IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 November 2025, accepted 17 December 2025, date of publication 24 December 2025,
date of current version 31 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3648053

RESEARCH ARTICLE

Finding Informative Skyline Results Over Incomplete Data With Threshold-Based Bucket Skyline Algorithm

ZHANG XIAOWEI[1], (Member, IEEE), HAMIDAH IBRAHIM[1], (Member, IEEE), FATIMAH SIDI[1], (Member, IEEE), SITI NURULAIN MOHD RUM[1], AND MUDATHIR AHMED MOHAMUD[2], (Member, IEEE)

[1]Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Selangor 43400, Malaysia
[2]Department of Computer Science, Faculty of Computing, SIMAD University, Mogadishu 2526, Somalia

Corresponding author: Hamidah Ibrahim (hamidah.ibrahim@upm.edu.my)

This work was supported in part by the Ministry of Higher Education Malaysia through the Fundamental Research Grant Scheme under Grant FRGS/1/2020/ICT03/UPM/01/1, and in part by Universiti Putra Malaysia.

ABSTRACT Skyline queries which return objects that are not dominated by any other objects, have been integrated into various real-world database applications. However, the presence of incomplete data in databases, may lead into cyclic dominance relation that none of the objects are considered as skylines, while the transitive dominance property may no longer hold. Although, these issues have received great attention, most of the existing skyline algorithms typically ignore the quality of the returned skyline results, thereby limiting the insights they offer. In this paper, a *Threshold-based Bucket Skyline Algorithm* (*TBSA*) is proposed with the aim to derive informative skyline results, i.e. results that are not too few and with low missing rate. *TBSA* organises the objects in the database into a two-layer structure, namely: *bucket* and *cluster*, to ensure only dominant objects are retained for further skyline analysis. Meanwhile, to minimise the tuning iterations, a *threshold prediction model* is constructed that identifies in advanced the subspaces to be excluded from the skyline computation. Extensive experiments have been carried out on the synthetic and real datasets and the results show that *TBSA* outperforms the previous approach with respect to the number of iterations, pairwise comparisons, skyline results, informative skyline results, and processing time.

INDEX TERMS Informative, skyline query, incomplete data, threshold prediction, bucket skyline algorithm.

I. INTRODUCTION

Skyline queries have drawn a lot of interest from the database community over the years. They are widely used in numerous real-world applications of multiple-criteria decision-making [1], [2]. The result of a skyline query consists of those objects, herein called the set of skylines, S, that are not dominated by any other objects in the database, D, where $S \subseteq D$. An object $o_i \in D$ is said to dominate another object $o_j \in D$, denoted by $o_i \prec o_j$, if and only if for all dimensions, the values of o_i are better or equal to o_j and there is at least one dimension where the value of o_i is better than o_j. As a result, only the preferred objects are maintained, i.e. S, while the uninteresting objects, $D - S$, are filtered/discarded during the skyline computation [3]. A significant number of skyline algorithms were proposed and studied extensively. These skyline algorithms can be categorised into two main categories, namely: non-index-based algorithms and index-based algorithms. The non-index-based algorithms also known as the naïve algorithms include *Block-Nested-Loop* (*BNL*) [4], *Divide-and-Conquer* (*D & C*) [4], *Sort-Filter-Skyline* (*SFS*) [5], *Linear Elimination Sort for Skyline* (*LESS*) [6], and *Sort and Limit Skyline algorithm* (*SaLSa*) [7]. Meanwhile, the index-based algorithms like *Bitmap and Index* [8], *Nearest Neighbour* (*NN*) [9], and *Branch-and-Bound Skyline* (*BBS*) [10] utilised the tree data structure like *R*-tree and *B*-tree to facilitate the searching of skyline objects. Although abundance of skyline algorithms can be found, almost all of

The associate editor coordinating the review of this manuscript and approving it for publication was Sawyer Duane Campbell.

TABLE 1. Samples of data with missing values.

ID	d_1	d_2	d_3	d_4	ID	d_1	d_2	d_3	d_4
$A1$	*	*	*	0.0144	$G3$	*	0.4426	0.9452	*
$A2$	*	*	*	0.115	$H1$	0.6131	*	*	0.9414
$A3$	*	*	*	0.5679	$I1$	0.542	*	0.9398	*
$A4$	*	*	*	0.1799	$J1$	0.4138	0.2672	*	*
$B1$	*	*	0.2191	*	$J2$	0.8182	0.7376	*	*
$C1$	*	0.4346	*	*	$K1$	0.6648	*	0.2112	0.5232
$D1$	0.542	*	*	*	$K2$	0.2904	*	0.0865	0.648
$D2$	0.5183	*	*	*	$K3$	0.6432	*	0.0809	0.8615
$E1$	*	*	0.3497	0.8014	$K4$	0.4559	*	0.6759	0.8429
$E2$	*	*	0.4498	0.8291	$L1$	0.1937	0.8186	*	0.6682
$F1$	*	0.8587	*	0.9518	$M1$	0.7272	0.8345	0.6856	*
$F2$	*	0.4047	*	0.6879	$M2$	0.1816	0.0406	0.1581	*
$F3$	*	0.5757	*	0.9886	$N1$	0.6928	0.2886	0.4933	0.3522
$F4$	*	0.4333	*	0.8879	$N2$	0.5327	0.0615	0.1563	0.8064
$G1$	*	0.1099	0.347	*	$N3$	0.7111	0.425	0.2914	0.9806
$G2$	*	0.1539	0.697	*					

TABLE 2. Analysis of the $|S|$ and S_β based on various values of β.

β	$\|S\|$	$\|S\|$:0	$\|S\|$:1	$\|S\|$:2	$\|S\|$:3	$\|S\|$:4	$\|S\|$:5	$\|S\|$:6	$\|S\|$:7	$\|S\|$:8	$\|S\|$:9	S_β
0	4815	4815	0	0	0	0	0	0	0	0	0	0
0.1	928	542	292	88	6	0	0	0	0	0	0	0.050
0.2	48	27	17	4	0	0	0	0	0	0	0	0.052
0.3	0	0	0	0	0	0	0	0	0	0	0	–
0.5	0	0	0	0	0	0	0	0	0	0	0	–
0.7	0	0	0	0	0	0	0	0	0	0	0	–
0.9	5	0	0	0	0	0	0	0	0	0	5	0.900

them assumed that the data are complete and available. This assumption is unrealistic as databases may suffer from quality issues in particular when the data have been gathered from different sources [11], [12]. The data may be incomplete and the presence of incomplete data may impact the computed skylines if they dominate some other objects with better quality. Furthermore, the incompleteness of data can cause cyclic dominance relation, which makes it impossible to classify any of the objects as skylines. While the transitive dominance property on which most skyline algorithms are based may no longer hold, which renders these algorithms ineffective [13], [14].

Table 1 presents a dataset with some samples of data with missing values which are denoted by the symbol '$*$'. By assuming that lower values are preferable and applying the dominance test over the objects $G1$, $F2$, and $K3$, the following can be observed; $G1 \prec F2$ as $G1$ is better than $F2$ in the second dimension and $F2 \prec K3$ as $F2$ is better than $K3$ in the fourth dimension. Based on the transitive dominance property, since $G1 \prec F2$ and $F2 \prec K3$, then $G1$ should dominate $K3$. However, $K3 \prec G1$ as $K3$ is better than $G1$ in the third dimension, causing a cyclic dominance relation and violating the transitive dominance property. Eventually, none of these objects, i.e. $G1$, $F2$, and $K3$, can be classified as skylines. The skylines, S, based on the given samples of data is given by $S = \{A1, M2\}$. The result is said not *informative* due to two reasons, that are: (i) the number of skyline objects is too few and (ii) although the object $A1 < *, *, *, 0.0144 >$ has the lowest value in the fourth dimension but three of its dimensions are with missing values. Evidently, the returned S does not offer insightful answers to the users' queries. Although, the issues of cyclic dominance relation and transitive dominance property have received great attention in the database community, most of the proposed skyline algorithms [18], [19], [20], [21], [23], [24], [25], [26], [27], [28], [29] do not deal with the quality of the returned skyline results, thereby limiting the insights they offer.

Hence, in this paper we attempt to achieve *informative skyline results*, S_I, for a given incomplete database. The S_I in our work is defined based on the following properties (i) the size of the returned skyline objects, $|S|$ – the number of skyline objects to be returned should not be too few or empty, and (ii) the missing value ratio of the returned skyline objects, S_β– the lesser the ratio of S_β, the more informative is the skyline results. To further clarify the above, we have randomly generated 10,000 objects with independent distribution, each object having 10 dimensions. The missing value ratio, β, is varied from 0 to 0.9 and the $|S|$ and S_β values are recorded. Table 2 presents the number of skylines, $|S|$, returned for each missing value ratio being investigated and the distribution

of the returned skylines according to the number of missing values, where $|S| : i$ denotes the number of skyline objects with i missing values. These are the typical results produced by most skyline algorithms like *Bucket* [18], *ISkyline* [18], and *SOBA* [22] when applied to incomplete data. For instance, $|S| = 4815$ when no missing values are observed in the dataset; while the value of $|S|$ reduced to 5 when the missing value ratio is the highest, i.e. 0.9. It is evident that the higher the missing value ratio, the smaller is the size of the returned skyline objects. Also, the returned skyline objects are less informative as almost all of the objects have missing values in most of their dimensions as depicted by the results of $\beta = 0.9$. Foremost, in some cases like $\beta = 0.3, 0.5$, and 0.7, no skylines are reported.

In this paper, we proposed an efficient skyline algorithm named *Threshold-based Bucket Skyline Algorithm* (*TBSA*). To solve the issue of noninformative skyline results, *TBSA* relies on the *bucket* and *cluster algorithms* with threshold *approach*. There are two main methods that can help users to get more informative skyline results from incomplete data. First, a threshold value, τ, can be utilised to control the size of the subspace in comparing objects across buckets. Due to the monotonicity property of skylines [22], the τ value can be tuned to achieve optimal size of the skyline results. Second, the informative skyline results can be discovered from incomplete data by relaxing the existing dominance notion through the use of variations of skyline queries like k-skyband queries [22], [32], k-dominant skyline queries [31], and top-k dominating queries [33], top-k frequent skyline queries [35]. In our work, the informative skyline results, S_I, is achieved by monitoring and controlling the $|S_I|$ and the S_β values. A threshold value, τ, is introduced to identify in advanced the objects with high incomplete ratio to be discarded from the skyline computation which indirectly control the $|S_I|$ and the S_β values. Nonetheless, identifying the optimal value of τ to be used is not straightforward as the S_I is affected by several factors [15], [16], [17]. To repeatedly adjust the τ value in a trial-and-error fashion or in an incremental manner until the targeted S_I is achieved is inefficient. Hence, in this study a *threshold prediction model* is constructed using machine learning techniques in which the optimal τ value is determined. The least informative objects are discarded until the targeted informative skyline results are obtained. In addition, objects in the database are organised into a two-layer structure, namely: bucket and cluster, to ensure only dominant objects are retained for further skyline analysis. This can significantly avoid unnecessary skyline computation. To summarise, this paper makes the following contributions.

- We have formally introduced the problem of *noninformative skyline results* and justify the significance of addressing the problem.
- We have proposed an efficient skyline algorithm, named *Threshold-based Bucket Skyline Algorithm* (*TBSA*), with the main aim to solve the issue of noninformative skyline results.
- We have constructed a *threshold prediction model* using machine learning techniques which include *multi linear regression* (*MLR*), *support vector regression* (*SVR*), *decision tree regression* (*DTR*), and *neural network regression* (*NNR*). The model is able to identify the optimal threshold value, τ, to be applied in achieving the intended informative skyline results, S_I.
- We have introduced a metric to measure the level of informative of a given set of skyline results.

The remainder of this paper is organised as follows. Section II presents the previous works that are related to this study. Meanwhile, Section III presents the necessary definitions and notations that are used throughout this paper. The phases that are involved in constructing the *threshold prediction model* are presented in Section IV. This is followed with Section V in which our proposed skyline algorithm *TBSA* is discussed in details. The performance of *TBSA* is portrayed in Section VI where the experimental results are reported. Finally, Section VII concludes our work.

II. RELATED WORK

For many years, the database community has valued research on databases with incomplete data as a significant area of study. In the following, we first present the existing skyline algorithms that are mainly designed to tackle the issues related to incompleteness of data. More specifically issues related to cyclic dominance relation and loss of transitive dominance property. Then, we highlight the variations of skyline queries that past studies have examined due to incomplete data with the aim to provide insightful results to the users.

A. SKYLINE QUERIES OVER INCOMPLETE DATA

The existing incomplete skyline algorithms can be classified into four categories based on the approach they used, that are: *replacement-based*, *bucket-based*, *sort-based*, and *table-scan-based approach*.

The *replacement-based* approach seeks to handle incomplete data by simply substituting them with predefined symbols or values. For instance, Khalefa et al. [18], the first to study skyline queries over incomplete databases, proposed replacing each incomplete dimension of an object with $-\infty$. This makes all objects complete which allows the use of conventional skyline algorithms to compute the $S_{sky-\infty}$ of current skylines. Then, the original form of $S_{sky-\infty}$ is restored by replacing $-\infty$ with incomplete dimension, and the final skylines are determined through pairwise comparisons. On the other hand, Arefin and Morimoto [19] replaces missing values with values that are not within the domain values to get the convex skyline sets. Although, the replacement approach significantly improves upon the naïve approach by limiting the exhaustive pairwise comparisons to only the $S_{sky-\infty}$ rather than the entire objects, it is still worse than *Bucket* and *ISkyline* algorithms as demonstrated by [18]. While the replacement-based approach correctly captures the multi-dimensional dominance relationships, some

objects may appear in the skyline solely because they exhibit superior values in one dimension, while other dimensions contain missing values that are replaced with predefined symbols or value. As a result, these objects do not provide full insight to users, leaving them unable to make well-informed decisions.

Meanwhile, the *bucket* approach organises the dataset into buckets based on the bitmap representation of the objects. Since each bucket contains objects with the same bitmap, the conventional skyline algorithms can be employed within each bucket to identify the local skylines. These local skylines also known as the candidate skylines are then compared through standard pairwise comparisons to determine the final skylines. However, this approach has two main drawbacks: (i) it incurs high computation cost due to exhaustive pairwise comparisons among candidate skylines, particularly when their number is large, and (ii) local skylines are identified independently within each bucket without leveraging information from other buckets preventing any reduction in the number of pairwise comparisons. To address these limitations, several enhancements to the bucket approach have been made, as discussed in [18], [20], [21] and [22]. For example, the work in [18] introduces *ISkyline* algorithm, which utilises virtual points and shadow skylines to accelerate the skyline computation. In contrast, Alwan et al. [20] employ a grouping strategy with the virtual k-dom skyline approach to minimise the search space by eliminating unnecessary pairwise comparisons in retrieving the skyline set. Meanwhile, Zhang et al. [21] proposed the *ISSA* algorithm, which starts by identifying the local skylines in each bucket. These skylines are then sorted in ascending order based on an aggregate score calculated from their non-missing dimensions. As a result, objects accessed earlier are more likely to dominate others when determining the final skylines. Additionally, Lee et al. [22] proposed the *SOBA* algorithm, which utilises both bucket-level and point-level methods. Buckets are first sorted in ascending order based on their bitmap representations to identify non-skyline objects at an early stage. Subsequently, objects within each bucket are rearranged to further refine the non-skyline objects. Nonetheless, the *bucket* approach becomes inefficient in high-dimensional data due to the excessive number of buckets generated. Although the bucket approach is the most popular and widely used method because it addresses both optimisation issues and problems of non-transitivity and cyclic dominance, none of the studies using this approach have examined the quality of the resulting skylines. An object with a single value superior to others will always be selected as part of the skyline, even if other attribute values are missing.

The *sort-based* approach aims at minimising the number of pairwise comparisons by pre-sorting the objects in descending order of each dimension and processing them in a round-robin fashion of these dimensions [23], [24], [25]. For instance, Bharuka and Kumar [23] developed the *SIDS* algorithm to perform early elimination of dominated objects by leveraging the pre-sorting strategy. However, *SIDS* is not well-suited for handling large-scale datasets. In contrast, Gulzar et al. [24] combined the pre-sorting with threshold-based method to effectively reduce both the number of comparisons and the overall search space in skyline computation. Meanwhile, Gulzar et al. [25] proposed the *Optimized Incomplete Skyline* (*OIS*) framework, which groups incomplete data into buckets and sorts them by non-missing dimensions in non-ascending order. A threshold then prunes irrelevant items, after which a traditional skyline algorithm identifies local skylines within each bucket. Finally, pairwise comparisons of these local skylines yield the final skyline results. However, the *sort-based* approach involves multiple scans over the objects before skyline objects can be realised, leading to high I/O costs, particularly with large datasets. In sorting-based skyline approaches, missing values can significantly affect the quality of the resulting skylines. Objects with missing attributes may be included in the skyline simply because their known values appear superior, leading to misleading results. Missing values also reduce the discriminative power of sorting, can cause overestimation of dominance, and may introduce inconsistencies in the skyline. As a result, users may make decisions based on incomplete or misleading information.

In order to deal with skyline queries on massive incomplete datasets, He and Han [26] proposed the *Table-Scan-based* (*TSI*) algorithm. It first performs a sequential scan with a pruning operation to eliminate unnecessary objects, then follows with a second scan to refine the candidate skylines and produce the final results. *TSI* was compared to *SOBA* and *SIDS* and demonstrated significantly better performance. Like other approaches, the table-scan method does not account for the quality of the resulting skylines. Consequently, users are not presented with truly informative objects, hindering effective decision-making.

Moreover, several studies have investigated the processing of skyline queries across diverse environments, including cloud-based incomplete databases [27] and incomplete distributed databases [28]. In a related effort, Dehaki et al. [29] introduced efficient algorithms aimed at preventing redundant skyline re-computation in dynamic and incomplete databases. Moreover, Swidan et al. [30] proposed an approach that integrates skyline query processing with missing value estimation in crowdsourced, incomplete databases, thereby ensuring the generation of high-quality skyline results. While these studies collectively address the challenge of data incompleteness, each primarily focuses on developing solutions tailored to the specific characteristics and requirements of its respective environment.

Although the above solutions [18], [19], [20], [21], [23], [24], [25], [26], [27], [28], [29] effectively address optimisation issues, cyclic dominance, and the loss of transitive dominance, they do not consider the quality of the resulting skyline. While the skyline results satisfy the dominance criteria, instances exist where objects with fewer non-missing

values are selected as skylines simply because one of their known attributes is superior to others. Conversely, some objects with complete information are excluded from the skyline because they are dominated by objects with a high proportion of missing values, even when the superior values show only marginal differences.

B. VARIATIONS OF SKYLINE QUERIES ON INCOMPLETE DATA

This section discusses the variations of skyline queries that are explored by previous studies to solve different aspects of skyline problems, such as *k-skyband queries*, *k-dominant skyline queries*, *top-k frequent skyline queries*, *top-k dominating skyline queries*, etc. The main aim of these various types of queries is to help users to find interesting/informative objects of a given incomplete dataset.

To address the challenge of skyline queries producing uncontrollable and often too few results, Lee et al. [22] introduced the *SOBA* algorithm. This ranking method fine-tunes two user-specific parameters: it adjusts the size of common subspaces among objects and relaxes the number of dominating objects. Initially, a threshold value is set and incrementally increased until a desirable number of skylines is returned. Also, the dominance notion is relaxed, similar to the concept used in the *k-skyband queries*, allowing meaningful skyline results to be returned. Nonetheless, the need for repeating the pairwise comparisons during each threshold adjustment makes the process computationally expensive. Meanwhile, Miao et al. [31] address the problem of *k-dominant skyline queries* on incomplete data (*IKDS*) by controlling the number of returned results and relaxing the traditional dominance to *k-dominance*. Three algorithms are proposed, namely: *Baseline*, *Dominance Ability Guided* (*DAG*), and *Bitmap Index Based* (*BIB*), which identify a subset of superior objects among the skyline objects. The *DAG* algorithm minimises the searching space, while the *BIB* algorithm employs the bitmap indexing to boost the search performance over incomplete dataset. Additionally, they proposed two efficient algorithms to tackle two interesting variants of *IKDS* queries, that are the *weighted dominant skyline* and the *top-£ dominant skyline query*.

Various studies have investigated ways to extract informative skyline results from incomplete datasets, using a dominance model based on how many objects each object dominates [32], [33]. Gao et al. [32], for instance, handle *k-SkyBand* skyline queries by retrieving objects that are dominated by at most k other objects. They use a *virtual point-based* method and *k-skyband algorithm for incomplete data* (*KISB*) and introduce new concepts like *expired skyline*, *shadow skyline*, and *thickness warehouse* to improve the search performance of processing the queries. These techniques are also extended to support *constrained skyline* and *group-by skyline queries* over incomplete data. In contrast, Miao et al. [33] focus on finding the k most significant objects that dominate the maximum number of objects in an incomplete dataset. To improve the efficiency of processing the *top-k dominating queries*, they use techniques such as *upper bound score pruning*, *bitmap pruning*, and *partial score pruning*.

