Bucketization based Flow Classification Algorithm for Data Stream Privacy Mining

G.Kesavaraj
Assistant Professor,
Department Of Computer Science and Applications,
Vivekanandha College Of Arts and Sciences For Women,
Tiruchengode,Namakkal-DT,TamilNadu,INDIA

S.Sukumaran Ph.D.,
Associate Professor
Department of Computer Science
Erode Arts and Science College,
Erode. Tamil Nadu,India

ABSTRACT
In recent years, data mining plays a major role in maintaining the huge volume of data from which it can derive the useful information. With the huge number of formation of data, the data wants to be lectured in a limit to the charge of growth. But it is complex to get over the set of meaningful information from the continuous set of data. Data-stream mining is a method which can discover important information from a huge contract of prehistoric data. For identification of useful information, the classification of continuous data streams is done. Current approaches in classifying the data streams are processed using supervised learning algorithms, which can be qualified with tagged data. Usually, manual classification of data is both expensive and time consuming. As a result, where massive amount of data emerge at a high speed, tagged data might be very sparse. Therefore, only a restricted amount of training data might be accessible for constructing the classification models, tend to badly trained classifiers. To overcome the issue, in this work, a novel technique is presented to build a classification set having both unlabeled and a small amount of labeled instances. This model is built by using the Flow Classification Algorithm (FCA). The FC algorithm is able to judge internally on set of marked data. Before classification, the correlation set of attributes in the each record set are grouped using bucketization technique. The superiority of models updated from them is enough for utilization of unlabeled records, or whether more set of labeled records are needed for classification is processed. Experimental evalaution is conducted to the proposed FC technqiue over its counterparts to find a set of diverse solution in terms of execution time, classification accuracy and security. Performance metrics for evaluation of proposed FCA technique shows that the security level is 10-15% high against existing work.

Keywords
Flow classification, bucketization, Frequent item sets.

1. INTRODUCTION
In today's framework, data mining has developed into a significant application owing to the large quantity of data and the crucial to haul out practical information from unrefined data. Numerous useful data prototypes can be chosen out, which assists expect effects of extraordinary scenarios. The information expanded from data mining can also be consequently utilized for diverse applications sorting from business organization to medicinal diagnosis. With process to data mining, main steps of KDD also comprise data cleaning, combination, collection, renovation, prototype evaluation, and knowledge production. Since data is normally speeded with missing values and noise, which creates them disjointed, data preprocessing has consequently turn out to be an significant step before data mining to progress the quality of the data.

Data stream has diverse uniqueness of data compilation to the conventional database representation. Such as the appointment of data stream incessant creation with time progresses and the data stream is active and the appearance of the data stream cannot be proscribed by the array. The data of Data stream can be interpret and route supported with the classification of arrival. The order of data cannot be misused to progress the outcome of conduct. Consequently, the processing of the data stream needs At first, every data element should be scrutinized at most one time, since it is idealistic to maintain the whole stream in the central memory. Second, every data element in data streams should be routed as fast as achievable. Third, the memory procedure for mining data streams should be enclosed even though new-fangled data elements are constantly produced. At last, the results produced by the online algorithms should be immediately accessible when user demanded.

Stream data classification is a demanding crisis as of two significant possessions: its infinite span and developing nature. Data streams might develop in numerous methods: the prior possibility allocation p(c) of a class c might vary, or the subsequent prospect allocation p (c|x) of the class might change, or both the prior and posterior possibilities might change. In either case, the confront is to construct a classification representation that is steady with the present notion. Conventional learning algorithms that need a number of passes on the training data cannot be openly functioned to the streaming situation, since the number of training instances would be infinite.

Manual cataloging of data is regularly expensive and time consuming. So, in a streaming atmosphere, where data emerge at a high speed, it might not be probable to physically tag all the data once they turn up. As a result, in practice, only a minute portion of each data chunk is expected to be tagged, leaving a main piece of the chunk as unlabeled. So, a very restricted quantity of training data will be accessible for the administered learning algorithms.

Published table, an effect of a data publishing algorithm terminates whether an isolation representation is correctly pleased or not. The privacy revelation devises a data publishing algorithm which is disparate by Algorithm in the structure of revelation. Eliminate the algorithm in the outline revelation to method and practical safe algorithm for privacy-

preserving data distributing. Exclusion of algorithm in the structure of disclosure guide a high level of value saved for the available table.

