ISSN:0975-9646

Speeding up of Search Engine by Detection and Control of Duplicate Documents on the Web

Rekha V R[1], Resmy V R[2]

[1]*Assistant Professor in IT, Department of IT,*
College of Engineering, Kidangoor, Kottayam, India.

[2]*Associate Professor, Dept. of Computer Science,*
Sarabhai Institute of Science and Technology, Vellanad, Thiruvananthapuram, India

Abstract— **World Wide Web has become the most populated database with increased number of users every day. This makes the search engines to produce duplicate data which has to be solved by de-duplication process. Various methods have been formulated in recent days to solve the issue but every method has one or other demerit that prevents it to be adapted successfully. Hence, in this paper, the patterns of the URLs are utilized to develop a framework for de-duplicating the web pages. The machine learning technique is used to study the pattern and precise rules are generalized. This helps in increasing the coverage. The pairwise rules are generated from URL pairs present in the duplicate clusters. When the web crawlers apply these rules, it normalizes the URLs. The normalized URLs are tokenized and pattern tree is constructed. This is performed over the selected clusters and thus the transformational rule proves efficient in avoiding redundancies in the search results. The feasibility of the proposed methodology is studied with an experimental setup with two different datasets. The results shows that the de-duplication is achieved with good efficiency. The comparative analysis is also made with the existing methodologies.**

Index Terms – **Pattern matching tree – URL – Tokenization, Data de-duplication – Machine Learning – Clustering**.

I. INTRODUCTION

A lot of data has been dumped every day into the World Wide Web (WWW) which makes fetching a desired detail through this web has become a tiresome activity. To recover from this, data mining provides a solution, through mining process which analyzes data and summarizes to useful non- redundant data. This process of data mining helps technologists and also the technology users to reduce costs and increase their profit. Moreover, data mining proves to be one of the best analytical tool to analyze, categorize and summarize data.

In real-time applications, major industries are using the data mining technology to their favor. The concept of data mining is used to correlate the internal and external factors. This gives clues to find out the sales impact, customer mindset, economic indicators etc. For instance, Blockbuster entertainment uses data mining to suggest products to their customers based on their video rental history in its database. Also Walmart uses the point-of-sale transactions data and stores them in its data warehouse. This enables the Walmart to identify the suitable merchandising opportunity and has transformed its supplier relationship.

The data mining can be performed with different level of analysis, namely,

- Artificial Neural Network(ANN)
- Genetic Algorithms(GA)
- Decision Trees
- Nearest neighbor method
- Rule induction
- Data Visualization

The technological infrastructure required for the data mining applications are ranging from a small space PC platform to the mainframes. The main parameters that should be considered are

- The size of the database and
- The query complexity.

Search engines are huge such databases with web page files generated from the existing external sources and are automatically assembled. The search engines can be classified as

- Individual search engine and
- Meta searchers

Here the individual search engine possesses its own database where information were compiled to be accessed by the users. While Meta searchers doesn't have their own databases but uses different individual search engines and displays the best results.

These Meta searchers use either one of the two ways to show their results.

(i) After removing the duplicated results from the results collected from various individual search engines, a merged single list of data is presented.

(ii) Without removing the duplicate data multiple lists are produced.

So this duplication is the undesired property which can be eliminated by the process of de-duplication. This helps to decrease the storage requirement as only distinct data is stored.

The redundant data is avoided by replacing a pointer to the distinct data copy. The disadvantage of duplication is that it affects crawling and relevance of the results. Hence rule induced mining is performed over the URL's in this proposal.

Since all the search engines strives to be the popular one based on the closest results that tends to provide. Also the

other main parameter to be considered is data duplication. The demerits if the data duplication are

- High storage space occupied
- Less efficient use of disk space
- Low Recovery Time Objectives (RTO)
- Need of tape backups
- Requires more transmissions

The existing system that is chosen to compare our result is Detecting Near-Duplicates for Web crawling. It uses simhash to address the large query.It also develops the hamming distance problem for both single and multi-queries online. However the existing system has the disadvantages like

- De-duplication performed after a web page downloads, hence
- High bandwidth usage during crawling.
- Costs high
- Limit of accuracy

Our focus in this paper is on efficient and large-scale de-duplication of documents on the WWW. Web pages which have the same content but are referenced by different URLs, are known to cause a host of problems. Crawler resources are wasted in fetching duplicate pages, indexing requires larger storage and relevance of results are diluted for a query.

