Data Deduplication Techniques for Big Data Storage Systems

Niteesha Sharma, A. V. Krishna Prasad, V. Kakulapati

Abstract*: *The enormous growth of digital data, especially the data in unstructured format has brought a tremendous challenge on data analysis as well as the data storage systems which are essentially increasing the cost and performance of the backup systems. The traditional systems do not provide any optimization techniques to keep the duplicated data from being backed up. Deduplication of data has become an essential and financial way of the capacity optimization technique which replaces the redundant data. The following paper reviews the deduplication process, types of deduplication and techniques available for data deduplication. Also, many approaches proposed by various researchers on deduplication in Big data storage systems are studied and compared.

Index Terms*: *Big data, storage, deduplication, redundancy, process.

I. INTRODUCTION

In a recent IDC survey, 80% of respondents indicated that in the next 24 months, there will be a significant increase in the number of different types of data in the form of GIF's, videos, sensors and other formats that need to be analyzed. In fact, IDC predicts that the amount of data volumes will increase by 175 zeta bytes worldwide by 2025[1]. This is really a challenging area for effectively storing such large volumes of data as it is very expensive for software organizations and domestic users. Moreover, deduplication studies were conducted by companies like Microsoft, IBM and Google and it was found that almost three quarters of digital data is redundant. This is because every user wants to keep their data secure and safe and that's the reason they keep n multiple replications of their data at various locations. The other reason for data redundancy can also be incremental backup of data since the data needs to be most secure and consistent. As a result, to overcome the replication problem, data deduplication, a highly effective and efficient method, is used to handle the challenges of freeing up storage space [2][3]. Data deduplication, an effective data compression approach, can be used to locate and remove the redundant data using cryptographic hash functions. The hash value generated is of fixed length [4]. Data deduplication compares the new block of data with the saved block of data using secure hash algorithms (SHA). The method to find out whether two data blocks are the same determines the hash value called as the fingerprint. The calculated hash value is compared with the stored fingerprints and if a match is found the new data block is not used for storage. If the fingerprint does not match then the new block of data is stored in the disk and the hash value is saved in the index table of fingerprint storage. Thus, deduplication reduces the amount of the data that has to be stored.

Manuscript published on 30 August 2019.
***Correspondence Author(s)**
Niteesha Sharma, IT Department, Anurag Group of Institutions, Ghatkesar, India.
Dr. A. V. Krishna Prasad, CSE Department, MVSR Engineering College, Hyderabad, India.
Dr. V. Kakulapati, IT Department, Sreenidhi Institute of Science and Technology, Ghatkesar, India.

II. DATA DEDUPLICATION

Data deduplication also called as single instance storage or intelligent compression. It is a capacity optimization technique which is used for eliminating the replicated data and also to enhance the effectiveness of storage. These methods make sure that only one instance of data is stored on devices like Disk, Tapes, and Files etc. The duplicated data blocks are therefore replaced with a pointer to the unique instance of data.

In this way deduplication can help various corporations to increase efficiency of data storage and incremental backups, reduce network bandwidth, manage data growth and reduce administrative costs.

A. Classification of Data Deduplication:

Deduplication can be categorized [5] into two types:

i) **Inline deduplication**: In this, deduplication is performed before the data write to the disk in the form of storage.

ii) **Post process deduplication**: This type of deduplication can be implemented after data stored on to the disk as this can save only disk space.

B. Deduplication based on Granularity:

Generally deduplication occurs at file or block level. The former approach is not efficient because it eliminates only the duplicate files.

i) **File-level deduplication**: In this method, deduplication evaluates the entire file as a chunk, which can be backup of multiple copies of files as storage. This method utilizes hash values for the creation of one index, which can be compared with the stored indexes. If the file is unique, it is stored onto the disk and the indexes are updated. If not only the address of the file is stored, which results in a unique copy of the file to be saved and the remaining copies are eliminated by address of the file.

ii)**Block-level deduplication**: Block level deduplication divides the incoming data (file) in to unique iterations of blocks. These blocks are broken into fixed size chunks and each chunk is processed using various hash algorithms like SHA, MD5 etc.

This process generates the hash values for chunks and if a file is updated only the relevant blocks of data is updated without changing the whole file, i.e., the changes do not constitute the whole file. This is more efficient than the File level deduplication. However, it takes more processing time as the number of indexes to be stored is more. Variable size block deduplication is an alternative for the above approach where the blocks are divided based on the variable length. This approach allows better data reduction ratios. However, more metadata information is generated as a tendency to become slower.

iii) **Byte-level deduplication:** This is also known as micro level deduplication**,** which can be applied on the received data chunks that are segregated into bytes after that deduplication techniques are applied.