Some studies identify the k most representative skyline results by evaluating the membership of objects in subspace skylines, assuming that objects found in more subspaces are more interesting [34], [35]. Chan et al. [34] proposed efficient approximate algorithms to identify the *top-k frequent skyline* points in large, high-dimensional complete datasets, where interesting objects are those that frequently appear as skylines across various subspaces. Meanwhile, Bharuka and Kumar [35] extended the *top-k frequent skyline* approach for complete datasets to incomplete datasets, retrieving k superior objects ranked by their fractional skyline frequency.

Probabilistic approaches have also been proposed for skyline computation on uncertain data [36] and incomplete data, where each object is assigned a probability of being a skyline, and the k objects with the highest probabilities are selected [37]. The work by Zhang et al. [13] investigates probabilistic skyline computation on three types of incomplete datasets—independent, correlated, anti-correlated—and enhances the skyline computation performance using *pruning strategy* and *sorting technique*.

Although the works discussed in this section [13], [22], [31], [32], [33], [35] aim to identify the top-k most interesting objects from a given incomplete dataset, they largely focus on a single property: dominance strength, defined by the number of objects an object dominates. However, other critical factors—such as the degree of incompleteness and the distance to optimal (i.e., deviation from the optimal object)—are neglected [38]. This paper seeks to address and incorporate these overlooked properties to enhance the identification of informative skyline results.

III. PRELIMINARIES

In this section, we present the necessary definitions and introduce the notations that are used throughout this paper. First, we provide the general definitions that have been established either formally or informally in the literatures [4], [10] and [39] based on the notations used in this paper (i.e. *Definition 1* till *Definition 7*), then we present the specific definitions that are related to our work in this paper (i.e. *Definition 8* till *Definition 11*). This is then followed by the descriptions of the problem tackled in this paper. Throughout this paper, we assume that all values of an object have a total order except the missing values, and smaller values are preferable than larger values.

Definition 1 (Dominance): Given a database, D, with m dimensions, $d = \{d_1, d_2, \ldots, d_m\}$ and n objects $\mathrm{D} = \{\mathrm{o}_1, \mathrm{o}_2, \ldots, \mathrm{o}_n\}$, o_i is said to dominate o_j denoted by $o_i \prec o_j$ where $i \neq j$ if and only if the following condition holds: $\forall d_k \in d, o_i.d_k \leq o_j.d_k \wedge \exists\, d_l \in d, o_i.d_l < o_j.d_l$. For instance, the object $N2 < 0.5327, 0.0615, 0.1563, 0.8064 >$ is said to dominate the object $N3 < 0.7111, 0.425, 0.2914, 0.9806 >$, i.e. $N2 \prec N3$, since it is better than $N3$ in all the dimensions.

Definition 2 (Transitive Dominance Property): Given the objects $o_i, o_j, o_k \in D$ where $i \neq j \neq k$, if $o_i \prec o_j$ and $o_j \prec o_k$, this implies that $o_i \prec o_k$. The transitive dominance property can significantly reduce the number of pairwise comparisons between objects.

Definition 3 (Incomparable): Given the objects $o_i, o_j \in D$ where $i \neq j$, if $o_i \not\prec o_j$ and $o_j \not\prec o_i$, then o_i and o_j are said to be incomparable. For instance, the object $N1 < 0.6928, 0.2886, 0.4933, 0.3522 >$ is better than $N2 < 0.5327, 0.0615, 0.1563, 0.8064 >$ in the fourth dimension while $N2$ is better than $N1$ in the first, second, and third dimensions. We can neither conclude that $N1 \prec N2$ nor $N2 \prec N1$. Hence, $N1$ and $N2$ are incomparable. Likewise, objects with missing values are incomparable if none of their non-missing values are over the common dimensions. For instance, the objects $C1 < *, 0.4346, *, * >$ and $K1 < 0.6648, *, 0.2112, 0.5232 >$ are incomparable.

Definition 4 (Skyline): An object $o_i \in D$ is a skyline of D if there is no other objects, say $o_j \in D$ where $i \neq j$, that dominates o_i. Correspondingly, a bucket skyline (cluster skyline), $o_i \in b_l$ ($o_i \in C_l$, respectively), is an object not dominated by other objects, say $o_j \in b_l$ ($o_j \in C_l$, respectively), where $i \neq j$ and b_l is a bucket (C_l is a cluster, respectively).

Definition 5 (Incomplete Database (Dataset):) A database D is said to be incomplete if at least one of its objects, say $o_i \in D$, has missing values in one or more of its dimensions, otherwise D is said to be complete. The example given in Table 1 reflects an incomplete dataset.

Definition 6 (Dominance over Incomplete Database:) Given an incomplete database, D, with m dimensions, $d = \{d_1, d_2, \ldots, d_m\}$ and n objects $D = \{o_1, o_2, \ldots, o_n\}$, o_i is said to dominate o_j where $i \neq j$ denoted by $o_i \prec o_j$ if and only if the following condition holds: $\forall d_k \in d', o_i.d_k \leq o_j.d_k \wedge$

$\exists d_l \in d', o_i.d_l < o_j.d_l$ where $d' \subseteq d$ in which both objects have no missing values on each d_r of d'. For instance, the object $E1 < *, *, 0.3497, 0.8014 >$ is said to dominate the object $E2 < *, *, 0.4498, 0.8291 >$, i.e. $E1 \prec E2$, where $d' = \{d_3, d_4\}$.

Definition 7 (Skyline over Incomplete Data): An object $o_i \in D$ is a skyline of D, where D is an incomplete database, if there is no other objects, say $o_j \in D$ where $i \neq j$, that dominates o_i. For instance, $A1$ and $M2$ are the skylines of the samples of data given in Table 1. Hence, the set of skylines is given by $S = \{A1, M2\}$.

Definition 8 (Informative Skyline Results): The set of skylines is considered to be informative, S_I, based on the following properties:

(a) the size of the returned skyline objects, $|S|$ – the number of skyline objects to be returned should not be too few or empty. In this regard, S_α is used to denote whether the $|S|$ is too few, $S_\alpha = 0$, or otherwise, $S_\alpha = 1$. Here, $|S|_u$ is expected which reflects the preferred minimum number of skyline objects to be returned as defined by the user. If $|S| \geq |S|_u$, then $S_\alpha = 1$. This means that the $|S|$ has fulfilled the number of skyline objects as indicated by the user. Otherwise, $S_\alpha = 0$ which means that the $|S|$ is too few than the specified $|S|_u$.

(b) the missing value ratio in the returned skyline objects, S_β – the lesser the value of S_β, the more informative is the skyline results. S_β represents the proportion of missing values in the returned skyline objects and is calculated as: $S_\beta = \frac{|*|}{|d| \times |S|}$ where $|*|$ is the total number of missing values in S, $|d|$ is the number of dimensions, and $|S|$ is the number of skyline objects. We use the notation S_I to indicate that the returned set of skylines is based on a subspace of the database D, while the set of skylines S is assumed to be derived based on the whole space of D.

Definition 9 (Level of Informative Skyline Results:) In this paper, we introduced a metric to measure the level of informative of a given S_I, denoted by L_{S_I} based on the properties defined in *Definition* 8. The following equation defined the metric L_{S_I}:

$$L_{S_I} = \omega_1 S_\alpha + \omega_2 \left[1 - S_\beta\right]$$
$$S_\alpha = \begin{cases} 1 & if\ |S| \geq |S|_u \\ 0 & otherwise \end{cases} \quad (1)$$

where S_α is either 0 or 1 which indicates either $|S|$ is too few or otherwise, respectively; S_β is the missing value ratio in the returned S; ω_i is a value between [0, 1], a weight value given to the i-th property with $\omega_1 + \omega_2 = 1$; and $|S|_u$ is the preferred number of skyline objects as defined by the user. The value of L_{S_I} lies between [0, 1] with values heading towards 0 indicate less informative and those towards 1 are more informative. For instance, the set of skylines for the samples of data given in Table 1 is $S = \{A1, M2\}$. Assume that $|S|_u = 4$ and $\omega_1 = \omega_2 = 0.5$, then $L_{S_I} = 0.5(0) + 0.5\,[1 - 0.5] = 0 + 0.25 = 0.25$, which reflects less informative. Meanwhile, given $S_\alpha = 1$, $S_\beta = 0$, and $\omega_1 = \omega_2 = 0.5$, then $L_{S_I} = 0.5(1) + 0.5\,[1 - 0] = 0.5 + 0.5 = 1$. Here, the returned set of skylines, i.e. S, is considered more informative because $|S|$ not only exceeds $|S|_u$, but is also complete.

Definition 10 (Least Informative Objects:) Given an incomplete database, D, an object $o_i \in D$ is said to be the least informative object if its o_{i_β} value is the highest. The objects $A1$, $A2$, $A3$, $A4$, $B1$, $C1$, $D1$, and $D2$ given in Table 1 are the least informative objects as compared to the other objects in the table since their β is the highest, i.e. 0.75.

Definition 11 (τ-Least Informative Objects): The threshold value, τ, is a value between [0, 1] that is used to identify the least informative objects to be filtered out from the skyline analysis. In general, objects with $1 - \beta \leq \tau$ will be omitted. For instance, given $\tau = 0.3$, then the objects $A1$, $A2$, $A3$, $A4$, $B1$, $C1$, $D1$, and $D2$ are omitted from the skyline analysis. τ controls the minimum size of the common subspace required to compare objects across buckets. It helps to filter out objects with a high ratio of missing values. If the size of the common subspace between two objects is less than or equal to $\tau \times |d|$, the objects are considered

incomparable, and the dominance test between them is skipped. For example, two objects are considered incomparable if the size of their common subspace is less than or equal to $\tau \times |d|$. Assume $\tau = 0.3$ and $|d| = 4$; in this case, the objects are deemed incomparable if the common subspace size is less than or equal to 1.2. Since the subspace size must be an integer, the objects are therefore incomparable when the common subspace size is 0 or 1. Lee et al. [22] proved that the size of skyline results exhibits a monotonicity property with respect to the threshold parameter. Using this property, if the size of the skyline result is too small, the threshold value is incrementally increased by $\frac{1}{|d|}$ in each iteration until skylines of a target size are obtained.

Problem Formulation – The skyline query algorithms exploit all kinds of data pruning and indexing methods based on the transitive dominance property to achieve the set of skylines, S. However, the transitive dominance property does no longer work in the presence of incomplete data. Enforcing the conventional skyline query algorithms over incomplete database will result in a set of skylines that is noninformative due to the number of skylines returned is too few or empty and/or the ratio of missing values in the returned skylines is too high. Hence, this paper aims to achieve *informative skyline results*, S_I, with high value of L_{S_I} for a given incomplete database. In satisfying the above aim, the following research questions are to be addressed: (i) What would be the optimal subspace to be considered in the skyline computation given an incomplete database D with size $|D|$, m number of dimensions, β missing value ratio, DD type of data distribution, and $|S|_u$ number of returned skylines as defined by the user? (ii) How can we control the value of $|S_I|$ and S_β to achieve a high value of L_{S_I}?

IV. THRESHOLD PREDICTION MODEL

The *informative skyline results*, S_I, with high L_{S_I} value for a given incomplete database can be achieved by monitoring and controlling the following parameters: (i) the size of the returned skyline objects, $|S|$ – the number of skyline objects to be returned should not be too few or empty, and (ii) the missing value ratio in the returned skyline objects, S_β – the lesser the value of S_β, the more informative is the skyline results. In our work, a threshold value, τ, is introduced to identify in advanced the least informative objects to be discarded from the skyline computation which indirectly control the $|S_I|$ and the S_β values. Since S_I is affected by several factors, tuning the τ value in a trial-and-error fashion or in an incremental manner until S_I is achieved is unwise and costly. Hence, in this work a *threshold prediction model* is constructed using machine learning techniques, which include: *multi linear regression* (*MLR*), *support vector regression* (*SVR*), *decision tree regression* (*DTR*), and *neural network regression* (*NNR*). Given an incomplete database, the model should be able to predict the optimal value of τ to be applied in achieving the *informative skyline results*, S_I, with high L_{S_I} value in lesser number of iterations. In constructing the *threshold prediction model*, the following phases are followed:

(i) *Features Identification* – Before the *threshold prediction model* can be constructed, the features to be incorporated into the model must be identified. In regard to this, we have generated three synthetic datasets with different data distributions (*DD*), namely: independent (*IND*), correlated (*COR*), and anti-correlated (*ANT*). Each dataset consists of 1,000,000 objects with 30 dimensions. We then conducted three analyses to investigate the effect of (i) missing value ratio, β, (ii) size of the dataset, $|D|$, and (iii) number of dimensions, $|d|$, on the size of the returned skylines, $|S|$. Each analysis is based on 10 runs in which the average results are reported. Figure 1 (a) presents the results of $|S|$ on the three types of data distributions when the missing value ratio, β, is varied from 0.1 to 0.9, while $|d|$ and $|D|$ are fixed to 10 and 1000, respectively. Obviously, $|S|$ decreases when the missing value ratio, β, increases. Interestingly, $|S| = 0$ when $\beta = 0.5$, 0.6, and 0.7. Furthermore, when $\beta = 0.1$ and 0.2, the returned $|S|$ on anti-correlated is larger than those returned by the independent and correlated datasets. Nevertheless, $|S|$ on these datasets is almost equal when β ranges from 0.3 to 0.9. Meanwhile, Figure 1 (b) presents the results of $|S|$ on the three different data distributions when the size of the dataset, $|D|$, is varied from 1,000 to 1,000,000. Here, we set $|d| = 10$ and $\beta = 0.2$. The results clearly show that $|S|$ decreases when the size of D increases. In particular, for the correlated dataset, $|S| = 66$ when $|D| = 1,000$ and $|S| = 2$ when $|D| = 10,000$, which is close to an empty result. Furthermore, when $|D| = 1,000$, the returned $|S|$ on the anti-correlated dataset is larger than the other two datasets; nonetheless these datasets showed lesser number of $|S|$, which is almost empty, when $|D| = 500,000$ and $|D| = 1,000,000$. On the other hand, $|S|$ increases when the number of dimensions, $|d|$, increases which is presented in Figure 1 (c) with the number of dimensions, $|d|$, varied from 2 to 30 while $|D|$ and β are fixed to 10,000 and 0.2, respectively. Here, when $|d|$ is varied from 2 to 10, the returned $|S|$ on these three datasets is almost equal. Nonetheless, when $|d|$ is varied from 15 to 30, the returned $|S|$ on the anti-correlated dataset is the largest as compared to the other datasets, namely: independent and correlated. From these analyses, it is obvious that $|S|$ is significantly affected by the β, $|D|$, $|d|$, and type of data distributions, i.e. independent, correlated, and anti-correlated. Thus, these parameters are chosen as the features of the *threshold prediction model*.

(ii) *Dataset Preparation* – In this phase, three main datasets based on different data distribution, *DD*, are prepared to be employed during the training and testing of the *threshold prediction model*. Each type of dataset, herein called the τ dataset, is with different numbers of objects which reflect different sizes, i.e. 1,000, 3,000, and 5,000. The τ datasets are prepared based on the three features identified in phase (i) that are size of the dataset, $|D|$, number of dimensions, $|d|$, and missing value ratio, β, with their values being randomly generated between 100 to 10,000, 2 to 15, and 0.1 to 0.9, respectively. By employing the *bucket* and *cluster algorithms* (explain in Section V), objects of the incomplete dataset

TABLE 3. The mean square errors of MLR, SVR, DTR, and NNR.

τ Dataset	*MLR*	*SVR*	*DTR*	*NNR*
1,000	0.174	0.173	**0.154**	0.279
3,000	0.181	0.184	**0.145**	0.262
5,000	0.185	0.185	**0.133**	0.189

are organised into a two-layer structure, namely: bucket and cluster. We then run the skyline algorithm over these clusters of objects. If the returned $|S| \geq |S|_u$, then $\tau = 0$. Otherwise, the τ value is incrementally increased with $\frac{1}{|d|}$ increment in each iteration, until the condition $|S| \geq |S|_u$ is satisfied. The final τ value is then recorded. We use a small increment of τ in each iteration to ensure that the returned $|S|$ is not less than and is close to the given $|S|_u$ value. Figure 2 shows the steps described above with some samples of the τ dataset with $|S|_u = 10$.

(iii) *Model Construction* – The *threshold prediction model* is constructed based on the regression model which include *multi linear regression* (*MLR*), *support vector regression* (*SVR*), *decision tree regression* (*DTR*), and *neural network regression* (*NNR*). The τ datasets with different numbers of objects prepared in phase (ii) are employed in this phase. These datasets are split into two sets with 80% used as a training set while 20% is used as a testing set. Each regression model is trained with the training dataset which is then followed by testing against the testing dataset. We analysed the results based on mean square error, as shown in Table 3. The results show that for all datasets, the *DTR* achieved the lowest error rate. Meanwhile, figures 3 (a), 3 (b), 3 (c), and 3 (d) show the fitting diagram of the *MLR*, *SVR*, *DTR*, and *NNR*, respectively. Figure 3 (c) shows that the predicted threshold values of *DTR* are between 0 and 1 which implies that the predicted values of *DTR* in most cases matched with the real test values. On the other hand, the predicted threshold values of the other models (*MLR*, *SVR*, and *NNR*) in some cases show a value that is either larger than 1 or smaller than 0. Based on these results, the *DTR* is selected and embedded into our *Threshold-based Bucket Skyline Algorithm* (*TBSA*).

V. THRESHOLD-BASED BUCKET SKYLINE ALGORITHM (*TBSA*)

This section explains our proposed skyline algorithm named *Threshold-based Bucket Skyline Algorithm* (*TBSA*) which is mainly designed to solve the issues related to processing skyline queries in the presence of incomplete data. *TBSA* aims at deriving informative skyline results, S_I, by employing the *bucket* and *cluster* algorithms with *thresholding approach*. We have designed *TBSA* in two main phases, namely: (a) Pre-processing phase – this phase aims at extracting the features of a given incomplete database and grouping the objects of the database into a two-layer structure, namely: bucket and cluster; and (b) Skyline Query Processing phase – this phase contains the necessary steps with the aim to derive the informative skyline results, S_I, given the number of skylines objects to be returned, $|S|_u$, as specified by the user. These phases are further explained in the following subsections. Figure 4 presents the *TBSA* flow diagram.

A. PRE-PROCESSING PHASE

The Pre-processing phase contains three main steps that are performed only once for a given incomplete database, $|D|$. These steps are: (i) Extract the features of the incomplete database, $|D|$, (ii) Organise the objects of the incomplete database, $|D|$, into buckets and derive the bucket skylines; and (iii) Organise the buckets into clusters and derive the cluster skylines. The two-layer structure formed in this phase is utilised in the second phase, *Skyline Query Processing*.

Extract the Features of the Incomplete Database, D– In this step, the features of the incomplete database, D, are extracted. These features are necessary for the *threshold prediction model* to determine the optimal threshold value, τ, to be employed in deriving the informative skyline results, S_I. We use the notation D_F to denote the set of features of the database D and is defined as follows: $D_F < DD, |D|, |d|, D_\beta >$ where DD is the type of data distribution, $|D|$ is the number of objects in D, $|d|$ is the number of dimensions of D, and D_β is the missing value ratio of D that is calculated by the following formula: $\frac{|*|}{|d| \times |D|}$ where $|*|$ is the number of missing values in D.

