International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 02 Issue: 04 | June-2015 www.irjet.net p-ISSN: 2395-0072

Handling Multiple K-Nearest Neighbor Query Verifications on Road Networks under Multiple Data Owners

S.Susanna[1], Dr.S.Vasundra[2]

[1] *Student, Department of CSE, JNTU Anantapur, Andhra Pradesh, India*

[2] *Professor, Department of CSE, JNTU Anantapur, Andhra Pradesh, India*

Abstract- ***The emergence of mobile devices with fast Internet connectivity and geo-positioning capabilities has led to a revolution in customized location-based services (LBS), where users are enabled to access information about points of interest (POI) that are relevant to their interests and are also close to their geographical coordinates. Probably the most important type of queries that involve location attributes is represented by nearest-neighbor (NN) queries, where a user wants to retrieve the k POIs (e.g., restaurants, museums, gas stations) that are nearest to the user's current location (kNN).This paper provides an efficient road network based k-nearest-neighbor query authentication technique which uses the neighbor's verification and network Voronoi diagram to prove the integrity of query results. Unlike previous work that verifies k-nearest-neighbor results in the Euclidean space, to do so, the existing approach [1] needs to verify both the shortest path and the distances from the query point to its kNN results on the road network. Further the technique handles the multiple data owners where the neighbours are divided into three groups and the data accessing is restricted within these groups. In-order to get the query executed the client has to go through an authentication process.***

Key Words: *Nearest Neighbour, Verification, Authentication, Voronoi Diagram, Road Network*

1. INTRODUCTION

In recent years, database outsourcing has gained tremendous popularity. In order to reduce operation and maintenance costs, an existing content distribution network such as an ad-hoc P2P/Grid computing environment may be used for database outsourcing. Database outsourcing involves three types of entities: data owners, service providers, and users. A data owner outsources its database functionality to one or more third parties which are called service providers (e.g., a cloud computing service) which have the computational power to support various query processing. Users issue their queries to the service providers. Database outsourcing has several advantages:

1) As the data owners store their data on the service providers, they do not need to have their own facilities to store and process the data.
2) Using third party service providers is a cheaper way to achieve scalability than fortifying the owner's data center and providing more network bandwidth for every user.
3) The database outsourcing model removes the single point of failure in the owner's data center, hence reducing the databases susceptibility to denial of service attacks and improving service availability.
4) The users can get the query results by a service provider which is close in terms of network latency without need to contact the data owners directly.

However, even though database outsourcing has several advantages, it poses several security challenges because we cannot completely trust the third party service providers which can be corrupted by adversaries.

- The first challenge is privacy. For instance, In an application to find nearby friends, the server stores the locations of the friends. If the location database is outsourced but not properly protected, unauthorized users may gain data access causing privacy bleaches to the data owners. In addition, not only the data stored in the server provider but also the query issued to the service provider is sensitive information that should be protected since the service provider can know the location of the users. In other application areas, user queries may disclose

private details about users such as shopping habits, political or religious affiliation, etc.

- The second challenge is authentication. In outsourced databases, the data owners delegate their database functionality like range query, kNN, proximity, top-k, SUM, etc to the service providers. If the service providers are compromised, they could return tampered results to the users. Authenticated query processing techniques guarantee authenticity and completeness of query results in outsourced systems. Authenticity ensures that all the results returned to users originate from the data owners and no spurious results are introduced. Completeness guarantees that all the results which satisfy the query are present in the result set. On the other hand, authentication can be used for location based access control. Location based access control is to give an access to an important information when a user is in a restricted area. In order to determine whether the user is in the restricted area, we can make the user to receive partial keys from several Location- based Service (LBS) devices only when the user is in the area. Then, when the keys are authenticated, the user is given an access to the information.
- The third challenge is recovery. Several protocols acknowledge the above authentication issue and provide authentication in the presence of malicious service providers. All these protocols deal with stealthy attacks where the malicious service providers try to modify the result without being detected. Such techniques can verify whether the result is correct or not, and in case they detect that the result has been tampered with, they raise an alarm. However, they cannot pinpoint the source of the attack. Hence, they cannot identify and remove the malicious service providers, leaving the network vulnerable to denial-of-service attacks. So, when the results are not correct, we need to detect the malicious service providers and give the correct results in the next round by excluding them.

