ISSN 2047-3338

Spatial Mining Process – Study on Implementation of Instance in Database Engine

Beeram Sathyanarayana[1], Paleti Lakshmi Kanth[2] and M. Srikanth[3]
[1]Kallam Haranadha Reddy Institute of Technology, India
[2]Vignan's Lara Institute of Technology and Science, Vadlamudi, Guntur, India
[3]Department of Computer Science and Engineering in RVR&JC College of Engineering, India

***Abstract*– In this article presenting the environment of spatial data mining and classifications of spatial databases to improve the quality in spatial objects that are spatial clustering, classification, and association. Framework for spatial data mining formulate the visibility nearest neighbor graphs, relations and objects differing in how to prune instances during the search process. We further propose the implementation of spatial database engine using the eleven spatial instances, here we describes only seven which finds the type of instance in different geometry or geography analysis. To design the spatial database engine requires the data types and measurements are analyzed, our work shows not only for mining process but also to implement the spatial objects in SQL specification version 1.1.0.**

***Index Terms*– Spatial Classifications, Outlier, Data Types and Measurements**

I. INTRODUCTION

SPATIAL data mining is the discovery of interesting relationships and characteristics that may exist implicitly in spatial databases. Spatial data mining aims to automate such a knowledge discovery process (Gueting R. H et al., 1994).

It plays an important role in: a) extracting interesting spatial patterns and features; b) capturing intrinsic relationships between spatial and non-spatial data; c) presenting data regularity concisely and at conceptual levels; and d) helping to reorganize spatial databases to accommodate data semantics, as well as to achieve better performance. Spatial data mining has deep roots in both traditional spatial analysis fields (such as spatial statistics, analytical cartography, and exploratory data analysis) and various data mining fields in statistics and computer science (such as clustering, classification, association rule mining, information visualization, and visual analytics).

Its goal is to integrate and further develop methods in various fields for the analysis of large and complex spatial data. Not surprisingly, spatial data mining research efforts are often placed under different umbrellas, such as spatial statistics, geocomputation, geovisualization, and spatial data mining, depending on the type of methods that a research focuses on. Data mining and knowledge discovery is an iterative process that involves multiple steps, including data selection, cleaning, preprocessing, and transformation; incorporation of prior knowledge; analysis with computational algorithms and/or visual approaches, interpretation and evaluation of the results; formulation or modification of hypotheses and theories; adjustment to data and analysis method; evaluation of result again; and so on (Fayyad et al., 1996). Data mining and knowledge discovery is exploratory in nature, more inductive than traditional statistical methods. It naturally fits in the initial stage of a deductive discovery process, where researchers develop and modify theories based on the discovered information from observation data (Miller & Han, 2009).

SECTION II

A) Spatial Classification

Spatial supervised classification is about grouping data items into categories according to their attribute values. ‘‘Supervised” classification needs a training dataset to the classification model, a validation dataset to optimize the configuration, and a test dataset to evaluate the performance of the trained model. Supervised methods include, for example, decision trees, artificial neural networks (ANN), maximum likelihood estimation (MLE), linear discriminant function (LDF), support vector machines (SVM), nearest neighbor methods and case-based reasoning (CBR). Spatial classification methods extend the general-purpose classification methods to consider not only attributes of the object to be classified but also the attributes of neighboring objects and their spatial relations (Ester et al., 1997; Koperski et al., 1998). A visual approach for spatial classification was introduced in (Andrienko & Andrienko, 1999), where the decision tree derived with the traditional algorithm C4.5 (Quinlan, 1993) is combined with map visualization to reveal spatial patterns of the classification rules.

Decision tree induction has also been used to analyze and predict spatial choice behaviors (Thill & Wheelerm, 2000). Artificial neural networks (ANN) have been used for a broad variety of problems in spatial analysis (Fischer, 1998; Fischer et al., 2003; Gopal, Liu and Woodcock, 2001; Yao & Thill, 2007). Remote sensing is one of the major areas that

commonly use classification methods to classify image pixels into labeled categories e.g., (Cleve et al., 2008).

Spatial prediction models form a special group of regression analysis that considers the independent and/or dependent variable of nearby neighbors in predicting the dependent variable at a specific location, such as the spatial autoregressive models (SAR) (Anselin et al., 2006; Cressie, 1983; Pace et al., 1998). However, spatial regression methods such as SAR often involve the manipulation of an n by n spatial weight matrix, which is computationally intensive if n is large. Therefore, more recent research efforts have sought to develop approaches to find approximate solutions for SAR so that it can process very large data sets (Griffith, 2004; Kazar, Sheikh Lilja et al., 2004; Smirnov & Anselin, 2001).