Organise the Objects of the Incomplete Database, D, into Buckets and Derive the Bucket Skylines – Given an incomplete database, D, with m dimensions, $d = \{d_1, d_2, \ldots, d_m\}$ and n objects $D = \{o_1, o_2, \ldots, o_n\}$, each object $o_i \in D$ is associated to a bucket b_l based on its bitmap representation. The bitmap representation of an object o_i is derived based on the following conditions: if $o_i.d_k \neq' *'$, then the k-th dimension of the bitmap representation of o_i is assigned the bit 1; and 0 otherwise. For instance, the bitmap representations of the objects $A1 < *, *, *, 0.0144 >$ and $N3 < 0.7111, 0.425, 0.2914, 0.9806 >$ are 0001 and 1111, respectively. Hence, objects with missing values in the same dimensions will have the same bitmap representation and they are grouped in the same bucket. Then, the set of bucket skylines, S_{b_l}, for each bucket, b_l, is derived by employing the conventional skyline algorithm. The complete set of buckets, $B = \{b_1, b_2, \ldots, b_l\}$, is produced depending on the number of distinct bitmap representations of the objects in the D collection. We use the following notation to represent a bucket, $b_l < bmr_{b_l}, b_{l_\beta}, S_{b_l}, |S_{b_l}| >$ where bmr_{b_l} is the bitmap representation of bucket b_l which is based on the bitmap representation of the objects in its collection, b_{l_β} is the missing value ratio of bucket b_l, S_{b_l} is the set of bucket skylines of b_l, and $|S_{b_l}|$ is the number of bucket skylines of b_l. We employ the bucket strategy for several reasons: (i) It is commonly used by previous works like [18], [20], [21], [22], [24], [25], [27], [28] and [29], to resolve the issues of cyclic dominance relation and loss of transitive dominance property, (ii) the conventional skyline algorithm can be easily applied over each bucket to derive the bucket skylines, hence ensuring that only the dominant objects of each bucket are retained

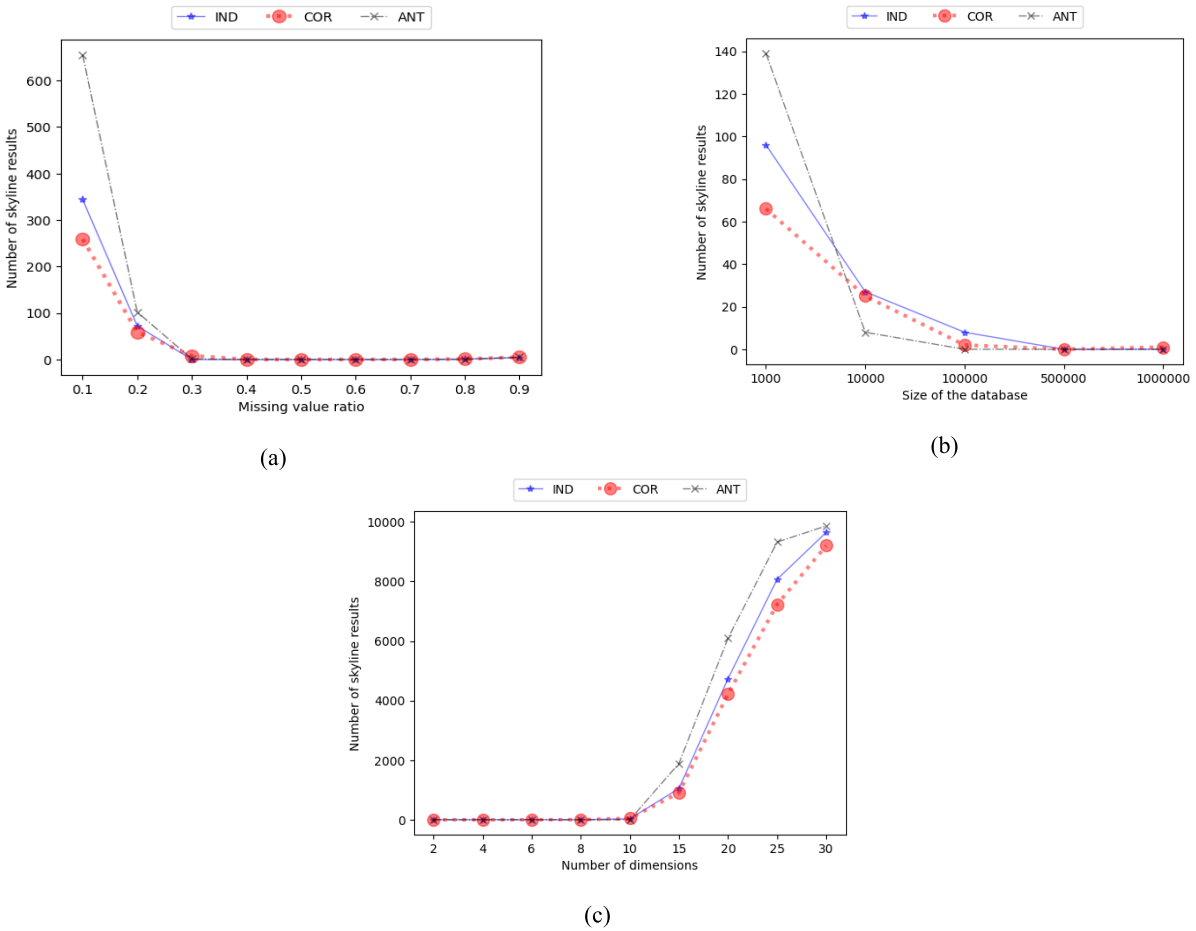
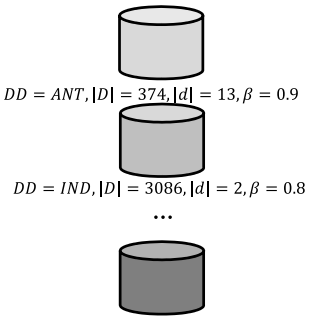


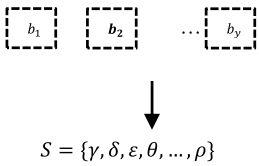
FIGURE 1. Analysis of $|S|$ with various values of β, $|D|$, and $|d|$ over different *DD*.

DD	$\|D\|$	$\|d\|$	β	τ
ANT	374	13	0.9	*
IND	3086	2	0.8	*
COR	9812	6	0.9	*
COR	5381	8	0.3	*
ANT	5635	6	0.3	*
IND	2605	3	0.7	*
IND	4737	11	0.1	*
ANT	5360	13	0.2	*
IND	6599	4	0.3	*
IND	6490	10	0.5	*

Construct the τ dataset with $|D|$, $|d|$, and β values generated randomly for different *DD*.



Build the incomplete datasets according to the features of each τ dataset.



Apply the *bucket* and *cluster* algorithms and the *skyline algorithm* to derive S of the incomplete dataset. If $|S| \geq |S|_u$, then $\tau = 0$.

DD	$\|D\|$	$\|d\|$	β	τ
ANT	374	13	0.9	1
IND	3086	2	0.8	0.5
COR	9812	6	0.9	1
COR	5381	8	0.3	0.375
ANT	5635	6	0.3	0.666
IND	2605	3	0.7	0.666
IND	4737	11	0.1	0
ANT	5360	13	0.2	0
IND	6599	4	0.3	0.75
IND	6490	10	0.5	0.7

Increment τ until $|S| \geq |S|_u$ and save the final τ value.

FIGURE 2. The steps of constructing the τ datasets.

for further skyline analysis; and (iii) the missing value ratio of each bucket, b_{l_β}, can be easily determined. Based on the samples of objects given in Table 1, the buckets that are derived are as shown in Figure 5. Altogether 14 buckets are derived, $B = \{b_1, b_2, \ldots, b_{14}\}$; while objects shaded in gray are the bucket skylines. The bucket b_1, for instance, is defined as $< 0001, 0.75, \{A1\}, 1 >$. The distributions of missing value ratio are as follows: 4/14 (28.6%) of the buckets have 0.75 missing values; 6/14 (42.9%) of the buckets have 0.5 missing values, 3/14 (21.4%) of the buckets have 0.25 missing values, and 1/14 (7.1%) of the buckets have no missing values.

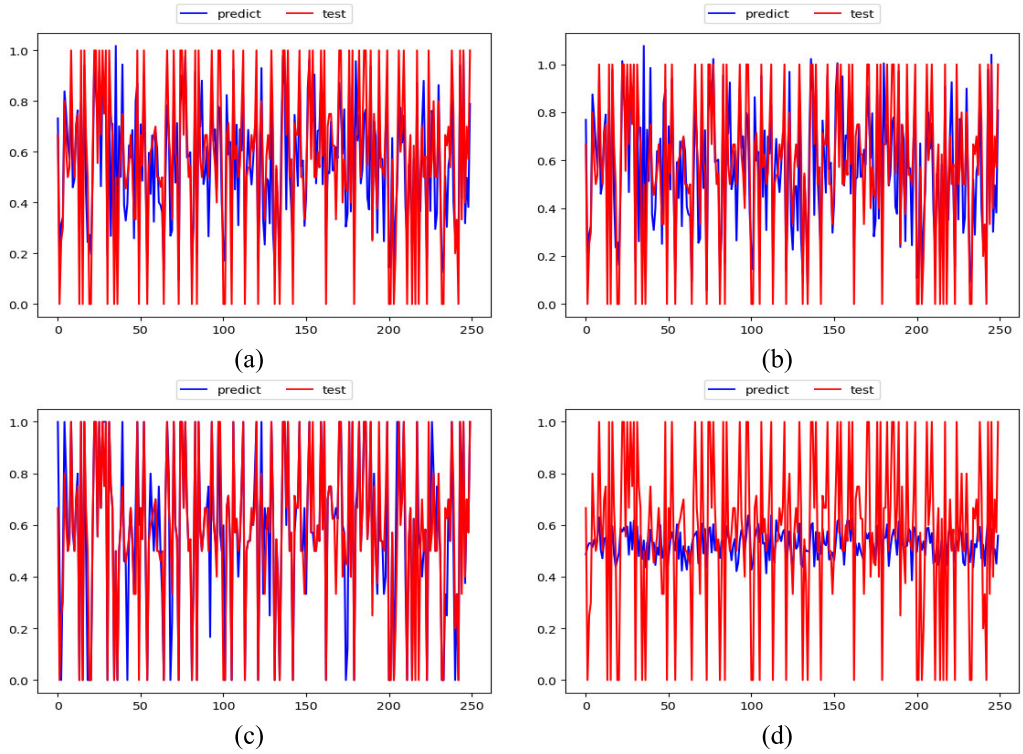


FIGURE 3. The fitting diagram of (a) MLR, (b) SVR, (c) DTR, and (d) NNR.

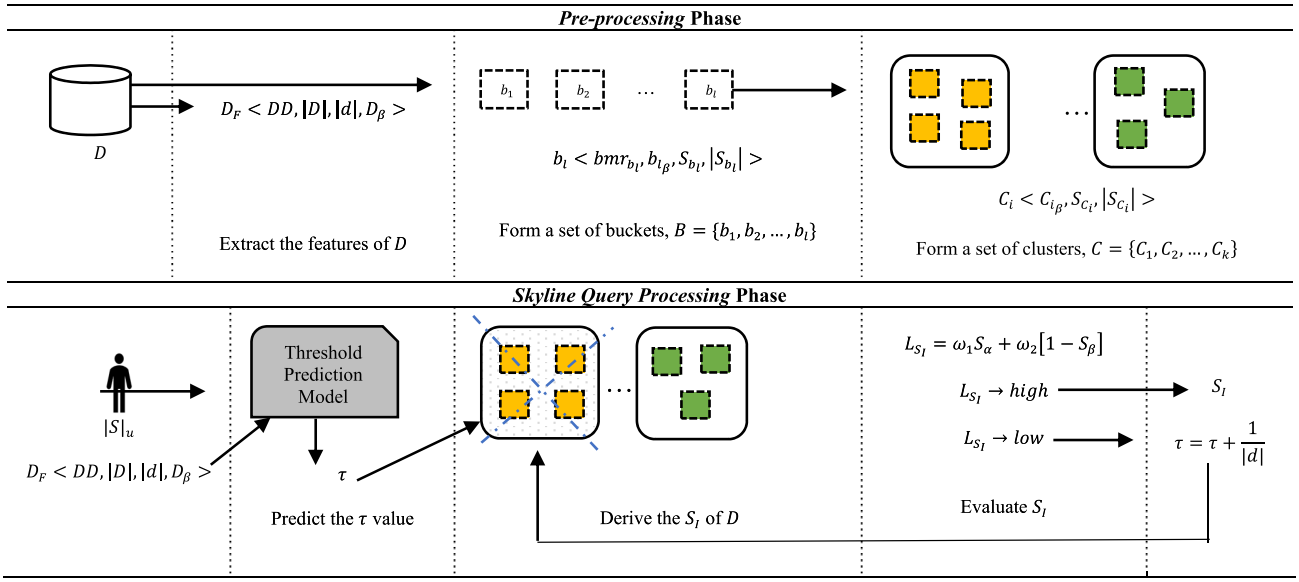


FIGURE 4. TBSA flow diagram.

Organised the Buckets into Clusters and Derive the Cluster Skylines – Given a set of buckets, $B = \{b_1, b_2, \ldots, b_l\}$, where each bucket contains the bucket skylines derived by Step (ii) above, this step focuses on grouping buckets that share the same missing value ratio, β, into the same cluster, C_i. Specifically, buckets with the same missing value ratio $\beta = 1 - \frac{i}{|d|}$ where i is the number of 1s in the bucket's bitmap representation are assigned to the same cluster. For example, consider $b_{1_\beta} = b_{2_\beta} = b_{3_\beta} = b_{4_\beta} = 1 - \frac{1}{4} = 0.75$, corresponding to bitmap representation with a single 1, i.e. 0001, 0010, 0100, 1000, respectively. As a result, these buckets are grouped into the same cluster, i.e. C_1 in this example. The number of clusters to be formed, $C = \{C_1, C_2, \ldots, C_k\}$, depends on the number of distinct missing value ratio of

the buckets in B. By clustering the buckets, it assists us in determining the subspace(s) that should be omitted in the process of deriving the informative skyline results, S_I. Also, objects in the cluster can be further refined to ensure that only the dominant objects of each cluster are retained for further skyline analysis. We use the following notation to represent a cluster, $C_i < C_{i_\beta}, S_{C_i}, |S_{C_i}| >$ where, C_{i_β} is the missing value ratio of cluster C_i, S_{C_i} is the set of cluster skylines of C_i, and $|S_{C_i}|$ is the number of cluster skylines of C_i. Figure 6 presents the results of grouping the buckets presented in Figure 5 into clusters. Since there are four distinct β that are 0.75, 0.5, 0.25, and 0, then four clusters, C_1, C_2, C_3, and C_4 are formed, respectively. The cluster skylines of a cluster, C_i, denoted by S_{C_i}, are derived by comparing the objects between the buckets of the cluster, C_i. Nonetheless, there are objects that are not comparable, for instance the object, o_i, with the bitmap representation 0011 is said not comparable to the object o_j with the bitmap representation 1100 since the non-missing values of these objects are not over the common dimensions (see Definition 3). Hence, to decide if the buckets b_k and b_l are comparable, a *revised bitwise* is derived by comparing each bit of both buckets' bitmap representations. If the i-th bit of both values are 1 then the i-th bit of the revised bitwise is assigned the value 1, and 0 otherwise. A revised bitwise with only 0s implies that these buckets are not comparable. For instance, the revised bitwise for the buckets b_1 and b_2 is 0000 which means they are not comparable. On the other hand, the revised bitwise of the buckets b_5 and b_6 is 0001 which means these buckets are comparable over the fourth dimension.

The cluster skylines, S_{C_i}, of a cluster, C_i, can then be derived by employing the conventional skyline algorithm over the comparable buckets of C_i. The objects shaded in grey as shown in Figure 6 are the cluster skylines of the clusters. The cluster C_1, for instance, is defined as $< 0.75, \{A1, B1, C1, D2\}, 4 >$.

B. SKYLINE QUERY PROCESSING PHASE

The *Skyline Query Processing* phase contains three main steps that are performed when a skyline query, q_p, and the required minimum number of returned skylines, $|S|_{u_p}$, are submitted by a user, u_p. The steps performed in this phase are as follows: (i) Get the threshold value, τ, from the *threshold prediction model*, (ii) Derive the informative skyline results, S_I, of the incomplete database, D, and (iii) Evaluate the *Level of Informative Skyline Results*, L_{S_I}, and refine the informative skyline results, S_I.

Get the Threshold Value, τ, from the Threshold Prediction Model – In this step, the features of the incomplete database, D, that are extracted in Step (i) of the *Preprocessing* phase are submitted to the *threshold prediction model* in which a threshold value, τ, is returned. The *threshold prediction model* utilised the $D_F < DD, |D|, |d|, D_\beta >$, of the database D in predicting the optimal τ to be considered in the derivation of the informative skyline results, S_I, of D.

Derive the Informative Skyline Results, S_I, of the Incomplete Database, D – The threshold value, τ, returned in Step (i) above is used in this step to determine the clusters (subspaces) that should not be considered in the derivation of the informative skyline results, S_I, of D to ensure that the number of returned skylines, $|S_I|$, meets the $|S|_{u_p}$ as specified by the user, u_p. The τ value is a value ≥ 0 and < 1. In general, if $1 - C_{i_\beta} \leq \tau$, then the cluster C_i is omitted.

The higher the value of τ, the more subspaces (clusters) are omitted. Here, the clusters with the τ-*Least Informative Objects* (see Definition 11) will be excluded from the skyline analysis. The list of clusters to be omitted, $\neg C$, is formally written as follows: $\{C_i | \frac{i}{|d|} \leq \tau\}$ where i is the number of non-missing values in C_i. For instance, if $|d| = 15$ and $\tau = 0.5$, then $\neg C = \{C_1, C_2, \ldots, C_7\}$. We use the notation C_τ to represent the set of clusters that is to be considered in the derivation of the S_I of D, i.e. $C_\tau = C - \neg C$.

Given the C_τ, the cluster skylines of C_τ are compared to derive the candidate skylines, S_C. Similar to the work by [18] each candidate skyline of S_C is then compared to the non-candidate skylines of the C_τ's clusters that are not within its own cluster also known as the shadow skylines, S_s, to derive the final skylines, S_I, of D. This is to ensure that exhaustive pairwise comparisons between objects are performed before the final skylines, S_I, of D can be realised.

Figure 7 (a) presents the cluster skylines of each cluster, S_{C_i}, that are derived in the final step of the *Preprocessing* phase. With the assumption that τ is 0 as predicted by the *threshold prediction model*, then the input to the skyline algorithm will be all the cluster skylines, $\{A1, B1, C1, D2, G1, K2, M2, N1, N2\}$, which then produces the candidate skylines, $S_C = \{A1, M2\}$ as shown in Figure 7 (b). Each candidate skyline is then compared to the shadow skylines that are not within its own cluster. For instance, $A1$ is compared to the objects of clusters C_2' and C_3', which are the shadow skylines of C_2 and C_3, respectively. This results in $\{A1, M2\}$ as the informative skyline results, S_I of D. Meanwhile, Figure 8 shows the results of applying the above steps with $C_\tau = \{C_2, C_3, C_4\}$, i.e. $\neg C = \{C_1\}$.

Inherently, a different τ value, for instance, $\tau = 0.5$, will result in $S_I = \{K2, M2\}$ since only the cluster skylines of the clusters C_3 and C_4 are analysed, i.e. $C_\tau = \{C_3, C_4\}$. More examples can be found in Table 4 that shows the possible values of τ and the S_I based on the example given in Figure 7.

Evaluate the Level of Informative Skyline Results, L_{S_I}, and Refine the Informative Skyline Results, S_I – The set of skyline results, S_I, produced in the previous step is evaluated by measuring its level of informative, L_{S_I}, as given by Equation (1). The L_{S_I} consists of two main properties, namely: the size of the returned skyline objects, $|S_I|$, and the missing value ratio in the returned skyline objects, S_β.The $|S_I|$ should not be too few or empty. If $|S_I| \geq |S|_u$, then it is said to have fulfil the number of skyline objects as requested by the user and the S_α is assigned the value 1; otherwise $S_\alpha = 0$ which indicates that the $|S_I|$ is fewer than the specified $|S|_u$. On the other hand, the lesser the value of S_β, the more

ID	d_1	d_2	d_3	d_4
A1	*	*	*	0.0144
A2	*	*	*	0.115
A3	*	*	*	0.5679
A4	*	*	*	0.1799

$b_1(0001), b_{1_\beta} = 0.75$

ID	d_1	d_2	d_3	d_4
B1	*	*	0.2191	*

$b_2(0010), b_{2_\beta} = 0.75$

ID	d_1	d_2	d_3	d_4
C1	*	0.4346	*	*

$b_3(0100), b_{3_\beta} = 0.75$

ID	d_1	d_2	d_3	d_4
D1	0.542	*	*	*
D2	0.5183	*	*	*

$b_4(1000), b_{4_\beta} = 0.75$

ID	d_1	d_2	d_3	d_4
E1	*	*	0.3497	0.8014
E2	*	*	0.4498	0.8291

$b_5(0011), b_{5_\beta} = 0.50$

ID	d_1	d_2	d_3	d_4
F1	*	0.8587	*	0.9518
F2	*	0.4047	*	0.6879
F3	*	0.5757	*	0.9886
F4	*	0.4333	*	0.8879

$b_6(0101), b_{6_\beta} = 0.50$

ID	d_1	d_2	d_3	d_4
G1	*	0.1099	0.347	*
G2	*	0.1539	0.697	*
G3	*	0.4426	0.9452	*

$b_7(0110), b_{7_\beta} = 0.50$

ID	d_1	d_2	d_3	d_4
H1	0.6131	*	*	0.9414

$b_8(1001), b_{8_\beta} = 0.50$

ID	d_1	d_2	d_3	d_4
I1	0.542	*	0.9398	*

$b_9(1010), b_{9_\beta} = 0.50$

ID	d_1	d_2	d_3	d_4
J1	0.4138	0.2672	*	*
J2	0.8182	0.7376	*	*

$b_{10}(1100), b_{10_\beta} = 0.50$

ID	d_1	d_2	d_3	d_4
K1	0.6648	*	0.2112	0.5232
K2	0.2904	*	0.0865	0.648
K3	0.6432	*	0.0809	0.8615
K4	0.4559	*	0.6759	0.8429

$b_{11}(1011), b_{11_\beta} = 0.25$

ID	d_1	d_2	d_3	d_4
L1	0.1937	0.8186	*	0.6682

$b_{12}(1101), b_{12_\beta} = 0.25$

ID	d_1	d_2	d_3	d_4
M1	0.7272	0.8345	0.6856	*
M2	0.1816	0.0406	0.1581	*

$b_{13}(1110), b_{13_\beta} = 0.25$

ID	d_1	d_2	d_3	d_4
N1	0.6928	0.2886	0.4933	0.3522
N2	0.5327	0.0615	0.1563	0.8064
N3	0.7111	0.425	0.2914	0.9806

$b_{14}(1111), b_{14_\beta} = 0$

FIGURE 5. The buckets formed based on the samples of data in Table 1.

