, the data
set will be unswerving for analyzing or any other operations with other similar data sets in the system. The inconsistencies data should be analyzed and detected or removed from the original database. Elastic Search is a schema less indexing technique. Hence hashing or indexing is similar to graphs in it. Elastic Search is able to achieve fast search responses because, instead of searching the text directly, it searches an index instead. This is like retrieving pages in a book related to a keyword by scanning the index at the back of a book, as opposed to searching every word of every page of the book. This type of index is called an inverted index, because it inverts a page-centric data structure (page->words) to a keyword-centric data structure (word->pages).Elastic Search uses Apache Lucene to create and manage this inverted index.

II. LITERATURE REVIEW

Most of the data cleaning research deals with schema translation and schema integration, data cleaning has received only little attention in the research community, but effective data cleaning methods may improve the effectiveness of data mining methods or the accuracy of any data mining techniques. Many researchers and their research work are focused on the problem of data cleaning and they suggest many methodologies, optimization techniques and variety of new algorithms have been proposed different algorithms to clean outlier data.

An effective data cleaning method is proposed by Wejje Wei, Mingwei Zhang, Bin Zhang Xiaochun Tang [1]. The authors have introduced a new methodology based on association rule mining method. This association rule mining method exploits the business rules provided by the association rules mined from multiple data sources and this methodology creates promising business rules for individual data sources.

YiqunLiu, MinZhang, LiyunRu, Shaoping Ma [2] has proposed a novel learning-based algorithm. This finds very good results with reducing the webpages. This is done by calculating user need. The methodology gives only the most matching websites and other sites are treated as outliers. The results show that how the retrieval target pages can be separated from minimal quality pages using query-independent features embedded in the cleansing algorithms.

Chaudhuri, S., Dayal [3] have proposed a new execution model and novel algorithms. In this approach users can involve and suggest data cleaning requirements and specifications declaratively which helps to perform the cleaning efficiently. This methodology uses a dummy a set of bibliographic references used to construct the Citesser Web Site.

A threshold based data cleaning method is proposed by Timothy Ohanekwu, C.I.Ezeife [4]. This algorithm uses a technique which eliminates the need to rely on match threshold, which is achieved by defining smart tokens.

Chris Mayfield et. al [5] has used a statistical method for integrated data cleaning and imputation. They focus on exploiting the statistical relationships between records in database; this methodology has a new approach to analyze the statistical dependencies between the records.

Kazi Shah Nawaz Ripon et. Al. [6], Proposed novel methodology for identifying and cleansing the identical tuples in a dataset. The algorithm is checked against the computational cost and they proved that this required less computational cost. Also they proposed the enhanced version of significant transitive rule.

Li Zhao, Sung Sam Yuan, Sun Peng and Ling Tok Wang They propose [7] a method based on based on the longest common subsequence. This method called LCSS and explained the possesses with its desired properties. This paper also includes two novel detection and correction methods, SNM-IN and SNM-INOUT, which are the optimized or enhanced version of detection and correction method called SNM.

Aye T.T[8] has explained the data cleaning algorithm eliminates inconsistent or unwanted or dirty items in the preprocessed data.

III. DATA CLEANSING APPROACH

A. *Data Cleaning Phases*

1) *Data analysis:* In this phase, the data is composed from different data sets or databases. Hence, there is a possibility getting of errors or bugs. In order to perceive and eradicate such type of bugs and inconsistencies in the data set a comprehensive data analysis is necessary.
2) *Description of transformation workflow and mapping rules:* This phase depends upon the number of data sources and large number of transformation steps may have to be involved. This is simply based on their degree of heterogeneity and the "dirtiness" of the data
3) *Verification:* The correctness and effectiveness of second phase should be tested and evaluated both manually and algorithmically on a sample or copy of the source data.
4) *Transformation:* Execution of the transformation steps is maintained by either running the ETL workflow for loading and refreshing a data warehouse or when answering queries on datasets or multiple tables
5) *Backflow of Cleaned Data:* After removing all the errors from the dataset, the resultant cleaned data should replace the dirty or unwanted data in the original sources in order to give the enhanced data and to avoid repetition of cleaning work for future data extractions. For data warehousing, the cleaned data is available from the data staging area.