III. PROPOSED METHODOLOGY

DATA DEDUPLICATION PROCESS:

This technique is used to remove the replicated data by storing only one instance of data. The main aim of deduplication is to identify the duplicate data and eliminate them.

The advantages of data deduplication include:

i) Reducing storage space

ii)Avoid the network traffic [5].

Here, we discuss the steps involved in the deduplication process as shown in the Figure.1.

A. **The Blocks of Data Stream:** The input file is split into small data chunks, which can be used as fingerprinted.

B. Fingerprinting: Each data block is created by utilizing several hash values also known as fingerprints functions like MD5, SHA-1.

C. Indexing: Indexing is utilized to compare the existing hash values with the newly created fingerprints for detecting the duplicated blocks. If any two data blocks have the same hash value, then this indicates that they are duplicated blocks, which can be eliminated and only one instance of data stored on the disk.

D. Writing: Only one instance can be stored into the storage device.

The following figure shows the process of data deduplication.

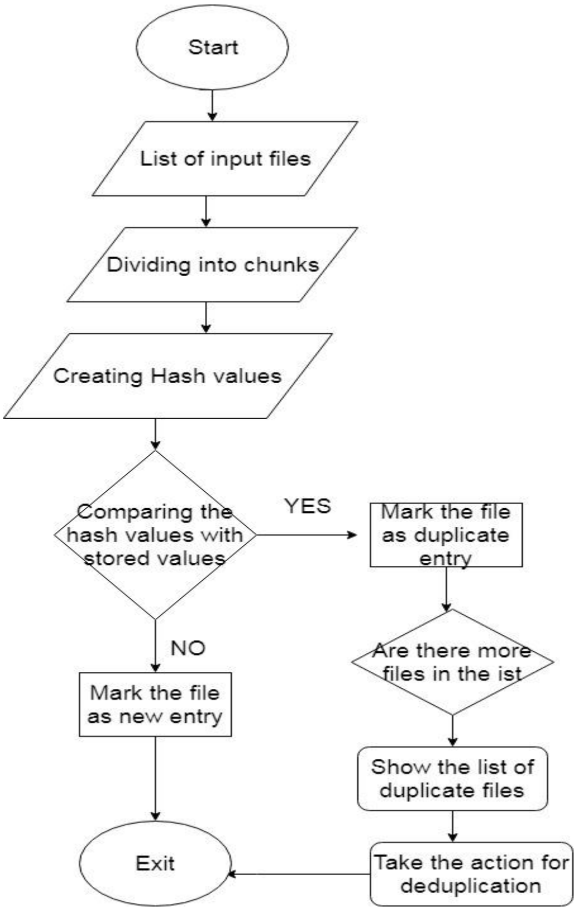
Start
List of input files
Dividing into chunks
Creating Hash values
Comparing the hash values with stored values
YES
Mark the file as duplicate entry
NO
Mark the file as new entry
Are there more files in the ist
Show the list of duplicate files
Take the action for deduplication
Exit

Fig 1: Flow chart representing Deduplication Process.

IV. RELATED WORK

Guohua Wang, et.al[6] adopted a clustering architecture based on Bloom Filter with multiple nodes where chunk level deduplication is done in parallel for all the nodes. The authors put forward a technique called “finger print summary” where in each is allowed to a compact summary of all the other chunks. In this paper SHA-1 algorithm is used to generate hash values.

X. Zhang, et.al[7] proposes the Similarity – Locality approach which finds duplicate segments by using Bloom Filters. As, deduplication system with a single node cannot adhere the needs of industries as well as enterprises, an approach of similarity – locality clustered deduplication is introduced in this paper. This optimizes the system by increasing the redundancy elimination ratio and throughput.

R. Zhou, et.al[5] in their paper have characterized three major sources of Big Data – Deployment of more nodes, increase of the size of data set and replication of the available data. In Global deduplication one ZFS pool is assigned to all VM disk images whereas in Local deduplication one ZFS pool is assigned to each VM disk image. Depending on this, the ZFS computes the deduplication ratio online and is equal to the number of bytes issued to ZFS pool by total number of bytes that is actually issued to the disks.

