

## RESEARCH ARTICLE

# A Performance Analysis of Prediction Techniques in Handling High-Dimensional Uncertain Data for the Application of Skyline Query Over Data Stream

MUDATHIR AHMED MOHAMUD<sup>1</sup>, (Member, IEEE), HAMIDAH IBRAHIM<sup>1</sup>, (Member, IEEE), FATIMAH SIDI<sup>1</sup>, (Member, IEEE), SITI NURULAIN MOHD RUM<sup>1</sup>, ZARINA BINTI DZOLKHI FLI<sup>2</sup>, AND ZHANG XIAOWEI<sup>1</sup>

<sup>1</sup>Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Selangor 43400, Malaysia

<sup>2</sup>Department of Computer and Networking, Faculty of Computing, Universiti Malaysia Pahang, Kuantan, Pahang 26300, Malaysia

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 the Universiti Putra Malaysia.

**ABSTRACT** The proliferation of high-dimensional data in many advanced database applications is a result of today's technological advancements. These data points that correspond to objects are often without a precise description, which make their representation uncertain. While the concept of data streaming is not new, its practical uses are only recently emerging. This research focuses on continuous range data—a type of uncertain data common in database applications—that do not have explicit representations of their exact values. Furthermore, the identification of skyline objects—one of the popular database applications—becomes more challenging when skylines are to be identified from a collection of continuously generated input data streams where objects might have imprecise values. This makes it imperative to determine which approach has the optimal accuracy for estimating or predicting the uncertain values and at the same time able to handle a massive streams of data that are continuously generated and analyze them almost instantly to provide accurate and timely responses. Given this, the following techniques are selected—*Linear Regression (LR)*, *k-Nearest Neighbour (k-NN)*, *Random Forest (RF)*, *Decision Trees (DT)*, and *Centre and Range Method (CRM)* and their effectiveness is evaluated in terms of execution time, precision, recall, F1-score, and root mean square error (RMSE). Additionally, in order to verify the accuracy of each prediction technique, the predicted data derived from its model is used to derive skyline objects, which are subsequently compared to the actual skyline results. An inaccurate prediction of a continuous range value would result in incorrect set of skyline objects.

**INDEX TERMS** Prediction techniques, uncertain data, high dimensional, skyline query, data stream.

## I. INTRODUCTION

Today's technological advancements have led to a proliferation of high-dimensional data in many advanced database applications. High-dimensional data refers to datasets with a large number of features or attributes that can be difficult to

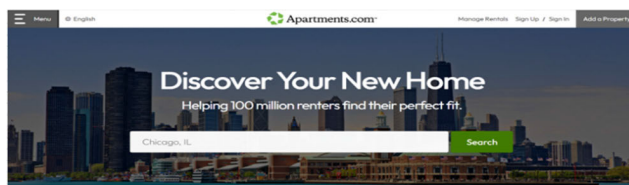
The associate editor coordinating the review of this manuscript and approving it for publication was Vlad Diaconita<sup>1</sup>.

deal with due to their complexity and the massive amounts of data they hold. Often, these data points may correspond to objects which are imprecisely described, making their representation deemed uncertain. Data uncertainty which is generally defined as the degree of unknown, unreliable, imprecise, and inaccurate places a significant demand on advanced techniques that are able to precisely predict the uncertainty [1], [2], [3], [4]. These techniques are essential in

many domains where uncertainty and high-dimensional data are unavoidable like medical logs, data mining applications, social survey dataset, robotics, gesture and speech recognition, agriculture, object detection and recognition, process optimization, supply chain optimization, surveillance, and intelligent transportation system.

While the concept of data streaming is not new, its practical uses are only recently emerging. Streaming data is essential for decision making applications. In contrast to traditional applications that produce data that are stored in finite persistent relations, a data stream is made up of a series of data elements that are time varying (time sensitive), continuous, real time, volatile, and unrepeatable [5]. Hence, every object  $o_i$  has a timestamp indicating the arrival time,  $arr(o_i)$ , and expiry time,  $exp(o_i)$ , of the object in the stream. Processing data stream is challenging for a number of reasons: (i) the objects in the stream arrive online, (ii) the system has no control over the order in which objects arrive to be processed, either within a data stream or across data streams, (iii) data streams are potentially unbounded in size, and (iv) once an object from a data stream has been processed it is discarded or archived, it cannot be retrieved easily unless it is explicitly stored in memory, which typically is small relative to the size of the data streams [6].

Quality issues in data may arise particularly when the data were gathered from disparate sources of data stream. The data might contain duplicates or can be out-of-date, insecure, inconsistent, inaccurate, or incomplete. This study focuses on data with continuous range values that lacked explicit representations of their exact values; a type of uncertain data that is typical in database applications. An example is shown in Figure 1 that presents a list of apartments in Chicago, United States of America which is taken from the Apartments.com website (<https://www.apartments.com>). Each apartment has specific values for the attributes *AP ID*, *monthly rent*, *bedrooms*, *bathrooms*, and *square feet*. The values 2,356 – 3,025, 4,190 – 4,250, 2,631 – 3,435, 3,840 – 4,010, and 1,956 – 2,492 of the *monthly rent* are examples of imprecise values which make these data uncertain.



AP ID	Monthly Rent (\$)	Bedrooms	Bathrooms	Square Feet
813	2,356 – 3,025	1	1	590
624	4,190 – 4,250	2	2	1,277
1308	2,631 – 3,435	1	1	700
1102	4,217	2	2	1,130
725	1,995	Studio	1	464
715	2,795	1	1	678
721	3,840 – 4,010	2	2	1,082
1001	6,595	3	2.5	1,494
805	1,956 – 2,492	Studio	1	576
809	2,030	Studio	1	508

FIGURE 1. Samples of apartments in apartments.com.

Apparently, handling the uncertainty and high-dimensional data requires sophisticated techniques. The knowledge of artificial intelligence (AI), particularly, machine learning (ML) is the key to intelligently analyze these data and develop the corresponding applications. ML has been widely adopted in recent years for a variety of purposes including but not limited to predicting missing values. While supervised learning, a machine learning approach, relies on labelled dataset to classify data and predict outcomes, unsupervised learning finds and classifies hidden patterns in the unlabeled datasets. Additionally, ML techniques are well trained to handle vast and complex volumes of data in numerous domains [7] and have the ability to predict values of continuous numerical and exact data. Nonetheless, they are unable to directly predict values of uncertain data resulting from repeated measurements, data staleness, measurement errors, or imprecise data [8].