B. *Data Cleansing Services*

Data cleaning services consist of the process of detecting, correcting errors and inconsistencies from a data set in order to improve its quality. Also data cleaning services aim to clean the data and bring uniformity to different data sets that have been merged from different sources. After cleansing, a data set should be consistent with similar data sets within the system.

1) *Import Data:* Dirty data is imported into the cleansing system in an Excel, CSV, or Tab-Separated Text file format.
2) *Merge Data Sets*: Data from several differently formatted is converted and combined into a common database.
3) *Rebuild Missing Data:* Missing information like Postal codes, states, country, area codes, gender and web address from email addresses is recreated.
4) *Standardize Data:* Data is combined, separated or modified to ensure that the same type of data exists in each column.
5) *Normalize Data*: comparable data is normalized (e.g. mister, Mr., mr are all converted to Mr. Or street, st., strt. are all converted to St.).
6) *De-Duplicate data:* By using a custom-built fuzzy-matching algorithm we will identify possible duplicates. This methodology will provide high accuracy matches with a tolerance for misspelling, missing values or different address orders.
7) *Verify & Enrich Data:* Data is validated against internal and external database sources and additional value-adding info is appended.
8) *Export Data*: Data can be exported in numerous formats like excel, csv, SQL database, XML, tiff, PDF, or as required in the same format that we receive.

Fig-1: Data cleansing Cycle.

C. Data Cleansing Tools

Data cleansing tools are the type of software application, which helps in cleaning the database. It accomplishes the action by identifying those parts of the data which are incorrect, incomplete, inaccurate, or irrelevant.

After identification those are replaced, deleted, or modified by the tools. Let's check the tools that are used for cleansing:

1) *Drake:* Drake is a simple, extensible, and text based process which systematizes the execution of commands in the niche of the principal content and its dependencies. Drake processes the data, based on the input and output, which automatically resolve the issues. The essential task of Drake involves the command to implement and the order of accomplishing the command.
2) *Open Refine:* Open Refine was a Google project and was called Google Refine. It is one of the strongest graphical user interfaces, which lets the user perform data manipulation. It can even convert the info from one format to another, by cleansing and transforming the unproductive information.
3) *Trifacta Wrangler:* It is the most interactive tool among the Data cleansing tools. It gathers all the unorganized, scrambled, and inappropriate information and transforms it into the neat data tables. The pattern obtained can be converted into any desirable format. It saves the hassle of manual formatting, and the time saved can be utilized in analyzing the output
4) *Data Cleaner:* The principal action of Data Cleaner tool is data profiling. It helps understand the pattern, analyze the quality of the document, and find the missing values and other negative characteristics of the information
5) *Datamartist:* It is a very user-friendly system, which allows incorporation of a variety of documents and sources in order to boost and renovate the existing database, rather than replacing it
6) *Tabula:* It is a convenient and a very acceptable cleansing tool, which converts the info that is embedded in the PDF into a spreadsheet. The task is accomplished without any human interference and is extremely beneficial for the marketers, data journalists, and scientists, financial analysts, etc
7) *Mo Data:* Most of the people, who use the data scrubbing tool for personal use, prefer this tool the most. It combines, cleanses, and produces the analytics cube from the contrasting CRP and ERP sources. This helps understand the document to be analyzed
8) *Winpure Clean & Match:* It is used to increase the accuracy of the customer data that elevates the quality of the service provided, hence leads to the success of the campaign. It rejects the duplicates, cleanses, and corrects the database, spreadsheet, and mailing lists.
9) *Data Ladder:* It matches the info by data matching, profiling, and removing the duplicates, etc., which supports the business to get the most out of the reliable and effective data
10) *TIBCO Clarity:* This mechanism involves data preparation and is used to profile, discover, standardize, and cleanses the raw data accumulated from various sources, which produces correct info. This leads to the accurate analysis and appropriate decision making.
11) *Star DQ Pro:* It prepares unique, accurate, up-to-date information and plays a great role in cleansing, de-duping, and monitoring the transactions.