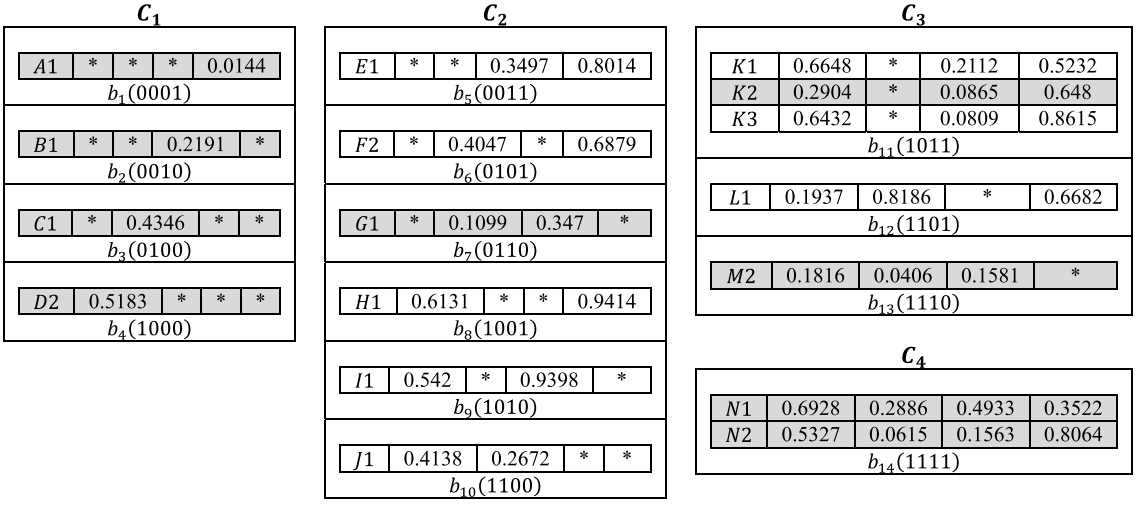


FIGURE 6. The clusters formed based on the buckets in Figure 5.

informative are the skyline results. Meanwhile, a weight value ω_i where $i \in \{1, 2\}$, a value between [0, 1] and $\omega_1 + \omega_2 = 1$, is given to the i-th property to designate the importance of the properties to the user. An equal value given to both ω_1 and ω_2, i.e. $\omega_1 = \omega_2 = 0.5$, implies that the two properties are equally important. The value of L_{S_l} lies between [0, 1] with values heading towards 0 indicate less informative and those towards 1 are more informative.

It is important to note that an object $o_i \in D$, with only one non-missing value, say $o_i.d_k = v \neq' *'$, while the v value of o_i is the lowest value in d_k, then the o_i is said to dominate the other objects, $o_j \in D$ where $i \neq j$ and $o_j.d_k \neq' *'$. Even though, the object o_i is the least informative object due to the highest missing value ratio (see Definition 10), based on the *Definition 6 Dominance over Incomplete Database*, the o_i is considered as the skyline object. Consequently,

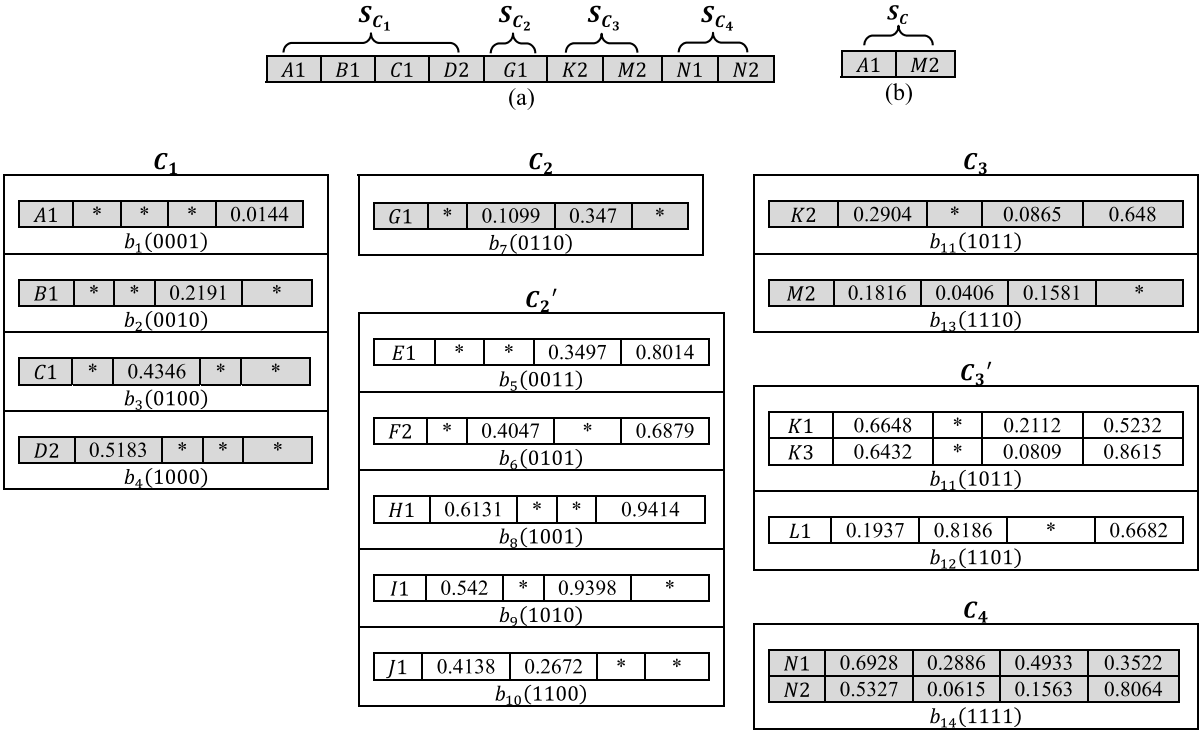


FIGURE 7. (a) The cluster skylines, S_{C_i} (b) The candidate skylines, S_C and (c) The cluster skylines and shadow skylines of each cluster with $C_\tau = \{C_1, C_2, C_3, C_4\}$.

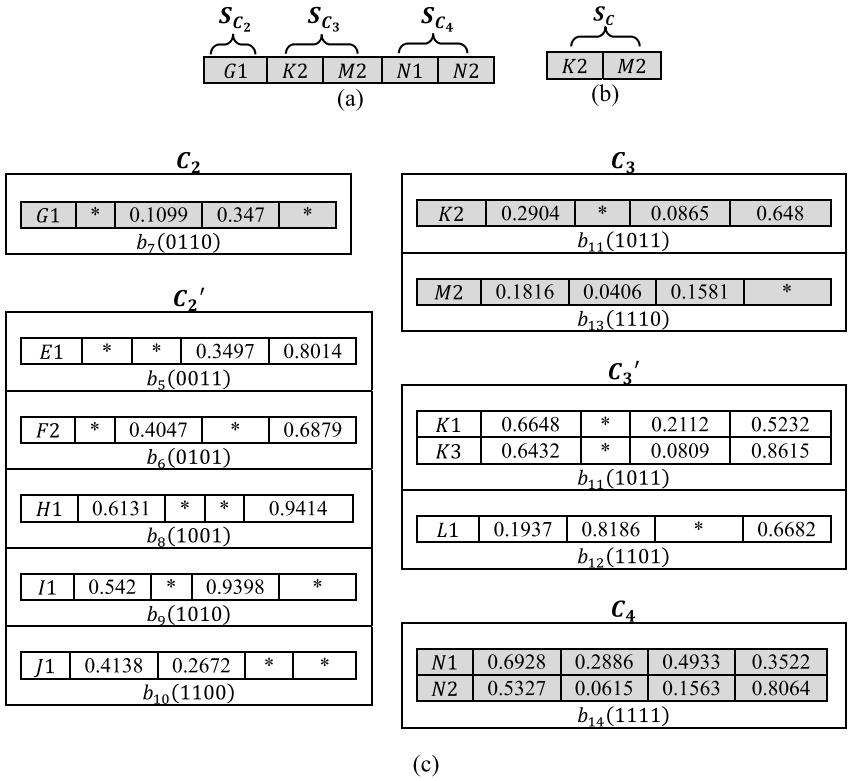


FIGURE 8. (a) The cluster skylines, S_{C_i} (b) The candidate skylines, S_C (c) The cluster skylines and shadow skylines of each cluster with $C_\tau = \{C_2, C_3, C_4\}$.

this results in a lesser number and less informative skyline results, i.e. low L_{S_l} value. For instance, the object $A1 < *, *, *, 0.0144 >$ has missing values in dimensions 1, 2, and 3 while the value 0.0144 is the lowest value in dimension 4. Hence, $A1$ dominates the other cluster skylines that are $K2$, $N1$, and $N2$ whose values in dimension 4 are higher than

TABLE 4. Examples of S_I with four possible ranges of τ with $|d| = 4$.

τ	$\neg C = \{C_i \mid \frac{i}{\lvert d \rvert} \leq \tau\}$	C_τ	S_I	$\lvert S_I \rvert$
$0 \leq \tau < 0.25$	$\neg C = \{\}$	$C_\tau = \{C_1, C_2, C_3, C_4\}$	$S_I = \{A1, M2\}$	2
$0.25 \leq \tau < 0.5$	$\neg C = \{C_1\}$	$C_\tau = \{C_2, C_3, C_4\}$	$S_I = \{K2, M2\}$	2
$0.5 \leq \tau < 0.75$	$\neg C = \{C_1, C_2\}$	$C_\tau = \{C_3, C_4\}$	$S_I = \{K2, M2\}$	2
$0.75 \leq \tau < 1$	$\neg C = \{C_1, C_2, C_3\}$	$C_\tau = \{C_4\}$	$S_I = \{N1, N2\}$	2

$A1$ even though their missing value ratio is lower than $A1$. This means to achieve an acceptable L_{S_I} value, the subspaces containing the least informative objects should be omitted in the skyline analysis.

On the other hand, an object $o_i \in D$ with non-missing values in all dimensions, i.e. $\forall d_k \in d, o_i.d_k \neq' *'$ and $o_{i_\beta} = 0$, is said to be more informative than those having at least a missing value. If the cluster, C_r, contains a collection of cluster skylines with $\beta = 0$, and $|S_{C_r}| \geq |S|_u$, then $L_{S_I} = 1$. However, limiting the skyline analysis over the cluster C_r alone would results in other useful objects not listed in the S_I. For instance, $S_I = \{N1, N2\}$ when C_4 is the only cluster considered in the skyline analysis. Even though, $N1$ and $N2$ have no missing values, but $K2 < 0.2904, *, 0.0865, 0.648 >$ is better than $N2 < 0.5327, 0.0615, 0.1563, 0.8064 >$ in dimensions 1, 3, and 4, while $M2 < 0.1816, 0.0406, 0.1581, * >$ is better than $N1 < 0.6928, 0.2886, 0.4933, 0.3522 >$ in dimensions 1, 2, and 3.

Hence, to balance between these two extreme cases, in our work, the clusters with the τ-*Least Informative Objects* (see Definition 11) are discarded from the skyline analysis, with the initial τ value obtained from the *threshold prediction model* as explained in Step (ii) above. If the L_{S_I} of the returned S_I is low, then the Step (ii) is repeated with a new τ value derived by adding $\frac{1}{|d|}$ to the current value of τ. For instance, if the initial value of τ is 0.5, then the new value of τ is $0.5 + \frac{1}{4} = 0.75$ with $|d| = 4$. This process is repeated until an acceptable L_{S_I} is obtained.

Table 5 presents samples of L_{S_I} with $\omega_1 = \omega_2 = 0.5$ and $|S|_{u_p} = 2$. Assume that $S_I = \{A1, M2\}$ is the result produced by Step (ii) with a τ value in the following range $0 \leq \tau < 0.25$ as given by the threshold prediction model. The $\omega_1 S_\alpha = 0.5$ implies that $|S_I| >= |S|_{u_p}$, which infers that the returned skyline results are not too few. Meanwhile, $\omega_2 S_\beta = 0.25$ implies that half of the values in the returned skyline results are missing. In order to retrieve more informative skyline results, the τ value can be adjusted. For instance, having $\tau = 0.3$ $(0.25 \leq \tau < 0.5)$ would return the $S_I = \{K2, M2\}$ with lesser missing values, i.e. 0.25, compared to the previous results.

Meanwhile, Table 6 shows samples of L_{S_I} with $\omega_1 = \omega_2 = 0.5$ and $|S|_{u_p} = 3$. Since, the maximum number of skylines that can be derived based on the given samples of data is 2, thus $S_\alpha = 0$ which states that $|S_I| < |S|_{u_p}$, i.e. the skyline results are too few and do not meet the required number as specified by the user, u_p. On the other hand, Table 7 presents samples of L_{S_I} with $\omega_1 = 0.75$, $\omega_2 = 0.25$, and $|S|_{u_p} = 2$.

The algorithms of the *Pre-processing* phase and *Skyline Query Processing* phase of *TBSA* are given in Figure 9 and Figure 10, respectively. The computational complexity of the *Skyline Query Processing Phase* in the *TBSA* Algorithm is $O\left(N^4\right)$, as derived below. In line 5, it is assumed that $|C_\tau|$ is $l \leq k$; therefore, this operation is executed l times. Line 6 is executed approximately $l \times (l-1) \approx l^2$ times. Similarly, line 7 is executed $l^2 \times \left|S_{C_k}\right|$ times, while line 8 is executed $l^2 \times |S_{C_k}| \times |S_{C_r}| \approx l^2 \times |S_{C_k}|^2$ times. Following a similar reasoning, steps 14 through 16 require approximately $|S_C| \times l \times \left|C'_r\right|$ operations. Assuming that each operation within the innermost loop executes in constant time $O(1)$, the overall time complexity can be expressed as $O(l^2 \times |S_{C_k}|^2) + O(|S_C| \times l \times \left|C'_r\right|)$. Simplifying this expression yields an overall complexity of $O(N^4)$.

VI. RESULTS AND DISCUSSION

In this section, we evaluate the performance of our proposed algorithm, *Threshold-based Bucket Skyline Algorithm* (*TBSA*) , which is designed mainly for finding informative skyline results, S_I, over incomplete dataset, D. To prove the efficiency of *TBSA*, several extensive experiments were conducted over two types of datasets, namely: synthetic and real datasets; that are widely adopted by previous works on skyline computation problem [39], [41], [42], [43]. The synthetic dataset is generated randomly with parameter settings shown in Table 8. On the other hand, the NBA dataset is a real dataset containing the basketball skills of 17,758 players during regular seasons in the period of 1983-2017 (http://www.basketballreference.com/). Each record has 17 dimensions that include number of games played, points scored, assists, etc. Since the NBA dataset is complete, we have randomly removed some of the values to meet the required settings as shown in Table 8. Each experiment is run 10 times and the average value of these runs is reported. In deriving the set of skylines of the synthetic dataset, it is assumed that lower values are preferable compared to higher ones; while the opposite is applied for the NBA dataset. In each experiment, different parameter settings are applied while the following performance measurements are recorded: (i) number of iterations, (ii) threshold values, (iii) number of pairwise comparisons, (iv) number of skyline results, (v) level of informative skyline results, and (vi) processing time.

TABLE 5. Examples of L_{S_I} with $\omega_1 = \omega_2 = 0.5$ and $|S|_{up} = 2$.

τ	$\neg C = \{C_i \mid \frac{i}{\lvert d \rvert} \leq \tau\}$	C_τ	S_I	$\lvert S_I \rvert$	$\omega_1 S_\alpha$	$\omega_2 S_\beta$	L_{S_I}
$0 \leq \tau < 0.25$	$\neg C = \{\}$	$C_\tau = \{C_1, C_2, C_3, C_4\}$	$S_I = \{A1, M2\}$	2	0.5	0.25	0.75
$0.25 \leq \tau < 0.5$	$\neg C = \{C_1\}$	$C_\tau = \{C_2, C_3, C_4\}$	$S_I = \{K2, M2\}$	2	0.5	0.375	0.875
$0.5 \leq \tau < 0.75$	$\neg C = \{C_1, C_2\}$	$C_\tau = \{C_3, C_4\}$	$S_I = \{K2, M2\}$	2	0.5	0.375	0.875
$0.75 \leq \tau < 1$	$\neg C = \{C_1, C_2, C_3\}$	$C_\tau = \{C_4\}$	$S_I = \{N1, N2\}$	2	0.5	0.5	1

TABLE 6. Examples of L_{S_I} with $\omega_1 = \omega_2 = 0.5$ and $|S|_{up} = 3$.

τ	$\neg C = \{C_i \mid \frac{i}{\lvert d \rvert} \leq \tau\}$	C_τ	S_I	$\lvert S_I \rvert$	$\omega_1 S_\alpha$	$\omega_2 S_\beta$	L_{S_I}
$0 \leq \tau < 0.25$	$\neg C = \{\}$	$C_\tau = \{C_1, C_2, C_3, C_4\}$	$S_I = \{A1, M2\}$	2	0	0.25	0.25
$0.25 \leq \tau < 0.5$	$\neg C = \{C_1\}$	$C_\tau = \{C_2, C_3, C_4\}$	$S_I = \{K2, M2\}$	2	0	0.375	0.375
$0.5 \leq \tau < 0.75$	$\neg C = \{C_1, C_2\}$	$C_\tau = \{C_3, C_4\}$	$S_I = \{K2, M2\}$	2	0	0.375	0.375
$0.75 \leq \tau < 1$	$\neg C = \{C_1, C_2, C_3\}$	$C_\tau = \{C_4\}$	$S_I = \{N1, N2\}$	2	0	0.5	0.5

TABLE 7. Examples of L_{S_I} with $\omega_1 = 0.75$, $\omega_2 = 0.25$, and $|S|_{up} = 2$.

τ	$\neg C = \{C_i \mid \frac{i}{\lvert d \rvert} \leq \tau\}$	C_τ	S_I	$\lvert S_I \rvert$	$\omega_1 S_\alpha$	$\omega_2 S_\beta$	L_{S_I}
$0 \leq \tau < 0.25$	$\neg C = \{\}$	$C_\tau = \{C_1, C_2, C_3, C_4\}$	$S_I = \{A1, M2\}$	2	0.75	0.125	0.875
$0.25 \leq \tau < 0.5$	$\neg C = \{C_1\}$	$C_\tau = \{C_2, C_3, C_4\}$	$S_I = \{K2, M2\}$	2	0.75	0.1875	0.9375
$0.5 \leq \tau < 0.75$	$\neg C = \{C_1, C_2\}$	$C_\tau = \{C_3, C_4\}$	$S_I = \{K2, M2\}$	2	0.75	0.1875	0.9375
$0.75 \leq \tau < 1$	$\neg C = \{C_1, C_2, C_3\}$	$C_\tau = \{C_4\}$	$S_I = \{N1, N2\}$	2	0.75	0.25	1

TABLE 8. Parameter settings.

Parameter	Synthetic	NBA
Size of the database, |D|	50K,100K,150K, 200K, 250K, 300K, **500K**, 1M	2K, 4K, 6K, 8K, 10K, 12K, 14K,16K, **17758**
Number of dimensions, $\lvert d \rvert$	2, 4, 6, 8, **10**, 15, 20, 30, 40, 50	6, 10, 15, **17**
Missing value ratio, β	0.1, 0.2, 0.3, 0.4, **0.5**, 0.6, 0.7, 0.8, 0.9	
Number of skyline results, $\lvert S \rvert_u$	5, **10**, 15, 20	
Weight values of ω_1 and ω_2	[0, 1.0], [0.25, 0.75], **[0.5, 0.5]**, [0.75, 0.25], [1.0, 0]	
Threshold value, τ	**0.5**	

The performance results of *TBSA* are compared to the approach proposed by Jongwuk Lee et al. [22] (herein called the JL approach) and the non-threshold *TBSA*. To retrieve meaningful skyline results, the JL approach [22] adjusts the minimum size of common subspaces in an incremental manner while the non-threshold *TBSA* is our algorithm *TBSA* without applying the thresholding approach. This is mainly to examine and validate the significant of employing the thresholding approach.

A. EFFECT OF THE SIZE OF DATASET, $|D|$

In this study, the effect of the size of dataset, $|D|$, on the performance of *TBSA*, JL approach, and non-threshold *TBSA* is investigated. The parameter settings of the synthetic dataset are as follows: the size of the dataset, $|D|$, is varied from 50K to 1M with the number of dimensions, $|d|$, fixed to 10, and the missing value ratio, β, set to 0.5. Meanwhile, the size of the dataset, $|D|$, for the NBA dataset is varied from 2K to its initial size, i.e. 177,58 with $|d| = 17$ and $\beta = 0.5$. For both datasets, the minimum level of informative skyline results, L_{S_I}, is set to 0.5; meanwhile ω_1 and ω_2 are fixed to 0.5, while the minimum number of skyline results, $|S|_u$, is set to 10. Figures 11 (a) – (e) present the performance of *TBSA*, JL approach, and non-threshold *TBSA* with regard to (i) number of iterations, (ii) number of pairwise comparisons, (iii) number of skyline results, (iv) level of informative skyline results, and (v) processing time, respectively.