The main goal of this study is to identify an effective prediction technique that could handle uncertain data in the form of continuous range values over data streams. Since the precise value of the continuous range values is not known, it is crucial to identify a technique that can most accurately predict or estimate the uncertain values. It is also imperative that the technique be capable of handling the unique characteristics of data streams, that are rapid data arrivals and strict response time constraints. Apparently, the prediction technique should be effective enough to manage a collection of continuously generated input data streams and analyze these data streams in close to real-time to offer accurate and fast response. In light of this, the following methods—*Linear Regression (LR)*, *k-Nearest Neighbour (k-NN)*, *Random Forest (RF)*, *Decision Trees (DT)*, and *Centre and Range Method (CRM)*—are chosen, and their performance is assessed in terms of execution time, precision, recall, F1-score, and root mean square error (RMSE). Furthermore, the predicted values of each technique that are based on a set of objects having imprecise values are analyzed to derive skyline objects. Skyline objects are objects that are not dominated by any other object in a given database. In other words, an object  $o_i$  dominates another object  $o_j$  when  $o_i$  is not worse than  $o_j$  on all dimensions and is better than  $o_j$  on at least one dimension of the database. This implies that an inaccurate prediction of a continuous range value would result in incorrect set of skyline objects. To the best of our knowledge, this is the first study to thoroughly examine prediction techniques in an effort to determine the most reliable and highest-performing technique in handling uncertain and high-dimensional data to be applied over data stream; where uncertain data take the form of continuous range values. The following are the substantial contributions made by this work:

- We have performed a thorough investigation and have highlighted the significance of identifying an effective prediction technique to be employed in data stream that demands accurate and fast response.
- We have conducted extensive analyses on five notable prediction techniques, namely: *Linear Regression (LR)*,

*k*-Nearest Neighbour (*k*-NN), Random Forest (RF), Decision Trees (DT), and Centre and Range Method (CRM) over uncertain and high-dimensional data; in which uncertainty is due to objects which are imprecisely described; a kind of uncertain data commonly found in database applications. The effective prediction technique is identified through the findings of the analyses which are primarily based on execution time, precision, recall, F1-score, and root mean square error (RMSE).

- To strengthen the findings, we further analyzed the selected prediction techniques in deriving skyline objects. In this analysis, precision, recall, and F1-score are measured by comparing the set of skyline objects obtained based on the predicted values of each technique to the actual set of skyline objects that is derived by employing the conventional skyline algorithm. The method with the highest precision, recall, and F1-score is said to be the most accurate prediction model.

The structure of this paper is as follows. The motivation behind this work is presented in Section II, with skyline queries as the application. Section III provides an overview of each technique that is being considered, and Section IV reports on the in-depth analyses we conducted to identify an effective and efficient technique for handling high-dimensional uncertain data. Section V the last section, contains a summary of the work and several suggestions for future enhancements.

## II. MOTIVATION

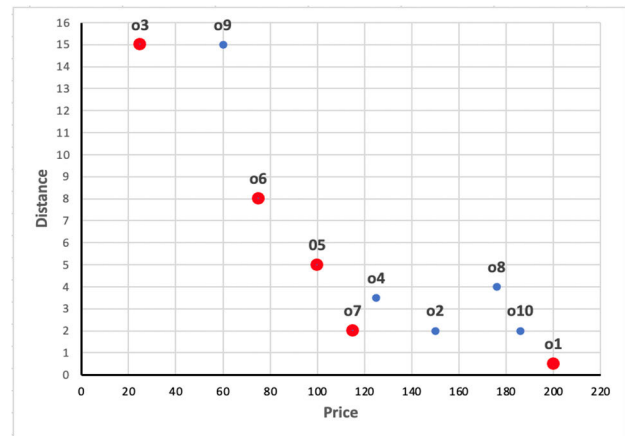
In this section, we explain the skyline queries that are widely used in multi-criteria decision support applications. We then highlight the challenges in deriving skyline objects for a given set of objects which are imprecisely described. We also give examples to rationalize the need for an effective prediction model that is not only accurate but efficient.

Skyline queries have received great attention in the database community during the past decades. The notion of skyline queries is to find a set of objects that is not dominated by any other objects; which can be formally written as follows: given a database,  $D$ , with  $m$  dimensions,  $d = \{d_1, d_2, \dots, d_m\}$  and  $n$  objects  $D = \{o_1, o_2, \dots, o_n\}$ ,  $o_i$  is said to dominate  $o_j$  denoted by  $o_i < o_j$  where  $i \neq j$  if and only if the following conditions hold:  $\forall d_k \in d, o_i.d_k \leq o_j.d_k \wedge \exists d_l \in d, o_i.d_l < o_j.d_l$ . For an example, consider the objects presented in Figure 2(a). Assume a user is interested in looking for hotels that are as cheap as possible and as close as possible to the city centre. Applying the skyline query on the given samples of objects would retrieve the following skyline results,  $S = \{o_1, o_3, o_5, o_6, o_7\}$  which are presented by red dots in Figure 2(b).  $o_2(150, 2)$  for instance, is not a skyline as it is being dominated by  $o_7(115, 2)$  since  $o_7$  has a lower price value than  $o_2$  with both having the same distance to the city center.

Processing skyline queries over databases with uncertainty imposes a number of challenges that negatively influence on

Object	Price	Distance
$o_1$	200	0.5
$o_2$	150	2
$o_3$	25	15
$o_4$	125	3.5
$o_5$	100	5
$o_6$	75	8
$o_7$	115	2
$o_8$	176	4
$o_9$	60	15
$o_{10}$	186	2

(a)



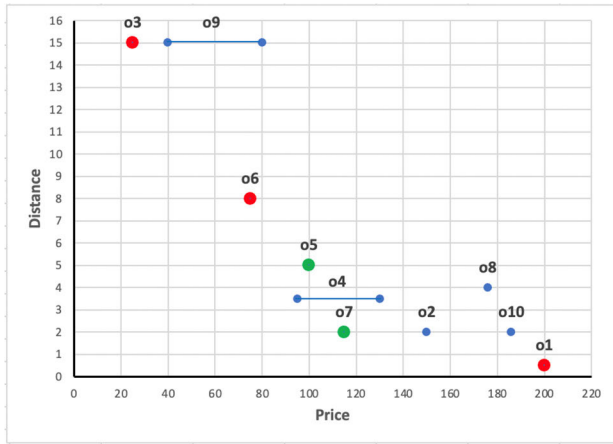
(b)

FIGURE 2. Results of skyline queries for the hotel database.

the skyline results. Figure 3 presents a similar samples of data as given in Figure 2; however in this example the price values of objects  $o_4$  and  $o_9$  are in the form of continuous range values (Figure 3(a)). Regardless of the price value of  $o_9$ ,  $o_3$  dominates  $o_9$ , i.e.  $o_3 < o_9$ , thus  $o_9$  is not a skyline. However, we cannot certainly conclude that  $o_4$  dominates  $o_5$  or vice versa without knowing the exact price value of  $o_4$ . Similarly, we are unable to determine which object is better between  $o_4$  and  $o_7$ , unless the price value of  $o_4$  is ascertain. This prompts us to investigate the prediction techniques that can accurately predict the value of continuous range values. It is worth noting that an inaccurate predicted value would impact the computed skylines if they dominate some other objects with better quality. Consider Table 1 that presents three different price values of  $o_4$ , i.e. lowest, average, and highest. If the price value of  $o_4$  is 95 (the lowest value in the range 95 – 130), then  $o_4 < o_5$  since  $95 < 100$  and  $3.5 < 5$  and  $o_4$  is one of skyline results. From the table, it is clear that different values will result in different set of skyline results, even if the values differ in a small fraction. Hence, it is important to ensure that the prediction technique to be employed can most accurately predict or estimate the uncertain values. Furthermore, deriving skyline objects over data stream is computational challenging due to the rapid data arrivals and strict response time constraints. Therefore, the prediction technique to be employed should strive for both quick response times and high accuracy.

