International Journal of Current Research and Modern Education (IJCRME)
ISSN (Online): 2455 - 5428
(www.rdmodernresearch.com) Volume I, Issue I, 2016

SECURE NEAREST NEIGHBOR SEARCH IN DATA MINING

V. Gokulakrishnan*, S. R. Saranya, M. Ajith Kumar*** & P. Arun Kumar*****

* Assistant Professor, Department of Computer Science & Engineering, Dhanalakshmi Srinivsan Engineering College, Perambalur
** UG Scholar, Department of Mechanical Engineering, Dhanalakshmi Srinivsan Engineering College, Perambalur

Abstract:

Multistep processing is commonly used for nearest neighbor (NN) and similarity search in applications involving high-dimensional data and costly distance computations. Today, many such applications require a proof of result correctness. In this setting, clients issue NN queries to a server that maintains a database signed by a trusted authority. The server returns the NN set along with supplementary information that permits result verification using the data set signature. An adaptation of the multistep NN algorithm incurs prohibitive network overhead due to the transmission of false hits, i.e., records that are not in the NN set, but are nevertheless necessary for its verification. In order to alleviate this problem, this paper presents a novel technique that reduces the size of each false hit and generalizes solution for a distributed setting, where the database is horizontally partitioned over several servers.

Index Terms: Query authentication, multistep nearest neighbors, similarity search, False Hits & Horizontally partitioned

1. Introduction:

DB be a D-dimensional data set. Each record P ∈ DB can be thought of as a point in the space defined by the D attribute domains, and in the sequel i use the term record and point interchangeably. Given a point Q, a nearest neighbor (NN) query retrieves the record {P ∈ DB: DST(Q,P) ≤ DST(Q,P') ∨ P' ∈ DB}, where DST(Q, P) denotes the distance between Q and P. Likewise, a kNN query returns the k closest points to Q. NN and kNN queries are common in similarity retrieval. Specifically, since similarity between records is inversely proportional to their distance, a kNN query returns the k most similar records to Q. The multistep framework has been proposed for NN and similarity retrieval in domains that entail high dimensional, expensive distance functions or a combination of both factors. In this paper, i focus on authenticated multistep NN search for applications that require a proof of result correctness. For instance, argues that the most cost-effective way for medical facilities to maintain radiology images is to outsource all image management tasks to specialized commercial providers. Clients issue similarity queries to a provider. The latter returns the result set and additional verification information, based on which the client establishes that the result is indeed correct, i.e., it contains exactly the records of DB that satisfy the query conditions, and that these records indeed originate from their legitimate data source. A similar situation occurs for data replication, i.e., when a data owner stores DB at several servers. Clients issue their queries to the server, but they wish to be assured that the result is the same as if the queries were sent to the original source of DB. In other cases, correctness is guaranteed by a trusted third party. For instance, notarization services have been proposed to safeguard against tampering in document databases. Authenticated query processing ensures the client that the received result complies with the validated DB. Initially, I study the problem assuming that the entire DB resides at a single server. Our first contribution is AMN, an adaptation of a multistep algorithm that is optimal in terms

of DST computations. AMN requires transmissions of false hits, i.e., records that are not in the result, but are nevertheless necessary for its verification. In addition to the network overhead, false hits impose a significant burden to the client, which has to verify them. The second contribution, C-AMN, alleviates this problem through an elaborate scheme that reduces the size of false hits. Finally, i consider a distributed setting, where the database is horizontally partitioned over several servers. Our third contribution, ID-AMN, incrementally retrieves data, gradually eliminating servers that cannot contribute results.

