Publishing High-Dimensional Micro Data Using Anonymization Technique

K. Kiruthika\textsuperscript{1}, M.S Kavitha\textsuperscript{2} & S. Gayathiri\textsuperscript{3}
\textsuperscript{1}PG Scholar, \textsuperscript{2}Assistant professor, \textsuperscript{1,2}Sri Eshwar College of Engineering, Coimbatore, India
\textsuperscript{3}Tech Mahindra, Design Associate Software Engineer, Bangalore, India

Abstract: Now a day’s society is experiencing very good growth in the count and variety of data collections having person-specific information as network connectivity, computer technology & disk storage space become increasingly affordable. Large databases is in use today’s society. The large amount of data available means that it is helpful to learn lot of individual information from public data. While doing, the privacy of the data should be persevered. The personal data may be misused, for various purposes. In order to improve these concerns, a number of techniques have newly been proposed to do the data mining tasks in a privacy-preserving manner. So, to preserve the privacy of the data, privacy preserving methods are introduced and one of the methods is 'anonymization', in which a record is anonymized by using different anonymization techniques. Privacy preserving in data publishing is most important research area in data security area. Privacy-preserving data publishing provides methods and tools for sharing useful information while preserving data privacy and also analyzed many methods used for data privacy and their reward and short comes are monitored very well. Many techniques have been designed for micro data publishing with privacy preserving, such as generalization and bucketization. Several works resulted that generalization loses some amount of information specifically for high dimensional data. So it’s not efficient for high dimensional data for privacy preserving microdata publishing. In Bucketization, it does not prevents membership exposes and also does not suitable for data that do not have a clear separation between Quasi-Identifying attributes and Sensitive attributes of data. In our paper, an efficient method for data anonymization known as slicing and k-anonymity is introduced in which data can be partitioned both vertically(column) and horizontally(row). Advantage of slicing is that it works on high-dimensional data. Also, slicing preserves better data utility and privacy than any other method. An efficient algorithm is developed for computing sliced data that follow l-diversity requirement. A random permutation is done to randomly order a set of objects, that is, to permutate the data for applying more privacy. Data slicing gives better utility than generalization and also no necessary for clear separation between Quasi-identifying and sensitive attributes. Experimental results using hospital dataset confirm that data slicing provides data utility than generalization and more effectual than bucketization including sensitive attributes. Experiments we use the hospital patient datasets propose that our approach achieves better utility and also efficiency than any other existing and baseline algorithms while satisfying of proposed security work.

Keywords: Privacy Preserving, Anonymization, Quasi Identifying Attributes, and Slicing, K-Anonymity, l-Diversity, Random Suffling, Bucketization.

1. INTRODUCTION

Data mining is the process of finding out interesting patterns from large amounts of data such as data warehouse, web or other information repositories. The data is mined to extract useful, interesting, and previously unknown information, but the Data Anonymization is different from mining. The collection of information by governments, corporations, and individuals has created an space that facilitates large-scale data mining and data analysis. The main issue for the data here ‘confidentiality of private data’. Confidentiality of private data means the sensitive or secret information of individuals should be protected in the published data. i.e. publishing data about individuals without leaking sensitive information about them is an important difficulty \cite{2}. So, to maintain privacy, first thing is to anonymize the data i.e. remove the personally identifying information like name, email-id, mobile number, etc before publishing.

1.1 Handling Large Data

Big Data where the information comes from various heterogeneous, autonomous sources with complex relationship and it is endlessly growing. More than 3.5 quintillion bytes of data are created daily and 90 percent data in the world today were produced within past three years \cite{1}. This shows that it is very difficult for big data applications to manage process and retrieve required data from very huge volume of data. There are various challenges with Big-Data. Data privacy challenge, which can be solved using different methods like key based encryption \cite{3} and
anonymization. To process and handle high dimensional distributed data is also one of the challenge. In this the data has different dimensions for example, in hospitals, patients data is stored in text, images, videos etc are used to store reports of X-ray, CT scan for detail examinations. MR-Cube method is used for efficient calculation of cube. Performing data analysis on such big data becomes very expensive as data is distributed, endlessly keeps growing and the nature of data is structured, unstructured and sometimes semi structured. Audio, Pdf, videos, images are example of unstructured data. That is for multidimensional microdata, data cube is one of the great tool. Method is used for efficient calculation of cube.