Object	Price	Distance
$o_1$	200	0.5
$o_2$	150	2
$o_3$	25	15
$o_4$	95 – 130	3.5
$o_5$	100	5
$o_6$	75	8
$o_7$	115	2
$o_8$	176	4
$o_9$	40 – 80	15
$o_{10}$	186	2

(a)



(b)

FIGURE 3. Results of skyline queries for the hotel database with continuous range values.

TABLE 1. Samples of predicted/estimated values and their effect on the skyline results.

Predicted/Estimated Value	Domination Result	Skyline Result
$o_4(95, 3.5)$	$o_4 < o_5, o_4 \nless o_7$	$S = \{o_1, o_3, o_4, o_6, o_7\}$
$o_4(112.5, 3.5)$	$o_4 \nless o_5, o_4 \nless o_7$	$S = \{o_1, o_3, o_4, o_5, o_6, o_7\}$
$o_4(130.5, 3.5)$	$o_7 < o_4$	$S = \{o_1, o_3, o_5, o_6, o_7\}$

III. BACKGROUND

This section provides an overview of each technique that is considered in this study, namely: *Linear Regression (LR)*, *k-Nearest Neighbour (k-NN)*, *Random Forest (RF)*, *Decision Trees (DT)*, and *Centre and Range Method (CRM)*.

A. LINEAR REGRESSION

Linear Regression (LR) is one of the most basic and widely used machine learning algorithms. It is a mathematical approach to predictive analysis with a wide range of applications, including face recognition, atmospheric applications, medical research, and others. The concept of linear regression that models and measures predicted effects across multiple input variables was initially proposed by Sir Francis Galton in 1894 [9]. It is a technique that establishes linear relationships between dependent and independent variables.

There are two main types of linear regression, namely: Simple Linear Regression (SLR) and Multiple Linear Regression (MLR). A simple linear regression model is a linear function that best represents the relationship between a dependent variable (output) and an independent variable (input). Equation (1) presents the mathematical representation of simple linear regression [10]:

$$y = \beta_0 + \beta_1x + e \tag{1}$$

where  $x$  and  $y$  are the independent and dependent variables, respectively,  $\beta_0$  is the  $y$ -intercept,  $\beta_1$  is the slope of the regression line, and  $e$  is the error term. The equation can be visualized, as shown in Figure 4.

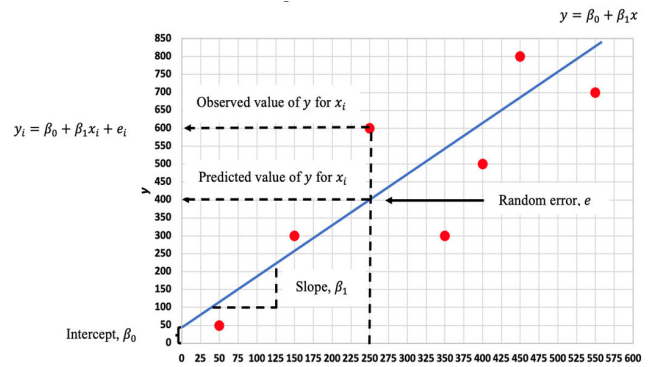


FIGURE 4. Visualization of Equation (1).

Meanwhile, a multiple linear regression establishes the relationship between two or more independent variables and the corresponding dependent variable. The independent variables can be either continuous or categorical. The mathematical representation of multiple linear regression is given by Equation (2) [11]:

$$y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \beta_nx_n + e \tag{2}$$

where  $y$  is the dependent variable,  $x_1, x_2, \dots, x_n$  are the independent variables,  $\beta_0$  is the  $y$ -intercept,  $\beta_1, \beta_2, \dots, \beta_n$  are the slopes of the regression line, and  $e$  is the error term.

To find the best fit in linear regression, the method of least squares is most commonly used. The regression line is formulated by maximizing the portion attributed to the regression while minimizing the residual of all data points. The residuals are the differences between the observed and predicted values of the dependent variable in the linear regression model. Furthermore, the difference between each data point and the mean outcome ( $mo$ ) is calculated by adding the vertical distance from the mean outcome line to the regression line (regression,  $r_1$ ) and the distance from the regression line to the data point (residual,  $r_2$ ). This means that the total distance ( $td$ ) of each point from the mean outcome value can be apportioned between the regression and the residual. A simplified illustration of the  $mo, r_1, r_2$ , and  $td$  are given in Figure 5; while Figure 6 presents the algorithm of the multiple linear regression.



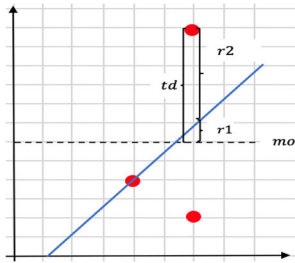


FIGURE 5. The  $mo$ ,  $r1$ ,  $r2$ , and  $td$ .

### B. K-NEAREST NEIGHBOUR

$k$ -Nearest Neighbour ( $k$ -NN) is a simple machine learning algorithm based on supervised learning technique that has been used in a variety of applications including text mining, agriculture, finance, medicine, image recognition, recommender system, etc. It was first developed by Evelyn Fix and Joseph Hodges in 1951, and later expanded by Thomas Cover [13].  $k$ -NN is a non-parametric algorithm, which means it makes no assumptions about the underlying data. It can handle both numerical and categorical data, making it a flexible choice for various types of datasets in classification and regression tasks. In  $k$ -NN classification, an object is assigned to a class by the majority vote (plurality vote) of its  $k$  neighbours, where  $k$  is a positive integer that is usually small. Meanwhile, in  $k$ -NN regression, the property value of an object is determined by averaging the values of its  $k$  nearest neighbours. To find the  $k$  nearest neighbours to a given object, a distance metric is used, such as *Euclidean Distance*, *Manhattan Distance*, and *Minkowski Distance*. Figure 7 illustrates the  $k$ -NN technique while Figure 8 shows its algorithm. In Figure 7, when  $k$  is set to 7, the new object is classified as Class A; while it is classified as Class B when  $k = 19$ . The new object is not classified as Class C for both cases simply because Class C has the minority vote as compared to the other two classes.

### C. RANDOM FOREST

Random Forest (RF) was introduced in 2001 by Leo Breiman [15], which is now used in a variety of applications, including consumer behaviour prediction, breast cancer diagnosis, prediction of road traffic congestion, etc. Random forest is a supervised learning technique that consists of unpruned classification or regression trees built from random data samples. It is one of the most widely used algorithms for both classification and regression tasks due to its simplicity and diversity, as well as its ability to handle binary, continuous, and categorical data. Random forest is an ensemble learning technique that combines multiple classifiers to solve a complex problem and improve the model's performance. Instead of relying on a single decision tree, random forest aggregates the results of each tree and predicts the final output based on the majority of prediction votes. Figure 9 depicts the random forest technique while Figure 10 shows its algorithm.