2. Experimental:

A. Multistep NN Framework:

The multistep NN framework is motivated by applications that entail expensive distance computations. Specifically, let DST(Q, P) be the actual distance between a query Q and a data point P ∈ DB. The applicability of the multistep framework rests on the existence of a filter distance metric dst, which is cheap to evaluate and satisfies the lower bounding property, i.e., for every possible Q and P: dst(Q,P) ≤ DST(Q, P). Multistep NN search was introduced in [11]. Subsequently, Seidl and Kriegel [19] proposed the algorithm of Fig. 1, which is optimal in terms of DST computations. In order to provide a concrete context, our explanation focuses on road networks [18], where DST and dst refer to the network and Euclidean distance, respectively. Compared to Euclidean distance (dst), network distance (DST) computations are significantly more expensive because they entail shortest path algorithms in large graphs. Moreover, the Euclidean kNN scan be efficiently retrieved using conventional NN search on a spatial index. Assuming that DB is indexed by an R*-Tree [1], the multistep kNN algorithm first retrieves the k Euclidean NNs of Q using an incremental algorithm (e.g., [7]). These points are inserted into a result set RS, and their network (DST) distances are computed. Let DSTmax be the network distance between Q and its current kth NN Pk. The next Euclidean NN P is then retrieved. As long as dst(Q,P) < DSTmax, the algorithm computes DST(Q, P) and compares it against DSTmax. If DST(Q,P) < DSTmax, P is inserted into RS, the previous Pk is expunged, and DSTmax is updated. The loop of Lines 5-9 terminates when dst(Q,P)>= DSTmax; because of the lower bounding property of the Euclidean distance, any point lying further in the Euclidean space cannot be closer than DSTmax in the network. Independently of the application domain, the algorithm performs the minimum number of DST computations. Specifically, in addition to RS, the DST distances are computed only for false hits, i.e., the set of points FH ={P ∈ DB-RS : dst(Q,P) <= DST(Q,Pk)}, where Pk is the final kth NN. The rest of the records are not accessed at all (if they reside in pruned nodes of the R*-Tree), or they are eliminated using their dst to Q.

B. High-Dimensional Similarity Search Using Multistep NN:

Several applications including image, medical, time series, and document databases involve high-dimensional data. Similarity retrieval in these applications based on low dimensional indexes, such as the R*-Tree, is very expensive due to the dimensionality curse. Specifically, even for moderate dimensionality (i.e., D = 20) a sequential scan that computes DST(Q, P) for every P ∈ DB is usually cheaper than conventional NN algorithms using the index. Consequently, numerous specialized structures have been proposed for exact and approximate kNN search in high dimensions. The GEMINI framework [6], [11] follows a different approach, combining multistep search with a dimensionality reduction technique that exhibits the lower bounding property. Specifically, each record P ∈ DB is mapped to a low-dimensional representation p in d dimensions (d << D). The resulting d-dimensional data set db is

indexed by an R*-Tree, or any low-dimensional index. The query Q is also transformed to a d-dimensional point q and processed using a multistep method. DST (resp. dst) computations involve high (low) dimensional points. The index prunes most nodes and records using the cheap, filter (dst) distances, whereas the expensive DST computations are necessary only for the points in result RS and false hit set FH. GEMINI is the most common approach for performing similarity search over high-dimensional data, and especially time series. Numerous dimensionality reduction methods have been used extensively including Discrete Fourier Transform (DFT), Singular Value Decomposition (SVD), Discrete Wavelet Transform (DWT), Piecewise Linear Approximation (PLA), Piecewise Aggregate Approximation (PAA), Adaptive Piecewise Constant Approximation (APCA), and Chebyshev Polynomials (CP). Their effectiveness is measured by the number of records that they can prune using only the low-dimensional representations (i.e., it is inversely proportional to the cardinality of FH). Ding et al. [5] experimentally compare various techniques, concluding that their effectiveness depends on the data characteristics.