Audio, Pdf, videos, images are example of unstructured data. That is for multidimensional microdata, data cube is one of the great tool. Method is used for efficient calculation of cube.

Hadoop is an open-source framework for storing large data and running applications on clusters of normal commodity hardware where MapReduce is used to process and HDFS to store data. It provides enormous storage for all kind of data, enormous processing supremacy and the capacity to handle virtually unlimited concurrent tasks or jobs. So, while using Hadoop for processing large data the performance of the system will be very effective compared to other. Various anonymization methods for privacy preserving in publishing of microdata have been reviewed in recent years. Micro data contains records and all those records includes info about an individual dimension and attribute days, months, and years are of temporal dimension.

Hadoop is an open-source framework for storing large data and running applications on clusters of normal commodity hardware where MapReduce is used to process and HDFS to store data. It provides enormous storage for all kind of data, enormous processing supremacy and the capacity to handle virtually unlimited concurrent tasks or jobs. So, while using Hadoop for processing large data the performance of the system will be very effective compared to other. Various anonymization methods for privacy preserving in publishing of microdata have been reviewed in recent years. Micro data contains records and all those records includes info about an individual dimension and attribute days, months, and years are of temporal dimension.

1.2 Previously Introduced Techniques

Existing techniques such as centralized algorithms for anonymization [9] think that all data process should fit in memory. There are also some distributed algorithms which is mainly for security integrating and anonymizing various data sources rather than scalability issue of TDS anonymization. In the proposed system we focus to preserve the privacy for data with hadoop framework, and also overcome the problems of M-privacy and secrecy methods with new anonymization and slicing technique. As there is greater need of sharing personal information through distributed database, the special concern should be taken to protect it from hackers. Invaders can be single entity or group of entities. Invaders can break privacy with the use of background information. Collaborative data publishing taken this as a multi-party computation problem. In this problem multiple providers want to compute an anonymized view of their data without publishing any private and sensitive information. A data recipient that might be an attacker, e.g., q0, attempts to collects more information about data records using the published data, D*, and background knowledge. For example, k-anonymity [10] protects against open attacks on identity by requiring each quasi identifier equivalence group (QI group) to have at least k records. L-Diversity requires that each QI group should have at least l sensitive values. Various privacy guarantees that the presence of a record can’t be inferred from a statistical data release with small assumptions on an attacker’s background information. For high dimensional data, slicing method is very useful because it separates the data in both vertical and horizontal manner. Encryption can also provide security, but the limitation is there could be loss of data utility. Our goal is to publish an anonymized view of integrated data, which will be protected to attacks. We improve the security and privacy of data with the help of slicing technique which satisfies privacy verification with good performance than provider aware (base algorithm) and encryption algorithm. Many microdata anonymization techniques have been introduced like generalization [3], Generaliation, bucketization, and suppression for ‘l-diversity’ [10].

II. PRIVACY MODELS

2.1 K-Anonymity

K-Anonymity means if the material for each person contained in the release cannot be known from at least k-1 individuals whose information also be release. For e.g., if you try to find a person from a released data, but the only information you have is their date of birth and gender. There are k people meet the same condition. This is said to be k-Anonymity. The database is expected to be K-anonymous where attributes are blocked or generalized until each row is equal with at-least k-1 other rows. K-Anonymity thus prevents definite database linkages. K-Anonymity assures that the data published is just right. K-anonymity has two techniques: generalization and Suppression. To protect defendant’s identity when releasing micro data, data holders sometimes remove or encrypt explicit identifiers, such as names and security numbers. [2]K-anonymity does not deliver guarantee of anonymity in de-identifying of data. Every free information often covers other data, for example: birth date, sex, and ZIP code that can be publicly obtainable and it cannot intend for publish openly. One of the concepts in micro data protection is [3] k-anonymity. One of the stimulating aspects of k-
Presently, there exist two broad categories of records that all contribute to a specific quasi-identifier. It resolves the short comes of k-anonymity systems: generalization and permutation based. An current generalization method would divide the data into disjoint groups of Contacts, such that each group covers sufficient records with l-distinct, well represented sensitive data. It assets both record and attributes linkage.