Ruay-Shiung, et.al [8] discussed about the storage space where HDFS plays a very important role. Since HDFS requires a quadruple amount of storage for saving the file the major disadvantage is expensive cost and the probability of redundant data getting large. Therefore the authors have proposed a dynamic deduplication to improve the usage of storage space which can save more data.

Naresh Kumar et.al[9] developed a bucket based deduplication system based on fixed size chunking level. To generate hash values MD5 algorithm is to be employed. To implement this technique the authors have used an open source Destor tool.

Q. Liu, et.al[10] presented a deduplication solution –Halodedu which is based on Hadoop and Local Database. A Map Reduce technique is adopted in which the map stage is used to realize the parallel deduplication and reduce stage for processing time. Data deduplication is conducted in parallel on deduplication nodes. In order to calculate the unique chunks, both MD5 and SHA-1 algorithms are used. Halodedu uses an index table to identify whether the chunk is duplicate or not. To speed up the searching Halodedu keep the index table in a local database such as MySQL. A comparative study of experiments shows the excellent ability of Halodedu on speed and scalability of the duplication process.

In [11] a deduplication strategy at file level is done using MD5 and SHA-1 algorithms. An experiment was conducted to compute hash values of various formats. The different sizes and results show the comparison of time complexity while using the unstructured file formats of MD5 algorithm with that of SHA-1 algorithm. The time complexity while using the structured and semi structured file formats of MD5 algorithm is low compared to SHA-1 algorithm. Based on the experimental results MD5 algorithm works efficiently compared to SHA-1 algorithm.

The TTTD is proposed in[12] variable size content-defined Chunking (CDC) that detects maximum redundancy than that by the fixed size chunking. For the efficient chunking Two Thresholds, Two Divisors (TTTD-P) algorithm using Genetic Evolution (GE) function is proposed. A Comparative analysis of TTTD, FAST CDC, Rabin CDC and TTTD-P is done in the HDFS environment. TTTD algorithm uses the maximum and minimum threshold values to detect duplicates. TTTD-S is an improvement of TTTD where it takes the average of minimum and maximum thresholds called as average parameter. TTTD-P is an improvement of the above as it detects more duplicity in data by using the optimal parameter found by Genetic Evolution[12].

Dongzhan, et.al[13] developed an integrated deduplication approach where SHA-3 “standard Keccak” for the hash computation is used. The fixed size block employing the client based in-line distributed deduplication approach for Hadoop is used. To address the scalability issues the overall deduplication procedure is implemented in MapReduce and HBase[13].

A proficient Cluster data deduplication approach called AR-dedupe is proposed in[14] . AR- Dedupe consists of four parts: A Backup client, metadata server, routing Server and deduplication server node. The Backup client first sorts the data accordingly and divides the large file into chunks and generates fingerprints. Metadata management server stores the metadata information of all the chunks. Routing server is responsible for load balancing. Deduplication server has n number of server nodes which stores unique chunk data. Two types of data sets were used one to extreme Binning and another to ∑-Dedupe. The experimental results generated indicate a gradual increase of 30% in the performance.

Naresh Kumar, et.al [15] presented a novel approach in data deduplication in which a hash based technique MD5 for deduplication of data is used and implemented in a distributed environment using Hadoop framework. Mapper and reducer are used to maintain the storage space and bucket approach is used for indexing of unique hash values.

A new Deduplication approach is proposed by Garg, et.al in[16] for semantic analysis of data. The data contains the same name, but in a different data format. This type of data duplication is not considered or solved using file/Block/Byte level data deduplication technique. Later the same mechanism is applied for the record level data duplication and experimental results show the maximum redundant data removal rate.

In [17] the data reliability is improved by backing up in-memory data periodically and for this NewSQL, an emerging database system for database backups is used,which results in a lot of redundant data. The traditional methods of data deduplication are not fully optimized and to overcome this, a new deduplication optimization method (DOMe) for NewSQL system backup is proposed. This method parallelizes the deduplication method based on the fork-join framework. The experimental results show that the deduplication performance is improved by parallelizing the CDC algorithm leading to an increase in throughput.

V. EXISTING TECHNIQUES ON DEDUPLICATION AND THEIR GAPS, PROS AND CONS

Deduplication can be implemented both at the source end (where data is created) as well as at the target end (where data is stored).

At the source end implementation, the elimination of redundant data is done at the source before it transfers to the backup device. This technique can decrease the amount of data sent over the network for backup. The Shorter backup window and less bandwidth are the advantages of this method whereas this method increases the amount of overhead on client/source side.