The problem statement of the proposed work is for a given set of duplicate clusters and their corresponding URLs

- Learning Rules from URL strings which can identify duplicates
- Utilizing learned Rules for normalizing unseen duplicate URLs into a unique normalized URL

Applications such as crawlers can apply these generalized Rules on a given URL to generate a normalized URL. The proposed work involves mining of the crawling logs and the transformational rule are extracted from the clusters of similar pages. The cluster URLs are then normalized. Instead of using every mined rule, the machine learning technique is introduced to de-duplicate web resource with ease.

The remaining sections of this paper is arranged as follows. The earlier works associated to the URL based de-duplication are discussed and the methodologies used to overcome the problems are presented in Section II. The descriptive details of the proposed rule extraction technique is given in the Section III. The experimental results based on the simulation and their comparative output results are provided in Section IV. Lastly, Section V summarizes and gives the conclusion to the work.

II. RELATED WORK

Poria et al [1] proposed a rule-based mining that exploited the knowledge and sentence dependency trees. It solved the issue in extracting aspect information from the reviews of the products. Though the detection accuracy was high it did not provide prediction of user classes and had not studied the result for different sample sizes. *Puzio et al* [2] presented ClouDedup for secure and efficient storage of data duplication. This involved high memory usage and high cost for computation. A bipartite graph was *Steorts et al* [3] recorded to detect the duplicate record in a file without any supervision. Its dimensionality was found to be high. *He et al* [4] gave techniques to overcome issues of deep-web sites which were entity oriented. It helped in effective URL de-duplication. *Rodrigues et al*[5] suggested DUSTER, for detecting and eliminating the redundant content of the web during crawling. The duplicate URLs are reduced 54% more than that of earlier techniques. *Dutta* [6] used crawl log mining and found the topics that are searched by the users frequently. It provided automated recommendations for the other users. *Shivaprasad et al* [7] performed Web Usage Mining(WUM) to identify the pattern of data search. It also reduced the noise and unnecessary data in those logs. *Sethi et al* [8] proposed a new algorithm for extracting data. Also the rule extraction was also tested for its feasibility. *Jain and Srivastava* [9]conducted a study on the effect rule extraction with ANN and GA and found to be robust and self-adaptive.

Cai et al [10] presented learning approach which was based on pattern tree. The work was acknowledged for its ability to implement rule based mapping and was employed by the processor. A set of URL samples from the targeted website was retrieved by the web crawler of the search engine. *Saha Roy et al* [11] formulated the issue of probabilistic de-duplication as a binary classification task for unknown visitors. *Zawoad et al*[12] Since the number of potential malicious URLs from diverse sources is large, URL de-duplication is needed for the efficient identification of malicious websites. URL De-duplication as-a-Service (UDaaS) was developed to help a URL analyst to deploy and manage a cloud-based distributed and parallel URL de-duplication infrastructure easily; this can improve the performance of malicious websites detection while reducing duplication and quantity of local storage requirements. *Wu et al* [13] presented HACE theorem to process the big data in terms of data mining. Creating a global model based on this theorem is a challenging task, however it served good for understanding the working of data mining. *Lin et al* [14] presented three algorithms namely SPC, DPC and FPC. These algorithms helped to understand the implementation result of apriori algorithm using Map reduce framework. The parallelization technique proved good for all sizes of datasets and for all cluster sizes. *Kolb et al* [15] proposed Dedoop, which was a powerful, high performance tool for de duplication using Hadoop. It was demonstrated for larger datasets. The proposed methodology was based on browser specifications that used machine learning to generate match classifiers. *Das Sarma* [16] presented a system called CBLOCK, to address the de-duplication challenges. The proposed CBLOCK framework is designed in such a way to learn hash functions derived from attribute domains. The blocking functions are represented as hierarchical tree structure constraints. The method was tested with two large datasets and the utility was proved. *Shim* [17] analyzed the efficiency of Map Reduce algorithms for web mining , machine learning. *Jangra and Singh* [18] addressed the challenges of redundant data using No-SQL. The authors proposed an effective storage. *Li et al* [19] suggested a baseline key management approach to de duplicate data