2. RELATED WORK

- Most similarity searching algorithms may be distilled to a simple formula applicable to kNN Query Processing or index structures. Further, range queries may be viewed as a special case of nearest neighbor queries where the final search radius is known at the start of the search. As The following algorithms perform similarity searches for the nearest neighbor to a query based on an index structure of fixed dimensionality.
- Generally, the search for a single nearest neighbor may be expanded to find k nearest neighbors by maintaining a list of neighbors found and using the distance between the neighbor that occupies the k^{th} distance related slot and the query point as a search range/radius value to search within. Each of the index structures described in the following sections may generally be used in conjuncture with one of the following search methods. However, developers will typically modify the algorithm to better suit the applicable structure.

2.1 Exact Searching Methods

- The most basic method to perform a k-NN similarity search involves using a range search algorithm. Begin with radius $r = \alpha : (\alpha > 0)$ centered at query point q. Increase α until at least k elements lie within the radius. The cost, in terms of page accesses, of this algorithm is similar to that of performing a range search. This cost however is greatly affected by the amount the value α is adjusted by every time the desired number of elements is not yet found. Too small, the performance cost will quickly grow; too large, the number of points returned will far exceed the desired number thus decreasing the usefulness of the solution.
- A more elegant approach was proposed for both general metric spaces and continuous data spaces (CDSs) [2, 3, 4] by beginning the search on any data structure using $r = \infty$. The search begins at the anchor/root of the data structure. Each time the query point q is compared to some element p, the search radius is updated such that
- $r = \min(r, D(q, p))$.
- The search then backtracks to the nearest split in the index where a new path has not yet been traversed and continues down the new path using the newly calculated radius. As the search continues, the possibility increases that an entire path of the index structure may not need to be searched due to the decreasing size of the radius r. Pruning heuristics are employed to determine if

a certain route is no longer necessary to search. Roussopoulos et al. [4] provided two heuristics for CDS based index structures, namely R-trees, named MINDIST and MINMAXDIST that invoke this pruning. MINMAXDIST is used in the breadth search of the covering rectangle of the current search path by determining the minimum maximum distance of all covering rectangles contained in the current one. This distance is used to update the current search radius such that r = min(r,MINMAXDIST).

- MINDIST is then used in the depth traversal to determine if any point of a covering rectangle associated with a new search path is within the search radius. If no point is within the search radius, that search path is pruned from the remainder of the search and thus the effective size of the database to search is decreased. This range reduction method may be improved further by attaining a smaller radius value earlier in the search. Several techniques have been used in both CDSs and general metric spaces [5, 6, 7]. The underlying idea of each technique is that certain paths may be identified by their high level statistics that will yield a closer point to the query point q earlier in the search. The most common application of this idea is to order potential search paths by either their MINDIST or MINMAXDIST values to q.
- The MINDIST ordering gives the optimistic approach that a lower MINDIST value is caused by a relatively closer object in the index structure. This may not always prove true in spatial index structures. Commonly, some point of a search path only exists at the top most layers. At higher levels within an index structure, points may actually be the result of the intersection of lines drawn from several points lower in the index structure. When this technique appears to suffer from this problem, the pessimistic approach using MINMAXDIST may be used instead. Here, search paths are ordered by the increasing value of their furthest point. Thus a search may correctly assume that it will at least not encounter any points further away than the MINMAXDIST.

2.2 Approximate Searching Methods

- Relaxing the precision of the query results may lead to even further reductions in time complexity. This is a reasonable procedure for many applications due to some approximation in the modalization of feature vectors for both general metric and CDS indexes. In addition to the query itself, a user specifies some query parameter to control how far away from the query point the search may progress. In this manner, the algorithm avoids the costly initial stages of a similarity search. On subsequent searches of similar databases, may decrease to approach zero. As decreases, the time complexity, along with the precision of the result decreases as well. A probabilistic algorithm was given for vector spaces by Yianilos et al. [8], using a method described as aggressive pruning to improve the performance. Here, the idea is to increase the number of branches that are pruned at the expense of possible points in the set of nearest neighbors. This process is controlled such that the probability of success is always known. Unfortunately, the data structure used is only useful in a very limited radius; in databases or with searches that could result in neighbors with distances beyond the possible radius, the algorithm is not able to guarantee a result of the true nearest neighbors. This form of similarity searching is described as Approximate Similarity Searching. This topic is not covered in detail in this paper, but is mentioned here for completeness. An in depth coverage may be found in [9].