B) Spatial Association

Spatial association rule mining was originally intended to discover regularities between items in large transaction databases (Agrawal et al., 1993). Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items i.e., items purchased in transactions such as computer, memorycard, milk, bread etc.). Let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. Let X be a set of items and a transaction T is said to contain X if and only if $X \subseteq T$. An association rule is in the form: X=>Y, where $X \subseteq I, Y \subseteq I$ and $X \cap Y = \phi$. The rule X => Y holds in the transaction set D with confidence c if c% of all transactions in D that contain X also contain Y. The rule X => Y has supports in the transaction set D if s% of transactions in D contain $X \cup Y$. Confidence denotes the strength and support indicates the frequencies of the rule. It is often desirable to pay attention to those rules that have reasonably large support (Agrawal et al., 1993). Similar to the mining of association rules in transactional or relational databases, spatial association rules can be mined in spatial databases by considering spatial properties and predicates (Appice, Ceci, Lanza, Lisi, & Malerba, 2003; Han & Kamber, 2001; Koperski & Han, 1995; Mennis & Liu, 2005).

A spatial association rule is expressed in the form A) B [s%, c%], where A and B are sets of spatial or non-spatial predicates, s% is the support of the rule, and c% is the confidence of the rule.

C) Spatial Clustering

Spatial clustering, regionalization and point pattern analysis Cluster analysis is widely used for data analysis, which organizes a set of data items into groups (or clusters) so that items in the same group are similar to each other and different from those in other groups (Gordon, 1996; Jain & Dubes, 1988; Jain et al., 1999). Many different clustering methods have been developed in various research fields such as statistics, pattern recognition, data mining, machine learning, and spatial analysis. Clustering methods can be broadly classified into two groups: partitioning clustering and hierarchical clustering. Partitioning clustering methods, such as K-means and self-organizing map (SOM) (Kohonen, 2001); divide a set of data items into a number of non-overlapping clusters. A data item is assigned to the ‘‘closest” cluster based on a proximity or dissimilarity measure. Hierarchical clustering, on the other hand, organizes data items into a hierarchy with a sequence of nested partitions or groupings (Jain & Dubes, 1988). Commonly-used hierarchical clustering methods include the Ward’s method (Ward, 1963), single-linkage clustering, average-linkage clustering, and complete-linkage clustering (Gordon, 1996; Jain & Dubes, 1988).

To consider spatial information in clustering, three types of clustering analysis have been studied, including spatial clustering i.e., clustering of spatial points, regionalization (i.e., clustering with geographic contiguity constraints), and point pattern analysis (i.e., hot spot detection with spatial scan statistics). For the first type, spatial clustering, the similarity between data points or clusters is defined with spatial properties (such as locations and distances). Spatial clustering methods can be partitioning or hierarchical, density-based, or grid-based. Readers are referred to (Han et al., 2001) for a comprehensive review of various spatial clustering methods.

D) Spatial Outlier Analysis

Shekhar, Lu and Zhang (2003) define a spatial outlier as a spatially-referenced object whose non-spatial attributes appear inconsistent with other objects within some spatial neighborhood. Note that, unlike spatial outliers, this definition does not imply that the object is significantly different than the overall database as a whole: it is possible for a spatial object to appear consistent with the other objects in the entire database but nevertheless appear unusual with a local neighborhood.

They develop a unified modeling framework and identify efficient computational structures and strategies for detecting these types of spatial outliers based on a single (non-spatial) attribute. More generally, geographic objects can also exhibit unusual spatial properties such as size and shape. Ng (2001) uses distance-based measures to detect unusual paths in two-dimensional space traced by individuals through a monitored environment. These measures allow the identification of unusual trajectories based on entry/exit points, speed and geometry; these trajectories may correspond to unwanted behaviors such as theft.