To start with, first the focus is made on classifying data stream and the occasion to examine each record using Flow Classification Algorithm. The algorithm is able to judge internally on set of marked data. The superiority of models updated from them is good enough for deployment on unlabeled records, or whether additional labeled records are required for classification is examined.

The rest of the paper is organized as follows: Section 2 describes the related literature works. Section 3 describes the Bucketization based FC algorithm technique. Section 4 describes the experimental evaluation and section 5 describes the performance results. Finally, section 6 ends with conclusion.

2. LITERATURE REVIEW

Current years have observed huge volumes of data to be composed on a huge size. Determined by common benefits, or by policies that need definite data to be available, there is a command for distributing unruffled data to the public. Data publishing has engendered much distress on individual privacy. Current work has revealed that diverse setting knowledge can carry a variety of threats to the seclusion of available data.

Statistical revelation power (also termed as privacy-preserving data mining) of micro-data is on discharging data sets with the answers of considered respondents confined in such a method that: (i) the respondents equivalent to the unrestricted records cannot be reproduced; (ii) the unconstrained data wait logically helpful. Generally, the confined data set is produced by either perturbing the unique data or by engendering simulated statistics of the creative data [2].

In privacy-preserving data mining , an extensively utilized technique for attaining data mining objectives as protecting privacy is supported with on k-anonymity. The most widespread technique for achieving observance with k-anonymity is to restore definite values with fewer specific but semantically reliable values. In [3], the author proposed a diverse technique for determining k-anonymity by splitting up the unusual dataset into numerous ridges such that all one of them sticks to k-anonymity. In [4], the author explained numerous features of the technique. First, numerous kinds of well-known data mining representations distributed a similar stage of representation quality over the geometrically disconcerted dataset as in excess of the creative dataset.

To execute PPRL, it is essential to affect string comparators that process in the privacy-preserving break. A number of privacy-preserving approaches have been presented, but slight research has evaluated them in the framework of a genuine record linkage request. The paper [5] performed a honorable and complete assessment of PPSCs in terms of three key possessions: Rightness of record connection calculation, computational difficulty, and safety. The general endeavor of the paper [6] is to expand a construction with algorithms and devices for privacy and security improved active data compilation, aggregation, and examination with advice loops.

In [7], the author commenced an adaptive fuzzy neural network frame for categorization of datastream utilizing a moderately managed learning algorithm. The structure comprises of a developing grainy neural network able of meting out non-stationary data streams utilizing a one-pass incremental algorithm. Recent representations of the classification trouble do not efficiently grip explodes of particular classes at diverse times. The representation for data stream classification analyzed the data stream categorization crisis from the point of view of a active technique in which concurrent training and test streams are utilized for active categorization of data sets [8].

Most existing data stream categorization methods disregard one significant feature of stream data: entrance of an original group. The author in [9] addressed a data stream categorization method that incorporates a new class discovery method into conventional classifiers, allowing automatic recognition of novel classes previous to the true tags of the novel class occurrences arrive.

In [10], the author presented an ensemble categorization structure, where each classifier is prepared with a new class detector, to speak to concept-drift and concept-evolution. The novel class recognition part is dogged by building it more adaptive to the developing stream and allowing it to notice more than one new class at a time. The paper [11] supported data stream categorization. At first, there is low error association and therefore high variety amongst component classifiers, which tends to elevated categorization correctness. The author in [12] measured the crisis of data stream categorization, where the data appear in a theoretically never-ending stream, and the chance to scrutinize each record is concise.

To enhance the privacy preservation of the data in the data stream, here, bucketization and flow classification algorithm is presented for the process of determining the labeled data in the dataset.

3. PROPOSED METHODOLOGY

The environment of data stream consists of diverse set of requirements for the process of determining the labeled data. The significant parts of the proposed data stream approach are defined as the following:

Practice the set of processed data one instance at a time,

Utilize a restricted amount of memory,

Process the data streams in limited time, and

Predict the charateristics of the data.

Consequently, data streams consist of numerous confronts for data mining algorithm design. The derived procedures should consume less number of resources to achieve the distribution of data streams over the set of data items.

The architecture of the proposed bucketization based FC algorithm is defined in fig 3.1.