### D. DECISION TREE

Decision Trees (DT), as described by [17], are among the most prevalent and functional classifiers. Decision tree is a non-parametric supervised learning algorithm, that can also solve data-fitting challenges such as regression and classifications, making them useful for fitting non-linear relationships. It is the most widely used tool for decision making and is used in different areas such as business, intrusion detection, and energy modelling. Decision trees are one of the most effective types of learning algorithms based on various learning techniques. They improve predictive models' accuracy, interpretability, and stability. The goal is to build a model that can predict the value of a target variable based on simple decision rules derived from the data features.

A decision tree is a decision support hierarchical model that creates a tree structure with each internal node, branch, and leaf node representing a test on an attribute, the test outcome, and a class label, respectively. It is constructed by recursively splitting the training data into subsets based on attribute values until a stopping criterion is reached, such as the tree's maximum depth or the smallest number of samples required to split a node. Several metrics have been introduced to determine which attribute to place at the root or at different levels of the tree as internal nodes, including *entropy*, *information gain*, *Gini index*, *gain ratio*, *variance reduction*, and *Chi-Square*. Figure 11 presents the algorithm of constructing a decision tree model.

### E. CENTRE AND RANGE METHOD

Centre and Range Method (CRM) is a simple yet powerful mathematical manipulation that measures the mid-point of an interval valued data [19]; herein called continuous range value. Several studies have adopted this method [19] and its application can be seen in various domains like estimation of internet link delays and radial basis function neural network. The following Equation (3) denotes the calculation of mid-point,  $yc_i$ , and Equation (4) displays the calculation of the range value,  $yr_i$ , of a given point,  $y_i$  [19]:

$$yc_i = \frac{(y_{Li} + y_{Ui})}{2} \quad (3)$$

$$yr_i = \frac{(y_{Ui} - y_{Li})}{2} \quad (4)$$

where  $yc_i$  is the mid-point value,  $y_{Li}$  and  $y_{Ui}$  are the lower bound and upper bound values of  $y_i$ , respectively, and  $yr_i$  is the range of the point,  $y_i$ . The value of  $yr_i$  is used to verify the values of  $y_{Li}$  and  $y_{Ui}$  where  $y_{Li} = yc_i - yr_i$  and  $y_{Ui} = yc_i + yr_i$ .

## IV. EXPERIMENTAL RESULTS and DISCUSSION

This section provides a detailed presentation of the experimental setup and results.

### A. EXPERIMENTAL SETTING

To fairly evaluate the performance of the prediction models constructed by the learning algorithms, namely: *Linear Regression (LR)*, *k-Nearest Neighbour (k-NN)*, *Random*

Input:	Dataset, $D$
Output:	Multiple Linear Regression (MLR) Model
1	Begin
2	Initialize the MLR parameters based on $D$
3	Fit a linear regression model
4	Choose inputs $x_1, x_2, \dots, x_n$ and output $y$
5	Calculate the regression coefficient $\beta_1, \beta_2, \dots, \beta_n$
6	Estimate the model using equation $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + e$
7	Choose the most influential factor that gives the most differs between input and output
8	Return the result of linear equation
9	End

FIGURE 6. The multiple linear regression model algorithm [12].

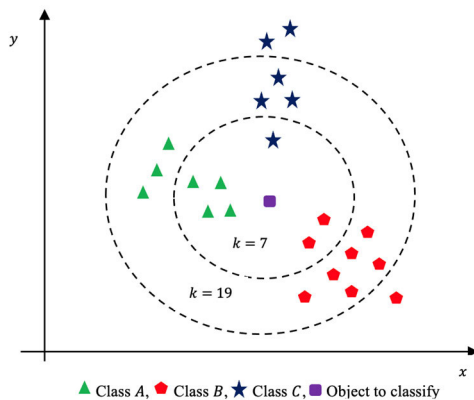


FIGURE 7. Illustration of  $k$ -NN technique.

Forest (RF), Decision Trees (DT), and the estimated model by Centre and Range Method (CRM), in predicting/estimating the continuous range values, several extensive experiments are designed. These experiments were conducted on Intel Core i5 PC with 1.80GHz processor and 8GB memory while the prediction techniques were implemented using Python programming language running on a 64bit Windows 11. Figure 12 presents the phases that are performed in preparing the datasets and constructing the prediction models. These phases include (a) data preparation phase, (b) CRM phase, and (c) machine learning phase. Each phase is further explained in the following paragraphs.

Two types of datasets are used in the experiments, namely: synthetic and real datasets. The parameter settings for these datasets are as shown in Table 2. The synthetic dataset which includes anti-correlated, independent, and correlated is generated in the same manner as in [20], [21], and [22]. The synthetic dataset comprises of  $n$  objects with  $d$  dimensions; where  $d$  ranging from 2 to 15 and  $n$  varying from 100,000 to 5,000,000 to accurately represent the massive amount of high-dimensional data that characterizes a data stream. Each object in the dataset represents a uniform random variable that is generated within the range of 1 to 100. The length of each continuous range value is from 1 – 100. Additionally, the National Basketball Association (NBA) dataset

(www.basketball-reference.com) which is commonly used by previous works like [3] and [22] is also employed in this study to evaluate the performance of the prediction models. NBA consists of 21,961 objects with 16 dimensions that represent a variety of statistical values associated with the NBA players from 1946 to 2009. Since all the values of the NBA dataset are exact values, we have introduced an additional dimension with continuous range values which were generated in the same manner as the synthetic dataset.

In the data preparation phase, three variations of datasets as shown in Figure 12(a) are prepared for both the synthetic and NBA datasets as the following:

- (i) A complete dataset,  $D_c$ , consisting of a collection of objects. For the synthetic dataset, the objects are formed with randomly generated values. These values are exact values while the size of the dataset,  $|n|$ , and the number of dimensions,  $|d|$ , are set according to the required parameter settings of each experiment which is further clarified in the following subsection. Based on the generated dataset,  $D_c$ , the skyline results of  $D_c$ ,  $S_{D_c}$ , are derived using the conventional skyline algorithm [20].  $S_{D_c}$  is the ground truth.
- (ii) For the synthetic dataset, an uncertain dataset,  $D_u$ , is formed based on  $D_c$  by replacing the exact values with continuous range values; while for the NBA dataset,  $D_u$  is formed by adding an additional dimension with continuous range values to  $D_c$ . The amount of uncertainty of  $D_u$  is based on the required parameter settings of each experiment.  $D_u$  is the dataset used by the CRM in generating the estimated data.
- (iii) An incomplete dataset,  $D_i$ , is formed based on  $D_u$  by removing the continuous range values. The LR,  $k$ -NN, RF, and DT, used  $D_i$  to generate the predicted data. This is due to the fact that these techniques cannot directly handle uncertain data in the form of continuous range values [8].