SECTION III

A) Spatial Data Mining Framework

Spatial data mining is based on spatial neighbourhood relations between objects and on the induced neighbourhood graphs and neighbourhood paths which can be defined with respect to these neighbourhood relations. Thus, a set of database primitives or basic operations for spatial data mining are introduced, which are sufficient to express most of the spatial data mining Algorithms. Similar to the relational standard language SQL, the use of standard primitives will speed-up the development of new data mining algorithms and will also make them more portable. Second, develop techniques to efficiently support the proposed database primitives (e.g., by specialized index structures) thus speeding-up all data mining algorithms which are based on our database primitives. Moreover, basic operations for spatial data mining can be integrated into commercial database management systems.

This will offer additional benefits for data mining applications such as efficient storage management, prevention

of inconsistencies, index structures to support different types of database queries which may be part of the data mining algorithms.

B) Spatial Neighborhood Relations, Spatial Neighborhood Graphs and their Operations

Spatial data mining are based on the concepts of neighborhood graphs and neighborhood paths which in turn are defined with respect to neighborhood relations between objects. There are three basic types of spatial relations: topological, distance and direction relations which may be combined by logical operators to express a more complex neighborhood relation. Spatial objects such as points, lines, polygons or polyhedrons are all represented by a set of points.

For example, a polygon can be represented by its edges) or by the points contained in its interior, e.g., the pixels of an object in a raster representation.

Topological relations: are based on the boundaries, interiors and complements of the two related objects and are invariant under transformations which are continuous, one-one, onto and whose inverse is continuous. The relations are: *A disjoint B*, *A meets B*, *A overlaps B*, *A equals B*, *A covers.*

Fig. 1 shows the Illustration of some topological distance and direction relations *B*, *A* covered-by *B*, *A* contains *B*, *A* inside *B*. A formal definition has been given by Egenhofer in 1991. Relations compare the distance of two objects with a given constant using one of the arithmetic comparison operators. If *dist* is a distance function, σ is one of the arithmetic predicates <, > or = , and *c* is a real number, then a distance relation *O1* distance c *O2* between the two spatial objects *O1* and *O2* holds if distance(*O1*, *O2*) σ *c*. To define the direction relations, e.g. *O2* south *O1*, we consider one representative point of the object *O1* as the origin of a virtual coordinate system whose quadrants and half-planes define the directions. To fulfill the direction predicate, *all points* of *O2* have to be located in the respective area of the plane.

Fig. 1 illustrates the definition of some direction relations using 2D polygons. Obviously, the directions are not uniquely defined but there is always a smallest direction relation for two objects *A* and *B*, called the *exact direction relation* of *A* and *B*, which is uniquely determined. In Fig. 2, for instance, *A* and *B* satisfy the direction relations *northeast* and *east* but the exact direction relation of *A* and *B* is *northeast*. By combining basic spatial relations via logical operators it is possible to define more complex spatial relations, e.g. "*O1* is north of *O2* and no more than 5 km away". Each such spatial relation induces a spatial neighborhood graph as defined in the following definition.

Definition 1: (*neighborhood graphs and paths*)
Let *neighbor* be a neighborhood relation and *DB* be a database of spatial objects. A n*eighborhood graph* is a graph with nodes *N* = *DB* and edges where an edge e = (*n1*, *n2*) exists if *neighbor(n1,n2)* holds. A *neighborhood path* of length *k* is defined as a sequence of nodes [*n1, n2, . . ., nk*], where *neighbor* (*ni*, *ni*+1) holds for all . We assume the standard operations from relational algebra like *selection, union*, *intersection* and *difference* to be available for sets of objects and sets of neighborhood paths (e.g., the operation *selection (set, predicate)* returns the set of all elements of a *set* satisfying the *predicate*). In addition, we introduce some operations which are specific to neighborhood graphs and paths and which are designed to support spatial data mining.
The following operations are briefly described:
neighbors: Graphs × Objects × Predicates → Sets_of_objects
paths: Sets_of_objects → Sets_of_paths;
extensions: Graphs × Sets_of_paths × Integer × Predicates → Sets_of_paths

The operation neighbors *(*graph*,* object*,* predicate*)* returns the set of all objects connected to *object* in *graph* satisfying the conditions expressed by the predicate. The operation paths(objects*)* creates all paths of length 1 formed by a single element of objects and the operation extensions(graph, paths, length, predicate*)* returns the set of all paths of the specified length in graph extending one of the elements of paths. The extended paths must satisfy the predicate predicate. The elements of *paths* are not contained in the result implying that an empty result indicates that none of the elements of paths could be extended. Because the number of neighbourhood paths may become very large, the argument predicate in the operations *neighbours* and *extensions* acts as a filter to restrict the number of neighbours and paths to certain types of neighbours or paths.