Limitations of k-anonymity

Limitations identified are: (1) it does not hide whether a given separate is in the database, (2) it reveals folks sensitive attributes , (3) it doesn’t protect against attacks based on background information,(4) mere knowledge of the k-anonymity algorithm can violate privacy, (5) it can’t be applied to high-dimensional data without whole loss of utility , and (6) special approaches are required if a dataset is anonymized and available more than once. It preserves record linkage.

2.2 I-Diversity

I-Diversity principle: A q-block is l-diverse if it has at-least l ‘well represented” values for the sensitive attribute S. A table is l-diverse if all other q-block is l-diverse. L-Diversity gives privacy conserving even when the data generator does not know what kind of knowledge is crazed by the competitor. The idea of l-diversity is the object that the values of those sensitive attributes are represented in every group. The l-diversity can be of two methods. First, Distinct l-Diversity Each one is equivalence class has at least l well-represented sensitive values Second, Entropy l-diversity Each equivalence class is not only must have enough different sensitive values, but also the diverse sensitive values must be allocated evenly enough. It resources the entropy of the distribution of sensitive values in each likeness class is at least log(l).Sometimes this may be too limiting. When some values are very common, the entropy of the whole table may be low. This leads to the low conservative notion of l-diversity. Declare if you have a group of k various records that can share every certain quasi-identifier. An hacker cannot identify the different based on the quasi-identifier. The identical for all value in the group. The circulation of target values within a group is to as “diversity”. It resolves the short comes of k-anonymity. The concept is “l-diversity”. Approximately if you have a group of k different records that all contribute to a specific quasi-identifier. That is well where an hacker cannot find the inaccessible based on the quasi-identifier. But if the value they are involved in the same for every value in the group. The supply of target values within a group is referred to as “l-diversity”. Presently, there exist two broad categories of l-
about specific individuals. It conserve both from attribute linkage and probabilistic attack.

**Limitations of T-Closeness**

There is no computational process to enforce t-closeness followed in. There is effective way till now of Joining with generalizations and suppressions or slicing. Gone co-relation among different attributes. This is since each attribute is generalized distinctly[11] and so we lose their need on each other.

Utility of data is hurt if we use very small t (And small t will result in increase in computational time).

### III. ANONYMIZATION TECHNIQUE

#### 3.1 Generalization

Generalization is one of the normally anonymized methods, which replaces quasi-identifier values with values that are less-specific but semantically reliable. Then, all quasi-identifier values in a group would be general to the whole group range in the QID space. If at-least[9] two transactions in a group have separate values in a certain column (i.e. 1 contains an item and the other does not), then all material about that item in the current collection is lost. The QID used in this process includes all likely items in the log. Due to the high-dimensionality of the quasi-identifier, with the no of possible items in the sort of thousands, it is likely that any generalization method would incur extremely high information loss, representing the data useless. In order for generalization to be actual, records in the same bucket must be close to all other so that simplifying the records would not lose too much data. However, in high-dimensional data, many data points have same distances with each other.

To perform data analysis asks on the generalized table, the data analyst has to make the constant distribution guess that every value in a widespread interval/set is equally possible, as no other allocation assumption can be justified. This meaning fully decreases the data utility of the generalized content. And also because each attribute is universal separately, correlations between different attributes are lost. To study attribute correlations on the generalized table, the analyst has to assume that every possible blend of attribute values is equal as possible. This will be an inherent problem of overview that protects effective analysis of attribute correlations.

#### 3.2 Bucketization

The first, which we term bucketization, is to divides the tuples in T into buckets, and then to separate the sensitive attribute from the non-sensitive ones by aimlessly permuting the sensitive attribute values within each bucket. The clean data then include the buckets with permuted sensitive data. We use bucketization as the method of creating the published micro data from the original table T, although all results hold for full-domain generalization as well. Partition the tuples into specified sized buckets (i.e., horizontally partition the table T according to some of the scheme), and within each bucket, Here applied an independent random permutation to the column including S-values. At last set of buckets, denoted by B, is then published. For eg, if the underlying table T, then the distributer might publish bucketization B. For added privacy, the publisher can completely mask the classifying attribute (Name) and may incompletely cover some of the other non-sensitive attributes (Age, Sex, Zip).While bucketization has better data utility than simplification, it has more limitations. First, bucketization does not prevent membership disclosure. Because bucketization distributes the QI values in their original forms, an challenger can find out whether an individual has a record in the distributed data or not. A micro data (e.g., census data) usually encloses many of the other attributes besides those three attributes. This means that the membership information of a large amount of individuals can be indirect from the bucketized table. Second, bucketization needs a clear separation between QIs and SAs. However, in many data sets, it is unclear those attributes are QIs and which are SAs. Third, by untying the sensitive attribute from the QI attributes, bucketization cracks the attribute correlations between the QIs and the SAs. Bucketization will first partitions tuples in the table into buckets and then separates the QI with the sensitive attribute by randomly permuting the very direct attribute values in each bucket. The anonymized data includes a set of buckets with permuted direct-attribute values. In bucketization has been used for anonymizing highly dimensional data. Their approach assumes a good separation between QIs and SAs. In addition, because the accurate values of all QIs are released, membership information is disclosed.