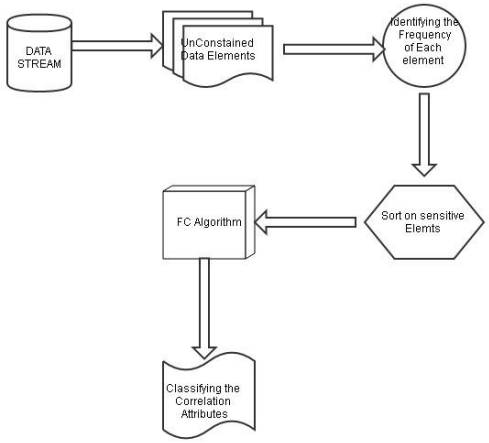

Fig 3.1:Process of bucketization based FC algorithm

From fig 3.1, it is being observed that the data stream classification is done by adapting the bucketization technique. With the Bucketing process, the data in the continuous data streams are grouped based on the frequency of occurrence of data in the dataset. After formation of group, the classfication under the data processing is done through the set of flow classification algorithm.

3.1 Data stream

Consider a set of items to be presented as, I={i1,i2,…,in} with n number of items. The set of items is presented as a subset of I. A transaction of the data in the data streams are defined as, T (id, iset), where id is the ID of the transaction, iset determined the set of items contained in the transaction. While performing the transaction of set of data streams, the data are sent in terms of infinite sequence of blocks referred as B1, B2,….,Bn, where each block is processed with a identifier. The length of the data stream is expressed as,

$$L\,(DS) = |B_1| + |B_2| + \ldots |B_N|$$

L- Length Bi i th Block

3.2 Formation of bucketized data from data streams

Bucketization is the process of defining the several records grouping based on their sensitive values. The apparent sensitive values of the attributes are identified and sorted based on the frequencies in ascending order. After sorting is done, the contiguous sensitive values are grouped into the similar bucket. Only the buckets contain at least ℓ distinct sensitive values which are kept after bucketing process completion. To add sensitive value, choose the relevant bucket to pick and processes it. After all sensitive values are bucketed; the larger buckets are splitted into set of smaller number of buckets. The splitting is order-preserving, i.e., the frequency of all impressionable values in the bucket is smaller than that of all sensitive values in the split bucket. After splitting, all the buckets are ordered by maximum frequency of their impressionable values in ascending order. At the end of the process, it returns a set of disjoint buckets and least ℓ distinct impressionable values.

The purpose of bucketizing the attributes is to establish the least cost splitting up of a multidimensional data set into B set of buckets (where the value of B is smaller than the number of data points in the dataset). The value of a bucket is calculated by the cost function. Naturally, the cost measure detains the compression of a bucket. A formation of bucketization is measured based on minimizing the average of total number of buckets of a bucket's weight (number of data points) and its perimeter (or sum of its extents along each dimension).

Consider a database DB with a set of tables T comprises with a set of records R. Each record consists of several set of attributes A with a set of values specified. Consider A = {a1, a2,…,an} be a set of attributes. Among these set, identify the sensitive attributes and grouped into a set of buckets B = {B1, B2,….,Bn}.

Table 1 Sample dataset

Name	Zip code	Age	Sex	Disease
Bob	14850	23	M	Flu
Charliey	14589	24	F	Flu
Dave	14863	25	M	Lung Cancer
Ed	14850	21	M	Lung Cancer
Frank	15478	27	M	Mumps
Gloria	14589	29	M	Flu
Hannah	14850	26	F	Flu

Table 1 describes the sample dataset comprises with set of sensitive and non-sensitive attributes. In the dataset, name, zip code, age, se are non-sensitive attributes. Disease is a sensitive attribute. With the set of sensitive attributes obtained, the buckets are created in which it arbitrarily produce each set of sensitive attribute values among each set of bucket formed.

Table 2 Bucketized table

Name	Zip code	Age	Sex	Disease
Bob	14850	23	M	Flu
Charliey	14589	24	F	
Gloria	14589	29	M	
Hannah	14850	26	F	
Dave	14863	25	M	Lung Cancer
Ed	14850	21	M	
Frank	15478	27	M	Mumps

Rather than specifying the sensitive attribute name as bucket name. Assign a name to sensitive values with a group name as group1, group2, and group n. Instead of putting flu, lung cancer, specify as group1, group2 and so on.