2.3 Unique Searching Methods

- The techniques described thus far cover universal proposals for performing k-NN similarity searches. There are however examples of k-NN search algorithms developed for specific indexes that are inapplicable in a generic sense. Such algorithms depend upon the structure developed to support them and are unable to be incorporated with common indexing techniques. Clarkson [10] proposes a method that alleviates the need to perform extensive backtracking by creating a GNAT-like data structure where points are inserted into multiple subtrees. The tree is constructed by first selecting representatives for the root(s) and then inserting each element u into not only the subtree of its closest representative

p, but also into the subtree of any other representative p 0 such that

- D(u, p0) ≤ 3 ∗ D(u, p).
- During a query on object αq, the search enters all subtrees such that
- D(αq, p0) ≤ 3 ∗ D(αq, p).
- As shown by Clarkson, this is enough to guarantee the retrieval of the nearest neighbor and could be extended to determine the set of k-NN.

3. PROPOSED MODEL

The proposed techniques contain the following functionalities.

- Data Owner / Client
- Server
- Group Manager

The data owner plays the part of owner as well the Client as he/she will be sending queries to other users in the same group.

The process of authentication starts with the registration of the owners where they have to enter the personalized password for getting in to the owner dialog.

The model of the system is depicted in figure 1.

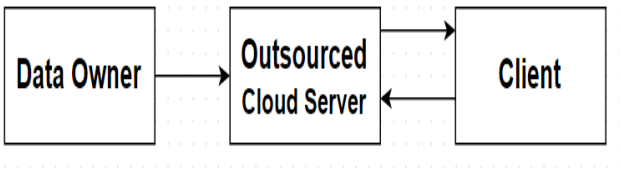


Fig-1: System Model

The system model comprises of three distinct entities: (1) the data owner; (2) the outsourced cloud service provider (for short cloud server, or simply server); and (3) the client.

The data owner has a dataset with n two-dimensional points of interest, but does not have the necessary infrastructure to run and maintain a system for processing nearest-neighbor queries from a large number of users. Therefore, the data owner outsources the data storage and querying services to a cloud provider. As the dataset of points of interest is a valuable resource to the data owner, the storage and querying must be done in encrypted form.

3.1. Privacy Model:

As mentioned previously, The normal road network define the distance between two points is measured by the road network distance instead of their Euclidean distance, assuming objects can only move along street systems. A road network system can be modeled as a weighted graph G(V,E,W) consisting of a set of vertices V ={p1, p2, . , pn} and a set of edges E ={e1, e2, ., em} connecting vertices to form a graph. W represents the cost of each edge in E

Figure 2 illustrate the original road network is represented as a graph where p1, p2, and p3 are points of interest and p4-p16 are intersections on the road network.*p*1, *p*2, and *p*3 are points of interest and *p*4-*p*16 are

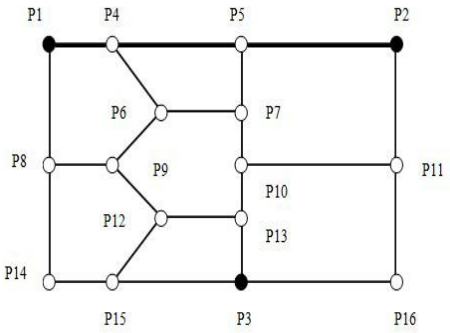


Fig-2: Road network

intersections on the road network. The network Voronoi cell *V(pi)* contains all points on edges that are closer to *pi* than to any other POIs. It is actually a shortest path tree generated from *pi*, and hence *pi* is also called the *generator* of *V(pi)*. Note that, different from the Voronoi Diagram in the Euclidean space where each Voronoi cell is a continuous area, the network Voronoi cell of each generator contains a set of road segments.