### IV. RELATED WORK

#### 4.1 Background

Anonymity is the process of changing one’s name or identity to unknown or concealed. It
serves valuable social purposes and empowers individuals as against organisations by limiting surveillance, but it is also used by wrong users, to hide their actions to allow anonymous access to services, which avoid tracking of user's personal detail and user behavior such as user location, salary details, account id’s, frequency of a service usage, and so on. If someone sends a file, there may be details on the file that leaves a trail to the sender. The sender's detail may be traced from the data logged after the file is sent.

A. Anonymity Vs. Security

Anonymity is a very good technique for protecting privacy. The decentralized & stateless design of the Internet is particularly adoptable for anonymous behavior. Although anonymous actions can ensure privacy, they should not be used for ensuring privacy protection as they also allow for harmful activities, such as spamming, harmful, and slander, attacks without any fear of reprisal. Security shows that one should be able to detect and catch individuals illegal behaviour, such as hijacking, hacking, conspiring for terrorist acts, and conducting fraud activities. After anonymization there is no need for security because no one can able to hack the personal details from this anonymized data.

B. Anonymity Vs. Privacy

Privacy and anonymity are not equal. The difference between privacy and anonymity is clear. Privacy corresponds to send an encrypted e-mail to another recipient. Anonymity is able to send the contents of the e-mail in plaintext, easily readable format but without any details that enables a reader of the message to identify the particular person who create it. Privacy is very important when the contents of a message are in issue, where anonymity is important when the identity of the author of a message is in issue.

4.2 Privacy Risks

Three types of privacy disclosure hazards are possible when publishing microdata.

4.2.1 Membership Disclosure

Membership disclosure means learning whether an individual is included in the database. The dataset which is to be disclosed is choose from a large population and the selection criteria are very sensitive, none needs to prevent adversaries, when person’s private record is included in the publically released dataset.

4.2.2 Identity Disclosure

Identity disclosure is linking an individual to a record in the dataset. Identity disclosure occurs when an individual’s record is associated to a particular record in the released table of dataset. Sometimes, one wants to protect against their identity disclosure when the adversary is uncertain of membership. In this situation, protection against membership release helps protect against identifying the entity. In other cases, some adversary may already know that an individual’s information is in the released dataset, in that case, membership disclosure protection is absent or is lacking.

4.2.3 Attribute Disclosure

Attribute disclosure occurs when new content about some individuals is leaked, i.e. the leaked data made it possible to infer the attributes of an individual person more correctly than it would be possible before the release. Attribute disclosure can occur with or without identity disclosure.

V. PROPOSED SYSTEM AND EXPERIMENTAL RESULT

5.1 Basic Ideas of Data Slicing

To improve the release of the patient data and to provide better data utility sliced data is more efficient when comparing with generalization and bucketization. In generalization, it is exposed that generalization loses considerable information mainly, for high dimensional data. In order to perform data analysis or data mining tasks on the generalized data, the data analyst make the uniform distribution assumption over every value in a generalized set is evenly possible and no other distribution assumption can be acceptable. This significantly reduces the data utility of the generalized table of data. In that generalized table of data each attribute is generalized separately, correlations between different attributes are vanished. This are the inherent problem of generalization. In case of bucketization, it has better data utility than generalization technique but does not prevent membership disclosure.