C. Authenticated Query Processing:

In authenticated query processing, a server maintains a data set DB signed by a trusted authority (e.g., the data owner, a notarization service). The signature sig is usually based on a public-key cryptosystem (e.g., RSA [16]). The server receives and processes queries from clients. Each query returns a result set RS _ DB that satisfies certain predicates. Moreover, the client must be able to establish that RS is correct, i.e., that it contains all records of DB that satisfy the query conditions, and that these records have not been modified by the server or another entity. Since sig captures the entire DB (including records not in the query result), in addition to RS, the server returns a verification object (VO). Given the VO, the client can verify RS based on sig and the signer's public key. VO generation at the server is usually performed using an authenticated data structure (ADS). The most influential ADS is the Merkle Hash Tree (MH-Tree) [15], a main-memory binary tree, originally proposed for single record authentication. Each leaf in the MH-Tree stores the digest of a record, calculated using a one-way, collision-resistant hash function h(.), such as SHA-1 [16]. An inner node stores a digest computed on the concatenation of the digests in its children. The trusted authority signs the root digest

3. Node Architecture and Protocols:

A. Authenticated Multistep NN:

Our work adopts the GEMINI framework because 1) it has been proven effective in non authenticated similarity retrieval, especially for numerous (i.e., D > 100) dimensions, where even high-dimensional indexes fail 2) it can be extended to authenticated query processing based on a low dimensional ADS, i.e., the MR-Tree, whereas, currently there are no authenticated high-dimensional structures; and 3) it is general, i.e., it can also be applied when the expensive distance computations are due to the nature of the distance definition (e.g., network distance), rather than the data dimensionality (in which case D = d). We assume a client-server architecture, where the server maintains data signed by a trusted authority. There are two versions of the signed data set: a D-dimensional DB and a d dimensional db (d << D), produced from DB using any dimensionality reduction technique that satisfies the lower bounding property. For instance, DB may be a set of high dimensional time series and db their low-dimensional representations obtained by DFT. There is a single signature sig, generated by a public key cryptosystem (e.g., RSA), that captures both DB and db. DST (dst) refers to the distance metric used in the D(d)-dimensional space. For ease of

illustration, we use Euclidean distance for both the DST and dst metrics. Nevertheless, the proposed techniques are independent of these metrics, as well as of the underlying dimensionality reduction technique. The proposed Authenticated Multistep NN (AMN) extends the multistep NN algorithm of [19] to our setting. As opposed to optimizing the processing cost at the server, the major objective of AMN (and any query authentication technique, in general) is to minimize 1) the network overhead due to the transmission of the VO, and 2) the verification cost at the client (which is assumed to have limited resources compared to the server).

B. Communication-Efficient AMN*:*

Depending on the dimensionality reduction technique, the values D, d, and k, and the data set characteristics, there may be numerous false hits in FH, each containing hundreds or thousands (i.e., D) values. Next, we propose communication-efficient AMN (C-AMN), which decreases the size of the false hits, significantly reducing the transmission and verification cost without compromising the security of AMN.

C. False Hit Reduction Algorithm:

Ideally, for each false hit P, Reduce FH should derive the subset SP with the minimum length. Intuitively, this task is at least as difficult as the Knapsack Problem; we need to select a subset of items (SP of P values), each assigned a cost (communication overhead) and a weight (distance DST(SQ, SP)), such that the sum of costs is minimized and the sum of weights exceeds DSTmax. An additional complication is that when we select one item, the cost of the rest changes (i.e., unlike knapsack, where the cost is fixed).

D. AMN in Distributed Servers:

In this setting, we assume that the database is horizontally partitioned and distributed over m (>1) servers. Specifically, each server Si stores a subset DBi such that: DB^1 U..U DB^m =DB and $DB^i \cap DB^j = \Phi <$ i,j <m,. In addition, S^i maintains an MR-Tree on the corresponding reduced data set db^i, which is signed by a signature sig^i. A query result comprises the kNNs over all servers. Minimization of transmissions (of the high-dimensional data) is particularly important for this setting, especially for large values of m. SD-AMN (short for simple distributed AMN), used as a benchmark in our experimental evaluation. Section 5.2 proposes ID-AMN (short for incremental distributed AMN), a more elaborate method, which quickly eliminates servers that cannot contribute results.