The CRM phase as shown in Figure 12(b) employed the CRM method to estimate the continuous range values of the  $D_u$ . Based on the estimated data produced by CRM method, the conventional skyline algorithm is employed to derive the skyline results,  $S_{CRM}$ . Meanwhile, in the machine learning

Input: Dataset, $D$
Output: $k$ -NN Model
1 Begin
2 Initialize the number $k$ of the neighbours
3 Calculate the distance of $k$ of the neighbours
4 Take the $k$ nearest neighbours
5 Based on the selected $k$ neighbours in Step 4, count the number of objects in each class
6 Assign the new object to the class with the maximum number of neighbours
7 Return the class label of the new object
8 End

FIGURE 8. The  $k$ -NN algorithm [14].

TABLE 2. Experimental parameter settings of the synthetic and NBA datasets.

Parameter	Dataset Type	
	Synthetic	NBA
Number of Objects, $ n $	0.1M, 0.5M, <b>1M</b> , 2M, 3M, 4M, 5M	2K, 4K, 6K, 8K, 10K, 12K, 14K, 16K, <b>21961</b>
Number of Dimensions, $ d $	2, 3, 4, 5, 6, 7, 8, 9, <b>10</b> , 15	6, 10, 15, <b>17</b>
Data Distribution (%)	10, 20, 30, 40, <b>50</b> , 60, 70, 80, 90	

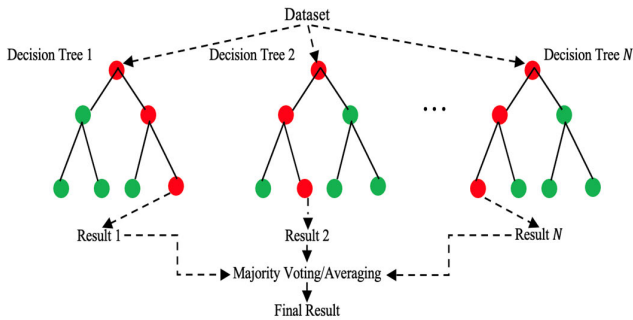


FIGURE 9. Illustration of random forest.

phase as depicted in Figure 12(c), the  $D_i$  is split into two sets, namely: training set and testing set, with distribution of 80% and 20%, respectively. The learning algorithm in the Figure 12(c) represents the LR,  $k$ -NN, RF, and DT. Each technique will construct a prediction model and generate the predicted data; which are then used to derive the skyline results by applying the skyline algorithm. We use the following notations,  $S_{LR}$ ,  $S_{k-NN}$ ,  $S_{RF}$ , and  $S_{DT}$ , to represent the skyline results of LR,  $k$ -NN, RF, and DT, respectively.

To provide a more reliable model evaluation, the 10-fold cross validation is employed. The 10 results from the folds are then averaged to produce the performance measure. The performance measurements used in our experiments are root mean square error (RMSE), execution time, precision (P), recall (R), and F1-score (F) as they are the most commonly used measurements in evaluating the performance of prediction models [21], [23], [24], [25]. These measurements are evaluated on different parameter settings that include the number of objects,  $|n|$ , the number of dimensions,  $|d|$ , and the percentage of uncertainty in the dataset as used in [3], [21], and [22]. RMSE is one of the typically used measures for

evaluating the quality of predictions which can be expressed by the following equation [26], [27]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n \|y(i) - \hat{y}(i)\|^2}{n}} \quad (5)$$

where  $n$  is the number of data points,  $y(i)$  is the  $i$ th measurement, and  $\hat{y}(i)$  is its corresponding prediction. In general, a lower RMSE is better than a higher one. We also measured the precision, recall, and F1-score of each model based on the skyline results they derived. Precision represents the fraction of true skyline results that are derived by the prediction model to the total number of skyline results in the dataset (i.e. the ground truth,  $S_{DC}$ ); while F1-score is the harmonic mean of recall and precision. The formula of precision, recall, and F1-score are given in equations (6), (7), and (8), respectively; with  $TP$ ,  $FP$ , and  $FN$  are the True Positive, False Positive, and False Negative, respectively [23].

$$P = \frac{TP}{TP + FP} \quad (6)$$

$$R = \frac{TP}{TP + FN} \quad (7)$$

$$F = 2 \times \frac{R \times P}{R + P} \quad (8)$$

As an example, the precision of CRM is given by,  $P_{CRM} = \frac{S_{DC} \cap S_{CRM}}{S_{CRM}}$  while recall as,  $R_{CRM} = \frac{S_{DC} \cap S_{CRM}}{S_{DC}}$ .

## B. EXPERIMENTAL RESULTS

We have carried out three primary analyses wherein execution time, RMSE, precision (P), recall (R), and F1-score (F) are used to assess the performance of LR,  $k$ -NN, RF, DT, and CRM with varying number of objects,  $|n|$ , number of

---

Input: Dataset,  $D$   
 Output: Random Forest (RF) Model

---

- 1 Begin
- 2 Select random  $k$  objects from the dataset
- 3 Build the decision trees based on the selected  $k$  objects
- 4 Repeat steps 1 and 2 for the  $N$  number of decision trees
- 5 For new objects, find the predictions of each decision tree, and assign the new objects to the category that wins the majority votes
- 6 End

---

FIGURE 10. The random forest algorithm [16].

---

Input: Dataset,  $D$   
 Output: Decision Tree (DT) Model

---

- 1 Begin
- 2 Let  $S$  be the root node which contains the complete dataset
- 3 Find the best attribute in the dataset using Attribute Selection Measure (ASM)
- 4 Divide the  $S$  into subsets that contains possible values for the best attributes
- 5 Generate the decision tree node, which contains the best attribute
- 6 Recursively make new decision trees using the subsets of the dataset created in Step 4
- 7 Continue this process until the nodes cannot be classified any further, at which point they are referred to as leaf nodes
- 8 End

---

FIGURE 11. The decision tree algorithm [18].

