International Journal On Engineering Technology and Sciences – IJETS™
ISSN (P): 2349-3968, ISSN (O): 2349-3976
Volume 1 Issue 6, October 2014

Spatio-Temporal Query Processing on Weighted Timestamp Data Environment

Ms. P.Maheswari
II Year M.E(CSE)
Shree Venkateshwara Hi-Tech Engg College, Gobi
mahe11191@gmail.com

Dr. T. Senthil Prakash
Professor & HOD
Shree Venkateshwara Hi-Tech Engg College, Gobi
jtyesp@yahoo.co.in

Ms. N. Narmatha
II Year M.E(CSE)
Shree Venkateshwara Hi-Tech Engg College, Gobi
narmatha.nallasamy@gmail.com

ABSTRACT- *Durable query scheme is used to find objects in historical time series databases. Durable queries can be extended for multidimensional time series analysis. Aggregate scoring function is used to rank the series at every time instance in top-k search. Euclidean distance measure is used in KNN queries at each timestamp between the reference series. The durable top-k (DTop-k) query operates on a historical database of 1D time series. Query window and durability threshold are used to retrieve objects in Dtop-K query model. Durable k nearest neighbor (DkNN) query considers at each time moment the k nearest neighbors of a reference series. Top-k event scanning (TES) is used to detect actions that are related to the time limit. TES indexes the timestamp and incrementally computes the snapshot top-k sets at each timestamp of the query window. Query Space Indexing(QSI) indexes the query space. The time series data analysis scheme is improved with noise elimination tasks. The durable query scheme is enhanced with weight based timestamp analysis process. Durable KNN query model is extended for time weight analysis mechanism.*

Keywords: *Timestamp, durable top-k, Top-k event scanning, Query Space Indexing*

1 INTRODUCTION

Spatio-temporal data mining is an emerging research area dedicated to the development and application of novel computational techniques for the analysis of large spatio-temporal databases. The main impulse to research in this subfield of data mining comes from the large amount of

- spatial data made available by GIS, CAD, robotics and computer vision applications, computational biology, and mobile computing applications;
- temporal data obtained by registering events (e.g., telecommunication or web traffic data) and monitoring processes and workflows.

Both the temporal and spatial dimensions add substantial complexity to data mining tasks.

First of all, the spatial relations, both metric (such as distance) and non-metric (such as topology, direction, shape, etc.) and the temporal relations (such as before and after) are information bearing and therefore need to be considered in the data mining methods.

Secondly, some spatial and temporal relations are implicitly defined, that is, they are not explicitly encoded in a database. These relations must be extracted from the data and there is a trade-off between precomputing them before the actual mining process starts (eager approach) and computing them on-the-fly when they are actually needed (lazy approach). Moreover, despite much formalization of space and time relations available in spatio-temporal reasoning, the extraction of spatial/temporal relations implicitly defined in the data introduces some degree of fuzziness that may have a large impact on the results of the data mining process.

Thirdly, working at the level of stored data, that is, geometric representations (points, lines and regions) for spatial data or time stamps for temporal data, is often undesirable. For instance, urban planning researchers are interested in possible relations between two roads, which either cross each other, or run parallel, or can be confluent, independently of the fact that the two roads are represented by one or more tuples of a relational table of Blines'' or Bregions''. Therefore, complex transformations are required to describe the units of analysis at higher conceptual levels, where human-interpretable properties and relations are expressed.

Fourthly, spatial resolution or temporal granularity can have direct impact on the strength of patterns that can be discovered in the datasets. Interesting patterns are more likely to be discovered at the lowest resolution/granularity level [2]. On the other hand, large support is more likely to exist at higher levels.

Fifthly, many rules of qualitative reasoning on spatial and temporal data (e.g., transitive properties for temporal relations after and before), as well as spatiotemporal ontologies, provide a valuable source of domain independent knowledge that should be taken into account when generating patterns. How to express these rules and how to integrate spatio-temporal reasoning mechanisms in data mining systems are still open problems.

2 RELATED WORKS

a Consistent and Durable Queries

Lee et al. [4] were the first to study the consistent top-k query, which is the special case of DTop-k with the durability threshold r fixed to 100 percent. In the example of consistent top-2 query with time period (0, 5) retrieves only object s2. The basic idea of the solution is to exhaustively verify every object in the data set against the query definition. For each object s, LHL first checks whether s belongs to the top-k set at timestamp t_b. If so, LHL continues to check if s is a top-k object at t_{b+1}; otherwise, it discards s and starts with another object. The process continues, until either s is eliminated, or after checking the rank of s at every timestamp in the query window. To accelerate snapshot top-k membership checking, LHL pre computes the rank of each object at every timestamp, and organizes this information into a sorted list, stored on disk in a compressed format. For instance, LHL associates the list (1, 2, 2, 3, 3) to object s_1, signifying that s1 ranks first at timestamp 0, second at time 1-2, and third at time 3-4. During query processing, LHL scans the rank list of an object linearly from t_b, until reaching either te or a value larger than k.