E. Simple Distributed AMN:

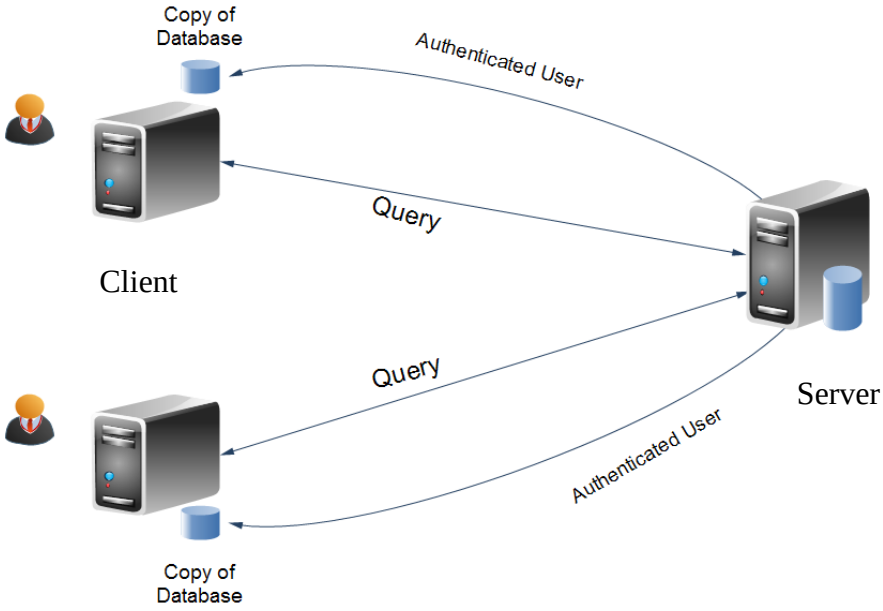
In SD-AMN, a client sends its kNN query Q to all servers. Each server S^i retrieves the partial result RS^i on the local DB^i using the conventional multistep algorithm, and generates a vector $kDST^i$ with the distance values of the Knn set RS^i in S^i. The client collects the vectors from the servers and determines the global kth nearest distance DST_{max} over all m .k collected distances. Then, it transmits a range q_R =(q,DST_{max}). Each server S^i executes qR using its MRTree and returns VO^i_R, RS^i, and FH^i. VO^i_R has the same meaning as in centralized processing, i.e., it is the VO of q_R. RS^i (resp. FH^i) is a set of results (resp. false hits), i.e., points of db^i that fall in qR and whose high-dimensional representations have distance from Q smaller (resp. larger) than DST_{max}. The size of FH can be reduced through C-AMN.

F. Incremental Distributed AMN:

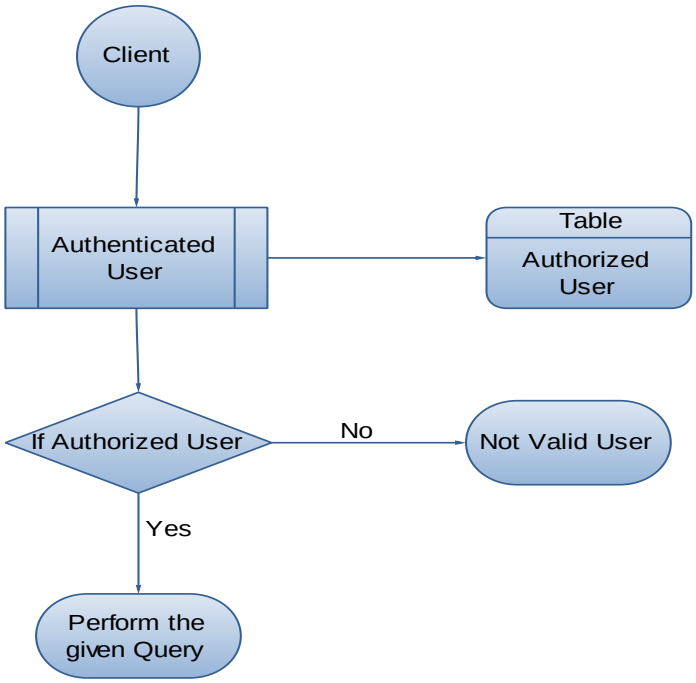
SD-AMN is optimal in terms of high-dimensional point transmissions because the client receives D- imensional representations only for points in qR. All these points (results and false hits) are necessary to establish correctness anyway. However, it must transmit Q to all servers. Moreover, each server Si has to compute RSi although none of