Figures 11 (a1) and 11 (a2) show that our proposed algorithm *TBSA* achieved the least number of iterations than the JL approach and the non-threshold *TBSA* for both datasets, i.e. synthetic and NBA. The *TBSA* typically entails one iteration, with the exception of the synthetic dataset with $|D| = 500$K and $|D| = 1$M, where two iterations are performed. On the other hand, both the JL approach and non-threshold *TBSA* show comparable performance with 8 and 9 iterations for the synthetic and NBA datasets,

Input: Incomplete database, D
Output: A set of clusters, $C = \{C_1, C_2, \ldots, C_k\}$

Begin
  Extract the features of D, $D_F < DD, |D|, |d|, D_\beta >$
  For each $o_i \in D$ do /* generate the bitmap representation, bmr, of o_i
    For each $d_k \in d$ do
    Begin
      If $o_i.d_k = '*'$, then $bmr_{o_i}.d_k = 0$
      Else $bmr_{o_i}.d_k = 1$
    End
  For each $o_i \in D$ do /* group the objects into buckets
  Begin
    If $bmr_{o_i} = bmr_{b_l}$, then $b_l = b_l \cup o_i$
    Else
    Begin
      $bmr_{b_{l+1}} = bmr_{o_i}$
      $b_{l+1} = b_{l+1} \cup o_i$
    End
  End
  For each $b_l \in B$ do /* derive the bucket skylines, S_{b_l}
  Begin
    For each $o_i \in b_l$ do
      For each $o_j \in b_l$ where $i \neq j$ do
      Begin
        If $o_i \prec o_j$, then $b_l = b_l - o_j$
        Else If $o_j \prec o_i$, then $b_l = b_l - o_i$
      End
    $S_{b_l} = b_l$
    Get $b_l < bmr_{b_l}, b_{l_\beta}, S_{b_l}, |S_{b_l}| >$
  End
  For each $b_l \in B$ do /* group the buckets into clusters
    If $b_{l_\beta} = 1 - \frac{i}{|d|}$, then $b_l \in C_i$
  For each $C_k \in C$ do /* get the cluster skylines, S_{C_k}
  Begin
    For each $b_l \in C_k$ do
      For each $o_i \in b_l$ do
        For each $b_r \in C_k$ where $l \neq r$ & b_l and b_r are comparable do
          For each $o_j \in b_r$ do
          If $o_i \prec o_j$, then
          Begin
            $b_r = b_r - o_j$
            $C_k = C_k - o_j$
          End
          Else If $o_j \prec o_i$, then
          Begin
            $b_l = b_l - o_i$
            $C_k = C_k - o_i$
          End
    $S_{C_k} = C_k$
    Get $C_i < C_{i_\beta}, S_{C_i}, |S_{C_i}| >$
    Get $C_i' = C_k - S_{C_k}$ /* get the shadow skylines, S_{SC_i}, of C_i
  End
End
Return $C = \{C_1, C_2, \ldots, C_k\}$ with each $C_i < C_{i_\beta}, S_{C_i}, |S_{C_i}| >$

FIGURE 9. The Pre-processing phase of the *TBSA* algorithm.

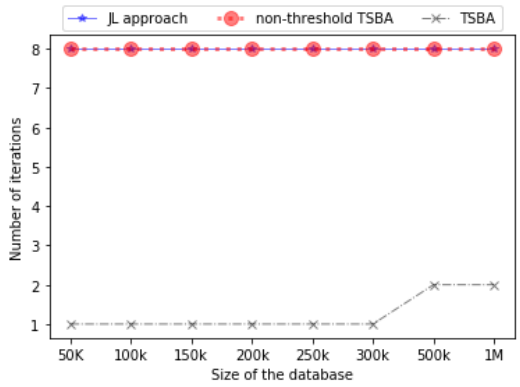
Input: The required minimum number of returned skylines, $|S|_u$; the features of D, $D_F < DD, |D|, |d|, D_\beta >$; the set of clusters, $C = \{C_1, C_2, \ldots, C_k\}$
Output: Informative skyline results, S_I

Begin
  Get the τ based on the $D_F < DD, |D|, |d|, D_\beta >$ by invoking the *threshold prediction model*
  Identify the $\neg C = \{C_i | \frac{i}{|d|} \leq \tau\}$ where i is the number of non-missing values in C_i
  Let $C_\tau = C - \neg C$
  For each $C_k \in C_\tau$ do /* derive the candidate skylines, S_C, based on the cluster skylines of C_τ
    For each $C_r \in C_\tau$ where $k \neq r$ do
      For each $o_i.S_{C_k} \in C_k$ do
        For each $o_j.S_{C_r} \in C_r$ where $i \neq j$ do
        Begin
          If $o_i \prec o_j$, then $S_{C_r} = S_{C_r} - o_j$
          Else If $o_j \prec o_i$, then $S_{C_k} = S_{C_k} - o_i$
        End
  $S_C = S_{C_p} \cup S_{C_{p+1}} \cup \ldots \cup S_{C_{p+s}}$ where C_p, C_{p+1}, …, C_{p+s} are clusters of C_τ
  For each $o_i \in S_C \wedge o_i \in C_k$ /* derive the final skylines based on the candidate skylines
    For each $C_r \in C_\tau$ where $k \neq r$ do
      For each $o_j \in C_r'$ /* shadow skylines of C_r
        If $o_j \prec o_i$, then $S_C = S_C - o_i$
  $S_I = S_C$
  Calculate the L_{S_I} of S_I /* Equation (1)
  If L_{S_I} is low, then $\tau = \tau + \frac{1}{|d|}$ and repeat Step (3)
End
Return S_I

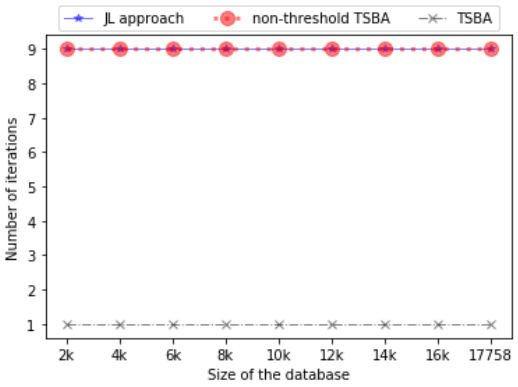
FIGURE 10. The Skyline Query Processing phase of the *TBSA* algorithm.

respectively. To further validate the performance of *TBSA*, the threshold value, τ, predicted by *TBSA* in each case is compared to the real τ value as shown in figures 11 (a3) and 11 (a4) for the synthetic and real datasets, respectively. Figure 11 (a3) shows that *TBSA* has predicted the accurate τ values when $|D| = 50$K until $|D|$ reaches to 300K for the synthetic dataset. However, when $|D| = 500$K and $|D| = 1$M, the τ values predicted by *TBSA* are smaller than the real τ values. As a result, the number of iterations performed to achieve the targeted level of informative skyline results, L_{S_I}, is more than 1 as shown in Figure 11 (a1). Furthermore, the τ values for the NBA dataset are larger than the real τ values for all sizes of $|D|$, i.e. 2K to 17,758; while the accuracy of the τ value predicted by *TBSA* is at its lowest level when the size of $|D| > 12$K. As a result, more skylines are being derived while the level of informative skyline results is high.

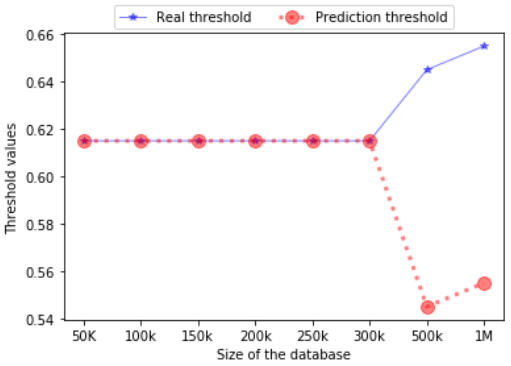
Figures 11 (b1) and 11 (b2) present the performance results of *TBSA*, JL approach, and non-threshold *TBSA* with regard to the number of pairwise comparisons for both datasets, namely: synthetic and NBA, respectively. In all cases, the number of pairwise comparisons of both *TBSA* and non-threshold *TBSA* is almost equal; with *TBSA* exhibiting the least number of pairwise comparisons among the approaches being compared. This is because *TBSA* which employed the thresholding approach is able to minimise the number of iterations performed by examining only those subspaces of the datasets that will yield the intended informative skyline



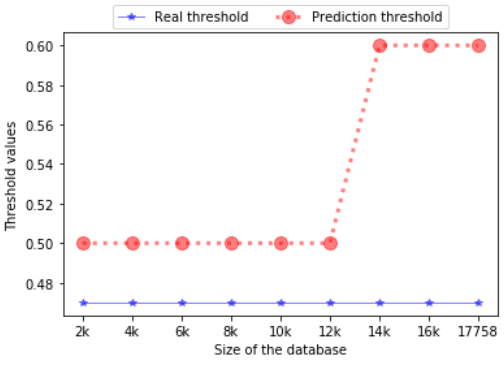
(a1) Synthetic Dataset



(a2) NBA Dataset

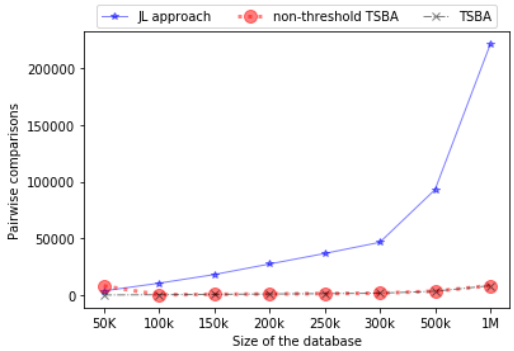


(a3) Synthetic Dataset

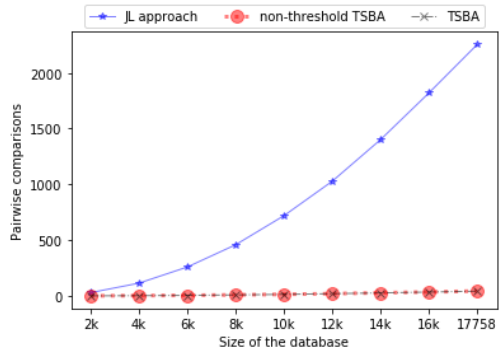


(a4) NBA Dataset

(a) Number of iterations

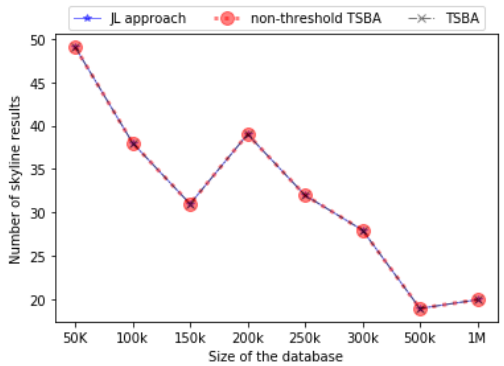


(b1) Synthetic Dataset

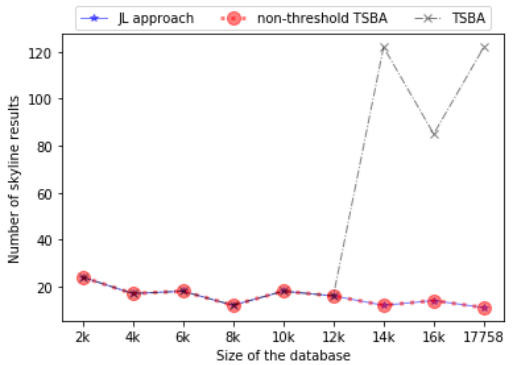


(b2) NBA Dataset

(b) Number of pairwise comparisons



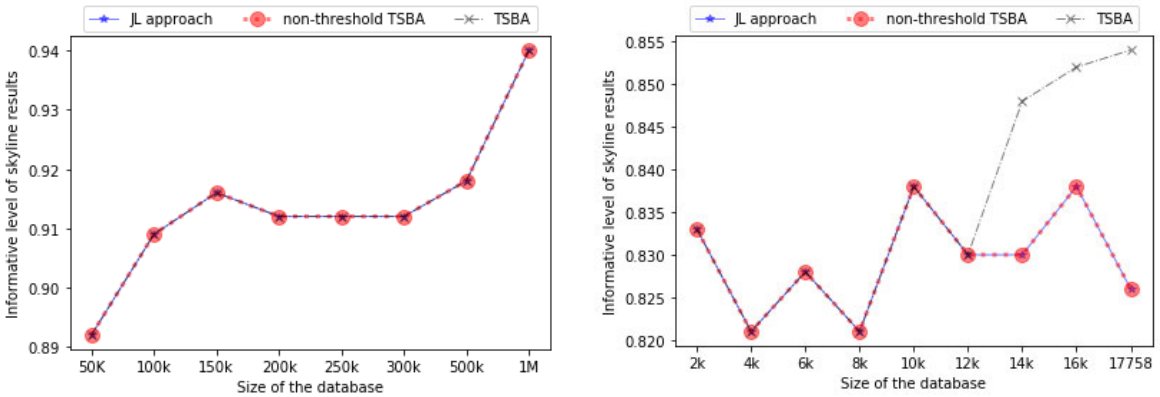
(c1) Synthetic Dataset



(c2) NBA Dataset

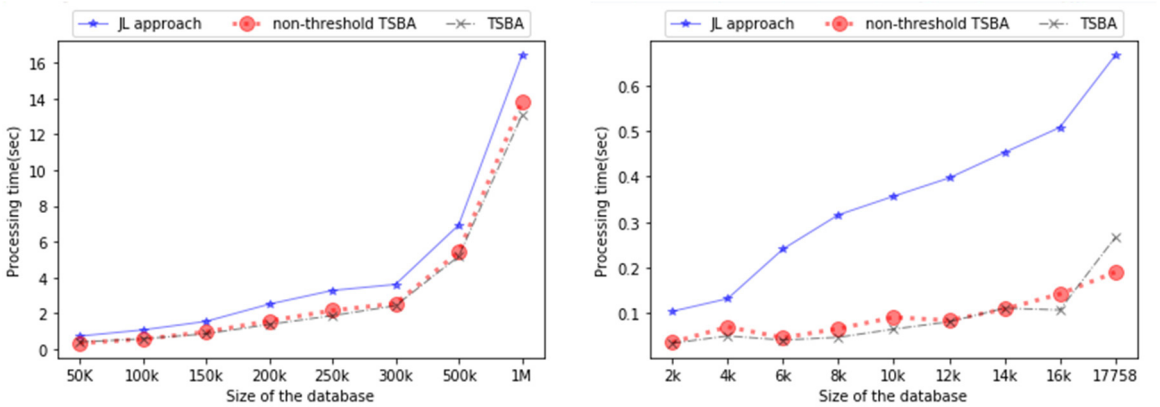
(c) Number of skyline results

FIGURE 11. The performance results based on the size of dataset, $|D|$.



(d1) Synthetic Dataset (d2) NBA Dataset

(d) Level of informative skyline results



(e1) Synthetic Dataset (e2) NBA Dataset

(e) Processing time

FIGURE 11. ***(Continued.)*** **The performance results based on the size of dataset, $|D|$.**

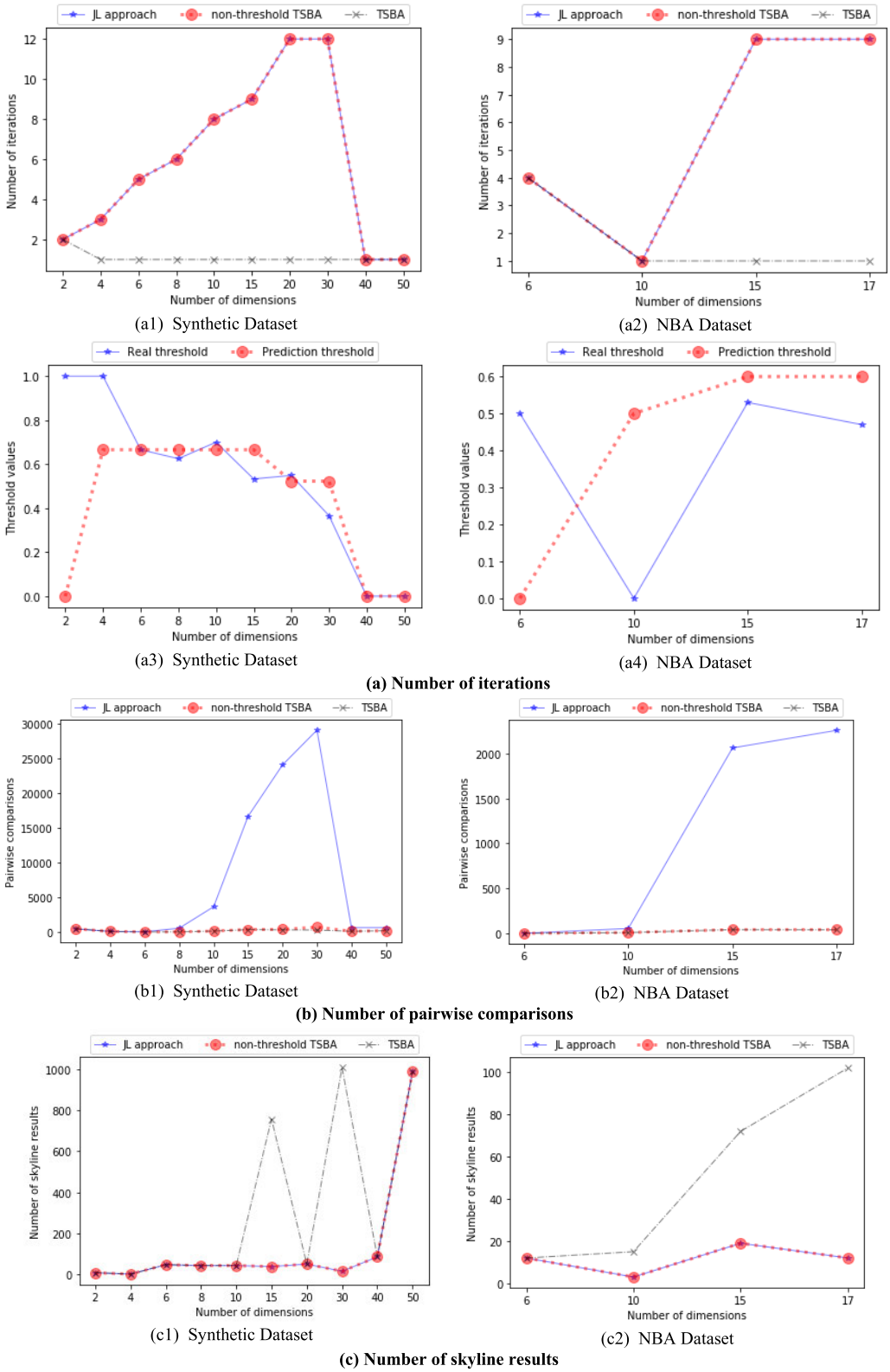
results, L_{S_I}. Meanwhile, the cluster skyline and shadow skyline that are part of the non-threshold *TBSA*'s solution made it superior to the JL approach.

On the other hand, the number of skyline results, $|S_I|$, derived by *TBSA*, JL approach, and non-threshold *TBSA* is the same for all cases as clearly depicted in figures 11 (c1) and 11 (c2). The only exception is the NBA dataset with $|D| > 12$K, from which *TBSA* derived a larger number of skyline results than the other two approaches. Nevertheless, in all cases, $|S_I| > |S|_u$; where $|S|_u = 10$. Similar trends can be seen in figures 11 (d1) and 11 (d2) where *TBSA*, JL approach, and non-threshold *TBSA* achieved the same level of informative skyline results, L_{S_I}, for most cases with the exception of the NBA dataset with $|D| > 12$K where *TBSA* demonstrates a higher L_{S_I}. This is due to (i) the τ values predicted by *TBSA* for these cases are higher than the real τ values (see Figure 11 (a4)); consequently the subspaces of the datasets with higher missing value ratio, β, are filtered out and (ii) $|S_I| > |S|_u$; which contribute to higher L_{S_I}. Furthermore, $L_{S_I} > 0.820$ for all cases which is higher than 0.5; the minimum value of L_{S_I} set for the experiment.

Figures 11 (e1) and 11 (e2) present the performance results of *TBSA*, JL approach, and non-threshold *TBSA* with regard to the processing time for both datasets, namely: synthetic and NBA, respectively. In all cases, *TBSA* exhibits the least processing time, while JL approach shows the highest. The only exception is the NBA dataset with $|D| = 17758$. Meanwhile, the non-threshold *TBSA* approach requires less processing time than the JL approach for both datasets mainly due to the utilisation of the cluster and shadow skylines.