LHL does not support DTop-k queries with r < 100%. To handle such cases, we extend LHL as follows: For each object s, we scan the part of its rank list from timestamp tb to te, and count the number of times that s is in the snapshot top-k sets. During the scan, if we find that s is outside the top-k set for more than (1-r). (t_e t_b) timestamps, we drop s since it cannot possibly reach the durability threshold r. The set of objects that pass the verification are reported as results. The main drawback of LHL is that it scales poorly with the number of objects, as each object initiates a list scan with at least one I/O read.

U et al. [10] studied durable top-k queries in the context of keyword search in web archives, where each object is a web document that gets edited or replaced over time. In addition to the parameters k, (t_b, t_e), and r, a durable query also involves a keyword list K_W. The score of a document version is calculated based on its relevance to the keywords in K_W with an IR model. There are important differences between our work. First, computing the relevance of a document to an arbitrary K_W is both hard and expensive. Therefore, preprocessing methods cannot be used to accelerate search in our work. Second, the data domain, i.e., versional documents, is quite special: the relevance of keywords to documents remains relatively constant in adjacent timestamps. When this assumption does not hold, for example, if all objects change values at every timestamp methods reduce to brute-force search. Hence, the solutions are tailored to a specific domain, and are not suitable for DTop-k queries in the general case.

b Temporal Queries

Numerous solutions (e.g., [1]) have been proposed for indexing time series to support similarity search. Such queries retrieve time series that are closest to a reference series, according to a certain distance measure. Two popular distance measures are 1) the euclidean distance in the space defined by considering each time instance as a dimension and 2) dynamic time warping (DTW) which improves robustness over the euclidean distance by allowing mapping of shifted sequence elements. The Gemini framework addresses the dimensionality curse in time-series indexing and search using dimensionality reduction; popular methods in this direction include Chebyshev polynomials, piecewise linear approximation [5], APCA and so on. These methods do not apply to durable queries, as they focus on an object's overall similarity to a query, rather than their properties at individual timestamps.

There is also a vast amount of existing work in indexing and searching trajectories, where each record contains the locations of a moving object at different timestamps. TrajStore [6] is a full-featured storage engine for trajectories using adaptive quad-tree indices. Sherkat and Rafiei [8] propose a new class of robust summaries for highdimensional time series. Chen et al. propose a new distance measure for multidimensional time series, as well as the corresponding indexing and searching methods. Another line of work focuses on spatiotemporal queries on moving objects trajectories. Yu et al. propose efficient methods for continuous nearest neighbor search, which continuously updates the nearest neighbor of a query as objects update their locations. Guting et al. [9] study a similar problem, termed TCkNN, but focus on retrieving the nearest neighbors during a historical period rather than the current ones. Specifically, a TCkNN query finds, at each timestamp during the given period, the NN of a

International Journal On Engineering Technology and Sciences – IJETS™
ISSN (P): 2349-3968, ISSN (O): 2349-3976
Volume 1 Issue 6, October 2014

reference trajectory. The technical focus is to organize trajectories into an R-tree-like structure and then take advantage of some pruning heuristics.
Compared to similarity queries, there is little work on top k queries for time series data, despite the importance of such queries. Although it is possible to simulate a top-k query by a k-NN query with an imaginary reference time series that has the largest domain value at each time moment, such a reduction is often "far from satisfactory"; methods designed for similarity search do not capture well the unique properties of top-k search. Li et al. [7] conducted a thorough study on the evaluation of snapshot top-k queries on continuous time series with a piecewise linear representation. The focus is clearly different from ours, both in terms of the query nature and the data model used.
Jestes et al. [12] study aggregate top-k queries on temporal data with a piecewise linear representation. The goal is to find the top-k objects with the highest aggregation scores in a given time interval. The focus and the data model are clearly different from ours, and their solutions do not apply to durable queries. Another piece of related work is the interval skyline query [3]. An object si dominates another sj, if an only if si is better than sj in at least one timestamp, and no worse in all other timestamps. The set of objects that are not dominated is then reported as the interval skyline. The interval skyline and the durable top-k, however, retrieve very different results. The former's result set includes objects with high values in a small number of timestamps, whereas the latter identifies objects with durable quality. For instance, s_1 is on the interval skyline as long as the query window contains timestamp 0 (where s_1 is the best object), regardless of its scores in other time instances. Consequently, the solutions are inapplicable to our problems. The probabilistic top-k query finds objects with high probability to be in the top-k set, is also remotely related to this work, since one can view each timestamp as a possible world, and calculate the probability for each object. On the other hand, the focus is clearly different from ours, and their methods do not apply to durable queries.
Finally, recent-biased time series have been studied in the context of online analysis of streaming data. The focus of this work is different, however, since we focus on offline queries over historical data. For instance, in the various application scenarios, it is generally more natural to consider timestamps within the query window as equally important, than giving higher weights to more recent time instances. For this reason, in the following, we focus on the equal-weight time series model, as is done in many existing work involving historical data, for example, [12].