This bucketization process partitions the set of attributes in the dataset so that interrelated attributes are specified in the same column. This process is very adaptable for enhancing the privacy over the set of data in the dataset. In terms of data service, grouping the set of interrelated attributes conserves the associations amongst those attributes. Usually, the connection of uncorrelated set of attributes provides less security in data of the dataset than the association of interrelated attributes since the associations of uncorrelated attribute values is much less common and thus more exclusive.

3.3 Flow classification algorithm for bucketized data.

To achieve the privacy over the set of correlated set of attributes in the dataset, in this work, flow classification algorithm is presented. While applying the FC algorithm to

the bucketized data, the privacy level over the data is high. The flow classification algorithm is a decomposition based algorithm, which presents very high research throughput with low memory. It achieves the equivalent lookups on every individual field of correlated set of attributes first. This step is usually processed by a lookup table so as to attain the best throughput presentation. The table size with the set of attributes depends on the number of bits presented in header field.

The entry given in the lookup table with set of bucketized records (bu) practically accommodates the set of filters for which the record ri covers up the total number of records i. Actually, in a grouped set of records, each unique set of filters is assigned as index. The identifier of the determined set of attributes are formed as aid and utilized for data stream classification. The connection of all the record table lookups is closely the set of marked data that matches with a given set of packet identification. The group of records are processed and assigned with an id by presenting a lookup table. Yet again, for the throughput routine, this step is chosen to be processed from a direct table lookup. The process of flow classification is shown fig 3.2.

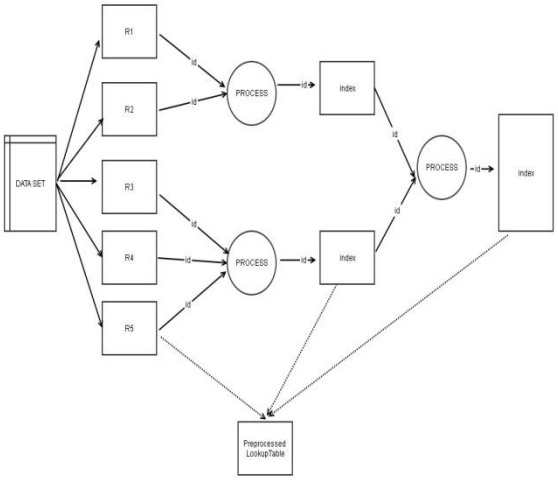

Fig 3.2: Process of FC algorithm

With the set of records maintained in the dataset, the bucketization of data is achieved through identifying and grouping the set of sensitive attributes. With the obtained set of grouped data, for enhancing the privacy, here, a FC algorithm is presented. The FC algorithm here will derive a lookup table for the set of each packet data in the dataset. By setting up the id for each type of data record value, the look up table is formed with the set of correlated attributes to process. Each column in the lookup table consists of bucket with values for each partitioned data. It is also provided a say that the occurrence of the value in each one of the FC algorithm checks the diversity of the correlated set of attribute value.

The pseudocode below describes the process of bucketization based FC algorithm.

*// **Bucketization***

Step 1: Extract the dataset from the database

Step 2: Divide the set of records present in the dataset.

Step 3: Identify the sensitive attributes

Step 4: Sort the remaining set of records based on the Occurrence of the sensitive attributes

Step 5: Group the similar set of records with set of buckets.

Step 6: Analyze the set of records in each bucket

Step 7: Compute the look up table

Step 8: Diversity maintains the set of buckets with matching attributes

Step 9: Combine the set of correlated attributes

Step 10: Display the set of secured data.

The objective of the FC algorithm is to ensure whether a lookup table satisfies bucketization i.e., l-diversity. For the presence of each set of attribute (a) in the group n, the FC algorithm sustains a list of matching buckets, in which, each element in the list comprise with one matching bucket b, with a matching probability and the allocation of sensitive values d (a,B). The FC algorithm scans each bucket b to trace the occurrence of each value v in bucket b.

To estimate the performance, the following metrics have been used to test it.

Execution time

The time taken to perform the bucketization process to transform the dataset into a bucketized table and then FC algorithm is applied. Finally, the classified set of data has been obtained. The total time taken to furnish the above said processes is defined in execution time.

Classification accuracy

Classification accuracy determines how all the correlated attributes in the dataset are effectively classified without any redundancy over the group formed.

Security

The security determines the privacy level of the classified data from the unauthorized access. It is measured in terms of percentage.