with security. The proposal also introduced Dekey construct, which enabled the storage of secure keys and managed to achieve good efficiency with security for cloud storage server. *Rafeeq and Kumar* [20] proposed the Open Stack Swift scheme for secure de-duplication from client-side. The presented scheme provided confidentiality, secured storage, outsourced data and shared them through cloud. The major demerit of the proposed work was that there was increase in un-structured files and name space gets populated. *Santhiya and Bhuvaneswari* [21] addressed the de-duplication issue using Uniform Resource locator. The web crawler used the ontology by which the domain specific data were extracted and clustered. SVM classifiers were used to exploit the keywords and URLs.

III. PROPOSED METHOD

The presented work provides few techniques to prevent data de-duplication in World Wide Web (WWW). The approach mine rules from the URLs. These rules are utilized for data de- duplication. The host specific tokens and delimiters are extracted from the URLs. Pairwise rule generation is performed and hence the source URL and target URL is selected. The rules are fine-tuned using generalization technique through machine learning.

Target Selection: The target URL is selected from the duplicate clusters of the URLs. This is done by selecting the shortest matching URL in the cluster. Another criteria for target selection is the length of the URL, which must be shorter. The other parameters that could be considered are the number of inlinks, minimum hop distance etc.

Source Selection: The statistical based ranking helps to decide the URL to be chosen as a source. The online page ranker is utilized for this task.

The following fig 1 illustrates the system architecture of the proposed system.

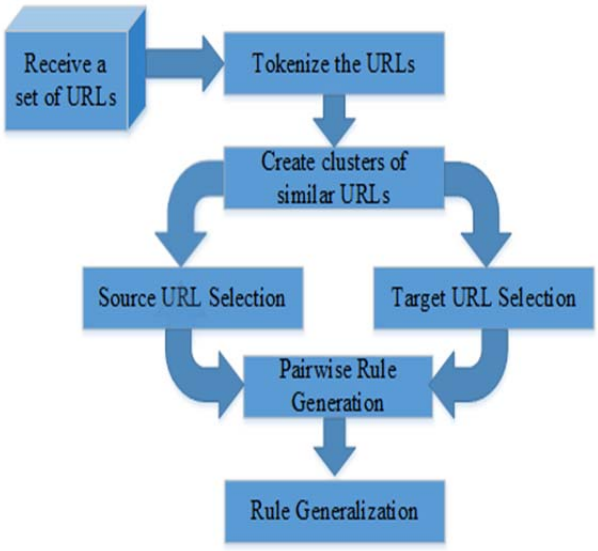

Fig 1. System Architecture of the proposed system

When the rules are precise, the results produced are accurate and de-duplicated. The basic modules of the proposed system are

1. URL Dataset Visualization
2. Tokenization
3. Clustering
4. Pair wise rule generation
5. Rule generalization
6. Comparison

Fig 2 presents the modules of the proposed work.