IV. DUPLICATE DETECTION

A. Duplicate Detection Tools

In the past decade, various data cleaning tools were sold out in market and they were available as public software packages mainly for duplicate record detection.

1) *Febrl:* The Febrl (Freely Extensible Biomedical Record Linkage) is an open-source data cleaning tool kit. It has two main components one for data standardization and second for duplicate detection**.** Data standardization relies mainly on hidden-Markov models and Supports phonetic encoding namely Soundex, NYSIIS, double metaphone to detect similar names
2) *Tailor:* Tailor is a flexible record matching toolbox. The main feature of this toolbox is that it enables users to apply different duplicate detection methods on the data sets. This tool is termed to be flexible because multiple models are supported.
3) *WHIRL:* WHIRL is an open source duplicate record detection system used for academic and research purposes. Similar strings within two lists identified using a token-based similarity metric.

B. Techniques to Match Individual Fields

The typographical variations of string data is one of the reasons of mismatches in database entries. Hence, duplicate detection typically relies on string comparison techniques to deal with typographical variations. Based on various types of errors, multiple methods have been developed for this task namely:

1) Character-based similarity metrics
2) Token-based similarity metrics
3) Phonetic similarity metrics
4) Numeric similarity metrics

These methods can be used to match individual fields of a record. In most real-life situations, records consist of multiple fields. Thus, duplicate detection problem becomes more complicated. There are two categories used for matching records with multiple fields namely Probabilistic approaches and supervised machine learning techniques Usage of declarative languages for matching and devise distance metrics for duplicate detection task

C. Finding Duplicate Record

Various methods that can be used to compare strings or individual fields and use a metric to understand their similarity. When applied to real world situations where data is multivariate and the number of fields is as dynamic as the data itself, this makes the field of duplicate detection more complicated. These approaches can be classified as

1) *Rule based Techniques* : Rule-based approaches can be considered as distance-based techniques where the distance of two records is either 0 or 1. Rule based approaches require an expert to devise meticulously crafted matching rules typically result in systems with high accuracy. At present the typical approach is to use a system that generates matching rules from training data and then manually tune the automatically generated rules.
2) *Active-Learning Based Techniques :* Problem with the supervised learning techniques is the requirement for a large number of training examples. This method suggested that by creating multiple classifiers trained using slightly different data or parameters it is possible to detect cases and then ask the user for feedback. The key innovation in this work is the creation of several redundant functions and the concurrent exploitation of their conflicting actions in order to discover new kinds of inconsistencies among duplicates in the data set.

D. Probabilistic Matching Models

Let A and B be representation of two tables having n comparable fields. In the case of duplicate detection problem each tuple pair is assigned to one of the two classes M and U. The class M contains the record pairs that represent the same entity (“Match”) and the class U contains the record pairs that represent two different entities (“Non-Match”). Each tuple pair is represented as a random vector x = [1...xn] T with n components that correspond to the n comparable fields of A and B. Let x be the comparison vector. x is the input to a decision rule that assigns x to U or to M. The assumption about x is it is a random vector whose density function will be differing / different for both classes.

1) *Naive Bayes rule Conditional independence*

a) *Assumption*: p (xi/M), p (xj/M) is independent if $i \neq j$.
b) *Goal:* To compute the distributions of p(x/M) and p(x/U).
c) *Naive Bayes rule,* Π Π : Using a training set of pre-labeled record pairs, the values of p (xi/M) and p (xi/U) are computed.

2) *Winkler Methodology:* The conditional independence is not a reasonable assumption so Winkler suggested a method to estimate p (x/M), p (x/U) using expectation maximization algorithm. Winkler suggested five conditions to make unsupervised EM algorithm to work well,

a) The data contain relatively large percentage of matches.
b) The matching pairs are "well-separated" from other classes.
c) The rate of typographical errors is low.
d) There are many redundant identifiers to overcome errors in other fields of the record.
e) The estimates computed under the conditional independence assumption result in good classification performance.