The definition of predicate may use spatial as well as non-spatial attributes of the objects or paths. For the purpose of KDD "leading *away*" from the start object. Conjecture that a spatial KDD algorithm using a set of paths which are crossing the space in arbitrary ways will not produce useful patterns. The reason is that spatial patterns are most often the effect of some kind of influence of an object on other objects in its neighborhood. Furthermore, this influence typically decreases or increases more or less continuously with increasing or decreasing distance. To create only "relevant" paths, special filter predicates which select only particular subsets of all paths, i.e., paths which are "*leading away*" from the start objects in a certain sense.

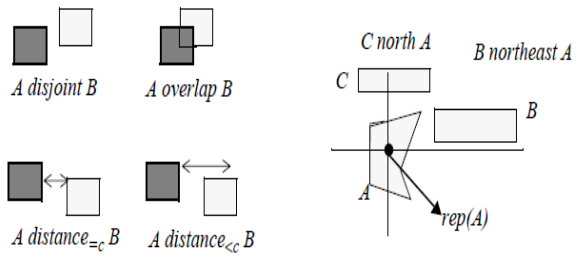

Fig. 1. Topological distance and direction relations

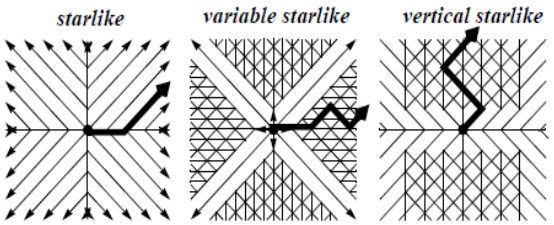

Fig. 2. The Illustration of some filter predicates

This approach also significantly reduces the runtime of the spatial data mining algorithms operating on neighborhood paths.

There are different possibilities to define filters for paths "*leading away*" from start objects. The filter starlike, e.g., is a very restrictive filter which allows only a small number of "coarse" paths. This filter is appropriate for many applications, and in terms of runtime it is most efficient. It requires that, when extending a path $p = [n1,n2,...,nk]$ with a node $nk+1$, the exact "final" direction of p may not be generalized. For instance, a path with final direction *northeast* can only be extended by a node of an edge with the exact direction northeast. The filter variable-starlike allows more "finegrained" paths by requiring only that, when extending p the edge $(nk, nk+1)$ has to fulfill at least the exact "initial" direction of p. For instance, a neighborhood path with initial direction north can be extended such that the direction *north* or the more special direction northeast is satisfied. The filter variable-starlike allows a more detailed spatial analysis than the filter *starlike*, but it increases the runtime of a data mining algorithm because more paths have to be processed by the algorithm.

Fig. 2 illustrates these filters when extending the paths from a given start object. It also depicts another filter vertical starlike which is less restrictive in vertical than in horizontal direction. This filter is appropriate when the vertical direction should be analyzed in greater detail than the horizontal direction.

IV. IMPLEMENTING SPATIAL DATABASE ENGINE

Spatial data represents information about the physical location and shapes of geometric objects (e.g point locations or countries, roads or lakes, etc). There are two types of spatial data: geometry data type supports the Euclidean (flat-earth) data, conforms to the open geospatial consortium features for SQL specification version 1.1.0.

Geometry and geography data types support seven spatial data instances, are instant able we can create and work with these instances in a database. These instances derive certain properties from their root data types that distinguish them as points, Line strings, Polygons, Multiplegeometry or geography instances in a Geometry Collection.

The Fig. 3 depicts the geometry hierarchy upon which the geometry and geography data types are based. The instant able types of geometry and geography are indicated in colour blue are point, multipoint, linestring, multilinestring, polygon,

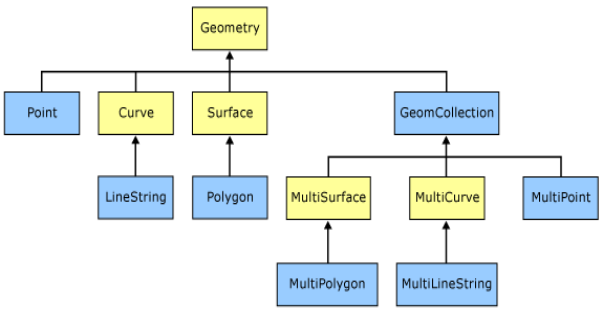

Fig. 3. The seven instance of geometry and geography data

multipolygon and geometry collection. The geometry and geography types can recognize a specific instance as long as it is a well-formed instance, even if the instance is not defined explicitly using the ST point form text() method, geometry and geography recognize the instance as a point as long as the method input is well-formed. If you define the same instance using the STGeom From text() method, both the geometry and geography data types recognize the instance as a point.