B. EFFECT OF THE NUMBER OF DIMENSIONS, $|d|$

We also study the effect of the number of dimensions, $|d|$, on the performance of *TBSA*, JL approach, and non-threshold *TBSA*. The parameter settings of the synthetic dataset are as follows: the number of dimensions, $|d|$, is varied from 2 to 50 with the size of dataset, $|D|$, fixed to 50,000, and the missing value ratio, β, set to 0.5. Meanwhile, the number of dimensions, $|d|$, for the NBA dataset is varied from 6 to its initial number, i.e. 17 with $|D| = 17,758$ and $\beta = 0.5$. The settings for the other parameters, namely: L_{S_I}, ω_1, ω_2, and $|S|_u$, are the same as in



(a1) Synthetic Dataset (a2) NBA Dataset

(a3) Synthetic Dataset (a4) NBA Dataset

(a) Number of iterations

(b1) Synthetic Dataset (b2) NBA Dataset

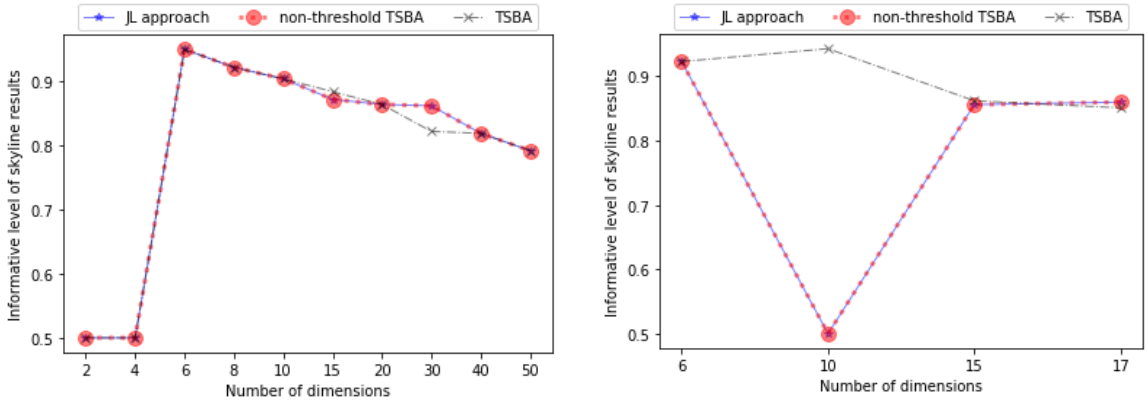
(b) Number of pairwise comparisons

(c1) Synthetic Dataset (c2) NBA Dataset

(c) Number of skyline results

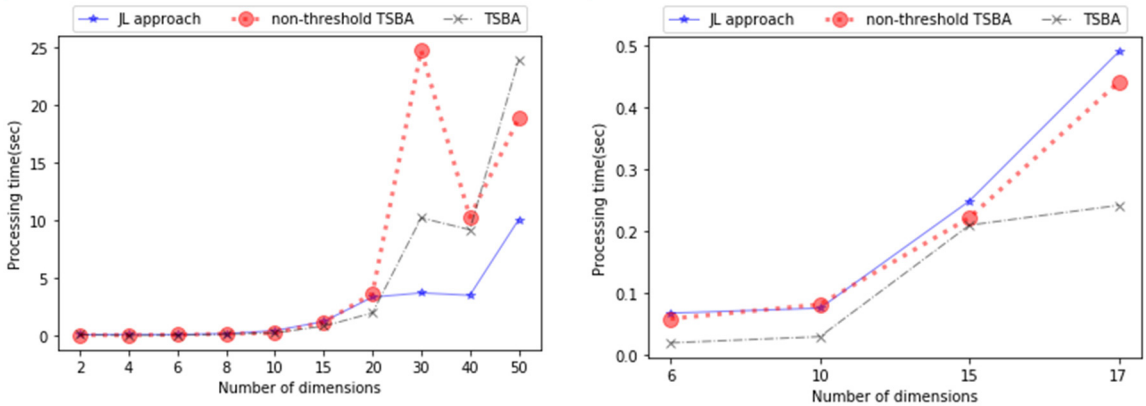
FIGURE 12. The performance results based on the number of dimensions, $|d|$.

Section VI-A. Figures 12 (a) – (e) present the performance of *TBSA*, JL approach, and non-threshold *TBSA* with regard to (i) number of iterations, (ii) number of pairwise comparisons, (iii) number of skyline results, (iv) level of informative skyline results, and (v) processing time, respectively.



(d1) Synthetic Dataset (d2) NBA Dataset

(d) Level of informative skyline results



(e1) Synthetic Dataset (e2) NBA Dataset

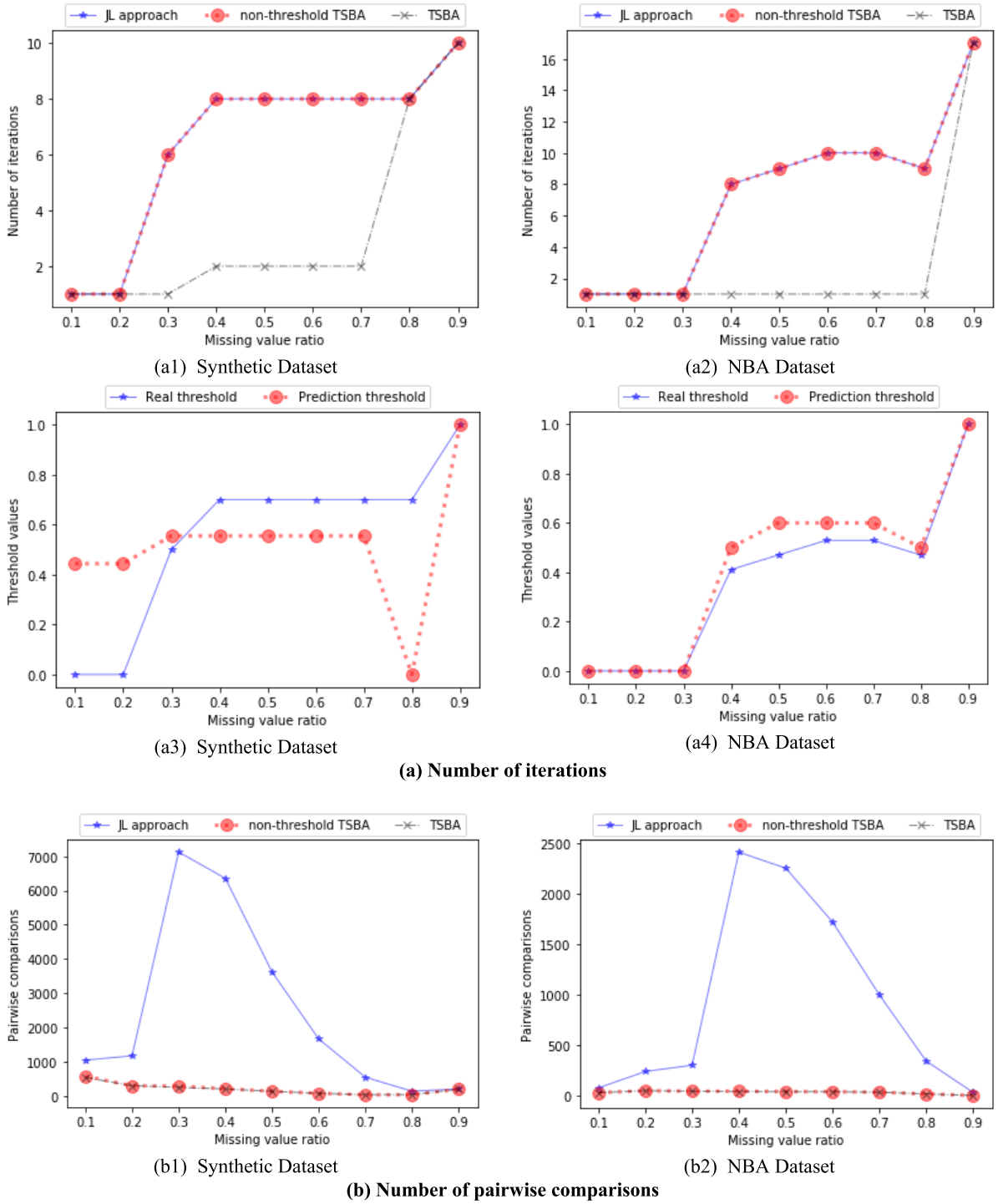
(e) Processing time

FIGURE 12. ***(Continued.)*** **The performance results based on the number of dimensions, $|d|$.**

Figures 12 (a1) and 12 (a2) demonstrate that, for both the synthetic and NBA datasets, the number of iterations performed by our proposed algorithm *TBSA* is one, which is the least number of iterations compared to the JL approach and the non-threshold *TBSA*. The only exceptions are the synthetic dataset with $|d| = 2$, 40, and 50 and NBA dataset with $|d| = 6$ and 10 in which the number of iterations performed by *TBSA* is the same as the other two approaches. Furthermore, the results indicate that the JL approach and the non-threshold *TBSA* have the highest number of iterations–12, for the synthetic dataset with $|d| = 20$ and $|d| = 30$ and 9 for the NBA dataset with $|d| = 15$ and $|d| = 17$. Meanwhile, as shown in figures 12 (a3) and 12 (a4) the τ value predicted by *TBSA* corresponds to three different cases, namely: (i) it is smaller than the real τ value as shown in the synthetic dataset with $|d| = 2, 4, 10$, and 20; and NBA dataset with $|d| = 6$, (ii) it is equal to the real τ value as shown in the synthetic dataset with $|d| = 6$, 40, and 50, and (iii) it is larger than the real τ value as shown in the synthetic dataset with $|d| = 8$, 15, and 30; and NBA dataset with $|d| = 10$, 15, and 17. It is evident that there is less likelihood of achieving the desired L_{S_I} in a single iteration when the τ value predicted by *TBSA* is smaller than the real τ value.

Figures 12 (b1) and 12 (b2) present the performance results of *TBSA*, JL approach, and non-threshold *TBSA* with regard to the number of pairwise comparisons for both datasets, namely: synthetic and NBA, respectively. In most cases, the number of pairwise comparisons of both *TBSA* and non-threshold *TBSA* is almost equal; with *TBSA* exhibiting the least number of pairwise comparisons among the approaches being compared with exception of the synthetic dataset with $|d| = 2$, 40, and 50; and NBA dataset with $|d| = 6$ where both the *TBSA* and non-threshold *TBSA* achieved the same number of pairwise comparisons; which support the results on the number of iterations performed by these approaches as shown in Figure 12 (a1). Nonetheless, the JL approach demonstrates the least number of pairwise comparisons for the synthetic dataset with $|d| = 2$.

On the other hand, the number of skyline results, $|S_I|$, derived by *TBSA*, JL approach, and non-threshold *TBSA* is the same for most cases of the synthetic dataset as shown in Figure 12 (c1) and NBA dataset with $|d| = 6$ as shown in



(a1) Synthetic Dataset (a2) NBA Dataset

(a3) Synthetic Dataset (a4) NBA Dataset

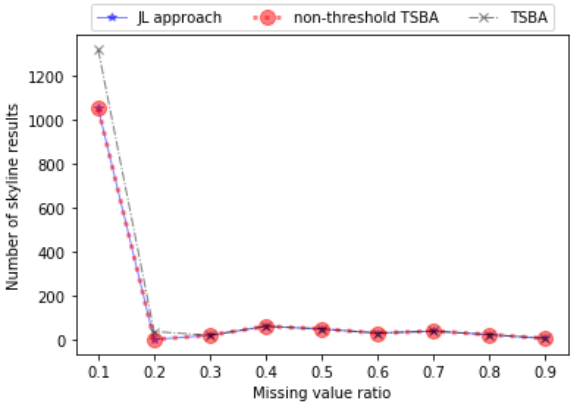
(a) Number of iterations

(b1) Synthetic Dataset (b2) NBA Dataset

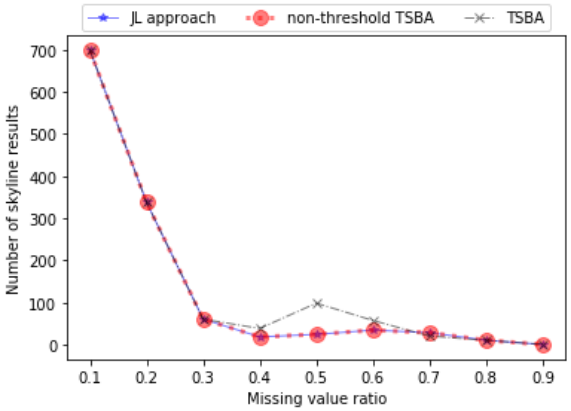
(b) Number of pairwise comparisons

FIGURE 13. **The performance results based on missing value ratio, β.**

Figure 12(c2). The only exceptions are the synthetic dataset with $|d| = 15$ and $|d| = 30$ and NBA dataset with $|d| > 6$, from which the *TBSA* derived a larger number of skyline results than the other two approaches. For instance, the JL approach and non-threshold *TBSA* derived $|S_I| = 38$ and $|S_I| = 15$ for the synthetic dataset with $|d| = 15$ and $|d| = 30$, respectively; while *TBSA* generated $|S_I| = 757$ and $|S_I| = 1008$. Mainly, this is because the τ value predicted by *TBSA* is greater than the real τ value, leading to the subspaces of the datasets with higher missing value ratio, β, being filtered out. Nevertheless, in all cases, $|S_I| > |S|_u$; where $|S|_u$ is set to 10. Similar trends can be seen in

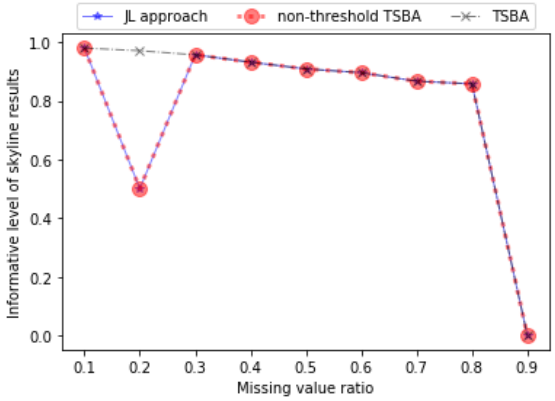


(c1) Synthetic Dataset

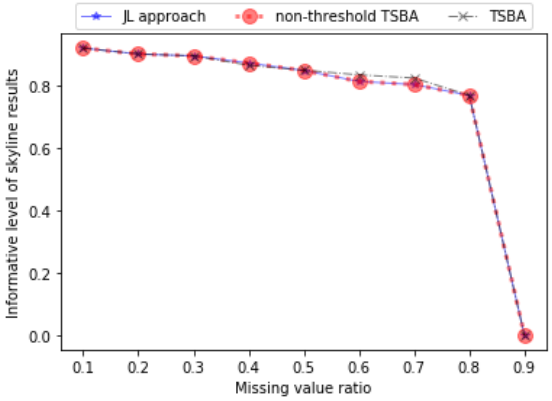


(c2) NBA Dataset

(c) Number of skyline results

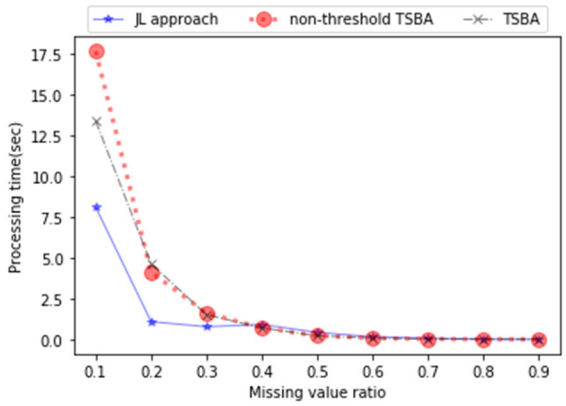


(d1) Synthetic Dataset

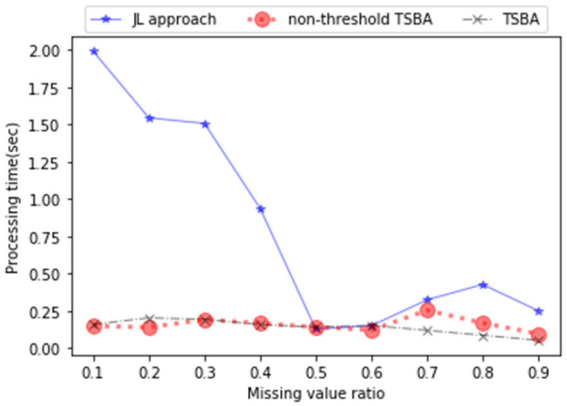


(d2) NBA Dataset

(d) Level of informative skyline results



(e1) Synthetic Dataset



(e2) NBA Dataset

(e) Processing time

FIGURE 13. ***(Continued.)*** **The performance results based on missing value ratio, β.**

figures 12 (d1) and 12 (d2) where *TBSA*, JL approach, and non-threshold *TBSA* achieved the same level of informative skyline results, L_{S_I}, for most cases with the exception of the synthetic dataset with $|d| = 15$ and NBA dataset with $|d| = 10$ and $|d| = 15$ where *TBSA* demonstrates a higher L_{S_I} while a lower L_{S_I} for synthetic dataset with $|d| = 30$ and NBA dataset with $|d| = 17$. Nonetheless, $L_{S_I} \geq 0.5$ for all cases which meets the minimum value of L_{S_I} set for the experiment, i.e. 0.5.

Figures 12 (e1) and 12 (e2) present the performance results of *TBSA*, JL approach, and non-threshold *TBSA* with regard to the processing time for both datasets, namely: synthetic

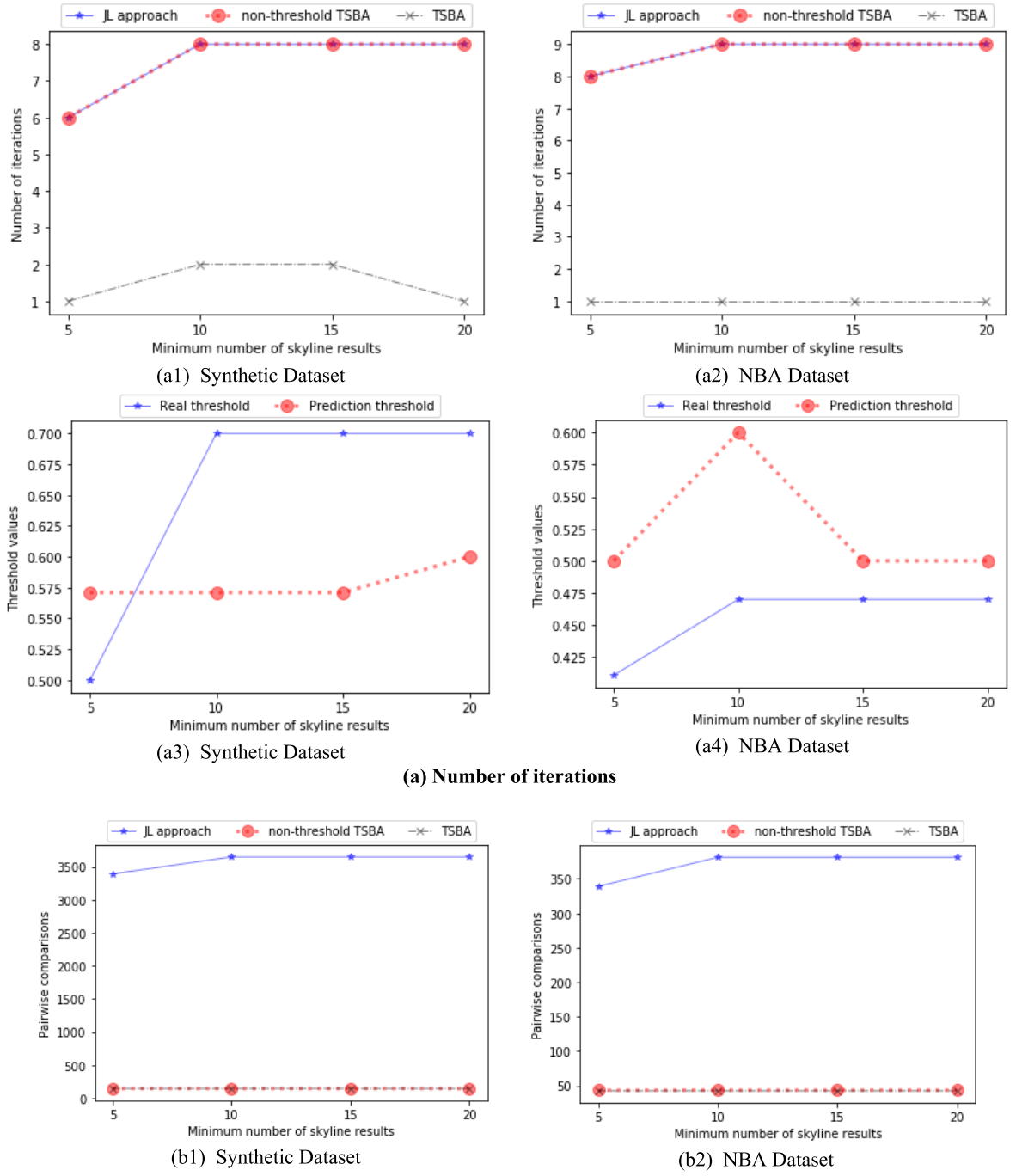
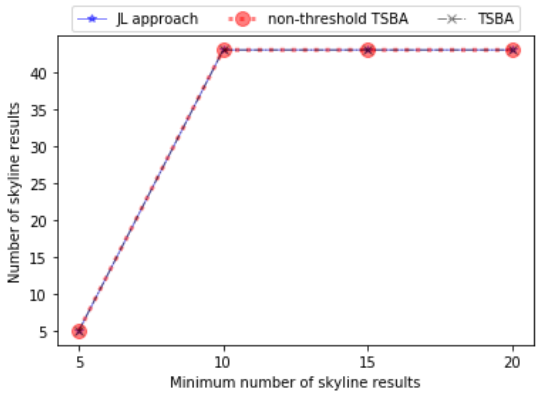


FIGURE 14. The performance results based on the number of skyline results, $|S|_u$.