3) *Bayes decision rule (for minimum error)*

a) *Assumption:* x be a comparison vector, randomly taken from the comparison space that corresponds to the record pair <α, β>.
b) *Goal:* To determine whether <α, β> €M or <α, β> €U.
c) *Decision rule*

$$<\alpha, \beta> \in M; \text{ if } p(M/x) \geq p(U/x) \quad U; \text{ otherwise} \tag{1}$$

The above decision rule (1) reveals that if the probability of the match class M, given the comparison vector x, is larger than the probability of the non-match class U, then x is classified to M and vice versa.

Bayes decision rule

$$M, \text{ if } l(x) = p(x/M) \geq p(U) <\alpha, \beta> \in p(x/U)\, p(M) \quad U, \text{ Otherwise} \tag{2}$$

The ratio l(x) = p (x/M) P(x/U) is called as likelihood ratio.
The ratio p(U)/p(M) denotes the threshold value of the likelihood ratio for the decision.
The above statement holds good only when the distributions of p(x/M), p(x/U) and the priors p(U) and p(M) are known.

d) *Naïve bayes rule*
Conditional independence: Assumption: $p(x_i/M)$, $p(x_j/M)$ are independent if $I \neq j$.
e) *Goal:* To compute the distributions of p(x/M) and p(x/U).
f) *Naive bayes rule:*

$$P(\underline{x}|M) = \prod_{i=1}^{n} p(x_i|M)$$

$$P(\underline{x}|U) = \prod_{i=1}^{n} p(x_i|U)$$

Using a training set of pre-labeled record pairs, the values of p (x_i/M) and p (x_i/U) are computed.

4) *Binary Model*

a) The probabilistic model can also be used without using training data
b) A Binary model for the values of x_i was introduced by Jaro such that:
x_i=1,if field i matches x_i = 0, else
c) He suggested to calculate the probabilities p (x_i = 1/M) using an expectation maximization (EM) algorithm and the probabilities p (x_i = 1/U) can be calculated by taking random pairs of records.

E. Unsupervised Learning

One way to avoid manual cataloging of the comparison vectors is to use clustering algorithms and group together similar comparison vectors. The idea behind most unsupervised learning approaches for duplicate detection is that similar comparison vectors correspond to the same class. The idea of unsupervised learning for duplicate detection has its roots in the probabilistic model.

The basic idea also known as training is to use very few labeled data and then use unsupervised learning techniques to appropriately label the data with unknown labels. Each entry of the comparison vector (which corresponds to the result of a field comparison) Then they partition the comparison space into clusters by using the Auto class clustering tool. The basic premise is that each cluster contains comparison vectors with similar characteristics. Therefore, all the record pairs in the cluster belong to the same class (matches, no matches or possible matches).

F. Supervised and Semi-supervised Learning

The supervised learning systems rely on the existence of training data in the form of records pairs, pre labeled as matching or not. One set of supervised learning techniques treat each record pair (a,b) independently similar to the probabilistic techniques. A well-known CART algorithm generates classification and regression trees. A linear discriminate algorithm generates a linear combination of the parameters for separating the data according to their classes and a "vector quantization" approach which is generalization of the nearest neighbour algorithms. The transitively assumption can sometimes result in inconsistent decisions. For example (a,b) and (a,c) can be considered matches but (b,c) not Partitioning such as "inconsistent" graphs with the goal of minimizing inconsistencies in an NP-complete problem.