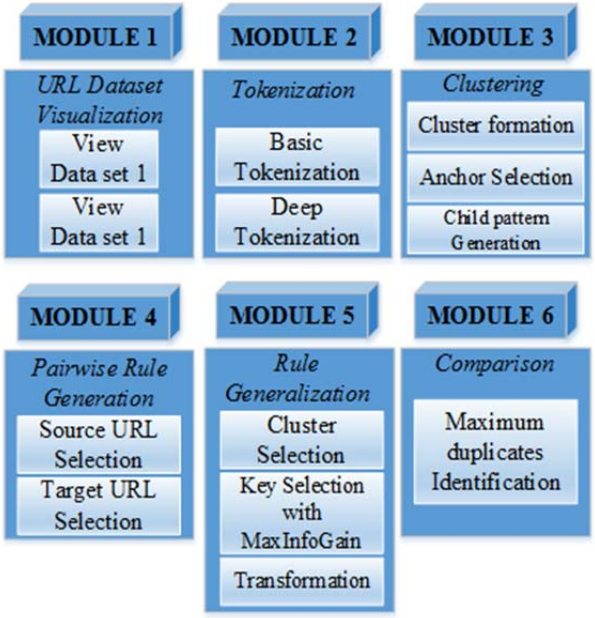

Fig 2.Modules of the proposed system

1. URL Dataset Visualization:

The work included two sets of data to achieve data de-duplication. The feasibility of data sets, both small and large data sets are used for experimentation are known. These datasets contains either the URLs of many websites or the URLs of many web pages. The pre-requisite to select these small and big data set is that it should contain at least 2 sized duplicate clusters in both data sets. Also it is characterized by the number of hosts, URLs and the duplicate clusters present in them. The collection Mixer is constructed with the data sets containing web pages and clusters. Based on the human judgment, the core content is collected.

The stop words are also chosen in order to get proper understanding and good performance for de-duplication.

2. Tokenization:

Basic tokenization is the process of parsing URL to extract tokens. The protocols, hostnames, query arguments and the path components are also extracted from the specified standard delimiters. Firstly, clusters are formed with the URLs in the datasets. Then, anchors are selected from the URL clusters formed in the previous step. The selected anchors are validated and if the anchors are found to be valid, then the child pattern is generated. If the anchors are not valid, they do not generate child pattern.

Then, the process of generating tokenized key value pairs and associates them to the original URL in order to generate deep tokenized URLs is known as Deep tokenization. The URL encodings are learnt by a specialized technique that doesn't require any supervision. This process is iterative defined and conducted as per the decision tree generated.

3. Clustering

The process of cluster formation with the URLs are known as Clustering. It is the basic step of module 3 in which the cluster is formed and is produced to the rule generalization module. The URLs which consists of more similarity in the web page content is termed as a duplicate cluster. The rules are generated for all the URL pair present in the duplicate clusters.

4. Pairwise Rule Generation:

Pairwise rule generation module is designed for generating pairwise rules from the URL pairs of the duplicate clusters. The transformational rules are framed in this module. This is the critical part of the work which decides the efficient working of de-duplication process. Here target URLs are used for generating transformation. The clusters have few URLs which are closest to the normal URL.

Out of these URLs, one is selected as source URL. The source URL is changed frequently based on the study. Now, using these source and target URLs the pairwise rule is generated. The learning of these pairwise rules generated through URLs out of duplicate cluster happens which are further generalized so as to normalize the unseen URLs as well.

5. Rule Generalization:

In the process of rule generalization, one of the cluster is selected from the cluster groups. A key is selected from the previously selected cluster. We know that all keys have information gains, so the key selection is made by studying the maximum information gains possessed by the key. Finally the transformation process of transmitting the source to the target is performed. Generalization is performed by generating a decision tree. This tree is constructed with the selected keys and their branch is formed with the key's matching pair else it is branched out with a wildcard.

Number of linear rules generated by the generalization technique. Only after rule generalization, the new values can be accommodated. The decision tree based generalization enabled the work to be error proof and robust. The so generated decision tree follows bottom-Up approach. The rules are used in online mode and hence the memory requirement to store these rules are minimal. The contexts as well as transformation format are generated by the rule generalization process. Thus it provides a compatible contexts as well as target URLs. The feasibility is improved as the iteration counts high. During the initial phase, the frequency is generated for each key. Then context generalization is performed. The generalization is performed over the contexts.

6. Comparison:

This is the ultimate step to present the non-redundant, de-duplicated data for the users. With two data sets in hand, the number of de-duplicated data are estimated. The comparison module is presented, where the maximum number of duplicates are detected. Thus the matches are identified and are produced for display without any duplicated data. There is a steady improvement as the proposed pairwise rule generalization is an iterative algorithm. The proposed work cross verifies both the data sets so as to provide the optimal results. Through comparison of both the data sets the data set with maximum number of de-duplicated data are known.

IV. PERFORMANCE ANALYSIS

The experimentation is performed over the presented algorithm with Windows XP Operating System, in Java language and used 6.9.1 version of Netbeans Integrated Development Environment. The algorithm is implemented using Pentium IV computer system. In order to verify the proposed algorithms, the data sets are segregated based on its content. The segregated data are clustered by using number of keyword, selected based its feasibility to satisfy the generalized rule. The adopted rule is fine tuned in every iteration, which makes the proposed technique applicable for various spheres.