the points of RSi may participate in the global result. ID-AMN avoids these problems by gradually eliminating servers that cannot contribute results. Specifically, ID-AMN incrementally retrieves distance values from servers to compute the final DST_{max}, postponing local NN computations at the servers until they are required

4. System Overview:



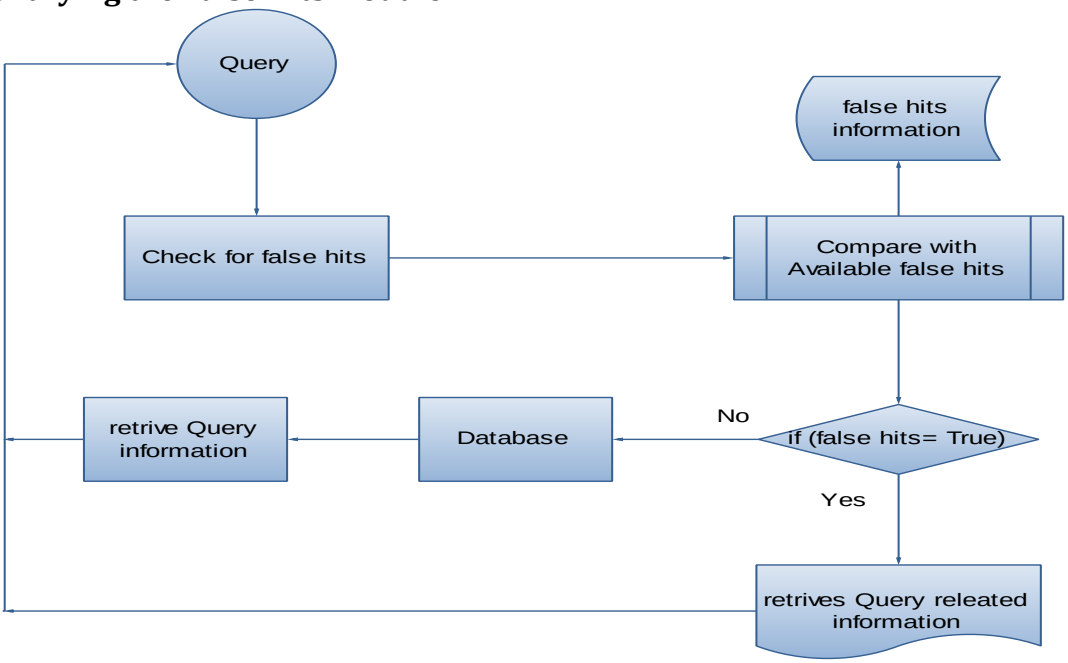
A. Authentication Process:



Authenticated multistep NN search for applications that require a proof of result correctness. Clients issue similarity queries to a provider. The latter returns the result set and additional verification information, based on which the client establishes that the result is indeed correct, i.e., it contains exactly the records of Database that satisfy the query conditions, and that these records indeed originate from their legitimate data source. Clients issue their queries to the closest (in terms of network latency) server,