G. Bigram Indexing

The Bigram Indexing (BI) method as implemented in the Febrl record linkage system allows for fuzzy blocking. The basic idea is that the blocking key values are converted into a list of Bigram (sub-strings containing two characters) and sub-lists of all possible permutations will rebuilt using a threshold (between 0.0 and 1.0). The resulting Bigram lists are sorted and inserted into an inverted index, which will be used to retrieve the corresponding record numbers in a block. The number of sub-lists created for a blocking key value both depends on the length of the value and the threshold. The lower the threshold the shorter the sub-lists but also the more sub-lists there will be per blocking key value resulting in more (smaller blocks) in the inverted index.

H. Distance-Based Techniques

Active learning techniques require some training data or some human effort to create the matching models. In the absence of such training data or the ability to get human input supervised and active learning techniques are not appropriate. One way of avoiding the need for training data is to define a distance metric for records which does not need tuning through training data.

V. SHARDS IN ELASTIC SEARCH

If we have a large number of documents a single node may not be enough because of RAM limitations, hard disk capacity, insufficient processing power, and inability to respond to client requests fast enough. In that case, data can be divided into smaller parts called shards. A shard is a separate Apache Lucene index. Each shard can be placed on a different server so that the data can be spread among the cluster nodes. When you query an index that is built from multiple shards, Elastic search sends the query to each relevant shard and merges the result in such a way that your application doesn't know about the shards. Having multiple shards can speed up the indexing.

Elastic Search spread data to several physical Lucene indices, to store information volumes that exceed abilities of a single server. Those Lucene indices are called shards and the process of this spreading is called sharding. Elastic Search can do this automatically and all parts of the index (shards) are visible to the user as one-big index. It is vital to tune this automation for a specific use case because the number of shard index is built or configured during index creation and cannot be changed later. Thus if we have an index with 100 documents and a cluster with 2 nodes, each node will hold 50 documents if the shard_number is 2.

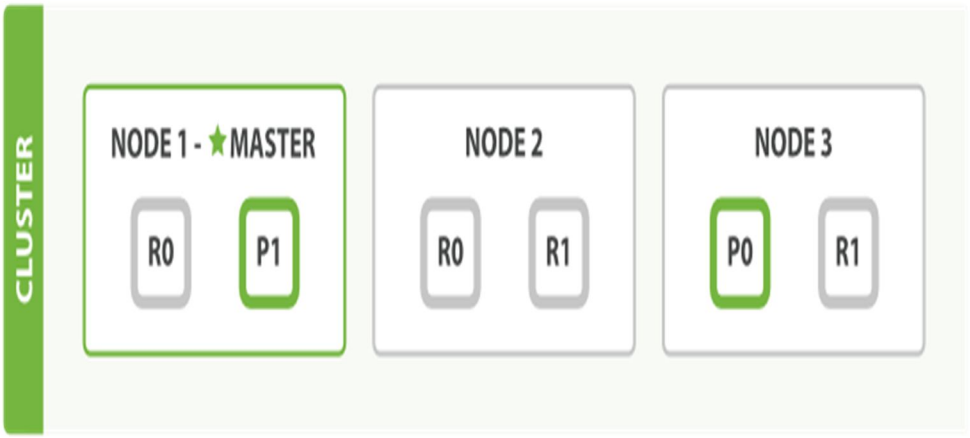


Fig-2: Elastic search Cluster

When data is written to a shard, it is published into new immutable Lucene segments on disk, and it becomes available for querying. This is referred to as a refresh.

As the number of segments increases, these are periodically consolidated into larger segments. This process is referred to as merging.

A node is an instance of Elasticsearch. When you start Elasticsearch on your server, you have a node. If you start Elasticsearch on another server, it's another node. You can even have more nodes on the same server by starting multiple Elasticsearch processes. Multiple nodes can join the same cluster. With a cluster of multiple nodes, the same data can be spread across multiple servers. This helps performance because Elasticsearch has more resources to work with. It also helps reliability: if you have at least one replica per shard, any node can disappear and Elasticsearch will still serve you all the data.