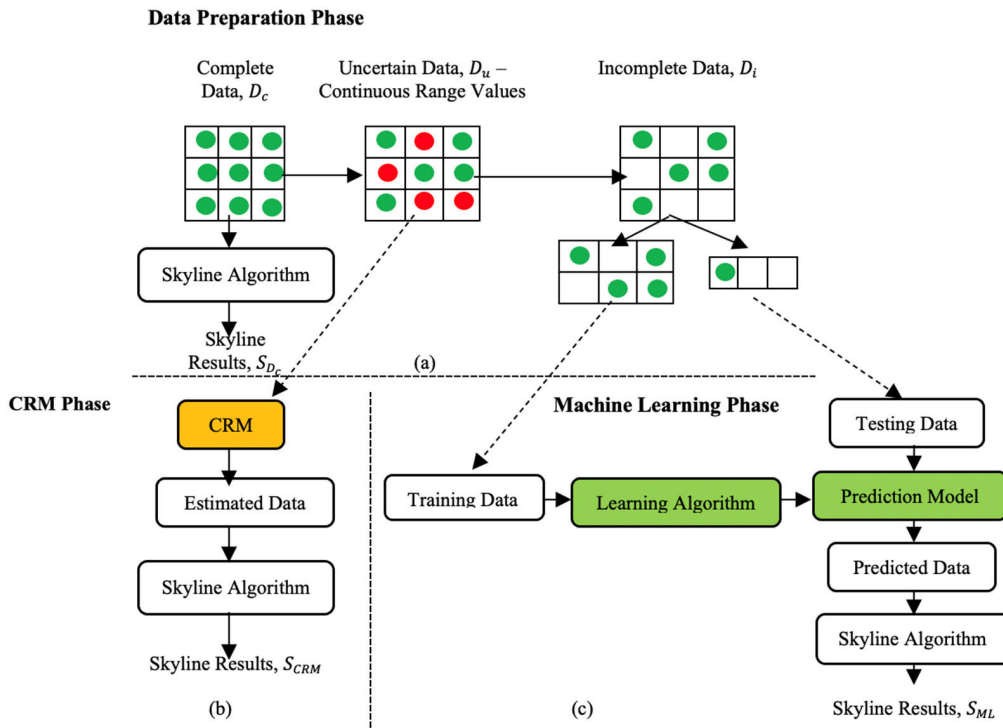


FIGURE 12. Phases of preparing the datasets and constructing the prediction models.

dimensions,  $|d|$ , and uncertainty distribution. The results of the analyses are presented in the following subsections.

### 1) PERFORMANCE WITH VARYING NUMBER OF OBJECTS

In this subsection, the effect of the number of objects,  $|n|$ , in the dataset on the performance of LR,  $k$ -NN, RF, DT, and CRM is investigated. This analysis evaluates the scalability of

the techniques being examined. The parameter settings of the synthetic dataset which include anti-correlated, correlated, and independent are as follows: the number of objects,  $|n|$ , is varied from 0.1M to 5M, the number of dimensions,  $|d|$ , is fixed to 10, while the uncertainty distribution is set to 50%. Meanwhile, the number of objects,  $|n|$ , for the NBA dataset is varied from 2K to its initial size, i.e. 21,961 with



17 number of dimensions, and 50% of uncertainty distribution. Figures 13 (a) – (e) present the performance of LR,  $k$ -NN, RF, DT, and CRM with regard to (i) execution time, (ii) RMSE, (iii) precision (P), (iv) recall (R), and (v) F1-score (F), respectively.

Figures 13 (a1), (a2), (a3), and (a4) clearly demonstrate that the techniques under investigation observed an increase in execution time as the number of objects,  $|n|$ , increased across all datasets. Nonetheless, in most cases, the execution time of both the CRM and LR shows a small increase, with both having nearly equal execution time. Meanwhile, the  $k$ -NN and RF show the highest execution time for the synthetic dataset (figures 13 (a1) – (a3)) and the NBA dataset (Figure 13(a4)), respectively.

Figures 13 (b1), (b2), (b3), and (b4) present the performance of LR,  $k$ -NN, RF, DT, and CRM with regard to RMSE. CRM shows the least RMSE values for all datasets, i.e. synthetic and NBA datasets. Meanwhile, for the independent dataset (Figure 13(b3)) and NBA dataset (Figure 13(b4)), DT shows the highest RMSE values followed by  $k$ -NN, RF, and LR. Nonetheless, for the anti-correlated dataset (Figure 13(b1)), the LR exhibits a sudden increase in RMSE at 0.5M and remains stable at 2M – 5M of objects; whereas the DT has the highest RMSE values for the same dataset.

Figures 13 (c), (d), and (e) present the performance of LR,  $k$ -NN, RF, DT, and CRM with regard to precision (P), recall (R), and F1-score (F), respectively. Here, the skyline results produced based on each prediction/estimated model, i.e.  $S_{LR}$ ,  $S_{k-NN}$ ,  $S_{RF}$ ,  $S_{DT}$ , and  $S_{CRM}$ , are compared to the actual skyline results,  $S_{DC}$ , as depicted in Figure 12 and precision, recall, and F1-score are calculated based on equations (6), (7), and (8), respectively. CRM achieved the highest precision values of more than 90% for both datasets, namely: the NBA dataset and the synthetic dataset which includes anti-correlated, correlated, and independent. This implies that the skyline objects retrieved by CRM,  $S_{CRM}$ , are mostly relevant. Meanwhile, LR achieved comparable precision to CRM for the anti-correlated (Figure 13(c1)) and correlated (Figure 13(c2)) datasets with 0.5M and 4M objects, respectively. In most cases, LR produces precision values greater than 80% for both datasets. Furthermore, the DT has the lowest precision values among the techniques being examined across all datasets.

On the other hand, the fraction of relevant skyline objects retrieved by CRM, as shown by the recall results (figures 13 (d1), (d2), (d3), and (d4)), is greater than 90% for both datasets, namely: synthetic and NBA. Meanwhile, DT has the lowest recall values for the anti-correlated dataset (Figure 13(d1)) and NBA dataset (Figure 13(d4)), whereas LR and  $k$ -NN have the lowest recall values for the correlated dataset (Figure 13(d2)) and independent dataset (Figure 13(d3)), respectively. As a result, CRM outperformed the other techniques in terms of recall.

The results of F1-score, which is a harmonic mean of precision and recall, show that CRM outperformed the other techniques, which are: LR,  $k$ -NN, RF, and DT.

Furthermore, LR and  $k$ -NN have the lowest F1-scores for the correlated dataset (Figure 13(e2)) and independent dataset (Figure 13(e3)), respectively; while DT has the lowest F1-score for the anti-correlated dataset (Figure 13(e1)) and NBA dataset (Figure 13(e4)).

Overall, with varying number of objects,  $|n|$ , in the dataset, the CRM method outperformed other prediction techniques in terms of execution time, RMSE, precision, recall, and F1-score.

## 2) PERFORMANCE WITH VARYING NUMBER OF DIMENSIONS

We also study the effect of the number of dimensions,  $|d|$ , on the performance of LR,  $k$ -NN, RF, DT, and CRM. The parameter settings of the synthetic dataset which include anti-correlated, correlated, and independent are as follows: the number of dimensions,  $|d|$ , is varied from 2 to 15, the number of objects,  $|n|$ , is fixed to 1M, while the uncertainty distribution is set to 50%. Meanwhile, the number of dimensions,  $|d|$ , for the NBA dataset is varied from 6 to 17 with 21,961 number of objects, and 50% of uncertainty distribution. Figures 14 (a) – (e) present the performance of LR,  $k$ -NN, RF, DT, and CRM with regard to (i) execution time, (ii) RMSE, (iii) precision (P), (iv) recall (R), and (v) F1-score (F), respectively.