The target end implementation is done at the back up device, i.e., at the target device. In this technique the whole data is sent through the network which requires more bandwidth; also more storage capacity is needed at the target end.

Deduplication can occur either at the whole file level or sub-file level. The chunking file is treated as a whole for checking redundancy, referred to as a single Instance storage, but if the same data is stored in two or more files, then this approach detects and removes the redundant copies of the identical files. This method is implemented with less cost and the deduplication effectiveness is also the lowest.

To overcome this, the sub-file level approach was introduced, which is implemented either in fixed size or variable size chunking (Content Defined Chunking).

The simplest approach is to cut/divide the data stream into fixed size (Static). In this approach if any part of the file needs to be modified (i.e. insertion/deletion takes place) then not only the chunk containing the modification is changed but also the remaining data chunks are also shifted. This leads to the boundary-shift problem those results in a significantly reduced duplicate identification ratio. For Example consider a File F1 as shown in the Fig 2, let us suppose that the deduplication uses a 10 byte fixed block size of data to divide the file into chunks as granularity. Likewise when we modify the file F1 in which we add some text "THIS IS" in the starting of the file F1, the deduplication process splits the original file (now F2) again into the same size blocks of 10 bytes. The block's split of file F2 is completely different from the previous file F1.This is because the contents of the file are shifted, which is referred to as the boundary shift problem in Static/Fixed size Chunking algorithm. This method seeks CPU Complexity and the deduplication effectiveness in the Middle.

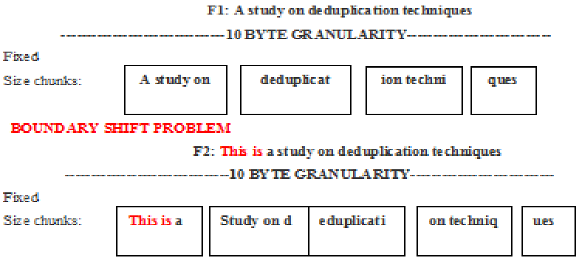
F1: A study on deduplication techniques
-----------------------------10 BYTE GRANULARITY-----------------------------
Fixed
Size chunks:
A study on
deduplicat
ion techni
ques
BOUNDARY SHIFT PROBLEM
F2: This is a study on deduplication techniques
-----------------------------10 BYTE GRANULARITY-----------------------------
Fixed
Size chunks:
This is a
Study on d
eduplicati
on techniq
ues

Fig 2: Fixed size chunking

To address this, content defined chunking (CDC) is used. CDC divides the input data based on the content itself. The CDC uses the sliding window protocol technique in order to generate the hash values on the content of files. Fig 3 shows how content defined chunking is used. Consider two files F1 (The original file) and F2 an updated version of F1, where we add some text in beginning or at the middle or end of file F1. The hash value generated is based on the Rabin algorithm. A chunk break point is decided when the hash value generated matches the other hash value, and then the data chunk is treated as redundant. This method seeks more of CPU complexity and the deduplication effectiveness is the highest.

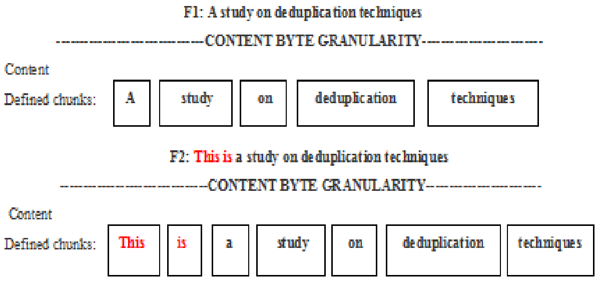
F1: A study on deduplication techniques
-----------------------------CONTENT BYTE GRANULARITY-----------------------------
Content
Defined chunks:
A
study
on
deduplication
techniques
F2: This is a study on deduplication techniques
-----------------------------CONTENT BYTE GRANULARITY-----------------------------
Content
Defined chunks:
This
is
a
study
on
deduplication
techniques

Fig 3: Content defined chunking

VI. A COMPARATIVE STUDY ON DATA DEDUPLICATION BY VARIOUS RESEARCHERS

S. No	Approach	Methodology	Challenges & Future Work
1.	Block level Deduplication[8]	Two tier deduplication: Prefilter & Postfilter	Data reliability without data storage space.
2.	Chunk Level Deduplication[10]	Unique data storage is done using HDFS; Map Reduce is used to realize parallel deduplication processing.	Deduplication ratio needs to be improved.
3.	Fixed size chunking algorithm[9]	HDFS is used for storage using buckets; Map Reduce is used to find the duplicates.	Results need to be refined with low computation time.
4.	Multi thread Frequency-Based chunking method.[12]	TTTD algorithm for both single and multi-threading.	More than one FBC thread causes bottle neck in CDC Chunking.
5.	Variable Chunking method.[15]	Improvement in storage efficiency of big data using hash based deduplication(MD5)	Avoiding using and maintenance of external hard drives for storing fingerprints.
6.	File Based Chunking[18]	RFD-HDFS is used for applications with no errors and FD-HDFS is used for applications with acceptable errors.	Storage space can be improved.