4. EXPERIMENTAL EVALUATION

An experimental evaluation is carried out to estimate the performance of the proposed bucketization based FC algorithm and is implemented in Java. The heart disease dataset is taken for experiments from UCI repository.

The heart disease dataset consists of 303 numbers of instances with 75 set of attributes. The associated set of tasks needed to process the dataset is defined as classification. The characteristics of attributes are determined as categorical, integer, real. This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. The attribute information is described in table 3.

Table 3 Attribute description

Attribute	Description
Age	age in years
Sex	sex (1 = male; 0 = female)
Cp	chest pain type
trestbps	resting blood pressure in mm Hg
chol	serum cholestoral in mg/dl
fbs	(fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
restecg	resting electrocardiographic results
thalach	resting electrocardiographic results
exang	exercise induced angina (1 = yes; 0 = no)
oldpeak	ST depression induced by exercise relative to rest
slope	the slope of the peak exercise ST segment
ca	number of major vessels (0-3) colored by flourosopy
thal	3 = normal; 6 = fixed defect; 7 = reversable defect
num	diagnosis of heart disease

The distribution of class under a set of instances is defined in table 4:

Table 4 Class distribution

Database	0	1	2	3	4	total
Cleveland	164	55	36	35	13	303
Hungarian	188	37	26	28	15	294
Switzerland	8	48	32	30	5	123
Long beach	51	56	41	42	10	200

With these set of data and classes, the experiments have been conducted for privacy preservation of patient data by adapting the bucketization based FC algorithm. The evaluation result will describe in the next section.

5. RESULTS AND DISCUSSION

The experiments have been evaluated and comparison results with the set of data are done with the existing DMPD approach [3]. The below table and graph describes the performance result of the proposed FC with the existing DMPD approach.

Table 5.1 No. of data vs. execution time

No. of data	Execution time (seconds)	
	FCA	DMPD
100	24	32
200	30	39
300	35	46
400	39	52
500	46	58
600	52	65

The execution time is measured based on the number of data present in the dataset. The value of the proposed bucketization based FC algorithm is compared with the existing DMPD approach illustrated in table 5. 1

Table 5.1 describes the execution time required to perform the classification to achieve the privacy based on the presence of the number of data in the dataset. Compared to the existing DMPD approach, the proposed bucketization based FC algorithm consumes less time to perform the classification even when number of data records increases in the dataset. In the proposed bucketization based FC algorithm, primarily transformed the original dataset into the bucketized set of data to achieve the classification in a minimal interval of time. But in the existing, only with the set of GA process, the classification is directly applied. So it consumes more time. The variance in the execution time is 10-12% less in the proposed bucketization based FC algorithm.

Table 5.2 No. of correlated attributes vs. classification accuracy

No. of correlated attributes	Classification accuracy (%)	
	based FCA	DMPD
10	56	24
20	63	30
30	68	36
40	72	45
50	76	56
60	82	67

The classification accuracy is measured based on the number of correlated attributes present in the dataset. The value of the proposed bucketization based FC algorithm is compared with the existing DMPD approach illustrated in table 5.

Table 5.2 describes classification accuracy is measured based on the number of correlated attributes present in the dataset. Compared to the existing DMPD approach, the proposed bucketization based FC algorithm has highest level of accuracy to perform the classification even when number of correlated set of attributes increases in the dataset. In the proposed bucketization based FC algorithm, primarily transformed the original dataset into the bucketized set of data to achieve the classification more accurate. But in the existing, only with the set of GA process, the classification is directly applied. So the accuracy is not defined in the set of attributes. The variance in the execution time is 7-10% high in the proposed bucketization based FC algorithm.

Table 6 No. of data vs. security

No. of data	Security (%)	
	FCA	DMPD
100	55	41
200	62	45
300	67	49
400	73	52
500	78	56
600	85	62

The security is measured based on the number of data present in the dataset. The value of the proposed bucketization based FC algorithm is compared with the existing DMPD approach illustrated in table 6.

Table 6 describes the security is measured based on the number of data present in the dataset. Compared to the existing DMPD approach, the proposed bucketization based FC algorithm achieved high level of security since it consumes less time and the high accuracy in classification.

Finally, it is being observed that the proposed bucketization based FC algorithm done the classification more accurate with the set of data in the dataset.