Clear idea about the spatial data types are:

A) Geography Data Type

The Point type for geography data type represents a single location where x and y represent longitude and latitude values respectively. The values for longitude and latitude are measured in degrees. Values for longitude always lie in the interval (-180, 180*)* and values inputted outside this range are wrapped around to fit in this range. For example, if 190 is inputted for longitude then it will be wrapped to the value -170. Values for latitude always lie in the interval [-90, 90] and values that are inputted outside this range will throw an exception.

A Multipoint is a collection of zero or more points. The boundary of a Multi Point instance is empty.

A Line String is a one-dimensional object representing a sequence of points and the line segments connecting them.

Fig. 4 contains examples of Line String 1 is a simple non-closed Line String instance, 2 is a non-simple, non-closed Line String instance, 3 is a closed simple Line String is a ring, 4 is also a closed non-simple Line String instance ring. A Multi Line String is a collection of more geometry or geography Line String instances.

A Polygon is a two-dimensional surface stored as a sequence of points defining an exterior bounding ring and zero or more interior rings.

A Polygon instance can be formed from a ring that has at least three distinct points. A Polygon instance can also be empty. The exterior and any interior rings of a Polygon define its boundary. The space within the rings defines the interior of the Polygon.

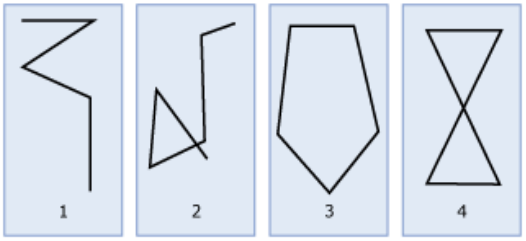

Fig. 4. Examples of Line String

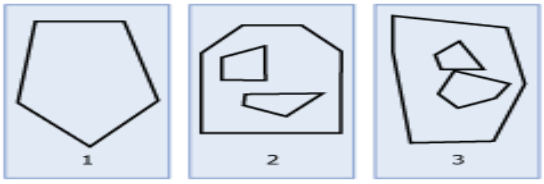

Fig. 5. Polygon instance

Fig. 5 depicts the 1 polygon instance whose boundary is defined by an exterior ring, 2 is a polygon instance whose boundary is defined by exterior ring and two interior rings, area inside the interior rings is part of the exterior of polygon instance and 3 is a valid polygon instance its interior rings intersect at a single tangent point.

A Multi polygon is a collection of more polygon instance.

B) Connecting Edges in Geometry

The defining data for Line String and Polygon types are vertices only. The connecting edge between two vertices in a geometry type is a straight line. However, the connecting edge between two vertices in a geography type is a short great elliptic arc between the two vertices. A great ellipse is the intersection of the ellipsoid with a plane through its center and a great elliptic arc is an arc segment on the great ellipse.

C) Measurements in Spatial Data Types

In the planar, or flat-earth, system, measurements of distances and areas are given in the same unit of measurement as coordinates. Using the geometry data type, the distance between (2, 2) and (5, 6) is 5 units, regardless of the units used. In the ellipsoidal or round-earth system, coordinates are given in degrees of latitude and longitude. However, lengths and areas are usually measured in meters and square meters, though the measurement may depend on the spatial reference identifier (SRID) of the geography instance. The most common unit of measurement for the geography data type is meters.

D) Orientation of Spatial Data

In the planar system, the ring orientation of a polygon is not an important factor. For example, a polygon described by ((0, 0), (10, 0), (0, 20), (0, 0)) is the same as a polygon described by ((0, 0), (0, 20), (10, 0), (0, 0)). The OGC Simple Features for SQL Specification does not dictate a ring ordering, and SQL Server does not enforce ring ordering. In an ellipsoidal system, a polygon has no meaning, or is ambiguous, without an orientation. For example, does a ring around the equator describe the northern or southern hemisphere? If we use the geography data type to store the spatial instance, we must specify the orientation of the ring and accurately describe the location of the instance.