Secondly bucketization publishes the QI values in their original forms, an attacker can easily find out whether an individual has a clear record in the published data or not. This shows that membership information of the majority individuals can be inferred from the bucketization
that when the data set contains QIs and one SA, bucketization has to crack their correlation; slicing, on the other hand, can cluster some QI attributes with the SA, protecting attribute correlations with the very responsive attribute. The key intuition that slicing provides privacy protection is that the slicing process ensures that for all tuple, there are generally many matching buckets. Slicing first partitions attributes into columns. Every column contains a subset of attributes. Slicing partition the tuples into buckets. Every bucket includes a subset of tuples. This horizontally partitions the table of content. Within each

This paper introduces a new technique, called DATA SLICING using the hadoop framework. This method both vertically and horizontally partition the data. Vertical partitioning is done through clustering attributes into columns based on the correlations among the attributes. Each column includes a subset of attributes that are highly correlated. Horizontal partitioning is made by grouping tuples into buckets. At last, between each bucket, values in every column are randomly permuted to crack the association between different columns. The core idea of data slicing technique is to crack the association across columns, but to maintain the association within each and every column. This reduces the dimensionality of the data and provides better data utility than bucketization and generalization method.

5.2 Overview of Slicing

Slicing partitions the data set into both vertically and horizontally. Vertical partitioning is done by separating attributes into columns according to the correlations among the attributes. Each column contains a subset of attributes that are extremely correlated. Horizontal partitioning is done by grouping tuples into buckets. At the end, within every bucket, values in each column are randomly (or sorted) permuted to disconnect the linking between different columns. The main idea of slicing is to crack the association cross columns, but to preserve the association within each column. This preserves better utility and reduces the dimensionality of the data than generalization and bucketization. Slicing preserves utility because it clusters highly correlated attributes together, and preserves the correlations between such attributes. Slicing protects secrecy of individuals because it cracks the associations between uncorrelated attributes, which are rare and thus identifying. Note down
separated bucket, values in each column are randomly sorted to break the linking between various columns. This algorithm includes three phases: attribute partitioning, (column) generalization, and tuple partitioning.

**Attribute Partitioning**

This method partitions attributes so that highly correlated attributes are in the single column. This is better for both utility and privacy. In terms of data usability, grouping highly correlated attributes preserves the correlations between those attributes. In terms of privacy, the grouping of uncorrelated attributes presents higher detection risks than the association of strongly correlated attributes because the grouping of uncorrelated attribute values is much less recurrent and thus more identifiable.

**Column Generalization**

Column generalization is required for identity the membership disclosure. This is not suggested for privacy protection, as in the protection. If a column value is unique in a column, a tuple with this unique column value can only have one similar bucket. In generalization/bucketization where every tuple can belong to only one equivalence-class/bucket.

**Tuple Partitioning**

The algorithm maintains two different data structures there are: 1) a queue of buckets Q and 2) a set of sliced buckets SiB. Initially, Q contains only one bucket which includes each tuples and SiB is empty. For every iteration, the algorithm removes a bucket from Q and divides the bucket into two buckets. If the sliced table of data after the split satisfies l-diversity, then the algorithm places the two buckets at the end of the queue Q or else, we cannot divide the bucket anymore and the algorithm puts the bucket into SiB. When Q becomes null, we have computed the sliced table of data. The set of sliced buckets is SiB.

**5.2 Experimental Setup**

The most common technique to preserve sensitivity is to modify the contents of the information owners before publishing the data using our approach itself is called Anonymization. In the most basic form of PPDP, the data holder has a table with : Explicit Identifier, Quasi Identifier, Sensitive Attributes, non-Sensitive Attributes, where Explicit Identifier is a group of attributes, such as name, id number and social security number, containing information that explicitly identifies record owners clearly, Quasi Identifier is a set of attributes that could potentially identify record owners, Sensitive Attributes includes sensitive person specific information such as disease, salary, area, age and disability status and Non-Sensitive Attributes include all attributes that do not fall into the previous three categories. Most works assume that the four sets of attributes are dissimilar. Major works assume that each record in the table represents a unique record owner.
Hadoop is well suited to large data because Hadoop works by breaking the data into pieces and assigning each "piece" to a particular cluster node for analysis. The data does not have to be similar because each piece of data is being managed by a separate process on a separate cluster of the hadoop node. A Hadoop cluster's parallel processing capabilities certainly assist with the speed of the analysis and also resilient to failure with more scalability.