Retrieval Number: J91290881019/19©BEIESP
DOI: 10.35940/ijitee.J9129.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

7.	Content Based File Chunking.[19]	Dual mode chunking is used. Invariability also called as MUCH.	---
8.	Fixed size Block[13]	Keccak (SHA-3) is used for Hash Computation.	Further optimization on chunking algorithm and I/O Operations. Migration of deduplication Framework to Spark Platform.
9.	Sorted Neighborhood method.[20]	Partition-Sort-Map-Reduce approach with adaptive sliding window, overlapping the boundary objects to find duplicates in adjacent partition.	Improve the current proposal.
10.	Fixed Size Blocking Algorithm[21]	Input data is text document. Each blocks hash Value is calculated using MD5 Algorithm.	Proposed system can be done on images and videos.
11.	Content defined chunking.[22]	File similarity and data locality.	--
12.	File level & Chunk Level[23]	Bloom Filter Array is used for search process.	---
13.	Chunk Level[24]	Finding same data stored on disks.	Efficient usage of computing resources.
14.	Fixed size chunking[25]	Unique Hash values are generated using SHA-2 algorithm	Apply the proposed work on real life applications, achieving less execution time.

VII. RESULT ANALYSIS

Increase in the unstructured data growth rate has imposed many challenges with respect to data storage. Data Deduplication is one of the emerging technologies which reduces both storage space but and the amount of redundant data being saved. In this paper, deduplication is classified based on backup and granularity and next the process and techniques involved in deduplication are discussed. Also, surveys of various techniques used on deduplication are compared in terms of certain performance parameters.

VIII. CONCLUSION

In our study we have compared the various Deduplication approaches and from these we observe that still more challenges need to be addressed. In the future, we will apply k-means clustering for better deduplication in HDFS and also incorporate better storage results. We apply some nature inspired algorithm for achieving an efficient de duplication process.

REFERENCES

1. D. Reinsel, J. Gantz, and J. Rydning, “The Digitization of the World From Edge to Core Mankind is on a quest to digitize the world,” no. November, 2018.
2. S. Singh and R. Singh, “A Viewpoint on Different Data Deduplication Systems and Allied Issues,” pp. 385–393, 2018.
3. G. Zhu, X. Zhang, L. Wang, Y. Zhu, and X. Dong, “An Intelligent Data De-duplication Based Backup System,” 2012.
4. G. Sun, Y. Dong, D. Chen, and J. Wei, “Data backup and recovery based on data de-duplication,” 2010.
5. R. Zhou, M. Liu, and T. Li, “Characterizing the efficiency of data deduplication for big data storage management,” Proc. - 2013 IEEE Int. Symp. Workload Charact. IISWC 2013, pp. 98–108, 2013.
6. G. Wang, Y. Zhao, X. Xie, and L. Liu, “Research on a clustering data de-duplication mechanism based on Bloom Filter,” 2010 Int. Conf. Multimed. Technol. ICMT 2010, no. 60573145, pp. 0–4, 2010.
7. X. Zhang and J. Zhang, “Data deduplication cluster based on similarity-locality approach,” Proc. - 2013 IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber, Phys. Soc. Comput. GreenCom-iThings-CPSCom 2013, pp. 2168–2172, 2013.
8. R. S. Chang, C. S. Liao, K. Z. Fan, and C. M. Wu, “Dynamic deduplication decision in a hadoop distributed file system,” Int. J. Distrib. Sens. Networks, vol. 2014, 2014.
9. N. Kumar, R. Rawat, and S. C. Jain, “Bucket Based Data Deduplication Technique for Big Data Storage System,” pp. 267–271, 2016.
10. Q. Liu, Y. Fu, G. Ni, and R. Hou, “Hadoop Based Scalable Cluster Deduplication for Big Data,” Proc. - 2016 IEEE 36th Int. Conf. Distrib. Comput. Syst. Work. ICDCSW 2016, pp. 98–105, 2016.
11. S. Ranjitha, P. Sudhakar, and K. S. Seetharaman, “A Novel and Efficient De-duplication System for HDFS,” Procedia Comput. Sci., vol. 92, pp. 498–505, 2016.
12. N. Kumar, S. Antwal, G. Samarthyam, and S. C. Jain, “Genetic optimized data deduplication for distributed big data storage systems,” 4th IEEE Int. Conf. Signal Process. Comput. Control. ISPCC 2017, vol. 2017–January, pp. 7–15, 2017.
13. D. Zhang, C. Liao, W. Yan, R. Tao, and W. Zheng, “Data Deduplication based on Hadoop,” 2017.
14. N. A. Academy, “AR-Dedupe : An Efficient Deduplication Approach for,” vol. 20, no. 1, pp. 76–81, 2015.
15. N. Kumar, P. Malik, S. Bhardwaj, and S. C. Jain, “Enhancing Storage Efficiency Using Distributed De-duplication for Big Data Storage Systems,” no. June, pp. 96–108, 2017.
16. S. Garg, “Semantic Analysis of Big Data by Applying De-duplication techniques.”