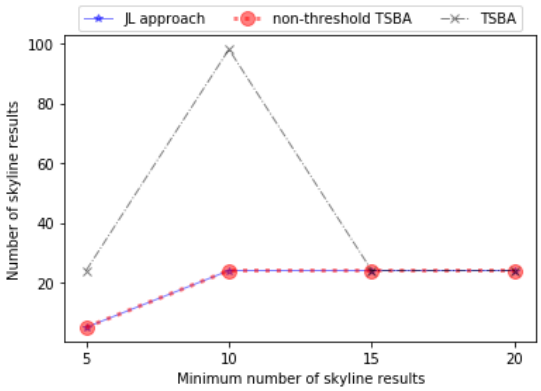
and NBA, respectively. In most cases, *TBSA* exhibits the least processing time, while the JL approach shows the highest. The only exception is the synthetic dataset with $|d| > 20$; in which the JL approach demonstrates the least processing time.

C. EFFECT OF MISSING VALUE RATIO, β

In this section, we report the performance of *TBSA*, JL approach, and non-threshold *TBSA* with regard to missing value ratio, β. The parameter settings of the synthetic dataset are as follows: the missing value ratio, β, is varied from 0.1 to 0.9 with the size of dataset, $|D|$, fixed to 50,000, and the number of dimensions, $|d|$, set to 10. Also, the missing value ratio, β, for the NBA dataset is varied from 0.1 to 0.9 with $|D| = 17,758$ and $|d| = 17$. The settings for the other parameters, namely: L_{S_I}, ω_1, ω_2, and $|S|_u$, are the same as in Section VI-A. Figures 13 (a) – (e) present the performance of *TBSA*, JL approach, and non-threshold *TBSA* with regard

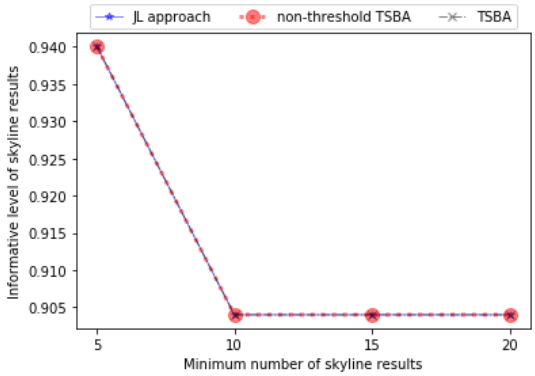


(c1) Synthetic Dataset

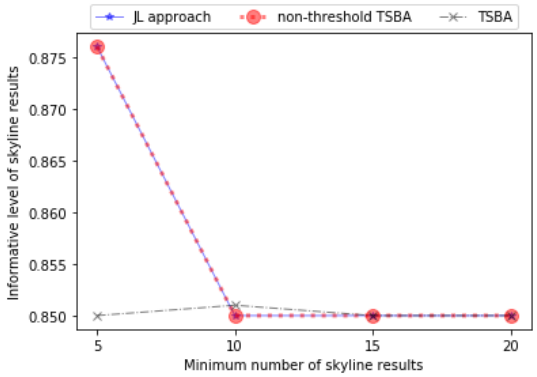


(c2) NBA Dataset

(c) Number of skyline results

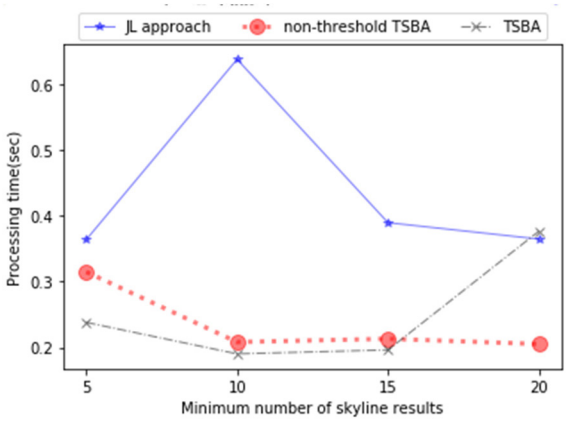


(d1) Synthetic Dataset

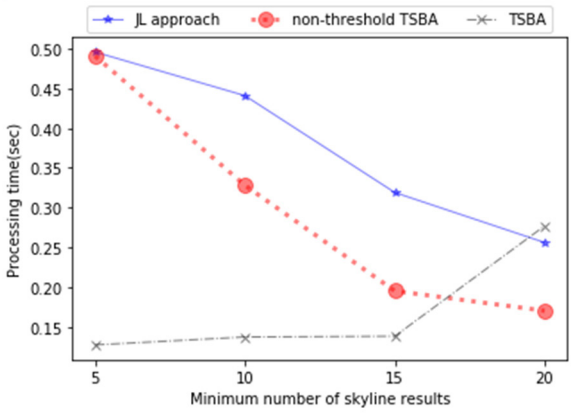


(d2) NBA Dataset

(d) Level of informative skyline results



(e1) Synthetic Dataset



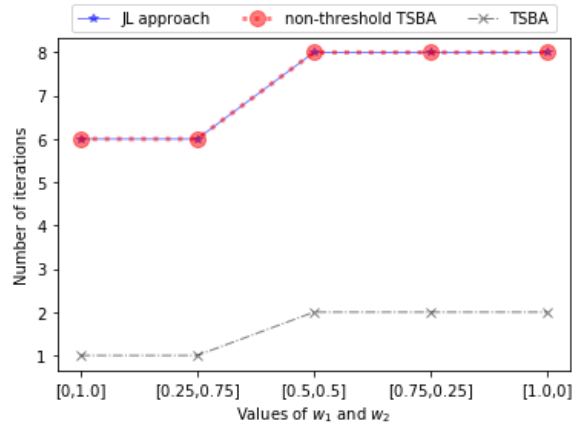
(e2) NBA Dataset

(e) Processing time

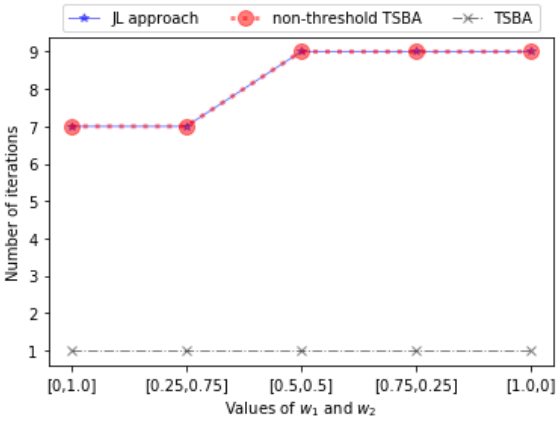
FIGURE 14. ***(Continued.)*** **The performance results based on the number of skyline results, $|S|_u$.**

to (i) number of iterations, (ii) number of pairwise comparisons, (iii) number of skyline results, (iv) level of informative skyline results, and (v) processing time, respectively.

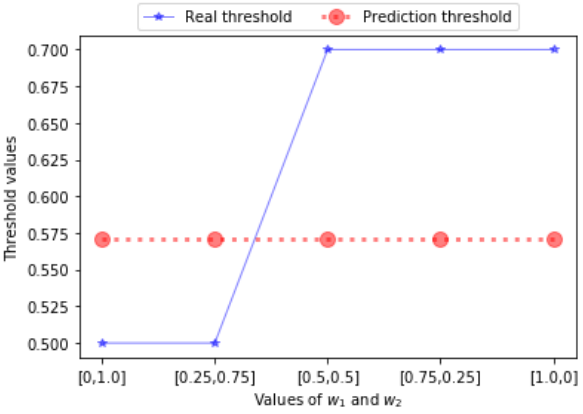
Figures 13 (a1) shows that our proposed algorithm *TBSA* achieved the least number of iterations than the JL approach and the non-threshold *TBSA* for the synthetic datasets when β is varied from 0.3 to 0.7. Nonetheless, these three approaches demonstrate the same number of iterations when $\beta = 0.1, 0.2, 0.8$, and 0.9 with 1, 1, 8, and 10 iterations, respectively. Similar trend is evident in Figure 13 (a2), in which all three approaches exhibit the same number of iterations for the NBA dataset when $\beta = 0.1 - 0.3$, and 0.9 with 1, 1, 1, and 16 iterations, respectively. Nevertheless, *TBSA* achieved the least number of iterations than the other two approaches



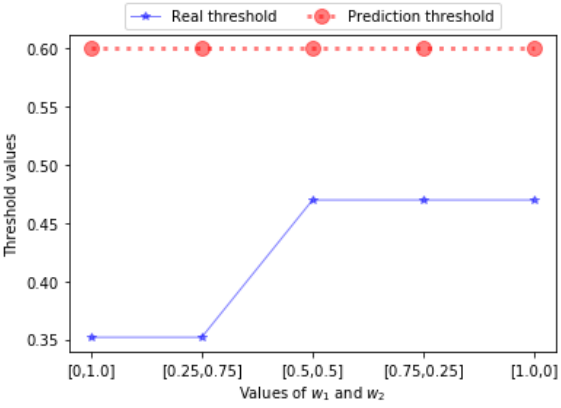
(a1) Synthetic Dataset



(a2) NBA Dataset

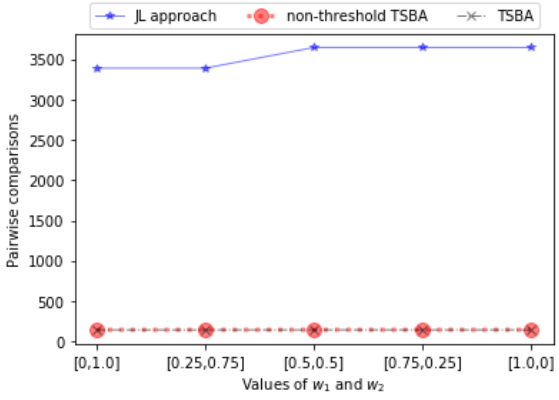


(a3) Synthetic Dataset

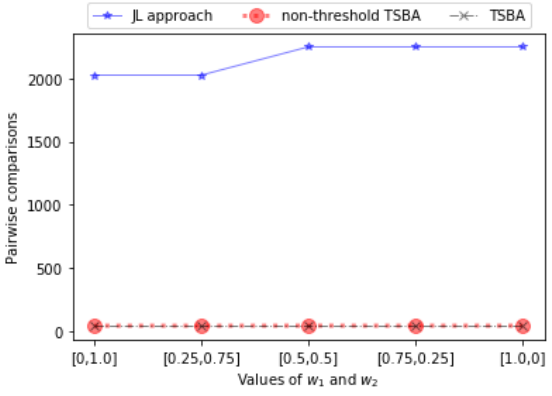


(a4) NBA Dataset

(a) Number of iterations



(b1) Synthetic Dataset

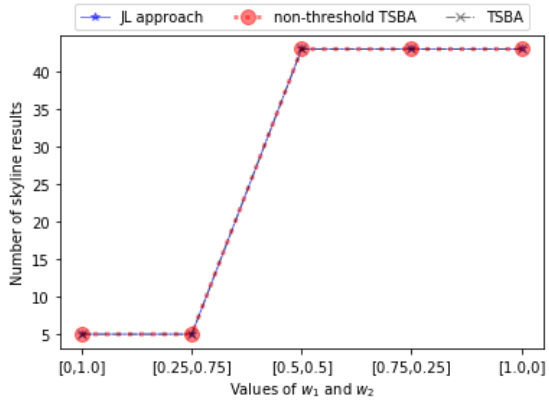


(b2) NBA Dataset

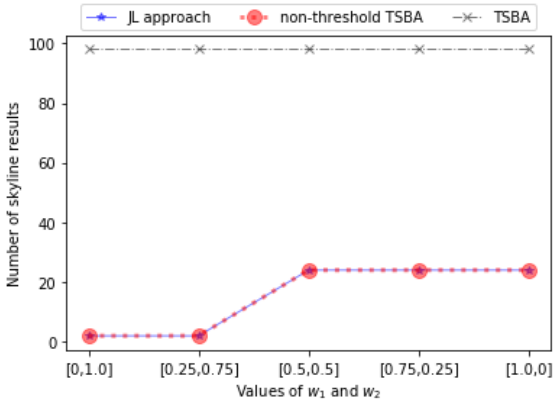
(b) Number of pairwise comparisons

FIGURE 15. The performance results based on the weight values of ω_1 and ω_2.

when $\beta = 0.4 - 0.8$. Meanwhile, Figure 13 (a3) shows that for the synthetic dataset, the τ value predicted by the *TBSA* in most cases, i.e. $\beta = 0.4 - 0.8$, is lower than the real τ value, except for the following $\beta = 0.1$, 0.2 and 0.3 where $\tau = 0.444$, 0.444, and 0.555, respectively; which are greater than the real τ value. Interestingly, the τ value predicted by *TBSA* is 0 (lowest range) when $\beta = 0.8$ and 1 (highest range) when $\beta = 0.9$. When the τ value predicted by *TBSA* is at the lowest range (i.e. 0) while the real τ value is at a higher range, more iterations are needed to achieve the intended level of informative skyline results, L_{S_I}, as shown in Figure 13 (a1) . However, when the τ value predicted by *TBSA*

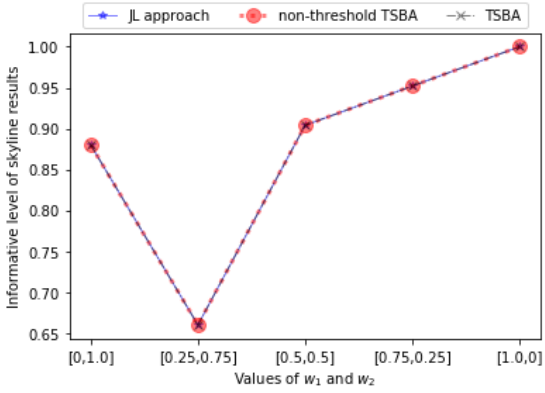


(c1) Synthetic Dataset

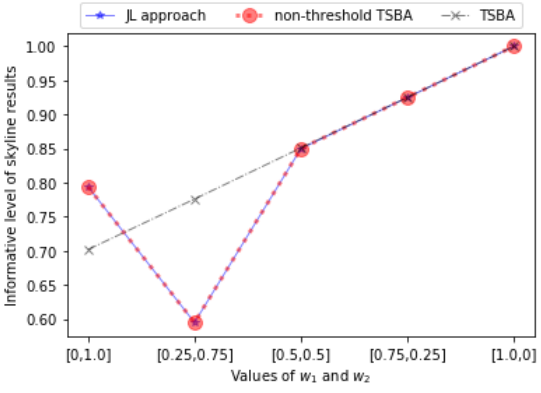


(c2) NBA Dataset

(c) Number of skyline results

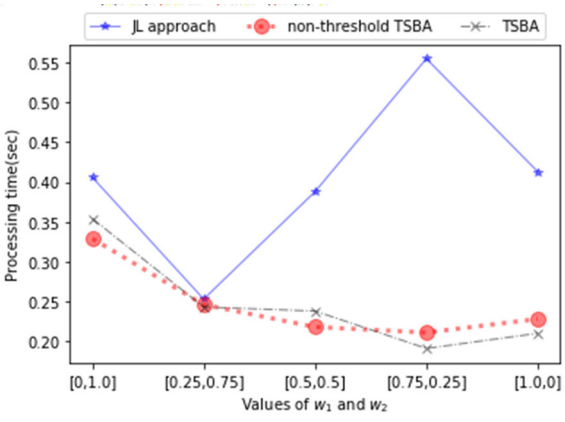


(d1) Synthetic Dataset

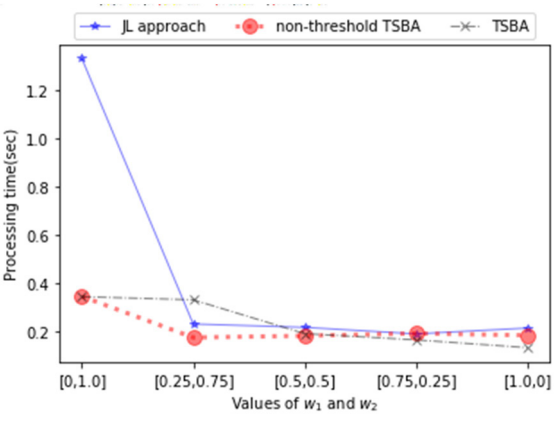


(d2) NBA Dataset

(d) Level of informative skyline results



(e1) Synthetic Dataset



(e2) NBA Dataset

(e) Processing time

FIGURE 15. ***(Continued.)*** **The performance results based on the weight values of ω_1 and ω_2.**

is at the highest range (i.e. 1), we cannot simply conclude that iteration is unnecessary and the set of skyline results is empty. Thus, for such case, the τ value is reset to 0. Meanwhile, for the NBA dataset, the τ value predicted by *TBSA* is higher than the real τ value when $\beta = 0.4-0.8$, and equal for the other β values as depicted in Figure 13 (a4). As a result, only a single iteration is performed as shown in Figure 13 (a2), with the exception of $\beta = 0.9$ where the τ value predicted by *TBSA* is at the highest range (i.e. 1); in which the same rule explained earlier is applied.

Figures 13 (b1) presents the performance results of *TBSA*, JL approach, and non-threshold *TBSA* with regard to the

number of pairwise comparisons for the synthetic dataset. In most cases, the number of pairwise comparisons of both *TBSA* and non-threshold *TBSA* is almost equal; with *TBSA* exhibiting the least number of pairwise comparisons among the approaches being compared with exception of $\beta = 0.8$ and 0.9, where both the *TBSA* and non-threshold *TBSA* achieved the same number of pairwise comparisons; which support the results on the number of iterations performed by these approaches as shown in Figure 13 (a1). Similar trends can be seen in Figure 13 (b2) for the NBA dataset. Nonetheless, the JL approach demonstrates the highest number of pairwise comparisons for most cases of both datasets.

On the other hand, the number of skyline results, $|S_I|$, derived by *TBSA*, JL approach, and non-threshold *TBSA* is the same for most cases of the synthetic dataset as shown in Figure 13 (c1) and NBA dataset as shown in Figure 13 (c2). The only exceptions are the synthetic dataset with $\beta = 0.1$ and $\beta = 0.2$ and NBA dataset with $\beta = 0.4 - 0.7$, from which the *TBSA* derived a larger number of skyline results than the other two approaches. For instance, the JL approach and non-threshold *TBSA* derived $|S_I| = 1050$ and $|S_I| = 1$ for the synthetic dataset with $\beta = 0.1$ and $\beta = 0.2$, respectively; while *TBSA* generated $|S_I| = 1318$ and $|S_I| = 37$. Similar trends can be seen in figures 13 (d1) and 13 (d2) where *TBSA*, JL approach, and non-threshold *TBSA* achieved the same level of informative skyline results, L_{S_I}, for most cases with the exception of the synthetic dataset with $\beta = 0.2$ and NBA dataset with $\beta = 0.4 - 0.7$ where *TBSA* exhibits a higher L_{S_I}. For instance, for the synthetic dataset with $\beta = 0.2$, the L_{S_I} achieved by the JL approach and non-threshold *TBSA* is 0.5 with $|S_I| = 1$; while *TBSA* gained $L_{S_I} = 0.971$ with $|S_I| = 37$. Although the skyline derived by the JL approach and non-threshold *TBSA* is complete with no missing values; however, $|S_I| = 1$ clearly indicates that the returned result is too few, which has an impact on the L_{S_I} value. Thus, there are cases where $|S_I| < |S|_u$; with $|S|_u$ set to 10; and $L_{S_I} = 0.5$. This is because the returned set of skylines, S_I, is complete with no missing values, which have an impact on the L_{S_I} value with $\omega_1 S_\alpha = 0$ and $\omega_2 \left[1 - S_\beta\right] = 0.5$.

Figures 13 (e1) and 13 (e2) present the performance results of *TBSA*, JL approach, and non-threshold *TBSA* with regard to the processing time for both datasets, namely: synthetic and NBA, respectively. In most cases, *TBSA* exhibits the least processing time. The only exception is the synthetic dataset with $\beta = 0.1 - 0.3$; in which the JL approach demonstrates the least processing time.

D. EFFECT OF THE NUMBER OF SKYLINE RESULTS, $|S|_u$

We also study the effect of the number of skyline results, $|S|_u$, on the performance of *TBSA*, JL approach, and non-threshold *TBSA*. Since, the aim of this study is to avoid having too few skyline results, the $|S|_u$ serves as a guide for the minimum results that must be returned. The parameter settings of the synthetic dataset are as follows: the number of skyline result, $|S|_u$, is varied from 5 to 20 with the size of dataset, $|D|$, fixed to 50,000, and the number of dimensions, $|d|$, set to 10. Meanwhile, the number of skyline results, $|S|_u$, for the NBA dataset is varied from 5 to 20 with $|D| = 17,758$ and $|d| = 17$. The settings for the other parameters, namely: L_{S_I}, ω_1, and ω_2, are the same as in Section VI-A; while the missing value ratio, β, is set to 0.5 for both datasets. Figures 14 (a) – (e) present the performance of *TBSA*, JL approach, and non-threshold *TBSA* with regard to (i) number of iterations, (ii) number of pairwise comparisons, (iii) number of skyline results, (iv) level of informative skyline results, and (v) processing time, respectively.