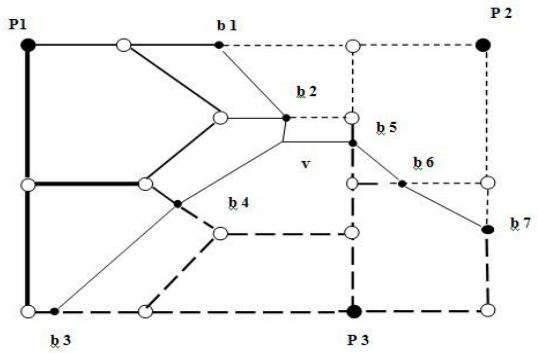


Fig-3: Network voronoi diagram

In figure 3 *V (p1)*, *V (p2)*, and *V (p3)* are represented by line segments with different styles separated by point's *b*1-*b*7. Since objects are restricted on road segments, the network Voronoi cell of a specific generator is unique. Given a set of points of interest, one can construct the network Voronoi diagram by expanding shortest path trees from each POI simultaneously until the shortest path trees meet. The meeting points, termed as *border points*, are also on the edges of the road network with the property that the costs (e.g., road network distances) from the meeting point to the two neighboring POIs are equal to each other.

This paper also implemented the multi data owner structure so that the data owners can be active simultaneously and the owners are divided into the groups and that is depicted in the below figure.

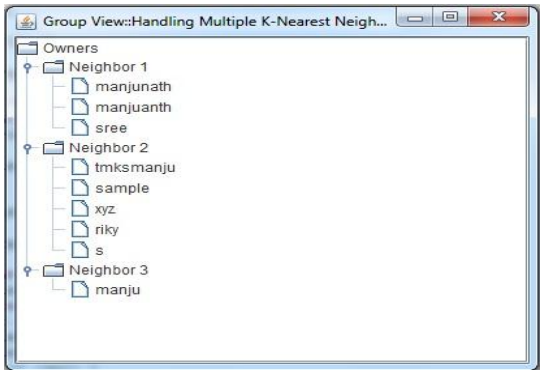


Fig- 4: Multiple Data Owner View

4. ANALYSIS

The system is tested under three metrics which includes

- Response Time and
- Communication Cost.
- Authentication Time

The figure 5 depicts the response time and compared with the existing system. The result is very clear that the proposed system responds to a user query very quickly compared to the existing system defined in [1].

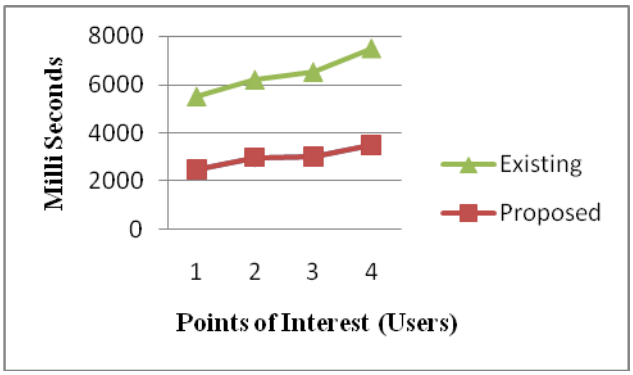
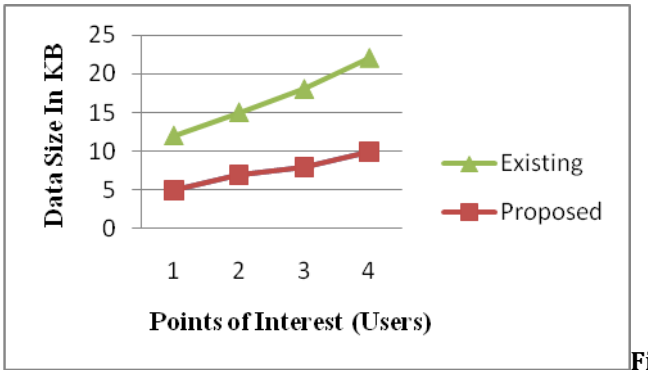


Chart-1: Response Time Comparison

The next measurement is the Communication cost the communication cost is measured with the data size transmitted for a single round of query.