The performance analysis proved that the proposed work is advantageous than the earlier works [22] because

- It is efficient in terms of indexing,
- It reduces the unnecessary usage of crawler resources
- It retrieves effective URLs which are more related to the requirement.

The research work presented here is tested for its feasibility based on the following metrics with the existing methodologies [23] SizeSpotSigs, AF_SpotSigs and SpotSigs.

- Performance
- Effectiveness
- Time Analysis
- Mixer Purity

The metrics are discussed with the graphs to support the advantage of the proposed pairwise rule generalization technique with respect to the existing methods.

1. Performance

Performance is the measure of achievement of the particular task verified against the previous attainments. The performance of the proposed method is compared with the performance of the existing methods. The analysis result is shown in Fig 3. The performance of the proposed pairwise rule generalization technique is found superior to that of the earlier methods.

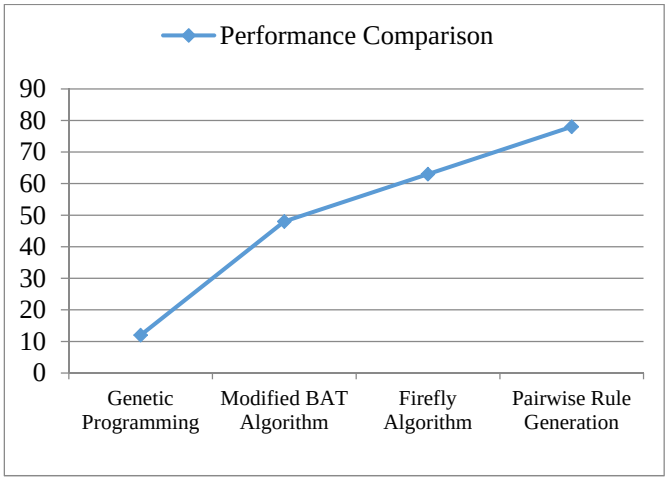

Fig 3. Comparison of performance

2. Effectiveness

The degree of compatibility of a system to the targeted issue, checked against various dataset is called as the effectiveness of the system. The effective detection of duplication leads to easier de-duplication process thereby reduces the noise content in the result. The proposed work is considered to be on par with the other methodologies in its effectiveness measure. The experimental proof is shown in the following fig. 4 in which the effectiveness of all the methods are comparatively studied.

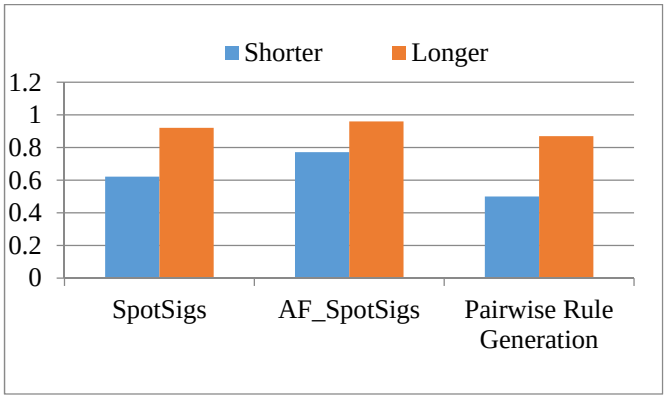

Fig 4. Analysis of effectiveness

3. Time Analysis

Any implementation is considered to be efficient, when the time delay is less. The proposed pairwise rule generation is taken experimentally to know the time taken to perform de-duplication of the webpage. The trial results are produced in the Fig 5. The figure illustrates the analysis of the time taken by the system. From the analysis, it is found that the least time is taken by the proposed work and the other earlier works had taken greater time to perform de-duplication.