Figures 14 (a1), (a2), (a3), and (a4) demonstrate that the execution time of the techniques under investigation increases across all datasets as the number of dimensions,  $|d|$ , increases. But in the majority of cases, the execution time for the CRM and LR both show a slight increase, with nearly equal execution time. For the NBA dataset (Figure 14(a4)) and the synthetic dataset (figures 14(a1) – a(3)), respectively, the  $k$ -NN and RF exhibit the highest execution time.

The results of LR,  $k$ -NN, RF, DT, and CRM in relation to RMSE are shown in figures 14 (b1), (b2), (b3), and (b4). CRM shows the least RMSE values for all datasets, i.e. synthetic and NBA datasets. In contrast, DT displays the highest RMSE values for the anti-correlated dataset (Figure 14(b1)), independent dataset (Figure 14(b3)), and NBA dataset (Figure 14(b4)); followed by  $k$ -NN and RF. However, the LR shows the highest RMSE values for the correlated dataset (Figure 14(b2)).

Figures 14 (c), (d), and (e) present the performance of LR,  $k$ -NN, RF, DT, and CRM with regard to precision (P), recall (R), and F1-score (F), respectively. Here, equations (6), (7), and (8) are used to calculate the precision, recall, and F1-score, respectively. These involve comparing the actual skyline results,  $S_{DC}$ , to the skyline results produced based on each prediction/estimated model, i.e.  $S_{LR}$ ,  $S_{k-NN}$ ,  $S_{RF}$ ,  $S_{DT}$ , and  $S_{CRM}$ . CRM exhibits the highest precision values for the anti-correlated dataset (Figure 14(c1)), starting at 60% for 2 dimensions and rising to above 90% as the number of dimensions increases. Comparable patterns are seen for the other approaches, which produce lower precision values of 30% at 2 dimensions and rise to above 80% as the number of dimensions increases. Nevertheless, CRM obtains the highest precision values for the synthetic datasets (anti-correlated

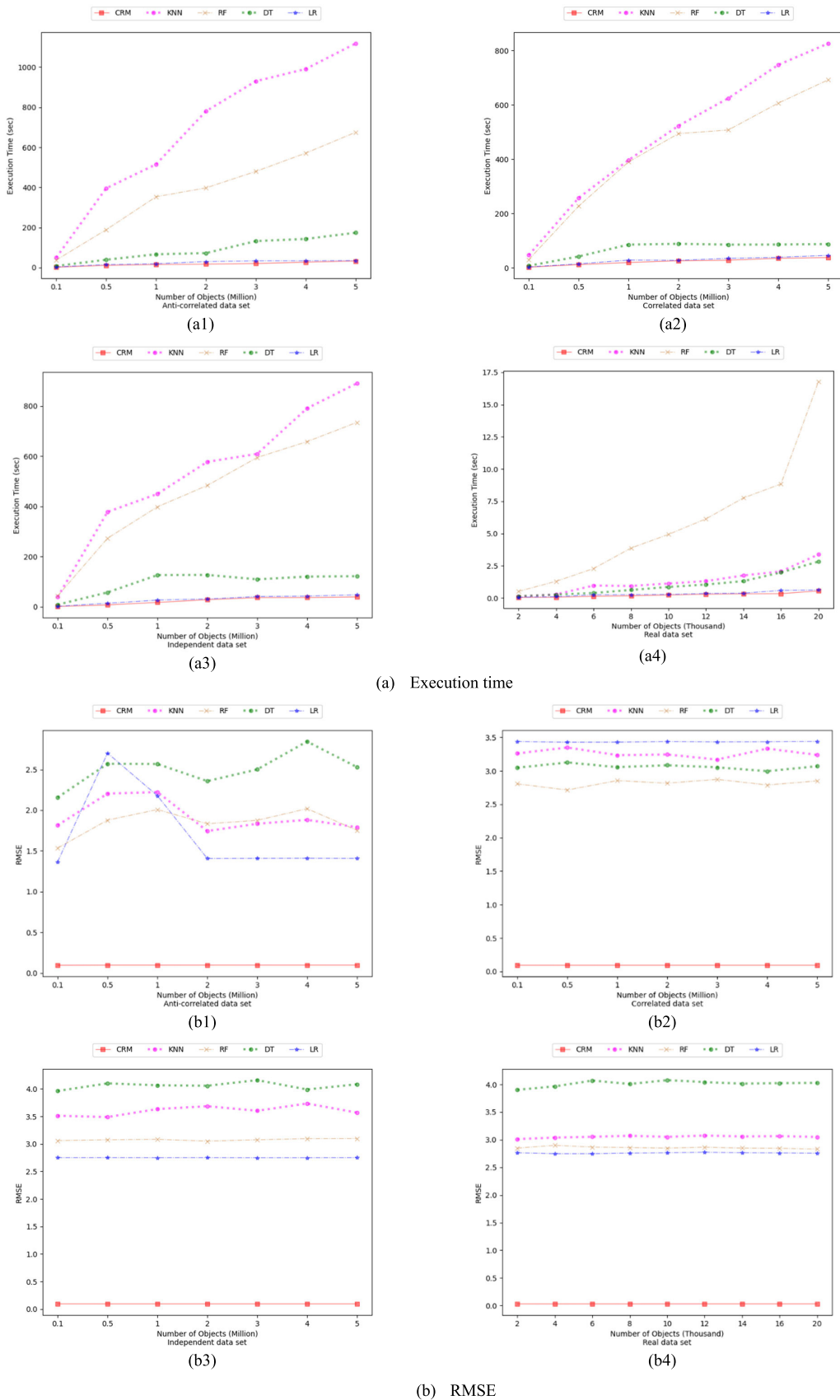


FIGURE 13. The performance of LR, k-NN, RF, DT, and CRM with varying number of objects,  $|n|$ .