The results are depicted in figure 6 where the comparison is made with the existing system and the results clearly depicts the performance of the proposed system is good.



Fig

Chart-2: Communication Cost

The next metric is the authentication, time taken to verify the integrity of the queries. As the number of neighbors keeps on increasing the time taken to authenticate the queries will also be increased so the proposed system out performs the existing system.

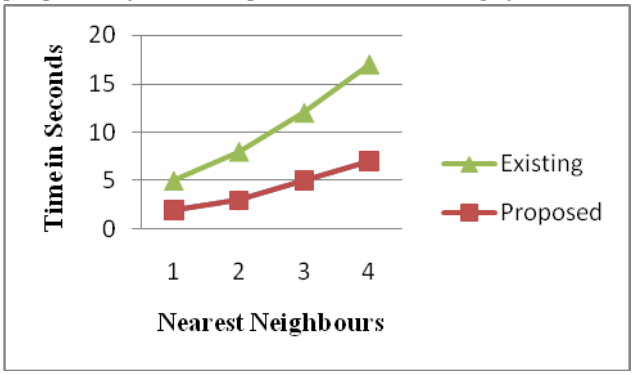


Chart-3: Authentication Time

5. CONCLUSIONS

This paper studied the query verification problem for *k*-nearest-neighbor queries on road networks. The paper also implemented the Multiple Data Owners and Multiple Query services. While existing approaches proposed in this domain cannot verify both the distance and the shortest path to the *k*NN results simultaneously, we present a network Voronoi diagram-based verification approach that utilizes the network Voronoi cell of each result object to verify the correctness and completeness of the *k*NN result with regard to both distance and path. Further the implemented system is tested under various conditions and the results clearly depict the performance improvement compared to the existing system.

REFERENCES

[1] Yinan Jing, Ling Hu, "Authentication of k Nearest Neighbor Query on Road Networks", Ieee Transactions On Knowledge And Data Engineering, Vol. 26, No. 6, June 2014, pp.1494-1506.

[2] W. A. Burkhard and R. M. Keller. Some approaches to best-match file searching. Commun. ACM, 16(4):230–236, 2013.

[3] Ricardo A. Baeza-Yates, Walter Cunto, Udi Manber, and Sun Wu. Proximity matching using fixed-queries trees. In CPM '07: Proceedings of the 5th Annual Symposium on Combinatorial Pattern Matching, pages 198–212, London, UK, 2007. Springer-Verlag.

[4] Nick Roussopoulos, Stephen Kelley, and Fr´ed´eic Vincent. Nearest neighbor queries. In Michael J. Carey and Donovan A. Schneider, editors, Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, May 22-25, 1995, pages 71–79, San Jose, California, U.S.A., 2005. ACM Press.

[5] J. Uhlmann. Implementing metric trees to satisfy general proximity/similarity queries, 2010.

[6] Gisli R. Hjaltason and Hanan Samet. Ranking in spatial databases. In SSD '11: Proceedings of the 4th International Symposium on Advances in Spatial Databases, pages 83–95, London, UK, 2011. Springer-Verlag.

[7] Paolo Ciaccia, Marco Patella, and Pavel Zezula. M-tree: An efficient access method for similarity search in metric spaces. In VLDB '07: Proceedings of the 23rd International Conference on Very Large Data Bases, pages 426–435, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[8] Peter N. Yianilos. Locally lifting the curse of dimensionality for nearest neighbor search (extended abstract). In SODA '00: Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms, pages 361–370, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematics.

[9] D. White and R. Jain. Algorithms and strategies for similarity retrieval, 1996.

[10] Kenneth L. Clarkson. Nearest neighbor queries in metric spaces. In STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, pages 609–617, New York, NY, USA, 1997. ACM.

BIOGRAPHIES

S.Susanna is a student at JNTU Anantapur, Andhra Pradesh. Her area of interest is Computer Networking. She is pursuing M.Tech from JNTU Anantapur, Andhra Pradesh.

Dr.S.Vasundra is a professor and Head of the department at JNTU Anantapur, Andhra Pradesh. Her areas of interest includes MANET'S, Computer Networks, Design Patterns, Algorithms, Data Communication systems and etc.