17. L. Wang, Z. Zhu, X. Zhang, X. Dong, and Y. Wang, “DOMe : A deduplication optimization method for the NewSQL database backups,” vol. i, pp. 1–17, 2017.
18. R. Sheu, S. Yuan, W. Lo, and C. Ku, “Design and Implementation of File Deduplication Framework on HDFS,” vol. 2014, 2014.
19. Y. Won, K. Lim, and J. Min, “MUCH : Multithreaded Content-Based File Chunking,” vol. 64, no. 5, pp. 1375–1388, 2015.
20. K. Ma, F. Dong, and B. Yang, “Large-Scale Schema-Free Data Deduplication Approach with Adaptive Sliding Window Using MapReduce,” 2015.
21. S. More and K. Devadkar, “EFFICIENT DATA SEARCHING AND RETRIEVAL USING BLOCK,” pp. 14–18, 2018.
22. L. Song, Y. Deng, and J. Xie, “Exploiting Fingerprint Prefetching to Improve the Performance of Data Deduplication,” pp. 849–856, 2013.
23. J. Zhang, S. Zhang, Y. Lu, and X. Zhang, “Hierarchical Data Deduplication Technology Based on Bloom Filter Array,” pp. 725–732.
24. S. Singh, “Next Level Approach of Data Deduplication in the Era of Big Data,” vol. 8, no. 4, pp. 71–74, 2017.
25. N. Kumar, “Secure Data Deduplication in Hadoop Distributed File Storage System,” vol. 7, no. 9, pp. 10–13, 2017.

AUTHORS PROFILE

Niteesha Sharma working as Assistant professor in Department of Information Technology at Anurag Group of Institutions, Ghatkesar is currently working towards Ph. D degree in Computer Science and Engineering at Osmania University. Her research interests include Data Intensive Science; data deduplication in Big Data etc. She has five papers in refereed journals out of which two are indexed in scopus.

Dr. A. V. Krishna Prasad working as a Professor in Department of Computer Science and Engineering at MVSR Engineering College, Hyderabad. He was awarded Doctoral degree in Computer Science from Sri Venkateswara University, Tirupati in 2012. He got several patents. He published several Books, Book Chapters, several papers in National, International Conferences and International journals. He is an Editorial Board Member and Reviewer for several International Journals. He is corporate trainer for Data Science and Analytics. His Interested Research areas are Mining, Data Science and Analytics**.**

Dr. V. Kakulapati is a well-known faculty in computer science at Hyderabad She is working as a Professor in the Department of Information Technology, Sreenidhi Institute of Science and Technology. She has two doctorate degrees and four PG Degrees. She is a member of various professional bodies like IEEE, ACM, CSTA, LMISTE, LMCSI, IACSIT, FIETE, Big data University, IOT Central, Data mining research group, Microsoft research group, IBM research group and few more. She is serving as an Editorial Board member to various International and National journals. Apart from these she is continuously serving couple of National & International Conferences as a review member and various International Journals including IJCT and Egyptian Informatics Journal. Track manager for ICTIS, ICCII and ICDECT conferences and member in Board of Studies. She is having more than 55+ publications in National, International Journals and conferences. Out of these 13 are Springer Chapters, 3 book chapters, 2 ACM, 2 IEEE and 1 in Elsevier.