A. Terms Related to Shards

1) *Cluster:* A cluster consists of one or more nodes which share the same cluster name. Each cluster has a single master node which is chosen automatically by the cluster and can be replaced if the current master node fails.
2) *Document:* A document is a JSON document which is stored in Elasticsearch. Each document is stored in an index and has a type and an id. A document is a JSON object which contains zero or more fields, or key-value pairs.
3) *Index:* An index is a logical namespace which maps to one or more primary shards and can have zero or more replica shards
4) *Node:* A node is a running instance of Elasticsearch which belongs to a cluster. Multiple nodes can be started on a single server for testing purposes, but usually you should have one node per serve
5) *Shard:* A shard is a single Lucene instance and a low-level unit which is managed automatically by Elasticsearch. An index is a logical namespace which points to primary and replica shards. Elasticsearch distributes shards amongst all nodes in the cluster, and can move shards automatically from one node to another in the case of node failure, or the addition of new nodes
6) *Replica:* In order to increase query throughput or achieve high availability, shard replicas can be used. A replica is just an exact copy of the shard, and each shard can have zero or more replicas
7) *Primary Shard:* Each document is stored in a single primary shard. When you index a document, it is indexed first on the primary shard, then on all replicas of the primary shard. By default, an index has 5 primary shards. You can specify fewer or more primary shards to scale the number of documents that your index can handle. one cannot change the number of primary shards in an index, once the index is created.
8) *Replica Shard:* Each primary shard can have zero or more replicas. A replica is a copy of the primary shard, and it is used to increase failover and to increase performance

VI. SEARCHING AN INDEX:\

For searching an index, Elasticsearch has to look in a comprehensive set of shards for that index which are either primary or replicas because primary and replica shards characteristically contain the same documents. Elasticsearch distributes the search load between the primary and replica shards of the index you're searching, making replicas useful for both search performance and fault tolerance

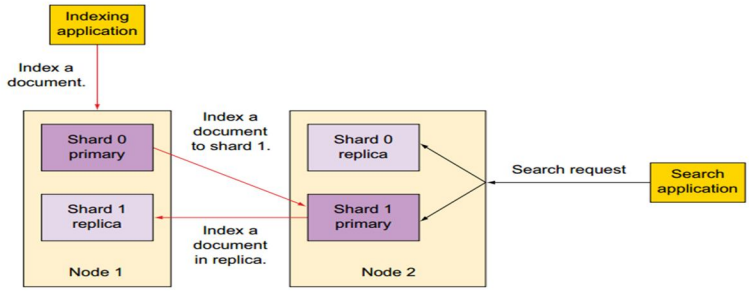


Fig-3: Indexing process using shards

Consider an index with two primary shards. Increase the capacity of the index by adding a second node. Adding more nodes would not add indexing capacity, but we could take advantage of the extra hardware at search time by increasing the number of replicas:

```
PUT /my_index/_settings
{
    "number_of_replicas":1
```

```
}
```

Having two primary shards, plus a replica of each primary, would give us a total of four shards: one for each node, as shown in Fig-4, “An index with two primary shards and one replica can scale out across four nodes”.

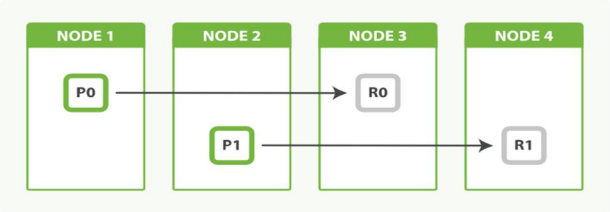


Fig-4: An index with two primary shards and one replica can scale out across four nodes

A. Balancing Load with Replicas

Search performance depends on the response times of the slowest node. So it is better to try to balance out the load across all nodes. If we added just one extra node instead of two, we would end up with two nodes having one shard each, and one node doing double the work with two shards.