6. CONCLUSION

A novel bucketization based FC technique is presented in this work to achieve the classification with the set of data in the

data stream. At first, the data in the dataset is bucketized by obtaining the set of buckets using bucketization technique. After bucketized set of data obtained, the FC algorithm is applied to claasifiy the data. Experimental evaluation is done with the heart disease dataset to analyze the performance. Performance results revealed that the proposed bucketization based FC Technique provides 13% accuracy in the classified data by consuming minimal interval of time compared to the existing DMPD approach.

7. REFERENCES

[1] Hui Wang, Ruilin Liu, "Privacy-preserving publishing micro-data with full functional dependencies", Elseiver Data & Knowledge Engineering 70 (2011) 249–268

[2] Josep Domingo-Ferrer, Ursula Gonzalez-Nicolas, "Hybrid microdata using microaggregation", Elseiver Information Sciences 180 (2010) 2834–2844.

[3] Nissim Matatov et. Al., "Privacy-preserving data mining: A feature set partitioning approach", Information Sciences 180 (2010) 2696–2720.

[4] Keke Chen, Ling Liu, "Geometric Data Perturbation for Privacy Preserving Outsourced Data Mining", ieee transactions knowledge and data engineering, 2012.

[5] Elizabeth Durham et. Al., "Quantifying the correctness, computational complexity, and security of privacy-preserving string comparators for record linkage", Information Fusion 13 (2012) 245–259.

[6] Li Xiong et. Al., "PREDICT: Privacy and Security Enhancing Dynamic Information Collection and Monitoring", International Conference on Computational Science, ICCS 2013.

[7] Leite, D.,Costa, P.;Gomide, F., "Evolving granular neural network for semi-supervised data stream classification", International Joint Conference on Neural Networks (IJCNN), 2010

[8] Aggarwal, C.C.,Jiawei Han;Jianyong Wang;Yu, P.S., "A framework for on-demand classification of evolving data streams", IEEE Transactions on(Volume:18 ,Issue: 5) Knowledge and Data Engineering, 2006.

[9] Masud, M.M.;et. Al., 'Classification and Novel Class Detection in Concept-Drifting Data Streams under Time Constraints', IEEE Transactions on(Volume:23 ,Issue: 6) Knowledge and Data Engineering, 2011.

[10] Masud, M.M. et. Al., 'Classification and Adaptive Novel Class Detection of Feature-Evolving Data Streams', IEEE Transactions on(Volume:25,Issue: 7) Knowledge and Data Engineering, 2013.

[11] Hashemi, S, Ying Yang;Mirzamomen, Z.;Kangavari, M., "Adapted One-versus-All Decision Trees for DataStream Classification",IEEE.Transactionson (Volume:21, Issue:5) Knowledge and Data Engineering, 2009

[12] Abdulsalam, H. et. Al., "Classification Using Streaming Random Forests", IEEE Transactions on Knowledge and Data Engineering, (Volume:23 ,Issue: 1), 2011.

[13] Ki-Seung Lee, "SNR-Adaptive Stream Weighting for Audio-MES ASR", IEEE Transactions on (Volume:55 ,Issue: 8) Biomedical Engineering, 20088.

ABOUT AUTHORS

Dr. S. Sukumaran graduated in 1985 with a degree in Science from Bharathiar University, Coimbatore. He obtained his Master Degree in Science and M.Phil in Computer Science also from the Bharathiar University. He received the Ph.D degree in Computer Science from the Bharathiar University. He has 25 years of teaching experience starting from Lecturer to Associate Professor. At present he is working as Associate Professor of Computer Science in Erode Arts and Science College, Erode, Tamilnadu. He has guided for more than 40 M.Phil research Scholars in various fields and guided one Ph.D Scholar. Currently he is Guiding 5 M.Phil Scholars and 8 Ph.D Scholars. He is member of Board studies of various Autonomous Colleges and Universities. He published around 15 research papers in national and international journals and conferences. His current research interests include Image processing and Data Mining.

G.Kesavaraj received B.Sc Computer science and M.Sc Computer Science Degree from Bharathiar University, Coimbatore. He pursuing Ph.D degree in Computer Science from the Manonmaniam Sundaranar University. He has 10 years of teaching experience. He is working as Assistant Professor of Computer Science in Vivekanandha College of Arts and Science for Women, Tiruchengode Namakkal DT, Tamilnadu. His research interests include Data Mining and Software Engineering