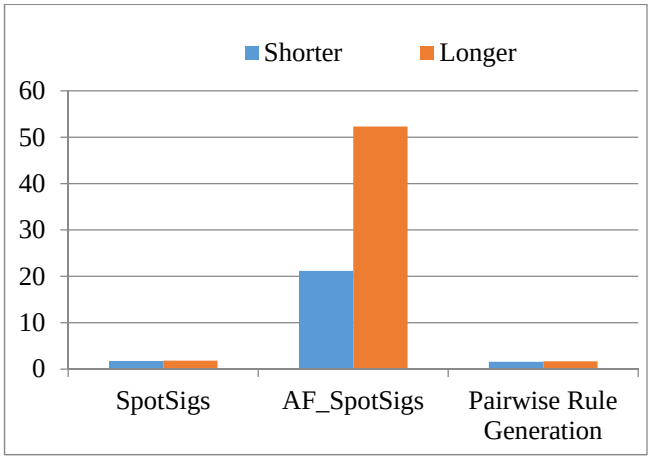

Fig 5. Time Analysis

4. Mixer Purity

Fig 6 shows the pictorial analysis of the mixer purity of the proposed pairwise rule generation with that of SizeSpotSigs, AF_SpotSigs and SpotSigs based on the cluster partition point. The result presents the lower value duplication present in the proposed work.

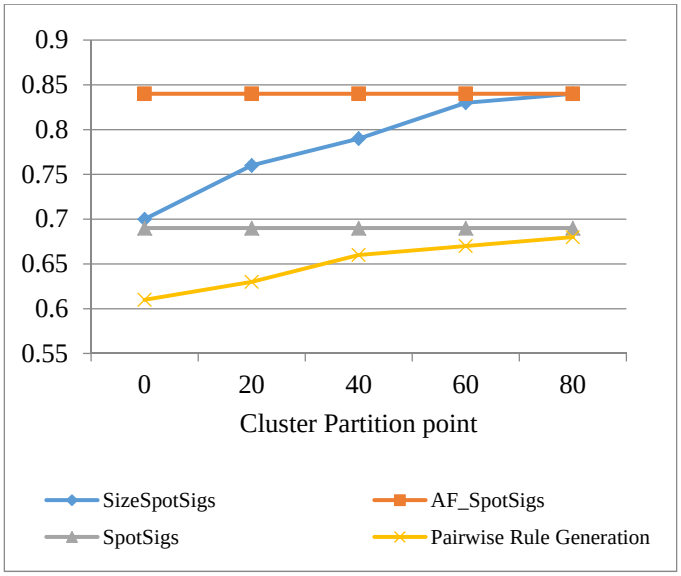

Fig 6. Analysis of Mixer Purity

V. Conclusion And Future Work

The main blocks of the web search through a search engine is crawling, relevance and indexing. These are affected due to large amount of duplication present in the web. This pose a serious issue for internet search engine users. Hence de-duplication becomes the need of the hour as there is lots of redundant data produced even by the autonomous users. Here a novel idea of de-duplication is performed by framing generalized rules for machine learning. The methodology is unique in the process of extracting unique URLs. Two data sets are used and their duplicated data are analyzed. Then, tokenization is performed over the datasets and tokenized URLs are clustered and pairwise rules are generated and the generated rules are generalized using decision tree. This ensures the precision of rules. Finally, the URLs are transformed and compared to give de-duplicated results. The performance of the presented approach is evaluated and found to be bandwidth efficient and precise. The methodology can be adapted for various other larger data sets.

References

[1] S. Poria, E. Cambria, L.-W. Ku, C. Gui, and A. Gelbukh, "A rule-based approach to aspect extraction from product reviews," in *Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP)*, 2014, pp. 28-37.

[2] P. Puzio, R. Molva, M. Önen, and S. Loureiro, "Block-level de-duplication with encrypted data," *Open Journal of Cloud Computing (OJCC)*, vol. 1, pp. 10-18, 2014.

[3] R. C. Steorts, R. Hall, and S. E. Fienberg, "SMERED: A Bayesian Approach to Graphical Record Linkage and De-duplication."

[4] Y. He, D. Xin, V. Ganti, S. Rajaraman, and N. Shah, "Crawling deep web entity pages," in *Proceedings of the sixth ACM international conference on Web search and data mining*, 2013, pp. 355-364.