3 DURABLE QUERIES ON SPATIO TEMPORAL DATA

The top-k query which selects k best objects based on their ranking scores, is a common approach to obtaining a small set of desirable objects from a large database. Recently, top-k search has been extended to databases that contain multiple versions of data objects, for example, web archives, trajectory data, time series and so on. Ranked retrieval in such applications may need to consider not only an object's value at one particular time instance, but also its overall quality during a time period [9], [10].
In this paper, we study in depth the problem of finding objects of consistent quality during a time interval. We first study the durable top-k (DTop-k) query that operates on a historical database where each object is a 1D time series, i.e., at each time instance, every series carries a single scalar. Given k, time interval ½tb; teÞ (called the query window), and percentage $0 < r \leq 1$ (called the durability threshold), a DTop-k query retrieves objects that appear in the snapshot top-k sets for at least [r ‘ (t_e = t_b)] timestamps during [t_b, t_e]. An example with four series s1- s4. Assuming higher scores are preferred, a durable top-2 query with [t_b, t_e) = [0; 4), r = 70% retrieves s1 and s2, since they appear in the top-2 set in at least 70 percent timestamps during (0, 4).
We also identify a natural extension of the DTop-k query: the durable k nearest neighbor (DkNN) query, which considers at each time moment the k nearest neighbors of a reference series sref . Consider again the example and the DkNN query with sref = s4, k = 1, [t_b, t_e) = [0, 4), r = 70%; i.e., we are interested in the sequences that are the nearest neighbor of s4 on at least 70 percent of the timestamps 0-3. The only series that qualifies this query is s3, since it is the NN of s4 75 percent of the time in (0, 4). As we show in the paper, the DkNN query is much more challenging compared to DTop-k, since the former is rather resistant to materialization and indexing.
Durable queries are useful in many real-world applications. For example, consider Google Zeitgeist, which presents weekly statistics of search keywords, each of which is associated with a time series of its search volumes. A DTop-k (resp. DkNN) query can be used to identify keywords that are frequently searched (resp. most related) during some time period, which may be further used by sociologists to understand the impact of certain historical events. A similar application is Twitter Trendsmap, which tracks frequently mentioned phrases and hashtags. In SciScope, a geospatial search engine built upon a wide-area sensor network, durable queries may be used by meteorologists to identify regions with consistently high environmental indices in particular

time windows. In general, durable queries may serve as fundamental tools in timeseries analysis; domain experts can use their results to better understand their data and trigger further investigation. We demonstrate some interesting examples in Section 6.

Durable queries can naturally be extended for multidimensional time series, where at each time moment every series carries an array of values. For top-k search, to rank the series at every time instance, we need to define an aggregate scoring function on the values in the individual dimensions (e.g., a linear function). For NN queries, we use a distance measure (e.g., euclidean distance) at each timestamp between the reference series sref and the sequences; for example, a police officer may investigate on vehicles consistently moving close to a pivot, for example, a suspect or a witness. An example 2D time series data set containing the positions of three moving objects s1-s3 at timestamps 0-3. Considering a DkNN query with k = 1 and a period (0, 4), object s1 satisfies the query for $r \leq 75\%$, since it is the snapshot NN of sref for timestamps 1-3.