SQL Server 2008 places the following restrictions on using the geography data type:

Each geography instance must fit inside a single hemisphere. No spatial objects larger than a hemisphere can be stored.

Any geography instance from an Open Geospatial Consortium (OGC) Well-Known Text (WKT) or Well-Known Binary (WKB) representation that produces an object larger than a hemisphere throws an Argument Exception.

The geography data type methods that require the input of two geography instances, such as STIntersection(), STUnion(), STDifference(), and STSymDifference(), will return null if the results from the methods do not fit inside a single hemisphere. STBuffer() will also return null if the output exceeds a single hemisphere.

E) Outer and Inner Rings Not Important in Geography Data Type

The OGC Simple Features for SQL Specification discusses outer rings and inner rings, but this distinction makes little sense for the SQL Server geography data type: any ring of a polygon can be taken to be the outer ring.

V. CONCLUSION

In this paper we clearly explain the spatial data mining for implementing the SQL data base engine, the spatial data design is having the two types of data types which are geometry or geography and geospatial. The geospatial information is open SQL specification version 1.1.0, geometry or geography spatial instances are collection of geometrics. Our analysis also represents the classifications of data mining which are applied for mining the process of data instances in spatial data bases like location, viewpoints, shapes, the spatial framework analysis finds the nearest neighbor visibility. This work describes mining activities and the different types measurements, data types in spatial data analysis.

REFERENCES

[1]. Znselin, L., Syabri, I., & Kho, Y. (2006). GeoDa: An introduction to spatial data analysis. Geographical Analysis, 38(1), 5–22.

[2]. Appice, A., Ceci, M., Lanza, A., Lisi, F. A., & Malerba, D. (2003). Discovery of spatial association rules in geo-referenced census data: A relational mining approach. Intelligent Data Analysis, 7(6), 541–566.

[3]. Bailey, T. C., & Gatrell, A. C. (1995). Interactive spatial data analysis. New York, NY, John Wiley and Sons, Inc Brimicombe, A. J. (2007). A dual approach to cluster discovery in point event data sets. Computers Environment and Urban Systems, 31(1), 4–18.

[4]. Cheng,T., & Wang, J. (2009). Accommodating spatial associations in DRNN for space–time analysis. Computers, Environment and Urban Systems, 33(6), 409–418

[5]. Cleve, C., Kelly, M., Kearns, F. R., & Morltz, M. (2008). Classification of the wildland– urban interface: A comparison of pixel- and object-based classifications using high-resolution aerial photography. Computers Environment and Urban Systems, 32(4), 317–326.

[6]. Guha, S., Rastogi, R., Shim, K., 1998. CURE: Efficient Clustering Algorithms for Large Databases. Proc. ACM SIGMOD Int. Conf. on Management of Data. Seattle, WA, pp.73-84.

[7]. Gueting R.H., 1994. An Introduction to Spatial Database Systems. The VLDB Journal 3(4), pp. 357-399.

[8]. Agrawal, Imielinski, Swami. Mining Association Rules between Sets of Items in Large Databases

[9]. Ester, Frommelt, Kriegel, Sander. Spatial Data Mining: Database Primitives, Algorithms and Efficient DBMS Support

Beeram Sathyanarayana Reddy an associate professor and Head of the Department for Computer Science and Engineering in Kallam Haranadha Reddy Institute of Technology affiliated to JNTU, Kakinada. He received his M.Tech degree in C.S.E. from acharya Nagarjuna University in the year 2008. He is an active member of C.S.I. since 2009. His research interests include cryptography and network security; data ware housing and data mining and image processing.

Paleti Lakshmi Kanth is currently working as Assistant Professor in Department of Information Technology, Vignan's Lara Institute of Technology and Science, Vadlamudi, Guntur. He received Masters of Technology degree from JNTU, Kakinada in Computer Science and Engineering. His research interests include Embedded Systems, Data Mining, Artificial Neural Networks, Network Security and Image Processing.

M. Srikanth an Assistant Professor in the Department of Computer Science and Engineering in RVR&JC College of Engineering. He received his M.Tech degree in C.S.E. from Satyabhama University. He is an active member of CSI since 2009. His research interests include data ware housing and data mining, image processing, cryptography and network security.