Figures 14 (a1) and 14 (a2) demonstrate that, for both the synthetic and NBA datasets, the number of iterations performed by our proposed algorithm *TBSA* is either one or two iterations, which is the least number of iterations compared to the JL approach and the non-threshold *TBSA* with 6 – 9 iterations. Meanwhile, Figure 14 (a3) shows that for the synthetic dataset, the τ value predicted by the *TBSA* in most cases is lower than the real τ value, except for $|S|_u = 5$. Here, the τ value predicted by the *TBSA* is 0.571 while the real τ value is 0.5. However, for the NBA dataset, the opposite can be seen where the τ value predicted by the *TBSA* is higher than the real τ value for all cases of $|S|_u$ being considered; as shown in Figure 14 (a4). It appears that when the τ value predicted by the *TBSA* is lower than the real τ value, more iterations are needed to reach the targeted level of informative skyline results, L_{S_I}.

Figures 14 (b1) and 14 (b2) present the performance results of *TBSA*, JL approach, and non-threshold *TBSA* with regard to the number of pairwise comparisons for the synthetic and NBA dataset, respectively. In most cases, the number of pairwise comparisons of both *TBSA* and non-threshold *TBSA* is almost equal; with *TBSA* exhibits the least number of pairwise comparisons among the approaches being compared. Obviously, the JL approach shows the highest number of pairwise comparisons for all cases of both datasets.

On the other hand, the actual number of skyline results, $|S_I|$, derived by *TBSA*, JL approach, and non-threshold *TBSA* is the same for all cases of the synthetic dataset as shown in Figure 14 (c1); while for the NBA dataset with $|S|_u = 5$ and $|S|_u = 10$, *TBSA* derived more skyline results as compared to the other two approaches as shown in Figure 14 (c2). Similar trends can be seen in figures 14 (d1) and 14 (d2) where *TBSA*, JL approach, and non-threshold *TBSA* achieved the same level of informative skyline results, L_{S_I}, for most cases with the exception of the NBA dataset with $|S|_u = 5$ and $|S|_u = 10$ where *TBSA* exhibits a lower and a higher L_{S_I}, respectively. Nonetheless, $|S_I| \geq |S|_u$ and $L_{S_I} > 0.850$ for all cases which is higher than 0.5; the minimum value of L_{S_I} set for the experiment.

Figures 14 (e1) and 14 (e2) present the performance results of *TBSA*, JL approach, and non-threshold *TBSA* with regard to the processing time for both datasets, namely: synthetic and NBA, respectively. In most cases, *TBSA* exhibits the least processing time. The only exceptions are the synthetic dataset

with $|S|_u = 20$ and the NBA dataset with $|S|_u = 20$; in which the non-threshold *TBSA* achieves the least processing time.

E. EFFECT OF THE WEIGHT VALUES OF ω_1 AND ω_2

In this section, the performance of *TBSA*, JL approach, and non-threshold *TBSA* with regard to the weight values of ω_1 and ω_2, is reported. ω_i is a weight given to the i-th property of the informative skyline results. The value of ω_i is between [0, 1] while $\omega_1 + \omega_2 = 1$. An equal value given to both ω_1 and ω_2, i.e. $\omega_1 = \omega_2 = 0.5$, implies that the two properties are equally important. The parameter settings of the synthetic dataset are as follows: the weight values of ω_1 and ω_2 are varied to represent different levels of importance among the properties of the informative skyline results: [0, 1.0], [0.25, 0.75], [0.5, 0.5], [0.75, 0.25], and [1.0, 0]; the size of dataset, $|D|$, is fixed to 50,000, and the number of dimensions, $|d|$, is set to 10. The same settings of ω_1 and ω_2 are applied for the NBA dataset with $|D| = 17,758$ and $|d| = 17$. For both datasets, the level of informative skyline results, L_{S_I}, is set to 0.5, the minimum number of skyline results, $|S|_u$, is set to 10, while the missing value ratio, β, is set to 0.5. Figures 15 (a) – (e) present the performance of *TBSA*, JL approach, and non-threshold *TBSA* with regard to (i) number of iterations, (ii) number of pairwise comparisons, (iii) number of skyline results, (iv) level of informative skyline results, and (v) processing time, respectively.

Figures 15 (a1) and 15 (a2) demonstrate that, for both the synthetic and NBA datasets, the number of iterations performed by *TBSA* is either one or two iterations, which is the least number of iterations compared to the JL approach and the non-threshold *TBSA* with 6 – 9 iterations. Meanwhile, Figure 15 (a3) shows that for the synthetic dataset, the τ value predicted by the *TBSA* in most cases is lower than the real τ value, except for the weights [0, 1.0] and [0.25, 0.75]. Here, the τ value predicted by the *TBSA* is 0.575 while the real τ value is 0.500. However, for the NBA dataset, the opposite can be seen where the τ value predicted by the *TBSA* is higher, i.e. 0.60, than the real τ value for all cases of ω_1 and ω_2 being considered; as shown in Figure 15 (a4). It is evident that more than a single iteration is required to reach the targeted level of informative skyline results, L_{S_I}, when the τ value predicted by the *TBSA* is smaller than the real τ value.

Figures 15 (b1) and 15 (b2) present the performance results of *TBSA*, JL approach, and non-threshold *TBSA* with regard to the number of pairwise comparisons for the synthetic and NBA dataset, respectively. In most cases, the number of pairwise comparisons of both *TBSA* and non-threshold *TBSA* is almost equal; with *TBSA* exhibiting the least number of pairwise comparisons among the approaches being compared. Obviously, the JL approach shows the highest number of pairwise comparisons for all cases of both datasets.

On the other hand, the actual number of skyline results, $|S_I|$, derived by *TBSA*, JL approach, and non-threshold *TBSA* is the same for all cases of the synthetic dataset as shown in Figure 15 (c1); while for the NBA dataset, *TBSA* derived more skyline results as compared to the other two approaches as shown in Figure 15 (c2). Similar trends can be seen in figures 15 (d1) and 15 (d2) where *TBSA*, JL approach, and non-threshold *TBSA* achieved the same level of informative skyline results, L_{S_I}, for most cases with the exception of the NBA dataset with the weights [0, 1.0] and [0.25, 0.75] where *TBSA* exhibits a lower and a higher L_{S_I}, respectively. Interestingly, for the NBA dataset with the weights [0, 1.0] and [0.25, 0.75], the JL approach and non-threshold *TBSA* return too few of S_I. Nevertheless, this has no effect on the L_{S_I} because the weights assigned to the property $|S_I|$, i.e. ω_1 are 0 and 0.25, respectively; which suggests that the rate of missing values of S_I is more important than the $|S_I|$ itself. Nonetheless, $|S_I| \geq |S|_u$ and $L_{S_I} > 0.600$ for all cases which is higher than 0.5; the minimum value of L_{S_I} set for the experiment.

Figures 15 (e1) and 15 (e2) present the performance results of *TBSA*, JL approach, and non-threshold *TBSA* with regard to the processing time for both datasets, namely: synthetic and NBA. In most cases, *TBSA* exhibits the least processing time. The only exceptions are the synthetic dataset with the weights [0, 1.0] and [0.5, 0.5] and the NBA dataset with the weights [0.25, 0.75]; in which the non-threshold *TBSA* achieves the least processing time.

VII. CONCLUSION

This paper proposes a *Threshold-based Bucket Skyline Algorithm* (*TBSA*), which mainly aims at finding informative skyline results of skyline queries over incomplete data. *TBSA* employs the *bucket* and *cluster algorithms* to reduce the effect of cyclic dominance relation and loss of transitive dominance property. Furthermore, to resolve the issues of insufficient and noninformative skyline results caused by the existence of incomplete data in the database, *TBSA* utilises a threshold prediction model, which explores the subspaces that have high potential to derive the informative skyline results and discards those subspaces with high incomplete ratio. This is to avoid a trial-and-error or an incremental tuning iterations in searching the potential subspaces to be examined which is unwise and costly. The threshold prediction model is constructed using *decision tree regression* (*DTR*) after a thorough analysis is conducted among the *multi linear regression* (*MLR*), *support vector regression* (*SVR*), *decision tree regression* (*DTR*), and *neural network regression* (*NNR*). Moreover, a metric is proposed to measure the level of informative skyline results derived by *TBSA* at each iteration that assists *TBSA* to make further refinement on the skyline results.

The work presented in this paper can be further enhanced by: (i) exploring other variations of skyline query formulations such as *top-k skyline queries* that rank and deliver the exact k skyline objects from the set of informative skyline results derived by *TSBA*; to meet the number of skylines as required by the user. *Range skyline queries* is another interesting queries that can be investigated which allow flexibility to the users to specify certain conditions on specific dimensions of the database; like ‘‘find those apartments whose price rate per night is between $100 and $200’’; hence, the properties

of the informative skyline results can be extended to include how close are the skyline objects derived by *TBSA* in satisfying the conditions as specified in the user's queries. (ii) *Threshold-based Bucket Skyline Algorithm* (*TBSA*) was initially proposed under the assumption that databases are saved in a single node; *TBSA* can be extended to other platforms like distributed database, data stream, etc; where data are not always available in a single source. Here, *TBSA* can be applied at each node and the local informative skyline results that are obtained at each node are integrated, and further refinement will have to be performed before the final global informative skyline results are realised.

ACKNOWLEDGMENT

All opinions, findings, conclusions, and recommendations in this article are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] J. Liu, J. Yang, L. Xiong, J. Pei, J. Luo, Y. Guo, S. Ma, and C. Fan, "Skyline diagram: Efficient space partitioning for skyline queries," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 1, pp. 271–286, Jan. 2021.

[2] X. Han, J. Wang, J. Li, and H. Gao, "Efficient computation of G-skyline groups on massive data," *Inf. Sci.*, vol. 587, pp. 300–322, Mar. 2022.

[3] L. Rui and L. Dominique, "Efficient skyline computation in high-dimensionality domains," in *Proc. Int. Conf. Extending Database Technol. (EDBT)*, 2020, pp. 459–462.

[4] S. Borzsony, D. Kossmann, and K. Stocker, "The skyline operator," in *Proc. 17th Int. Conf. Data Eng.*, Apr. 2001, pp. 421–430.

[5] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, "Skyline with presorting," in *Proc. 19th Int. Conf. Data Eng.*, Mar. 2003, pp. 717–719.

[6] P. Godfrey, J. R. Shipley, and J. Gryz, "Maximal vector computation in large data sets," in *Proc. 31st Int. Conf. Very Large Data Bases (VLDB)*, 2005, pp. 229–240.

[7] I. Bartolini, P. Ciaccia, and M. Patella, "SaLSa: Computing the skyline without scanning the whole sky," in *Proc. 15th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, vol. 2006, pp. 405–414.

[8] K. Tan, P.-K. Eng, and B. C. Ooi, "Efficient progressive skyline computation," in *Proc. 27th. Int. Conf. Very Large Data Bases (VLDB)*, 2001, pp. 301–310.

[9] D. Kossmann, F. Ramsak, and S. Rost, "Shooting stars in the sky: An online algorithm for skyline queries," in *Proc. 28th Int. Conf. Very Large Databases (VLDB)*, 2002, pp. 275–286.

[10] D. Papadias, Y. Tao, G. Fu, and B. Seeger, "Progressive skyline computation in database systems," *ACM Trans. Database Syst.*, vol. 30, no. 1, pp. 41–82, Mar. 2005.

[11] X. Miao, Y. Gao, G. Chen, B. Zheng, and H. Cui, "Processing incomplete k nearest neighbor search," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 6, pp. 1349–1363, Dec. 2016.

[12] X. Miao, Y. Gao, L. Zhou, W. Wang, and Q. Li, "Optimizing quality for probabilistic skyline computation and probabilistic similarity search," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1741–1755, Sep. 2018.

[13] K. Zhang, H. Gao, X. Han, Z. Cai, and J. Li, "Modeling and computing probabilistic skyline on incomplete data," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 7, pp. 1405–1418, Jul. 2020.

[14] W. Ren, X. Lian, and K. Ghazinour, "Skyline queries over incomplete data streams," *VLDB J.*, vol. 28, no. 6, pp. 961–985, Dec. 2019.

[15] S. Chaudhuri, N. Dalvi, and R. Kaushik, "Robust cardinality and cost estimation for skyline operator," in *Proc. 22nd Int. Conf. Data Eng. (ICDE)*, 2006, pp. 64–73.

[16] Z. Zhang, Y. Yang, R. Cai, D. Papadias, and A. Tung, "Kernel-based skyline cardinality estimation," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2009, pp. 509–522.

[17] X. Miao, Y. Wu, J. Peng, Y. Gao, and J. Yin, "Efficient and effective cardinality estimation for skyline family," in *Proc. ACM Manage. Data*, 2023, pp. 104–125.

[18] M. E. Khalefa, M. F. Mokbel, and J. J. Levandoski, "Skyline query processing for incomplete data," in *Proc. IEEE 24th Int. Conf. Data Eng.*, Apr. 2008, pp. 556–565.

[19] M. Shamsul Arefin, "Skyline sets queries for incomplete data," *Int. J. Comput. Sci. Inf. Technol.*, vol. 4, no. 5, pp. 67–80, Oct. 2012.

[20] A. A. Alwan, H. Ibrahim, and N. I. Udzir, "A framework for identifying skylines over incomplete data," in *Proc. 3rd Int. Conf. Adv. Comput. Sci. Appl. Technol.*, Dec. 2014, pp. 79–84.

[21] K. Zhang, H. Gao, H. Wang, and J. Li, "ISSA: Efficient skyline computation for incomplete data," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2016, pp. 321–328.

[22] J. Lee, H. Im, and G.-W. You, "Optimizing skyline queries over incomplete data," *Inf. Sci.*, vols. 361–362, pp. 14–28, Sep. 2016.

[23] R. Bharuka and P. S. Kumar, "Finding skylines for incomplete data," in *Proc. 24th Australas. Database Conf.*, 2013, pp. 109–117.

[24] Y. Gulzar, A. A. Alwan, N. Salleh, I. F. A. Shaikhli, and S. I. M. Alvi, "A framework for evaluating skyline queries over incomplete data," *Proc. Comput. Sci.*, vol. 94, pp. 191–198, Jan. 2016.

[25] Y. Gulzar, A. A. Alwan, and S. Turaev, "Optimizing skyline query processing in incomplete data," *IEEE Access*, vol. 7, pp. 178121–178138, 2019.

[26] J. He and X. Han, "Efficient skyline computation on massive incomplete data," *Data Sci. Eng.*, vol. 7, no. 2, pp. 102–119, Apr. 2022.

[27] Y. Gulzar and A. A. Alwan, "CIDS: An efficient algorithm for processing skyline queries for partially complete data in cloud environment," *IEEE Access*, vol. 10, pp. 66449–66466, 2022.

[28] A. A. Alwan, H. Ibrahim, N. I. Udzir, and F. Sidi, "Processing skyline queries in incomplete distributed databases," *J. Intell. Inf. Syst.*, vol. 48, no. 2, pp. 399–420, Apr. 2017.

[29] G. B. Dehaki, H. Ibrahim, F. Sidi, N. I. Udzir, A. A. Alwan, and Y. Gulzar, "Efficient computation of skyline queries over a dynamic and incomplete database," *IEEE Access*, vol. 8, pp. 141523–141546, 2020.

[30] M. B. Swidan, A. A. Alwan, S. Turaev, H. Ibrahim, A. Z. Abualkishik, and Y. Gulzar, "Skyline queries computation on crowdsourced-enabled incomplete database," *IEEE Access*, vol. 8, pp. 106660–106689, 2020.

[31] X. Miao, Y. Gao, G. Chen, and T. Zhang, "K-dominant skyline queries on incomplete data," *Inf. Sci.*, vols. 367–368, pp. 990–1011, Nov. 2016.

[32] Y. Gao, X. Miao, H. Cui, G. Chen, and Q. Li, "Processing k-skyband, constrained skyline, and group-by skyline queries on incomplete data," *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4959–4974, Aug. 2014.

[33] X. Miao, Y. Gao, B. Zheng, G. Chen, and H. Cui, "Top-k dominating queries on incomplete data," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 252–266, Jan. 2016.

[34] C.-Y. Chan, H. V. Jagadish, K.-L. Tan, and A. K. H. Tung, "On high dimensional skylines," in *Proc. 10th Int. Conf. Extending Database Technol. (EDBT)*. Berlin, Germany: Springer, 2006, pp. 478–495.

[35] R. Bharuka and P. S. Kumar, "Finding superior skyline points from incomplete data," in *Proc. 19th Int. Conf. Manage. Data*, Ahmedabad, India, 2013, pp. 35–44.

[36] P. Bosc, A. Hadjali, and O. Pivert, "On possibilistic skyline queries," in *Proc. Int. Conf. Flex. Query Answer. Syst. (FQAS)*, 2011, pp. 412–423.

[37] S. Elmi, M. A. Bach Tobji, A. Hadjali, and B. Ben Yaghlane, "Selecting skyline stars over uncertain databases: Semantics and refining methods in the evidence theory setting," *Appl. Soft Comput.*, vol. 57, pp. 88–101, Aug. 2017.

[38] D. Belkasmi, A. Hadjali, and H. Azzoune, "On fuzzy approaches for enlarging skyline query results," *Appl. Soft Comput.*, vol. 74, pp. 51–65, Jan. 2019.

[39] C. Bourahla, R. Maamri, and S. Brahimi, "Skyline recomputation in big data," *Inf. Syst.*, vol. 114, Mar. 2023, Art. no. 102164.

[40] A. Vlachou, C. Doulkeridis, J. B. Rocha-Junior, and K. Nørvåg, "Decisive skyline queries for truly balancing multiple criteria," *Data Knowl. Eng.*, vol. 147, Sep. 2023, Art. no. 102206.

[41] B. Yin, X. Wei, and Y. Liu, "Finding the informative and concise set through approximate skyline queries," *Expert Syst. Appl.*, vol. 119, pp. 289–310, Apr. 2019.

[42] K. Mouratidis, K. Li, and B. Tang, "Marrying top-k with skyline queries: Relaxing the preference input while producing output of controllable size," in *Proc. Int. Conf. Manage. Data*, Jun. 2021, pp. 1317–1330.

[43] M. Bai, Y. Han, P. Yin, X. Wang, G. Li, B. Ning, and Q. Ma, "S_IDS: An efficient skyline query algorithm over incomplete data streams," *Data Knowl. Eng.*, vol. 149, Jan. 2024, Art. no. 102258.

ZHANG XIAOWEI (Member, IEEE) received the bachelor's and master's degrees in computer science and technology from Zhengzhou University of Light Industry, China, in 2006 and 2009, respectively. He is currently pursuing the Ph.D. degree with Universiti Putra Malaysia (UPM), Malaysia. His research interests include database systems and data science.

HAMIDAH IBRAHIM (Member, IEEE) received the Ph.D. degree in computer science from the University of Wales Cardiff, U.K., in 1998. She is currently a Full Professor with the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM). Her current research interests include databases (distributed, parallel, mobile, biomedical, and XML), focusing on issues related to integrity maintenance/checking, ontology/schema/data integration, ontology/schema/data mapping, cache management, access control, data security, transaction processing, query optimization, query reformulation, preference evaluation–context-aware, information extraction, concurrency control, and data management in mobile, grid, and cloud.

FATIMAH SIDI (Member, IEEE) received the Ph.D. degree in management information systems from Universiti Putra Malaysia, Malaysia (UPM), in 2008. She is currently a Professor of the discipline of computer science with the Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM). Her current research interests include knowledge and information management systems, data and knowledge engineering, databases, and data warehouses.

SITI NURULAIN MOHD RUM received the Diploma degree from Universiti Teknologi Mara (UiTM), the bachelor's degree from Universiti Teknologi Malaysia (UTM), and the master's and Ph.D. degrees from the University of Malaya (UM). She is currently a Lecturer with the Department of Computer Science and Information Technology, Universiti Putra Malaysia. Before joining academia, she spent over 15 years as an IT professional, working on several IT projects, including software development, database and data center migration, virtualization infrastructure development, and procurement of hardware that includes servers and storage for the data center. She has published several journal articles and presented research papers at international conferences. She is also actively involved in doing research and consulting in the IT field. Her research interests include, but are not limited to, the areas of database processing, artificial intelligence, social media analytics, and data sciences.

MUDATHIR AHMED MOHAMUD (Member, IEEE) received the B.Sc. degree in information technology from the Faculty of Computing, SIMAD University, Mogadishu, Somalia, and the M.Sc. degree in computer science from Universiti Putra Malaysia (UPM). He is currently a Lecturer with the Faculty of Computing, SIMAD University. His research interests include query optimization, preference query evaluation, data science, database integration, data stream, machine learning, data mining, and information systems.