To our knowledge, currently there is a very narrow selection of solutions for the DTop-k query, and no previous work on the DkNN query. The only existing solutions for DTop-k (reviewed in Section 2) employ either brute-force search, or techniques that are limited to specific domains. To fill this gap, we propose an efficient method called top-k event scanning (TES). TES exploits the fact that real-world time series typically exhibit a certain degree of smoothness, meaning that the changes in the top-k set at adjacent timestamps are usually small, if at all. TES indexes these changes and incrementally computes the snapshot top-k sets at each timestamp of the query window. To efficiently support DkNN queries on 1D time series, we extend the methodology of TES, and propose an efficient solution, query space indexing (QSI), that indexes the query space. Going one step further, we extend QSI to handle multidimensional top-k and k-NN queries. Extensive experiments using real and synthetic data confirm that the proposed methods significantly outperform previous ones, often by large margins.

4 PROBLEM STATEMENT

Durable query scheme is used to find objects in historical time series databases. Durable queries can be extended for multidimensional time series analysis. Aggregate scoring function is used to rank the series at every time instance in top-k search. Euclidean distance measure is used in KNN queries at each timestamp between the reference series. The durable top-k (DTop-k) query operates on a historical database of 1D time series. Query window and durability threshold are used to retrieve objects in Dtop-K query model. Durable k nearest neighbor (DkNN) query considers at each time moment the k nearest neighbors of a reference series. Top-k event scanning (TES) is used to detect actions that are related to the time limit. TES indexes the timestamp and incrementally computes the snapshot top-k sets at each timestamp of the query window. TES scheme is extended with Query Space Indexing (QSI) model to support DKNN queries on 1D time series. Query space indexing (QSI) indexes the query space. QSI is extended to handle multidimensional top-k and k-NN queries. The following problems are identified from the existing system.

- Timestamp weights are not considered
- Pruning strategies are not used for timestamp analysis
- Time weight analysis is not carried out in DKNN query
- Noise elimination is not performed

5 DURABLE TOPK AND DURABLE KNN QUERIES

a Durable TopK Queries

This section focuses on DTop-k processing in settings where each object s is associated with a single value (i.e., its score) at each timestamp t. In other words, the top-k scores of the objects at all timestamps are known before query time. In practice, the value of k is usually only a fraction of the total number of objects in the database. Hence, we use k_{max} to denote the largest supported value of k in the target application. For the ease of presentation, we assume that all series in the data set are sufficiently long to cover the query window, i.e., each of them has a value at every timestamp during (t_b, t_e). If a series starts after tb or terminates before t_e, we simply put a (conceptual) value of 1 on each of its undefined timestamps. Meanwhile, we use $\Delta min = [r. (t_e - t_b)]$ to denote the minimum number of timestamps for which an object should satisfy the corresponding snapshot top-k query to appear in the DTop-k results. Besides LHL described another naıve solution (referred to as NAI) for the DTop-k query is to compute the snapshot top-k results at every timestamp, and report the objects that appear in no less than Δmin snapshot top-k sets. Clearly, this technique has a high cost when the query window is long. In the following, we present a novel solution TES that significantly outperforms both LHL and NAI.

b Durable KNN Queries

This section studies the evaluation of DkNN queries on 1D time series. Recall from Section 1 that a DkNN query contains a reference series sref , which has te tb values, one for each timestamp in the query window [11]. Hence, there is a vast space of possible DkNN queries, making effective materialization much more difficult. A naı¨ve approach (referred to as NAI) is to simply compute the snapshot k-NN results at every timestamp, and combine them to answer the DkNN query. NAI is clearly inefficient, because 1) the k-NN set may not change at every timestamp and 2) snapshot k-NN computations are expensive as they cannot be precomputed as in the DTop-k case. In the following, we describe a novel solution, namely query space indexing, which indexes the results of all possible DkNN queries, and stores them compactly.

6 SPATIO-TEMPORAL QUERY PROCESSING ON WEIGHTED TIMESTAMP DATA

The time series data analysis scheme is improved with noise elimination tasks. The durable query scheme is enhanced with weight based timestamp analysis process. The system is enhanced with pruning strategies to fetch data in required levels. Durable KNN query model is extended for time weight analysis mechanism. The system is designed to perform query processing on spatio temporal databases. The query processing scheme is improved with noise reduction mechanism. Timestamp weight assignment mechanism is used to analyze the temporal relationships. The system is divided into six major modules. They are spatio temporal database, timestamp weight analysis, query window analysis, indexing process, event detection using durable queries and event detection using weighted durable queries.