5.3 Experimental Output for Slicing

Slicing partitions the dataset in both horizontal and vertical manner. Vertical partitioning means grouping the attributes into columns according to the correlations of the attributes. Attribute partitioning enables slicing process the highly-dimensional data. Then horizontally partition is performed through grouping the tuples into buckets and within every bucket, values in each column are randomly permuted to crack the connectivity between different columns. Slicing breaks the association across columns, but protects the associations with every column. Slicing gives better privacy guard, that the slicing ensures that for all tuple there are multiple similar buckets. Given a tuple $t$, $c$ is the count of columns and $v$ is the value of the column in dataset and a bucket is a similar bucket from dataset for $t$ iff for each column value that come out atleast once in the $i$th column of the bucket of our dataset. Any bucket that has tuple as the original tuple is the matching bucket. At the same time similar bucket contains other tuples with same values but not all the buckets. Slicing prevents membership disclosure by splitting the QI attributes into different columns and correlations among those different columns are broken down. Every columns attribute values are randomly permuted. None can tell the tuple is the original dataset. Slicing can be used to prevent attribute disclosure according to the privacy requirement of l-diversity slicing. If that bucket is l-diverse then there will be 1 or more represented values for SAs. Let ‘$T$’ is the data which has been sliced by applying slicing technique. This method has two phases: vertical and horizontal partitioning of the dataset to be anonymized.
5.5 Vertical Partitioning of The Data

Vertical partitioning of the dataset includes the subset of attributes made as columns. Table T contains d attributes $A = \{A_1, A_2, \ldots, A_d\}$ and C columns $c_1, c_2, \ldots, c_n$. Our method grouping the highly correlated attributes are in the same column to preserve the data utility that means data should maintain privacy and data is useful for analysis. The association of uncorrelated attributes in the dataset provide more identification risk. The highly correlated attribute values are more recurrent than uncorrelated attributes and this reduces identification risks.

In slicing technique where each attribute is exactly in one single column. In this approach we consider the sensitive attribute to combine at different sets to provide superior anonymity. By duplicating the attribute in more column provides better data utility than other but this discharges more attribute correlations among them. But the privacy implications in this must be carefully retained.

5.6 Random Permutation

Random Permutations remembers that a permutation $\pi$ is one-to-one and onto function $\pi : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$. A permutation defines a reordering of the elements $1, \ldots, n$. It can be specified by the new values. e.g. $(2 6 3 5 1)$ specifies a permutation where $\pi(1) = 2$, $\pi(2) = 6$, $\pi(3) = 3$, $\pi(6) = 5$ and $\pi(5) = 1$. There are $n!$ permutations of $n$ given elements. That’s so many to create them by numbering all and randomly choose one. Random permutations are useful in randomized algorithms, cryptography and simulations. So its helpful to have efficient algorithms for generating them in random. But more than this, assured algorithms give us good characterizations of random permutations for use in proofs. Creating a random permutation is like “unsorting”. A sorting program can make an arbitrary permutation of elements to place them in ascending order, descending order or in any other predefined order. If the swaps in a sorting program are changed by random swapping in a consistent method, we attain a randomly permute generator. If done correctly, it will generate random permutations with the even distribution. Thus the information will be permuted to prevent from identifying the accurate user.
VII. CONCLUSION

In an increasingly more data-driven society, personal information is often collected and distributed with ease. In this examination, we have presented an overview of recent technical advances in defining and protecting individual’s privacy and confidentiality in micro data publishing. We proposed the slicing method that analyzes the data characteristics and uses those characteristics in anonymization. In this method, we achieve better privacy and data utility by making use of correlation between attributes of the data. In particular, we have focused on organizations such as financial organizations, hospitals, and government agencies that compile huge data sets, and must balance the privacy of individual participants for which the aggregate data can be considered. While technology plays a critical role in individual privacy protection for personal data of organizations, it does not solve the problem in its entirety. While analyzing the other techniques, Slicing overcomes the limitations of other methods such as generalization and bucketization and also preserves better utility while protecting against privacy threats, consider slicing where each attribute is in accurately one column. An extension is the notion of overlapping slicing, that duplicates an attribute in more than one column. Another super advantage of slicing is that it can handle high-dimensional data.

VIII. ACKNOWLEDGMENTS

I would like to express my gratitude towards my guide Mrs. M.S Kavitha, assistant professor, Department in Of Computer Science and Engineering at Sri Eshwar College Of Engineering, for his kind co-operation and encouragement. I would like to thank my family as well, for their continuous support and encouragement.

IX. REFERENCE