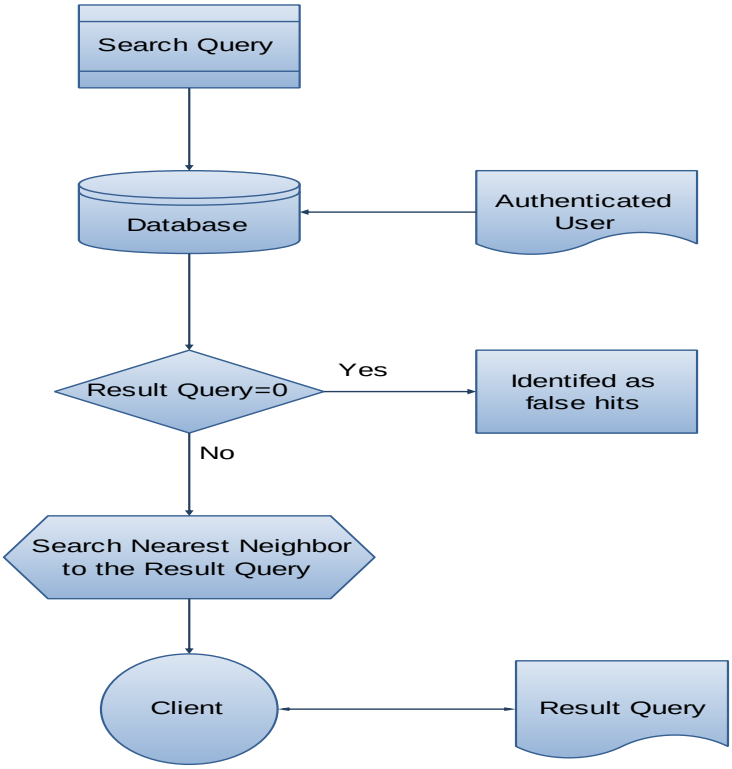
but they wish to be assured that the result is the same as if the queries were sent to the original source of Database. Authenticated query processing ensures the client that the received result complies with the validated Database.

B. Identifying the False Hits Module:



Records that are not in the Database, but are nevertheless necessary for its verification. In addition to the network overhead, false hits impose a significant burden to the client, which has to verify them. It initializes a false hit set False Hit (FH) = 0, and a result set Record set (RS) ={P1 . . . . Pk}, and computes DST_{max}, i.e., the DST of the current k^{th} NN P_k. The goal of AMN is to return to the corresponding client the kNNs of Q, in a verifiable manner.

C. Nearest Neighbor Search:



The server receives and processes queries from clients. Each query returns a result set RS_DB that satisfies certain predicates. Moreover, the client must be able to establish that Record Set (RS) is correct, i.e., that it contains all records of Database that satisfy the query conditions, and that these records have not been modified by the server or another entity. There may be numerous false hits in FH information, each containing hundreds or thousands of values. Next, we communication-efficient AMN (C-AMN), which decreases the size of the false hits, significantly reducing the transmission and verification cost without compromising the security of AMN. The main concepts of C-AMN algorithm, for false hit reduction.

5. Results:

A. Single Server:

The measures of interest are the communication overhead, and the CPU cost at the server and the client. We assess the communication overhead based on the verification information sent to the client. The transmission of the query and the result is omitted because it is necessary in any method. The CPU cost is measured in terms of the elementary distance computations. Specifically, D elementary computations are required to derive the Euclidean distance of two D dimensional points. We exclude the I/O cost at the server because it is identical for both AMN and C-AMN (and similar to that of the conventional multistep algorithm) since in any case, we have to retrieve the low-dimensional NNs using the MR-Tree. For each experiment, we select a random data point as the query, and report the average results over 10 queries.

B. Distributed Servers:

We compare SD-AMN and ID-AMN considering that the database is horizontally partitioned over m servers. Recall that the methods first collect distance information, based on which they determine the range that contains the result. The NNs and the false hits are obtained during the verification of this range, which is identical in SD-AMN and ID-AMN. Thus, when measuring the communication cost, we focus on their differences, which regard the transmission of query points and the distance information. The CPU overhead is based again on elementary distance computations. Finally, due to the identical verification process, the client cost is similar, and the corresponding experiments are omitted.

6. Discussion and Conclusion:

The importance of authenticated query processing increases with the amount of information available at sources that are untrustworthy, unreliable, or simply unfamiliar. This is the first work addressing authenticated similarity retrieval from such sources using the multistep kNN framework. We show that a direct integration of optimal NN search with an authenticated data structure incurs excessive communication overhead. Thus, we develop C-AMN, a technique that addresses the communication specific aspects of NN, and minimizes the transmission overhead and verification effort of the clients. Furthermore, this paper proposes ID-AMN, which retrieves distance information from distributed servers, eliminating those that cannot contribute results.

7. References:

1. N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, “The R- Tree: An Efficient and Robust Access Method for Points and Rectangles,” Proc. ACM SIGMOD, 1990.
2. K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, “When is “Nearest Neighbor” Meaningful?,” Proc. Int’l Conf. Database Theory (ICDT ’99), 1999.
3. R. Bryan, “The Digital Revolution: The Millennial Change in Medical Imaging,” Radiology, vol. 229, pp. 299-304, Nov. 2003.

4. Y. Cai and R. Ng, "Indexing Spatio-Temporal Trajectories with Chebyshev Polynomials," Proc. ACM SIGMOD, 2004.
5. H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures," Proc. Int'l Conf. Very Large Data Base Endowment (VLDB '08), vol. 1, pp. 1542-1552, 2008.
6. C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast Subsequence Matching in Time-Series Databases," Proc. ACM SIGMOD, 1994.
7. G. Hjaltason and H. Samet, "Distance Browsing in Spatial Databases," Trans. Database Systems (TODS '99), vol. 24, no. 2, pp. 265-318, 1999.
8. H. Jagadish, B. Ooi, K. Tan, C. Yu, and R. Zhang, "I-Distance: An Adaptive B+-Tree Based Indexing Method Nearest Neighbor Search," Trans. Database Systems (TODS '05), vol. 30, no. 2, pp. 364- 397, 2005.
9. H. Kellerer, U. Pferschy, and D. Pisinger, Knapsack Problems. Springer, 2004.
10. E.J. Keogh, C.A. Ratanamahatana, "Exact Indexing of Dynamic Time Warping," Knowledge and Information Systems, vol. 7, no. 3, pp. 358-386, 2005.
11. F. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Protopapas, "Fast Nearest Neighbor Search in Medical Image Databases," Proc. Int'l Conf. Very Large Data Base Endowment (VLDB '96), 1996.
12. A. Kundu and E. Bertino, "Structural Signatures for Tree Data Structures," Proc. Int'l Conf. Very Large Data Base Endowment (VLDB '08), 2008.
13. F. Li, M. Hadjieleftheriou, G. Kollios, and L. Reyzin, "Dynamic Authenticated Index Structures for Outsourced Databases," Proc. ACM SIGMOD, 2006.
14. C. Martel, G. Nuckolls, P. Devanbu, M. Gertz, A. Kwong, and S. Stubblebine, "A General Model for Authenticated Data Structures," Algorithmica, vol. 39, no. 1, pp. 21-41, 2004.
15. R. Merkle, "A Certified Digital Signature," Proc. CRYPTO, 1989.
16. A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography. CRC Press, 1996.
17. H. Pang and K. Mouratidis, "Authenticating the Query Results of Text Search Engines," Proc. Int'l Conf. Very Large Data Base Endowment (VLDB '08), 2008.
18. D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao, "Query Processing in Spatial Network Databases," Proc. Int'l Conf. Very Large Data Base Endowment (VLDB '03), 2003.
19. T. Seidl and H.-P. Kriegel, "Optimal Multi-Step k-Nearest Neighbor Search," Proc. ACM SIGMOD, 1998.
20. R.T. Snodgrass, S.S. Yao, and C. Collberg, "Tamper Detection in Audit Logs," Proc. Int'l Conf. Very Large Data Base Endowment (VLDB '04), 2004.
21. R. Tamassia and N. Triandopoulos, "Efficient Content Authentication over Distributed Hash Tables," Proc. Int'l Conf. Applied Cryptography and Network Security (ACNS '07), 2007.