The historical data values are managed in spatio temporal database module. Timestamp weight analysis module is designed to assign weight values for timestamp data. Query window analysis is performed to separate data based on query boundaries. Index process module is used to arrange the query window data values. Event detection is performed using durable queries with DTopK and DKNN methods. Event detection with weight information is carried out with timestamp weight values.

a Spatio Temporal Database

Spatio temporal database is used to maintain the location and time details. Traffic information are stored in the spatio temporal database. Location information are updated with distance details. User and activity details are updated in the database.

b Timestamp Weight Analysis

Timestamp indicates the date and time of activity. Time values are divided into different slots. Fixed and variable intervals are used in the slots. Weight values are assigned with time priority information.

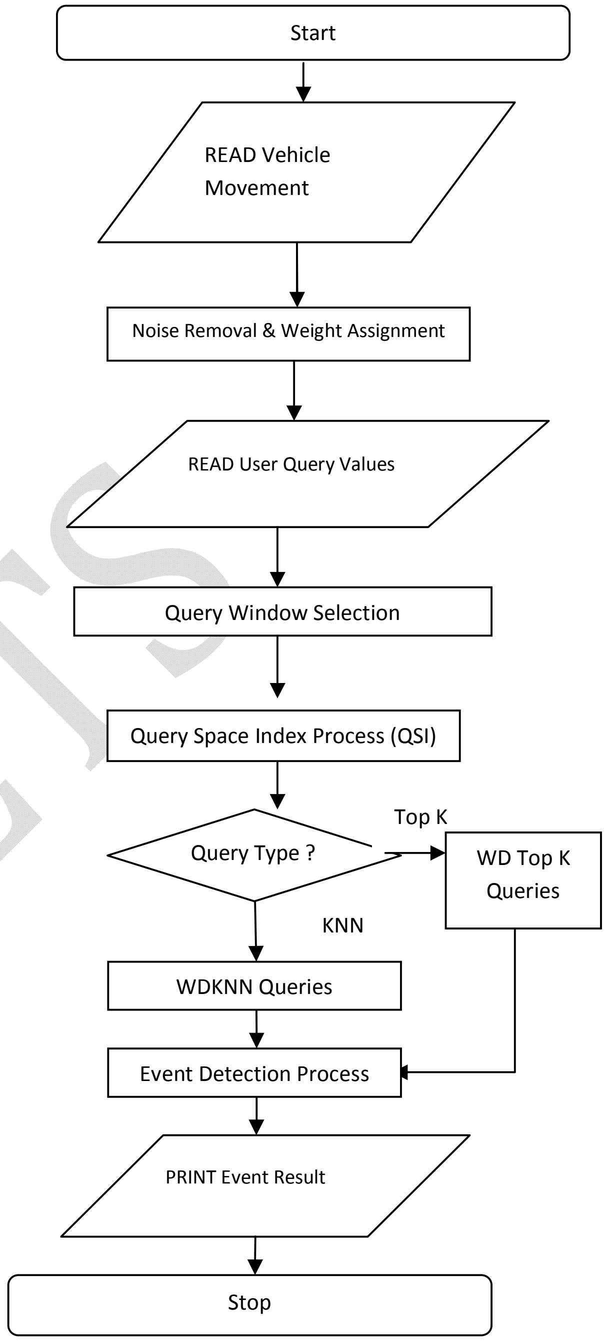


Fig. No:1 Spatio-Temporal Query Processing on Weighted Timestamp Data

c Query Window Analysis

Query window is the partitioned database of spatio temporal database. Query window is constructed with time boundary values. Time boundary values are collected from the user. Spatio temporal database is divided with query boundary levels.

d Indexing Process

Indexing process is applied to arrange the activity details. Similarity analysis is used for the indexing process. Indexing process is applied on the query window. Query Space Indexing (QSI) scheme is used for the indexing process.

e Event Detection using Durable Queries

Event detection process is performed on the indexed data values. Durable Top K (DTopK) query scheme is used to select top level events. Durable K Nearest Neighbor (DKNN) search scheme is used to select similar data values. Events are identified with similarity and threshold values.

f Event Detection using Weighted Durable Queries

Timestamp weight values are used in the weighted durable query process. Weighted Durable Top K (WDTopK) query scheme is used to discover events with top priority. Weighted Durable K Nearest Neighbor (WDKNN) search scheme is used to search similar activities with timestamp weight values. Noise reduction process is used to improve the accuracy query results.

7 CONCLUSION

Durable queries are used for historical time series analysis. Durable top-k (DTop-k) and nearest neighbor (DkNN) queries are used analyze time stamped sequences of values or locations. Indexing and query evaluation techniques are used to improve the DTop-k and DkNN queries. The durable query scheme is enhanced with noise elimination and weight based timestamp analysis mechanism. The system supports high scalability in spatio temporal analysis. Query retrieval accuracy is improved in the system. One dimensional and multi dimensional data query are supported by the system. Weight based timestamp analysis mechanism is used to improve the event detection accuracy.