By allocating two replicas instead of one, we end up with a total of six shards, which can be evenly divided between three nodes, as shown in Figure, “Adjust the number of replicas to balance the load between nodes”:

```
PUT /my_index/_settings
{
   "number_of_replicas":2
}
```

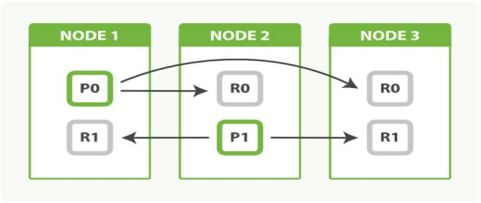


Fig-5: Adjust the number of replicas to balance the load between nodes

The fact is that node 3 holds two replicas and no primaries are not important. Replicas and primaries do the same amount of work; they just play slightly different roles. There is no need to ensure that primaries are distributed evenly across all nodes.

VII. CONCLUSION

Selecting the best data cleansing tools helps in tackling the issues and eradicates the undeserving information, which leads to increased productivity and success rate. In this Paper we have offered a wide-ranging review of the existing techniques used for detecting non identical duplicate entries in database records. As database systems are becoming more and more commonplace data cleaning is going to be the cornerstone for correcting errors in systems which are accumulating vast amounts of errors on a daily basis. Research in databases emphasizes relatively simple and fast duplicate detection techniques that can be applied to databases with millions of records. Such techniques typically do not rely on the existence of training data and emphasize efficiency over effectiveness. Most of the duplicate detection systems available today offer various algorithmic approaches for speeding up the duplicate detection process. Duplicate record detection techniques are crucial for improving the quality of the extracted data. Right now the situation with shard placement is not bad although it is not ideal either.

REFERENCES

[1] A Data Cleaning Method Based on Association Rules. Weijie Wei.Mingwei Zhang, Bin Zhang, Xiaochun Tang College of Information Science and Engineering, Northeastern University, Shenyang 110004, P. R. China. 2Department of telecommunications, NEUSoft Group Ltd, Shenyang 110004, P. R. China.

[2] Yiqun Liu, Min Zhang, Liyun Ru, Shaoping Ma. Data Cleansing for Web Information Retrieval using Query Independent Features. Journal of the American Society for Information Science and Technology (JASIST), Volume 58, Issue 12, Pages 1884-1898, 2007

[3] Umeshwar Dayal, Surajit Chaudhuri, “An overview of data warehousing and OLAP technology” ACM Sigmod Record 26(1), 65-74

[4] A Token-Based Data Cleaning Technique for DataWarehouseSystems. Timothy E.OhanekwuSchool of Computer Science,University of Windsor. C.I. Ezeife. School of Computer Science,. University of Windsor,. Windsor, Ontario,. Canada N9B 3P4, Mar 15, 2002

[5] A Statistical Method for Integrated Data Cleaning and Imputation. Chris Mayfield Jennifer Neville, Purdue University, Sunil Prabhakar, Purdue University, 2009. Report Number. 09-008

[6] Kazi Shah Nawaz Ripon, Ashiqur Rahman and G.M. AtiqurRahaman"A Domain-Independent Data Cleaning Algorithm for Detecting SimilarDuplicates",Journal Of Computers, VOL. 5, NO. 12, December 2010 . P.P 1800-1809