[5] K. W. L. Rodrigues, M. Cristo, E. S. de Moura, and A. S. da Silva, "Learning url normalization rules using multiple alignment of sequences," in *String Processing and Information Retrieval*, 2013, pp. 197-205.

[6] A. K. Dutta, "AUTOMATED RECOMMENDATION OF INFORMATION TO THE MEDIA BY THE IMPLEMENTATION OF WEB SEARCHING TECHNIQUE," *International Journal of Computer Science and Information Security*, vol. 13, p. 56, 2015.

[7] G. Shivaprasad, N. S. Reddy, and U. D. Acharya, "Knowledge Discovery from Web Usage Data: An Efficient Implementation of Web Log Preprocessing Techniques," *International Journal of Computer Applications*, vol. 111, 2015.
[8] K. K. Sethi, D. K. Mishra, and B. Mishra, "Novel algorithm to measure consistency between extracted models from big dataset and predicting applicability of rule extraction," in *Conference on IT in Business, Industry and Government (CSIBIG), 2014* 2014, pp. 1-8.
[9] N. Jain and V. Srivastava, "Data Mining techniques: A survey paper," *IJRET: International Journal of Research in Engineering and Technology*, vol. 2, pp. 2319-1163, 2013.
[10] R. Cai, L. Zhang, J.-M. Yang, Y. Ke, X. Fan, and W.-Y. Ma, "Pattern tree-based rule learning," ed: Google Patents, 2013.
[11] R. Saha Roy, R. Sinha, N. Chhaya, and S. Saini, "Probabilistic deduplication of anonymous web traffic," in *Proceedings of the 24th International Conference on World Wide Web Companion*, 2015, pp. 103-104.
[12] S. Zawoad, R. Hasan, G. Warner, and A. Skjellum, "UDaaS: A Cloud-based URL-Deduplication-as-a-Service for Big Datasets," in *IEEE International Conference on Big Data and Cloud Computing (BdCloud)2014*, 2014, pp. 271-272.
[13] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, pp. 97-107, 2014.
[14] M.-Y. Lin, P.-Y. Lee, and S.-C. Hsueh, "Apriori-based frequent itemset mining algorithms on MapReduce," in *Proceedings of the 6th international conference on ubiquitous information management and communication*, 2012, p. 76.
[15] L. Kolb, A. Thor, and E. Rahm, "Dedoop: efficient deduplication with Hadoop," *Proceedings of the VLDB Endowment*, vol. 5, pp. 1878-1881, 2012.
[16] A. Das Sarma, A. Jain, A. Machanavajjhala, and P. Bohannon, "An automatic blocking mechanism for large-scale de-duplication tasks," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 1055-1064.
[17] K. Shim, "MapReduce algorithms for big data analysis," *Proceedings of the VLDB Endowment*, vol. 5, pp. 2016-2017, 2012.
[18] A. Jangra, V. Bhatia, U. Lakhinaza, and N. Singh, "An efficient storage framework design for cloud computing: Deploying compression on de-duplicated No-SQL DB using HDFS," in *1st International Conference on Next Generation Computing Technologies (NGCT), 2015* 2015, pp. 55-60.
[19] J. Li, X. Chen, M. Li, J. Li, P. P. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," *IEEE Transactions on Parallel and Distributed Systems. ,* vol. 25, pp. 1615-1625, 2014.
[20] M. Rafeeq and C. S. Kumar, "Reliable Secure Data Storage in the Cloud Environments and De duplication," *International Journal of Computer Science and Engineering*, vol. 3, pp. 1086-1091, 2015.
[21] J. Santhiya and K. Bhuvaneswari, "Searching and Classification of List of Keywords and URLs Using SVM Classifier," *International Journal of Scientific and Engineering Research*, vol. 5, pp. 44-48, 2014.
[22] R. Gayathri and A. Malathi, "Exploration of data mining techniques in record deduplication," *IJSR*, vol. 2, pp. 216-19, 2013.
[23] X. Mao, X. Liu, N. Di, X. Li, and H. Yan, "SizeSpotSigs: An effective deduplicate algorithm considering the size of page content," in *Advances in Knowledge Discovery and Data Mining*, ed: Springer, 2011, pp. 537-548.