REFERENCES

[1] V. Athitsos, P. Papapetrou, M. Potamias, G. Kollios, and D. Gunopulos, "Approximate Embedding-Based Subsequence Matching of Time Series," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2008.

[2] Sajib Barua and Jörg Sander, "Mining Statistically Significant Co-location and Segregation Patterns", IEEE Transactions On Knowledge And Data Engineering, Vol. 26, No. 5, May 2014

[3] B. Jiang and J. Pei, "Online Interval Skyline Queries on Time Series," Proc. IEEE Int'l Conf. Data Eng. (ICDE), 2009.

[4] M.L. Lee, W. Hsu, L. Li, and W.H. Tok, "Consistent Top-K Queries over Time," Proc. 14th Int'l Conf. Database Systems for Advanced Applications (DASFAA), 2009.

[5] Q. Chen, L. Chen, X. Lian, Y. Liu and J.X. Yu, "Indexable PLA for Efficient Similarity Search," Proc. 33rd Int'l Conf. Very Large Data Bases (VLDB), 2007.

[6] P. Cudre´-Mauroux, E. Wu, and S. Madden, "TrajStore: An Adaptive Storage System for Very Large Trajectory Data Sets," Proc. IEEE 26th Int'l Conf. Data Eng. (ICDE), 2010.

[7] F. Li, K. Yi, and W. Le, "Top-k Queries on Temporal Data," VLDB J., vol. 19, pp. 715-733, 2010.

[8] R. Sherkat and D. Rafiei, "On Efficiently Searching Trajectories and Archival Data for Historical Similarities," Proc. VLDB Endowment, vol. 1, no. 1, pp. 896-908, 2008.

[9] R.H. Gu¨ ting, T. Behr, and J. Xu, "Efficient K-Nearest Neighbor Search on Moving Object Trajectories," VLDB J., vol. 19, pp. 687-714, 2010.

[10] L.H. U, N. Mamoulis, K. Berberich, and S. Bedathur, "Durable Top-K Search in Document Archives," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2010.

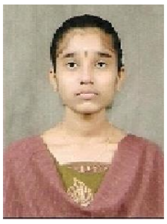
[11] Yinan Jing, Ling Hu, Wei-Shinn Ku and Cyrus Shahabi "Authentication of k Nearest Neighbor Query on Road Networks", IEEE Transactions On Knowledge And Data Engineering, Vol. 26, No. 6, June 2014

[12] J. Jestes, J.M. Phillips, F. Li, and M. Tang, "Ranking Large Temporal Data," Proc. VLDB Endowment, vol. 5, pp. 1412-1423, 2012.

Ms.P.Maheswari pursuing M.E(CSE) degree in Shree Venkateshwara Hi-Tech Engineering College,Erode,India in 2014 and B.E(CSE) degree from Kathir College Of Engineering,Coimbatore,India in 2013.she has published 3 National conferences, 6 workshops.Her research interests include data mining,databases.She is a Member of Computer Society Of India(CSI).

Dr.T.Senthil Prakash received the Ph.D. degree from the PRIST University, Thanjavur, India in 2013 and M.E(CSE) degree from Vinayaka Mission's University, Salem , India in 2007 and M.Phil.,MCA.,B.Sc(CS) degrees from Bharathiyar University, Coimbatore India, in 2000,2003 and 2006 respectively, all in Computer Science and Engineering. He

is a Member in ISTE New Delhi, India, IAENG, Hong Kong..IACSIT, Singapore SDIWC, USA. He has the experience in Teaching of 10+Years and in Industry 2 Years. Now He is currently working as a Professor and Head of the Department of Computer Science and Engineering in Shree Venkateshwara Hi-Tech Engineering College, Gobi, Tamil Nadu, and India. His research interests include Data Mining, Data Bases, Artificial Intelligence, Software Engineering etc.,He has published several papers in 17 International Journals, 43 International and National Conferences.



Ms.N.Narmatha pursuing M.E(CSE) degree from Shree Venkateshwara Hi-tech Engineering college, Erode, India in 2014 and B.Tech(IT) degree from Nandha Engineering college, Erode, India in 2011.she has attended national conference on MLP based intrusion detection using neural network and one workshop. Her research interests include data mining, software testing.