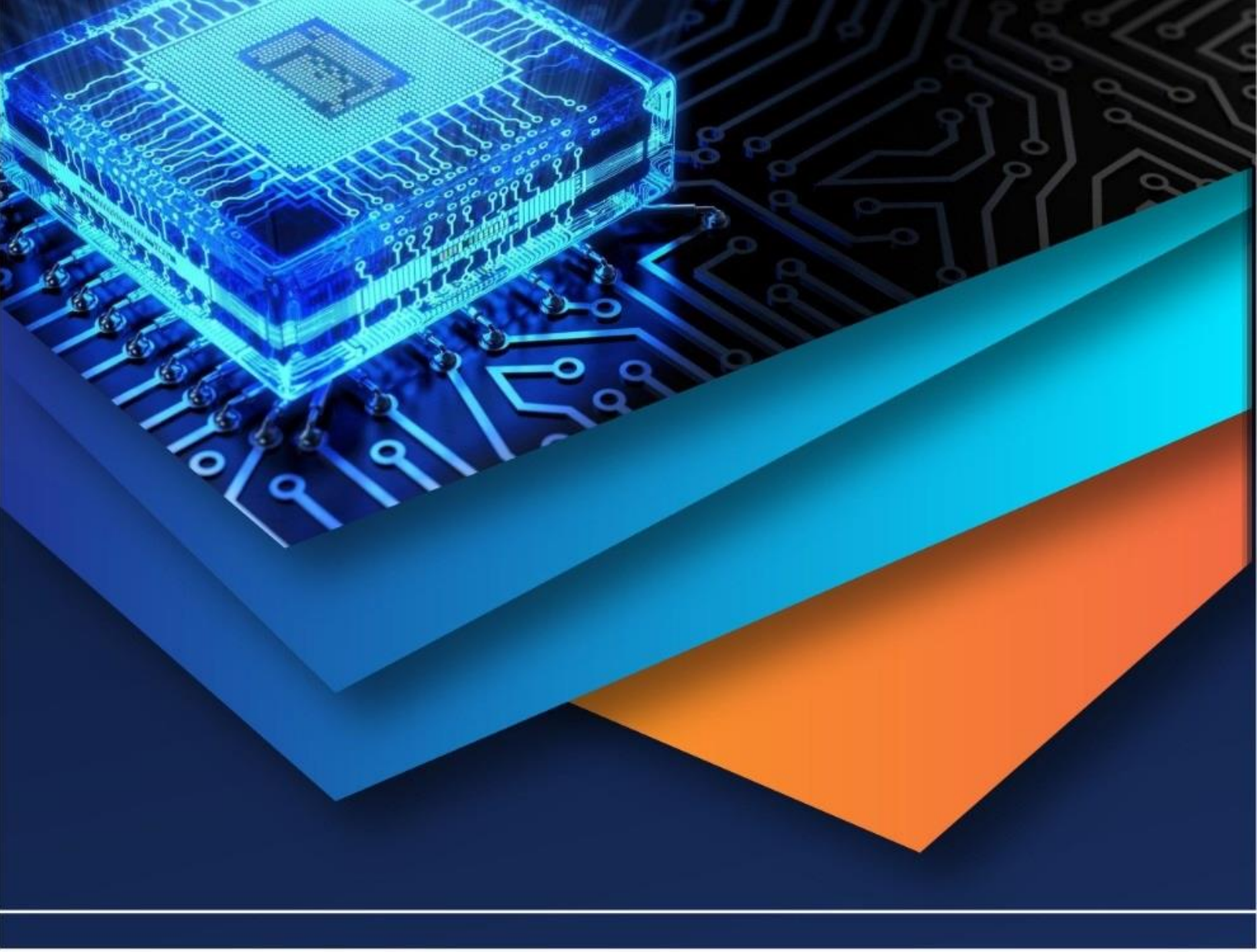
[7] Li Zhao, Sung Sam Yang, Sum Peng and Ling Tock Wang " A New Efficent Data Clencing Method" Springer - DEXA 2002 P.P 484 -804.

[8] T.T. "Web log cleaning for mining of web usage patterns" Computer Research and Development (ICCRD), 2011 3rd International Conference(IEEE Expore) Vol-2 P.P 490 – 494.

BIOGRAPHIES

Subhani Shaik received his B.Tech in computer science and Information Technology through JNTU, Hyderabad, India. He had his Master of Technology from JNTU, Kakinada, A.P, India. He has 12 years of experience in teaching. He is presently working as an Assistant Professor in St.Mary's Group of Institutions Guntur, Chebrolu. His research interest includes Data Mining and Big Data Management. Now he is pursuing his Ph.D in Computer Science in Acharya Nagarjuna University, Guntur, A.P, India.

Dr. Nallamothu Naga Malleswara Rao is working as Professor in the Department of Information Technology at RVR & JC College of Engineering with 26 years of Teaching Experience in the academics. His research interest includes Computer Algorithms, Compilers, and Image Processing.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY