Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9990193, 12 pages
https://doi.org/10.1155/2022/9990193

Research Article

H2SA-ALSH: A Privacy-Preserved Indexing and Searching Schema for IoT Data Collection and Mining

Guangjun Wu[1] **Bingqing Zhu**[1,2] **Jun Li**,[1,2] **Yong Wang**[1] **and Yungang Jia**[3]

[1] *Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100080, China*
[2] *School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100080, China*
[3] *National Computer Network and Information Security Management Center of Tianjin, Tianjin 300100, China*

Correspondence should be addressed to Yong Wang; wangyong@iie.ac.cn

Received 15 June 2021; Accepted 14 March 2022; Published 18 April 2022

Academic Editor: Zhaolong Ning

Currently, smart devices of Internet of Things generate massive amount of data for different applications. However, it will expose sensitive information to external users in the process of IoT data collection, transmission, and mining. In this paper, we propose a novel indexing and searching schema based on homocentric hypersphere and similarity-aware asymmetric LSH (H2SA-ALSH) for privacy-preserved data collection and mining over IoT environments. The H2SA-ALSH collects multidimensional data objects and indexes their features according to the Euclidean norm and cosine similarity. Additionally, we design a c-k-AMIP searching algorithm based on H2SA-ALSH. Our approach can boost the performance of the maximum inner production (MIP) queries and top-k queries for a given query vector using the proposed indexing schema. Experiments show that our algorithm is excellent in accuracy and efficiency compared with other ALSH-based algorithms using real-world datasets. At the same time, our indexing scheme can protect the user's privacy via generating similarity-based indexing vectors without exposing raw data to external users.

1. Introduction

In recent years, Internet of Things (IoT) technology has been applied to a wide range of applications [1, 2], mainly driven by the rising number of Internet-connected devices that already amount to several billion [3]. The devices of IoT [4] aim to connect everyday objects, such as humans, plants, and even animals, to the Internet to enable interactions among these objects [5]. Applications of IoT have been widely developed in medical healthcare [6, 7], vehicular networks [8, 9], and industrial IoT [10]. With the widespread popularity of IoT, a massive amount of data is generated and widespread at a relatively fast speed.

Thus, applications in different IoT domains have seen an explosion of information generated from heterogeneous devices every day. Recently, the data collection and mining over IoT data streams have increasingly incurred research interests [11–15].

However, due to weak privacy and security protection in IoT devices, some smart applications of IoT expose sensitive data and user privacy to security threats [16]. Thus, data mining over raw data will collect and expose user-sensitive information. As with stream data mining [17], interesting knowledge, regularities, or high-level information, they can easily introduce privacy protection policies. At present, MIP (maximum inner production) search is prominent, and it was used in a wide range of applications, such as matrix factorization-based recommendation systems [18–20], multiclass label prediction [21, 22], SVM classification [23], and even deep learning [24]. However, it is time-consuming to conduct the MIP search in high-dimensional space. Moreover, it may cause user's privacy leakage. A query system needs to collect the raw data from devices of the IoT system. Many types of research try to construct an appropriate approximate structure for the search. It is usually called approximate maximum inner product (AMIP) search [25–29], in which a given query q and a data object $x \in D$, D is the set of target objects, and the AMIP algorithms will compute the approximate maximum inner product results for q in D.

The techniques of AMIP are often based on locality-sensitive hashing (LSH) [30], which can solve the AMIP problem in sublinear time. Currently, many algorithms based on LSH are also proposed, such as L2-ALSH [28], Sign-ALSH [31], Simple-ALSH [27], XBOX [32], and *H*2-ALSH [33]. Additionally, many data mining tasks over massive datasets are also applied using LSH based algorithms [34–37] to accelerate the MIP search. The common AMIP data mining algorithms, such as L2-ALSH [28] and Simple-ALSH [27], converts AMIP searching problem into c-ANN searching problem. Recently, some novel methods proposed to solve the high-dimensional AMIP search problems by introducing approximate features into the indexing vectors.

Motivated by the promising techniques, we can extract target features from raw data objects of IoT devices and conduct the maximum similarity search between the input vector q and the set of extracted indexing vectors, and only the object of the matched features is needed to be transmitted to the user for a final decision. Thus, it will protect the user's privacy from sensitive information collection and exposure to third-parties query services [38, 39]. The contributions of the paper are as follows:

(1) We propose a novel privacy-preserved indexing and searching schema, termed H2SA-ALSH for high-dimensional data objects collection and mining. The indexing scheme is based on homocentric hyperspheres and similarity-aware algorithm (H2SA). The searching is applied to compute the cosine similarity between a query vector and data objects. The proposed schema can support AMIP search, top-k search, etc., without exposing raw data privacy

(2) We optimize the proposed indexing solution to fit IoT data collection and mining. In the process of IoT data collection, we establish an incremental indexing mechanism, which indexes an input item immediately, when a data item arrives. For IoT data mining, we design SRP-LSH to accelerate the search by filtering the low-similarity objects. Moreover, the algorithm is not sensitive to the data, i.e., it presents acceptable performance over different distribution datasets

(3) We conduct comprehensive experiments to evaluate the H2SA-ALSH indexing and searching scheme using three real-world data sets. The experimental results show that the proposed approach is more accurate and efficient than the state-of-the-art algorithms. As a result, a searching query will not be directly conducted over the raw data objects in IoT environments

2. Problem Definition

In the section, we briefly present preliminaries of the proposed techniques and state our research problem formally. Then we use the common notations in AMIP literature and present the MIP and corresponding AMIP searching problem formally.

Definition 1. Maximum inner product (MIP) search. For a data collection T that already received n data objects and an arbitrary query $q \in R^d$, the MIP search aims to find $t \in R^d$ that satisfies

$$t = \operatorname{argmax}_{t \in T} \langle q, t \rangle. \quad (1)$$

Definition 2. The c-approximate maximum inner product (c-AMIP) search. Given an approximate ratio $c(0 < c < 1)$, the goal of the c-AMIP search is to construct an approximate structure, and a user can find the approximate result t, $t \in T$, which satisfies the following condition for a query $q \in R^d$, i.e., $\langle t, q \rangle \le c\langle t^*, q \rangle$, t^* is the accurate result of the MIP search.

In the paper, we convert the c-AMIP search problem to the c_0-ANN problem. The c_0-ANN problem aims to find the nearest neighbour according to the Euclidean distance. The definition of the c_0-ANN problem is as follows:

Definition 3. Given an approximate ratio $c_0(c_0 > 1)$, and for a query vector $q \in R^d$, c_0-ANN aims to find data object t, $t \in T$, which satisfies the following formula:

$$\|t, q\| \le c_0 \|t^*, q\|, , \quad (2)$$

where t^* is accurately obtained by the MIP search.

The LSH is a common method to solve the c_0-ANN problem. We use the definition of the nearest neighbour whose distance measure is measured as $\mathrm{Sim}(q, p)$ to depict the LSH paradigm. Let h be a hash function that maps an item to a hash value, and the corresponding definition is as follows.

Definition 4. When a hash family H meets the following conditions, it can be called (S_0, cS_0, p_1, p_2) sensitive. For multidimensional data objects x and y, the hash function h from H satisfies:

(1) If $\mathrm{Sim}(x, y) \le S_0$, then, the probability of $h(x) = h(y)$ is at most p_1

(2) If $\mathrm{Sim}(x, y) \ge cS_0$, then, the probability of $h(x) = h(y)$ is at least p_2

where $c < 1$ and $p_1 > p_2$, respectively.

We adopt the common LSH technology to solve the ANN search problem, and similar data objects have higher probability of getting the same hash function results than those with lower similarity. Thus, the LSH can solve the nearest neighbour and similarity problems of multidimensional data even in linear time [40].

Furthermore, we transform the AMIP search problem into the nearest neighbour problem via asymmetric locality sensitive hashing (ALSH). There have been some researches on ALSH technologies [27, 30, 31, 32, 33]. In this paper, we use the QNF transformation [32]. For a data object $t = (o_1, o_2, \cdots, .o_d)$ and a query $= (q_1, q_2, \cdots, q_d)$, the transformation is as follows:

$$P(t) = \left[o_1, o_2, \cdots, o_d\, ; \sqrt{M^2 - \|t\|^2}\right], \tag{3}$$

$$Q(q) = [\lambda q_1, \lambda q_2, \cdots, \lambda q_d\, ; 0], \text{where } \lambda = \frac{M}{\|q\|}. \tag{4}$$

In formulas (3) and (4), the constant M is used to present the largest Euclidean norm among the data collection T. The maximum Euclidean norm may constantly change when it collects more data from IoT devices. In our schema, we assign an appropriate M to each ALSH unit, and the M is the maximum Euclidean norm. For a data object t, $\|t\| \leq M^2$. Through the QNF transformation, the AMIP search problem can be converted into the nearest neighbour search problem. The following formula can be used in the transform:

$$\|Q(q) - P(t)\|^2 = M^2 + \lambda^2\|q\|^2 - 2\lambda\langle t, q\rangle. \tag{5}$$

In Equation (5), for a query q, M^2 and $\lambda^2\|q\|^2$ are constant, so we have

$$\operatorname*{argmax}_{t\in T} \langle t, q\rangle \Longleftrightarrow \operatorname*{argmin}_{t\in T} \|Q(q) - P(t)\|^2. \tag{6}$$

The $\operatorname*{argmin}_{t\in T} \|Q(q) - P(t)\|^2$ is the nearest neighbour search problem, and it can be solved by the $L2$-LSH technology quickly. We will present the signed random projections LSH (SRP) and $L2$-LSH, where similarity measurement methods are the correlation similarity and the $L2$ distance, respectively. When the distance is the correlation similarity, let θ be the angle of two multidimensional vectors, A and B be the multidimensional vector, where $0 \leq \theta \leq 180$. The distance of correlation similarity is

$$d(A, B) = \theta = \arccos\left(\frac{\langle A, B\rangle}{|A| * |B|}\right). \tag{7}$$

The correlation similarity is $1 - d(A, B)$, and the SRP-LSH can solve the maximum correlation similarity search. The procedure can be depicted as follows: first, a random vector v with $v_i \sim N(0, 1)$ is obtained. The random vector determines a hash function h_v, and the hash function h_v will return dualistic results. If $\langle v, x\rangle < 0$, then, $h_v(x) = 0$, else $h_v(x) = 1$. The LSH family H is formed by several random vectors. By the SRP-LSH, we can conclude

$$\Pr[h_v(A) = h_v(B)] = 1 - \frac{d(A, B)}{\pi}, \tag{8}$$

i.e.,

$$\cos(A, B) = \cos(\pi(1 - \Pr[h_v(A) = h_v(B)])). \tag{9}$$

Now, we briefly propose the indexing schema based on the asymmetric LSH scheme for high-dimensional AMIP search. We also adopt the $L2$-LSH and SRP-LSH. The indexing features from IoT devices were calculated by the Euclidean norm and cosine similarity among the data. More details, when a data object t comes, the schema calculates the t's Euclidean norm and keeps the feature into an exact block according to the cosine similarity. The exact block and the exact bucket determine the data item's storage unit. When conducting a query, the schema adopts QNF transformation and searches the c-AMIP results through the $L2$-LSH, precisely through the QALSH [32]. We have kept the block partition principle of $H2$-ALSH. The blocks are divided by the Euclidean norm of the data objects with the division ratio. Besides, we consider another factor determining the inner product which is the angle between the given query and the data objects. We use SRP-LSH to divide one block into buckets, so the buckets are the minimum storage unit in our schema. The overview of our indexing schema is shown in Figure 1.

When we conduct the AMIP search, we traverse the blocks in order, traversing blocks from a large Euclidean norm to a small block. Then, we traverse from high similarity to low similarity according to the cosine similarity within one block.

In our schema, the calculation can focus on the data objects that can be considered as candidates, which have a higher possibility of becoming the AMIP search results, and the search process finishes when there is no necessary to traverse the rest data objects. Thus, filtering the unnecessary data objects allows the schema to reach a remarkable time performance.

Our work is different from the article [32], in which the data object is treated as the static items, and all data are only divided into buckets by Euclidean norm. Our schema considers IoT environments, where the data is updated frequently, and we cannot sort the whole static sets. Instead, the input object will be inserted into our H2SA-ALSH unit when it comes. The indexing construction does not decrease the accuracy of the following queries. Therefore, our indexing schema is more appropriate for IoT scenarios where the features are dynamically generated through distributed devices and applications.

3. Indexing Construction

Given a continuous object series T, and an incoming object $t_i \in T$, we first calculate the Euclidean norm $\|t_i\|$. To effectively divide the blocks $[S_1, S_2, \cdots, S_K]$, we introduce the interval rate b. Given an AMIP search approximation rate c and the query angle β in the bucket, c_0 is the

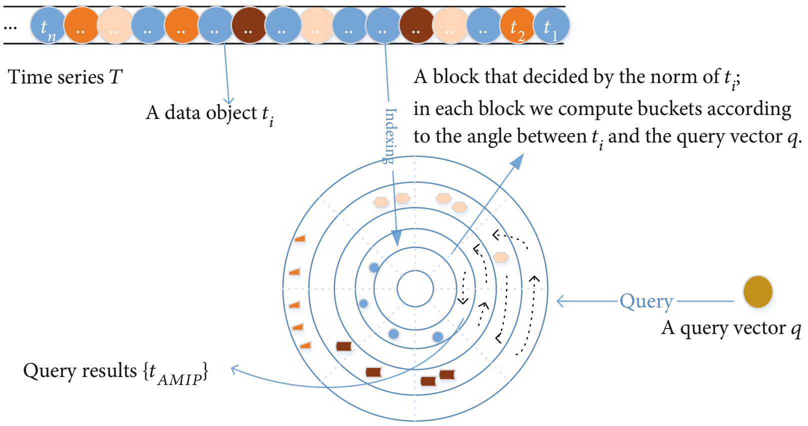

...
t_n
..
t_2
t_1
Time series T
A data object t_i
Indexing
A block that decided by the norm of t_i;
in each block we compute buckets according
to the angle between t_i and the query vector q.
Query
A query vector q
Query results $\{t_{AMIP}\}$

Figure 1: Overview of the method.

approximation rate of ANN, and the b can be expressed as follows:

$$b = \sqrt{\left(1 - \frac{1-c}{c_0^4 - c}\right) \cdot l}, \qquad (10)$$

where $l = (1 - \beta_0 \cdot (\tan \beta + 1/3 \cdot \tan^3 \beta))$.

We present explanations about b and use S to represent blocks. We assign a data object t into different blocks and different buckets. There are several buckets B in S_i, and different buckets represent the classification of different objects according to the cosine similarity. Every indexing unit has a unique identifier that consists of a block identifier (S) and bucket identifier (B). The schema determines the specific bucket identifier of the data object according to the hash family of SRP-LSH. Assuming that the hash family of SRP-LSH uses k_s hash functions, the bucket number can be expressed as k_s bits. The bucket M_k can be initialized later. All data objects that satisfy $bM_k < t_i < M_k$ will be assigned into the block S_k. When putting the data objects into the buckets, the number of buckets gets larger. We set a threshold N_0, and the bucket will use QNF to convert the d -dimensional data into the $(d+1)$-dimensional if the number reaches N_0 and then builds QALSH indexing. For the buckets which number of data objects is less than the threshold N_0, the raw feature is stored directly. When dividing the block, the schema will determine the first block based on the first data object t_0 of T. The maximum norm M_{base} of this block is t_0, and the block will set as the benchmark block. Then, we can determine other data objects' blocks. For the subsequent data, we can calculate the specific block based on the norm t_i and the benchmark M_{base}. The process can be presented as the following Algorithm 1.

4. Similarity-Aware AMIP Searching

To respond to the arbitrary maximum inner product query q , we first need to calculate the Euclidean norm q and then we set the MIP value as $\varphi = -\infty$. Since the maximum norm of the data objects in the first block is the largest one, it is most likely that the block contains the MIP data object. Thus, we traverse the block from S_1 to S_K. Each block contains many buckets according to the angle similarity. Moreover, the MIP data objects are most likely to have high cosine similarity with the query q, and the traversal of the buckets is performed in ascending sequence as the angle.

For a block S_i, the AMIP process can be described as the three main stages. First, for a query q and block S_i, we first calculate a deadline condition ub. All $t \in S_i$ satisfy $bM_{i-1} < \|t\| \leq M_i$, and $\langle t, q\rangle = \|t\|\|q\|\cos a\beta \leq \|t\|\|q \leq M_i \cdot q\|$. For a block, we can have $ub = M_i \cdot \|q\|$. In the AMIP algorithm, we consider the effect of data norm $\|t\|$, and the angle θ between the query q and $t \in D$. Since $\langle t, q\rangle = \|t\|\|q\|\cos \beta$, within each bucket, we use SRP-LSH to estimate the cosine similarity β^* between q and $t \in B_i$. If the similarities of the buckets satisfy the given similarity, the schema will conduct the AMIP search process. The cosine similarity calculation will cause errors, and in the later section, we will demonstrate the specific error. Then, we use these two deadlines ub and given cosine similarity to AMIP in the buckets. (1) Before starting to traverse the block M_i, the schema will stop traversing the rest blocks if $ub \leq \beta_0$ and then the algorithm will return the AMIP data object. (2) If $ub > \varphi$, we traverse the buckets in the block, and if the cosine similarity does not satisfy the given similarity, the schema skips the buckets and traverses other buckets.

In the process of cosine similarity searching, we apply hashing banding to improve the calculation accuracy. For details, an identifier of a bucket can be represented by ks bits. When we use the hashing banding, in which the ks bits are divided into ks/bs bands, and each band has bs bits. For a query q, the SRP-LSH hash functions will calculate ks bits, which are also divided into ks/bs bands. If one of ks/bs bands from q is the same as the corresponding band of the bucket's band, we term it as having a hash similarity collision, and the angle meets our calculation requirement. The total AMIP searching algorithm can be described in Algorithm 2.

5. Theoretical Analysis

5.1. Accuracy Analysis

```
Input: a time series $T$ with objects $t_1, t_2, \cdots, t_k,$an interval ratio $b$, and a threshold $N_0$
Output: The number of disjoint K, K disjoint sets with blocks $\{S_1 = \{B_1, \cdots\}, S_2 = \{B_1, \cdots\}, \cdots S_K = \{B_1, \cdots\}\}$.
k = 1;
Compute $t_i, t_i \in T$;
$M_{base} = t_i$;
While!End$(T)$do
        $S_i, M_i =$ get_Block $(M_{base}, t_i)$;
        Compute the bucket $B_j$ of the block $S_i$ using $SRP$-LSH hash family;
        $P(t_k) = (t_k, \sqrt{M_i^2 - o_j{}^2})$;
        $B_j = B_j \cup P(t_k)$;
        If$|B_j| == N_0$then
                Build hash tables for $b_j$ of using ALSH;
        End
        If$|B_j| > N_0$then
                Insert into QALSH indexing of $B_j$;
        End
End
K = $|S|$;
Return K, $\{S_1 = \{B_1, \cdots\}, S_2 = \{B_1, \cdots\}, \cdots S_K = \{B_1, \cdots\}\}$.
```

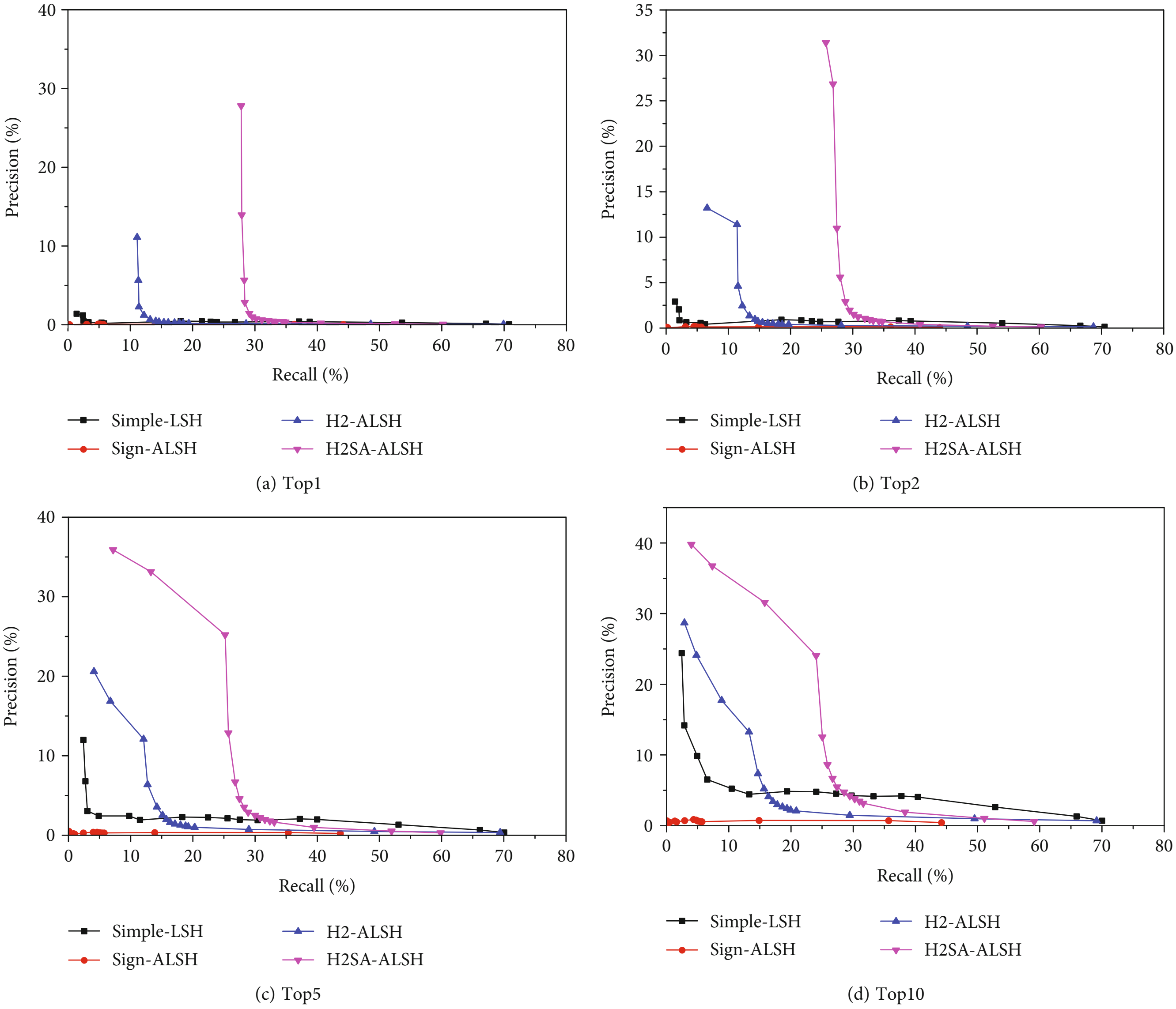
Algorithm 1: Indexing of H2SA-ALSH.

```
Input: The query q, threshold $N_0$, the number of disjoint sets $K$, and the structure of $H2SA$-ALSH: $\{S_1 = \{B_1, \cdots\}, S_2 = \{B_1, \cdots\},$
$\cdots S_K = \{B_1, \cdots\}\}$;
Output: The approximate MIP objects $t_{\text{AMIP}} \subset C$.
Compute q;
C = $\phi$;
$\varphi = -\propto$;
for $i = 1\,; i \leq K\,; i++$do
        $ub = M_i \|q\|$;
        If$ub < \varphi$then
            Break
        End
        For$B_j \in S_i$do
            Compute the hash value $B(q)$ of $q$ using $SRP$-LSH hash family;
            If the $B(q)$ causes hash collision with $B_j$ using "hash banding" technique then
                If$|B_j| < N_0$then
                        $\{t\} = linear\_scan(B_j, q)$;
                Else
                        $\lambda = M_i/\|q\|$;$Q(q) = (\lambda q_1, \lambda q_2, \cdots, \lambda q_d\,; 0)$;
                        $\{t\} = QALSH(Q(q))$;
                End
            End
            $C = C \cup \{t\}$;
            $(\varphi, t_{AMIP}) = update\,(C)$;
        End
End
Return$t_{AMIP}$.
```

Algorithm 2:c-AMIP search of H2SA-ALSH.

Theorem 5. *Set the approximation of c-AMIPS to $c(0 < c < 1)$, and the approximate value of c_0-ANN is c_0. By setting b, the probability that the result returned by Algorithm 2 meets c-AMIP is $(1/2 - 1/e)$.*

Proof. According to the paper [33, 37], we know that the probability that QALSH returns the result of c_0-ANN is at least $(1/2 - 1/e)$. If we fix the QALSH error rate is $1/e$, then the AMIP algorithm that searches for MIP will return a



(a) Top1

(b) Top2

(c) Top5

(d) Top10

Figure 2: Recall on year datasets.

result of c-AMIP. Below, we focus on proving that the AMIPS meets c-AMIP. □

We first derive the expression $\langle t,q\rangle/\langle t^*,q\rangle$, assuming t^* is the MIP for a query q, and in the block S_i, $bM_i < t^* \le M_i$, $\lambda = M_i/\|q\|$. According to the previous formula, we have

$$Q(q) - P(t^*)^2 = 2M_i^2 - \frac{2M_i}{\|q\|}\langle t^*, q\rangle. \tag{11}$$

As with c_0-ANN, according to [33], for $Q(q)$ and $P(t)$, QALSH returns a result of c_0^2-ANN, which is $\|Q(q)-P(t)\| / \|Q(q)-P(t^*)\| \le c_0^2$, let β^* be the angle of t^* and q. Combining the above formula, we have

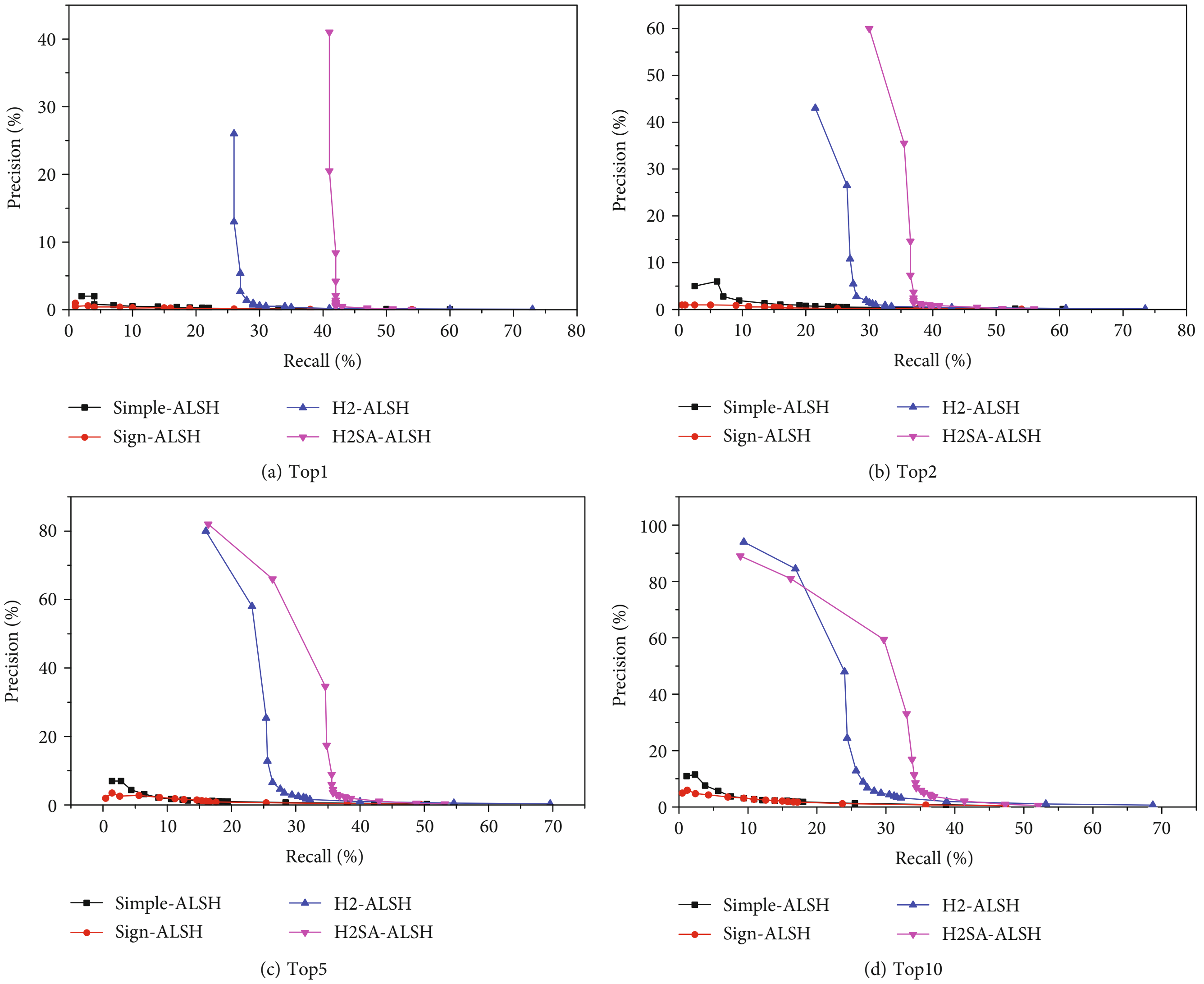
$$\frac{\langle t,q\rangle}{\langle t^*,q\rangle} \ge c_0^4 - \frac{\left(c_0^4-1\right)\cdot M_i\cdot\|q\|}{\langle t^*,q\rangle} \ge c_0^4 - \frac{c_0^4-1}{b\cos\beta^*}. \tag{12}$$

By SRP-LSH, we know

$$E(\cos\,(t,q)) = \cos\,(1-\Pr[h_v(t)=h_v(q)], \tag{13}$$

$$E\left(\frac{\langle o,q\rangle}{\langle o^*,q\rangle}\right) = c_0^4 - \frac{c_0^4-1}{b}E\left(\frac{1}{\cos\beta^*}\right) = c_0^4 - \frac{c_0^4-1}{b\cos(1-\Pr[h_v(o^*)=h_v(q)])}. \tag{14}$$

Now we try to calculate $E(1/\cos\beta^*)$, assuming α is the angle variable that changes as a threshold for a query, and β represents the angle between q and t, where $t \in M_-(i)$, $\alpha\epsilon$ $[0,\pi]$. For theoretical analysis, we assume the angles of data items obey the uniform distribution, i.e., $\Pr[\beta>\alpha] = 1-\Pr[\beta\le\alpha] = 1-\alpha/\pi$, and β is the angle of the smaller similarity bucket traversed in M_i for the q. Then, we assume that the number of data items for a block M_i is,



(a) Top1

(b) Top2

(c) Top5

(d) Top10

Figure 3: Recall on Sift datasets.

then

$$\Pr[\beta < \alpha] = \Pr\left[\beta_1 < \alpha, \beta_2 < \alpha, \cdots, \beta_{\frac{sn\alpha}{\pi}} < \alpha\right] = \left(1 - \frac{\alpha}{\pi}\right)^{sn\alpha/\pi}. \tag{15}$$

Thus, we can get the cumulative density function of α as follows:

$$F_\beta(\alpha) = \Pr[0 \le \beta \le \alpha] = 1 - \left(1 - \frac{\alpha}{\pi}\right)^{sn-\alpha/\pi}. \tag{16}$$

Also, we can calculate the deviation of $F_\beta(\alpha)$ to get the probability density function:

$$f_\beta(\alpha)\& = F'_\beta(\alpha) = -\frac{n}{\pi} \cdot \left(1 - \frac{\alpha}{\pi}\right)^{n\alpha/\pi} \left(\ln\left(1 - \frac{\alpha}{\pi}\right) - \frac{\alpha}{\pi - \alpha}\right). \tag{17}$$

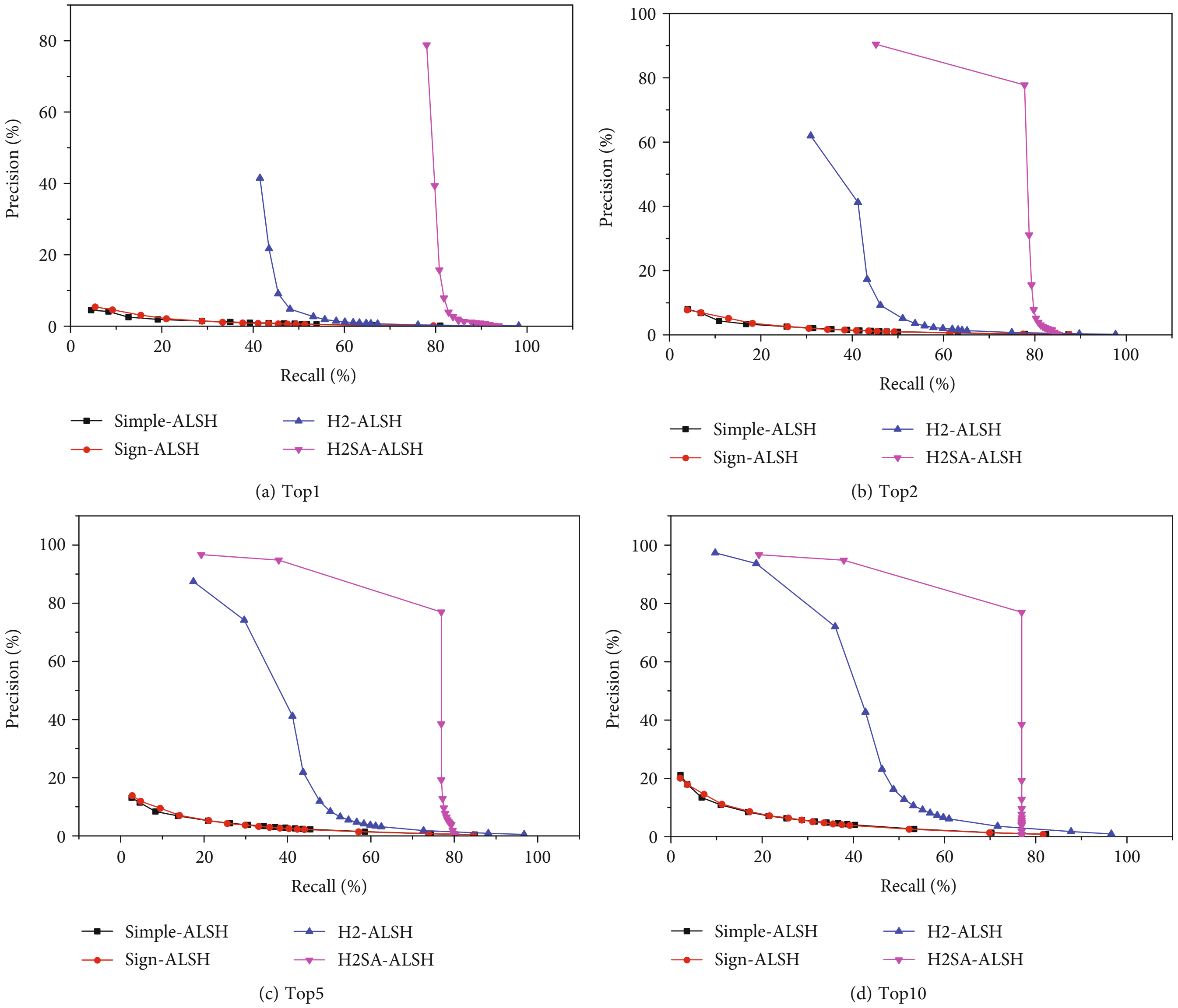
Assuming β_0 as the threshold, $\beta \in [0, \beta\_0]$, we have

$$\begin{aligned} E\left(\frac{1}{\cos \beta^*}\right) &= \int_0^{\beta_0} \frac{1}{\cos \alpha} f_{\beta_0}(\alpha) d\alpha = \int_0^{\beta_0} \frac{1}{\cos \alpha} d\left(F_\beta(\alpha) - 1\right) \\ &= \left[\frac{1}{\cos \alpha}\left(-\left(1 - \frac{\alpha}{\pi}\right)^{sn\cdot\alpha/\pi}\right)\right]_0^{\beta_0} + \int_0^{\beta_0} \left(1 - \frac{\alpha}{\pi}\right)^{5n\cdot\alpha/\pi} \cdot \frac{\sin \alpha}{\cos^2\alpha} d\alpha \\ &\le 1 - \int_0^{\beta_0} \left(1 - \frac{\alpha}{\pi}\right)^{2nc\cdot\alpha/\pi} d\alpha \cdot \int_0^{\beta_0} \frac{\sin^4\alpha}{\cos^4\alpha} d\alpha \le 1 - \beta_0 \cdot \int_0^{\beta_0} \frac{1}{\cos^4\alpha} d\alpha \\ &= 1 - \beta_0 \cdot \left[\tan \alpha + \frac{1}{3} \cdot \tan^3\alpha\right]_0^{\beta_0} = 1 - \beta_0 \cdot \left(\tan \beta + \frac{1}{3} \cdot \tan^3\beta\right). \end{aligned} \tag{18}$$

Finally, we have

$$E\left[\frac{\langle t, q\rangle}{\langle t^*, q\rangle}\right] \ge c_0^4 - \frac{c_0^4 - 1}{b^2} E\left[\frac{1}{\cos \beta}\right] \ge c_0^4 - \frac{c_0^4 - 1}{b^2}\left(1 - \beta_0 \cdot \left(\tan \beta + \frac{1}{3} \cdot \tan^3\beta\right)\right). \tag{19}$$

Let l be $(1 - \beta_0 \cdot (\tan \beta + 1/3 \cdot \tan^3\beta))$. We can depict the



(a) Top1

(b) Top2

(c) Top5

(d) Top10

Figure 4: Recall on Mnist datasets.

interval rate of the block b as

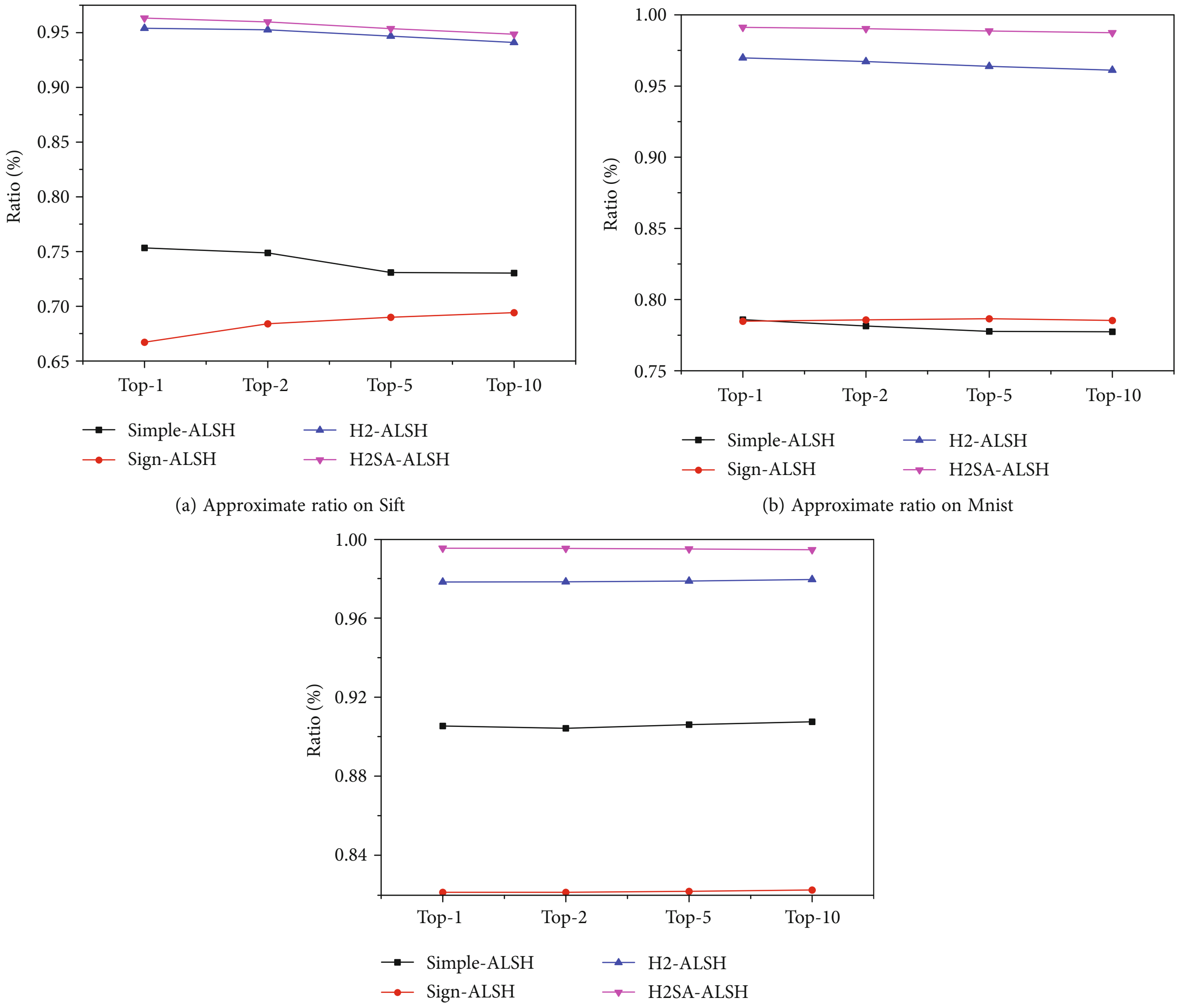
$$b = \sqrt{\left(1 - \frac{1-c}{c_0^4 - c}\right) \cdot l}. \tag{20}$$

5.2. Complexity Analysis. In this section, we conduct an analysis of the space and time complexity of our algorithm.

Theorem 6. *Given an approximation ratio $c (0 < c < 1)$ for a c -AMIP search, we use $O(nd + n \log n)$ space to construct indexing structure and cost $O(n \log n)$ time at most for a c-AMIP object searching.*

Proof. The storage structure of the H2SA-ALSH does not have essential differences with the *H*2-ALSH, and we also use QALSH to store and index the data. Algorithm 1 has two parts of overhead: the space of cost by arrived data $T(O(nd))$ and space cost by indexing LSH (QALSH). According to *H*2-ALSH [33], the space overhead of QALSH hash table is $O(n \log n)$. Thus, the space overhead of Algorithm 1 is $O(nd + n \log n)$. To answer a c-AMIP query, in the worst case, Algorithm 2 needs to check objects in all disjoint units, and the schema searches all the units and the search will cost $O(n \log n)$ query time. □

More details, the overhead of $O(n \log n)$ for query time represents the worst case. For the real data sets, the H2SA-ALSH will filter out most of data units, even the data is random distribution or even skewed. The H2SA-ALSH will stop in the first few blocks, and in one block, the schema only searches a few buckets. Therefore, the average query time of a c -AMIP object will be much better than in the worst case.



(a) Approximate ratio on Sift

(b) Approximate ratio on Mnist

(c) Approximate ratio on year

FIGURE 5: Approximation ratio evaluation.

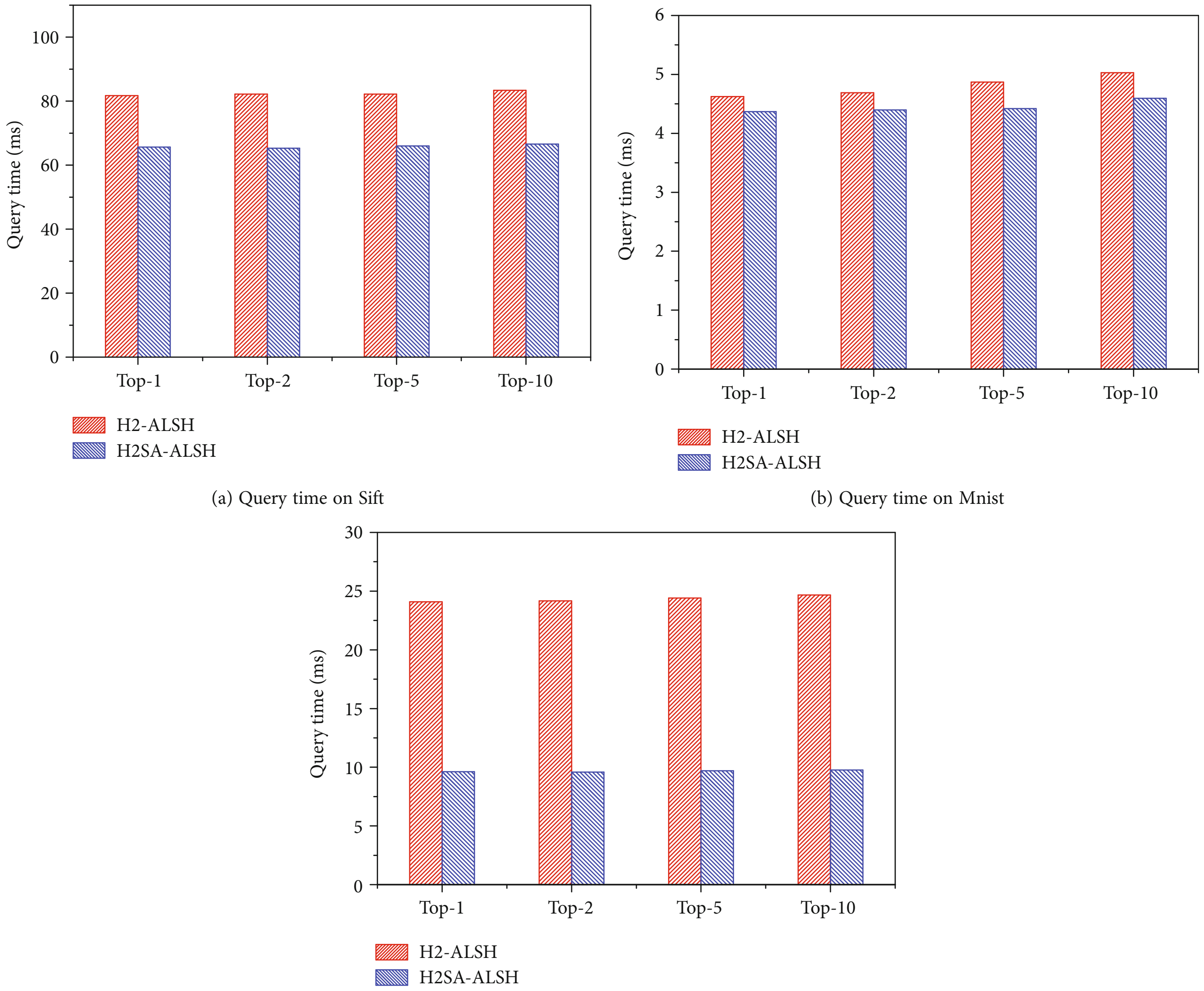
6. Experimental Evaluation

We conduct experiments on three real-world data sets (Mnist [41], Sift [42], and YearPredictionMSD [43] (be termed as Year)) and compare our algorithm with three state-of-the-art AMIP algorithms. The experiments mainly evaluate the precision of AMIP results, the time efficiency of constructing the index, and the query efficiency. We run all the experiments on an Intel Xeon E5-2620 machine with eight cores and 32 GB of memory. All the algorithms in the experiments are implemented by the C++ language and run on Centos 7 OS.

The main evaluation metrics of the experiments are the recall and precision of the AMIP results, overall approximation ratio, and running time of AMIP search. To evaluate the performance of our algorithm, we compare our approach with Simple-ALSH [27], $H2$-ALSH [32], and Sign-ALSH [31]. The experiment verified the performance of all methods for 0.5-k-AMIP search by varying k from 1 to 10 to show the evaluation results of recall and precision. Thus, we get the top-k MIP objects by 0.5-AMIP. Figures 2–4 describe the recall and precision curves of the evaluation. We can see from the curves of Figures 2–4, the H2SA-ALSH is better than those of other algorithms in the top-k searching ($k = 1, 2, 5, 10$), which means that the H2SA-ALSH can obtain more precise search results compared with other algorithms (Simple-ALSH, sign-ALSH, and $H2$-ALSH).

Furthermore, we use the metric of approximation ratio to evaluate the precision of the search results. For the approximate c-k-AMIP search, we set the given approximation ratio c to be 0.5. Then, we compare the approximation ratios of our algorithm with other algorithms. The comparison is conducted under c-k-AMIP searching using top-k searching ($k = 1$, 2, 5, and 10).

The approximation ratio is expressed as $(\langle o, q\rangle / \langle o^*, q\rangle)$, whose value is less than 1. The overall approximation ratio is the average approximation of all queries that can show precision. Additionally, when the ratio is greater, we can obtain



(a) Query time on Sift

(b) Query time on Mnist

(c) Query time on year

Figure 6: Query time evaluation.

better AMIP search results. As shown in Figure 5, the overall approximation ratios of all algorithms are higher than the approximation ratio = 0.5. Our algorithm has a better approximation ratio than all the other algorithms, which means that our algorithm will reach better precision for an arbitrary query.

To examine the query efficiency, we evaluate our algorithm performance on approximate object searching. We compare the average computation time for a query with the latest $H2$-ALSH algorithm. Figure 6 shows that the average query time of our algorithm is less than the time used in $H2$-ALSH over the three data sets. Especially in the year dataset, the query efficiency of our approach improves nearly 60% compared with $H2$-ALSH.

7. Conclusion

In the paper, we propose a novel indexing and searching schema, termed as H2SA-ALSH, in IoT environments. The H2SA-ALSH can construct indexing for multidimensional data objects according to the Euclidean norm and cosine similarity without collecting the raw data objects. At the same time, the extracted indexing features are built with approximate disturbance elements into the features. By collecting and indexing the disturbed features on the fly, we design a c-k-AMIP searching algorithm, to achieve accurate and efficient maximum inner product searching and top-k searching for a given vector. Experiments demonstrate the accuracy and efficiency improvement of our approach compared with three AMIP-based algorithms using real-world data sets.

Data Availability

The authors declare that all the data and materials in this manuscript are available.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant 2021YFB3101305 and by the National Natural Science Foundation of China under Grant 61931019.

References

[1] S. Wang, H. Chen, and Y. Wang, "Collaborative caching for energy optimization in content-centric internet of things," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 1, pp. 230–238, 2022.

[2] Z. Ning, P. Dong, X. Wang et al., "Mobile edge computing enabled 5G health monitoring for internet of medical things: a decentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 463–478, 2020.

[3] J. Soldatos, N. Kefalakis, M. Hauswirth et al., *Openiot: Open Source Internet-of-Things in the Cloud [M]//Interoperability and Open-Source Solutions for the Internet of Things*, Springer, Cham, 2015.

[4] Y. Lu and L. D. Xu, "Internet of Things (IoT) cybersecurity research: A review of current research topics," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2103–2115, 2018.

[5] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): a vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[7] Z. Ning, P. Dong, X. Wang et al., "Mobile edge computing enabled 5G health monitoring for internet of medical things: a decentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 463–478, 2021.

[8] G. Wu, S. Wang, Z. Ning, and B. Zhu, "Privacy-preserved EMR information publishing and sharing: a blockchain-enabled smart healthcare system," *IEEE Journal of Biomedical and Health Informatics*, 2021.

[9] Z. Ning, P. Dong, X. Wang et al., "Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks," *IEEE Transactions on Mobile Computing*, vol. 21, 2020.

[10] Z. Ning, S. Sun, X. Wang et al., "Blockchain-enabled intelligent transportation systems: a distributed crowdsensing framework," *IEEE Transactions on Mobile Computing*, p. 1, 2021.

[11] Z. Ning, S. Sun, X. Wang et al., "Intelligent resource allocation in mobile blockchain for privacy and security transactions: a deep reinforcement learning based approach," *SCIENCE CHINA Information Sciences*, vol. 64, no. 6, pp. 1–16, 2021.

[12] M. G. De Francisci, A. Bifet, L. Khan, J. Gama, and W. Fan, "Iot big data stream mining," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 2119-2120, San Francisco, California, USA, 2016.

[13] S. Chen, H. Xu, D. Liu et al., "A vision of IoT: applications, challenges, and opportunities with China perspective," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 349–359, 2014.

[14] G. Wu, Y. Wang, S. Wang, B. Li, and Y. Liu, "A sketching approach for obtaining real-time statistics over data streams in cloud," *IEEE Transactions on Cloud Computing*, 2020.

[15] G. Wu, X. Yun, S. Wang et al., "Accelerating real-time tracking applications over big data stream with constrained space," in *Database Systems for Advanced Applications*, G. Li, J. Yang, J. Gama, J. Natwichai, and Y. Tong, Eds., vol. 11446 of Lecture Notes in Computer Science, pp. 3–18, Springer, Cham, 2019.

[16] G. Wu, Z. Zhao, G. Fu et al., "A fast kNN-based approach for time sensitive anomaly detection over data streams," in *International Conference on Computational Science*, pp. 59–74, Cham, 2019.

[17] Y. Lu and L. Da Xu, "Internet of things (IoT) cybersecurity research: a review of current research topics," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2103–2115, 2019.

[18] E. Ikonomovska, S. Loskovska, and D. Gjorgjevik, "A survey of stream data mining," in *Proceedings of 8th National Conference with international participation, ETAI*, pp. 19–21, Boston, MA, 2007.

[19] N. Koenigstein, P. Ram, and Y. Shavitt, "Efficient retrieval of recommendations in a matrix factorization framework," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 535–544, Maui, Hawaii, USA, 2012.

[20] H. Li, T. N. Chan, M. L. Yiu et al., "FEXIPRO: fast and exact inner product retrieval in recommender systems," in *Proceedings of the 2017 ACM International Conference on Management of Data*, pp. 835–850, Chicago, Illinois, USA, 2017.

[21] N. Srebro, J. D. M. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," *Advances in Neural Information Processing Systems*, vol. 17, pp. 1329–1336, 2004.

[22] T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, "Fast, accurate detection of 100, 000 object classes on a single machine," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1814–1821, Portland, Oregon, USA, 2013.

[23] P. Jain and A. Kapoor, "Active learning for large multi-class problems," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 762–769, Miami, FL, USA, 2009.

[24] T. Joachims, T. Finley, and C. N. J. Yu, "Cutting-plane training of structural SVMs," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.

[25] R. Spring and A. Shrivastava, "Scalable and sustainable deep learning via randomized hashing," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 445–454, Halifax, NS, Canada, 2017.

[26] A. Auvolat, S. Chandar, P. Vincent, H. Larochelle, and Y. Bengio, "Clustering is efficient for approximate maximum inner product search," 2015, https://arxiv.org/abs/1507.05910.

[27] R. Guo, S. Kumar, K. Choromanski, and D. Simcha, "Quantization based fast inner product search," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pp. 482–490, Cadiz, Spain, 2016.

[28] B. Neyshabur and N. Srebro, "On symmetric and asymmetric lshs for inner product search," in *InInternational Conference on Machine Learning*, pp. 1926–1934, Lille, France, 2015.

[29] A. Shrivastava and P. Li, "Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS)," 2014, https://arxiv.org/abs/1405.5869.

[30] S. Vijayanarasimhan, J. Shlens, R. Monga, and J. Yagnik, "Deep networks with large output spaces," 2014, https://arxiv.org/abs/1412.7479.

[31] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 604–613, Dallas, Texas, USA, 1998.

[32] A. Shrivastava and P. Li, "Improved asymmetric locality sensitive hashing (ALSH) for maximum inner product search (MIPS)," 2014, https://arxiv.org/abs/1410.5410.

[33] Y. Bachrach, Y. Finkelstein, R. Gilad-Bachrach et al., "Speeding up the xbox recommender system using a Euclidean transformation for inner-product spaces," in *Proceedings of the 8th ACM Conference on Recommender systems*, pp. 257–264, Foster City, Silicon Valley, California, USA, 2014.

[34] Q. Huang, G. Ma, J. Feng, Q. Fang, and A. K. Tung, "Accurate and fast asymmetric locality-sensitive hashing scheme for maximum inner product search," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1561–1570, London, United Kingdom, 2018.

[35] B. Zheng, Z. Xi, L. Weng, N. Q. V. Hung, H. Liu, and C. S. Jensen, "PM-LSH," *Proceedings of the VLDB Endowment*, vol. 13, no. 5, pp. 643–655, 2020.

[36] R. Spring and A. Shrivastava, "Mutual information estimation using LSH sampling," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pp. 2807–2815, Yokohama, Yokohama, Japan, 2020.

[37] W. Liu, H. Wang, Y. Zhang, W. Wang, L. Qin, and X. Lin, "EI-LSH: an early-termination driven I/O efficient incremental c-approximate nearest neighbor search," *The VLDB Journal*, vol. 30, no. 2, pp. 215–235, 2021.

[38] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing - STOC '02*, pp. 380–388, Montreal, Quebec, Canada, 2002.

[39] G. Wu, S. Wang, Z. Ning, and J. Li, "Blockchain Enabled Privacy Preserving Access Control for Data Publishing and Sharing in the Internet of Medical Things," *IEEE Internet of Things Journal (IOT-J)*, 2021, early access.

[40] Q. Huang, J. Feng, Y. Zhang, Q. Fang, and W. Ng, "Query-aware locality-sensitive hashing for approximate nearest neighbor search," *Proceedings of the VLDB Endowment*, vol. 9, no. 1, pp. 1–12, 2015.

[41] Y. LeCun, C. Cortes, and C. J. C. Burges, *The MNIST database of handwritten digits*, 1998, New York, USA. http://yann.lecun.com/exdb/mnist/.

[42] L. Amsaleg and H. Jegou, "Datasets for approximate nearest neighbor search," http://corpus-texmex.irisa.fr.

[43] D. Dua and C. Graff, "UCI machine learning repository," (2017), http://archive.ics.uci.edu/ml.