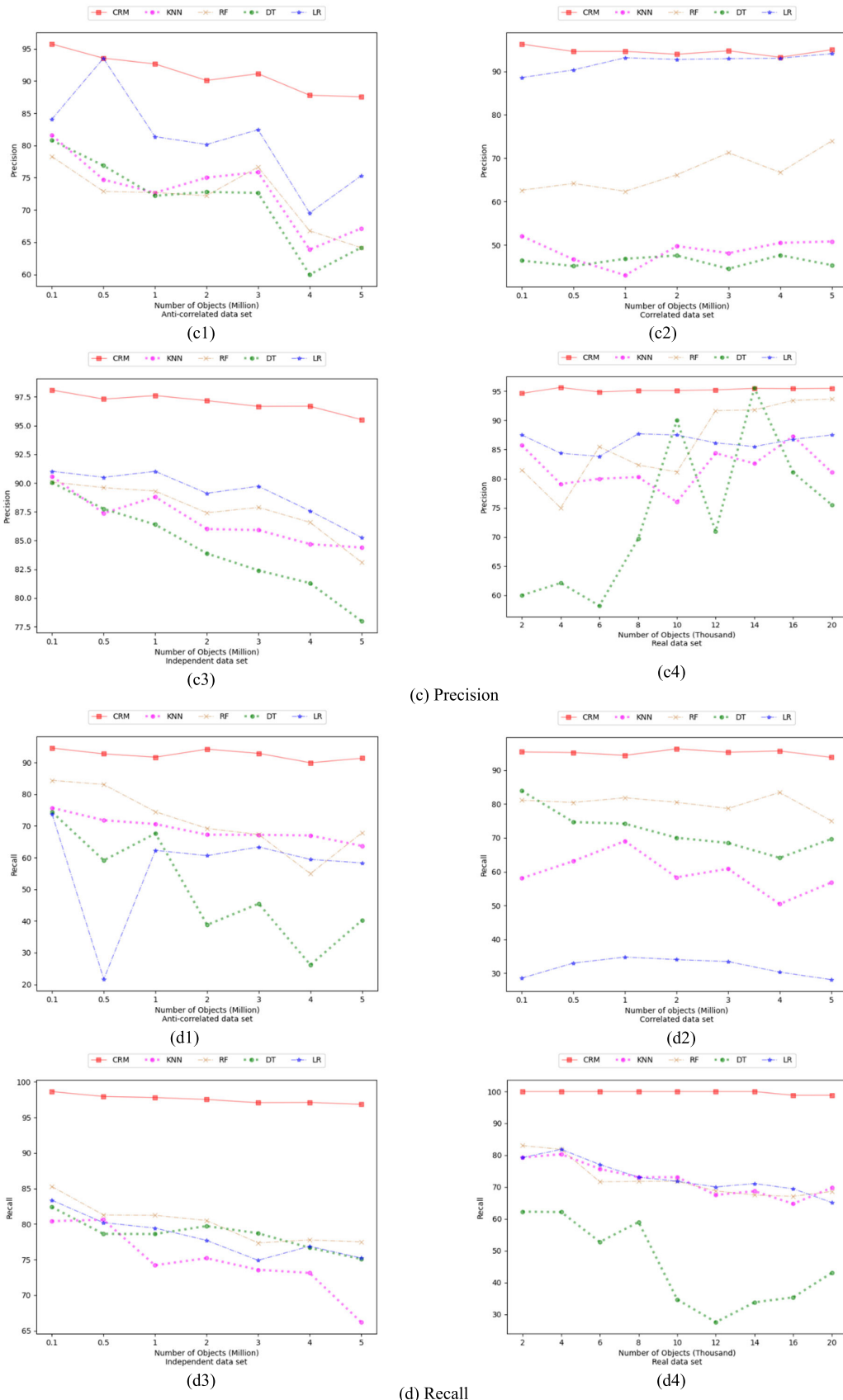


FIGURE 13. (Continued.) The performance of LR, k-NN, RF, DT, and CRM with varying number of objects,  $|n|$ .

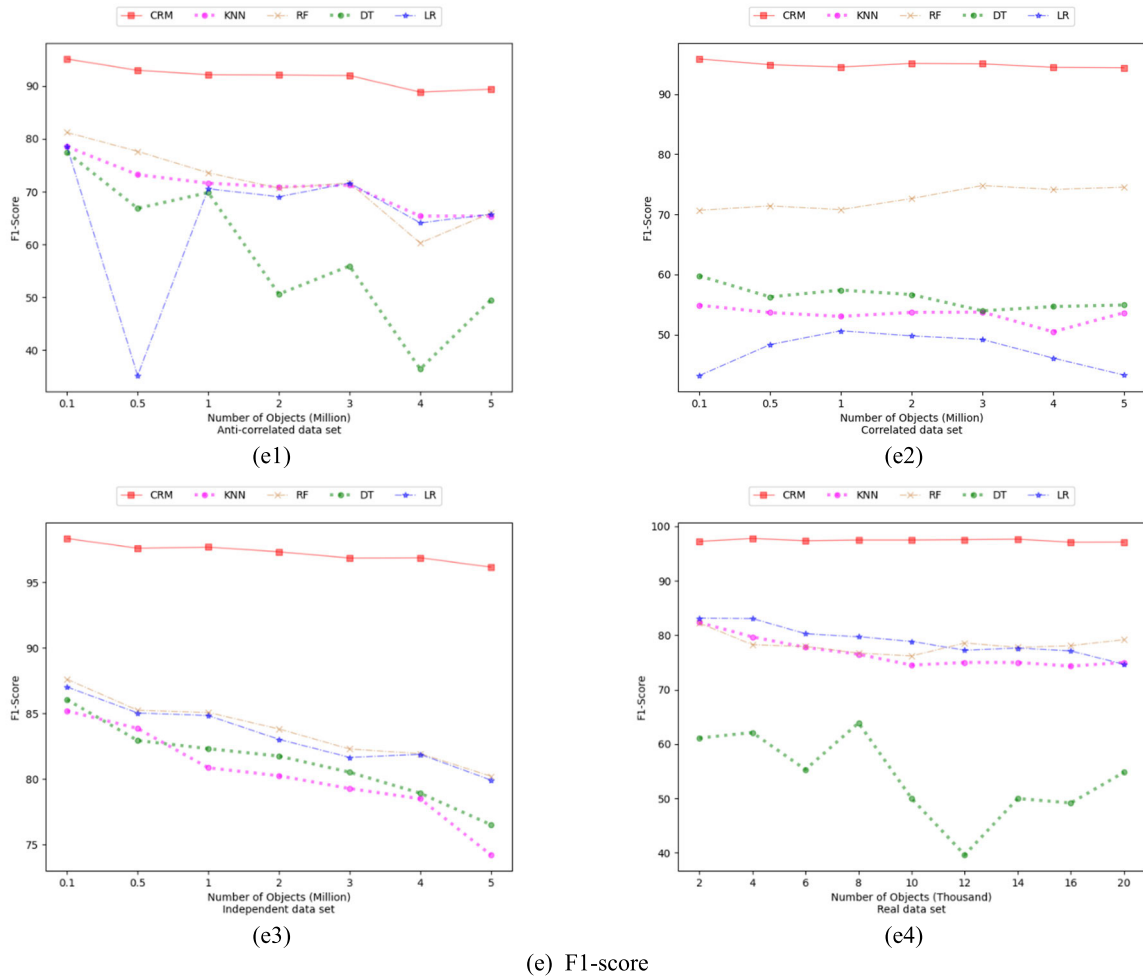


FIGURE 13. (Continued.) The performance of LR, k-NN, RF, DT, and CRM with varying number of objects,  $|n|$ .

(Figure 14(c1)), correlated (Figure 14(c2)), and independent (Figure 14(c3)), followed by LR,  $k$ -NN, RF, and DT. This suggests that the majority of the skyline objects retrieved by CRM,  $S_{CRM}$ , are relevant. The NBA dataset (Figure 14(c4)) shows a similar pattern, although the precision values of all techniques decreased at dimension 10 and increased at dimension 15 and almost stable at dimension 17.

Meanwhile, the recall results (figures 14 (d1), (d2), (d3), and (d4)) demonstrate that, for both datasets—the synthetic and NBA—the fraction of relevant skyline objects retrieved by CRM ranges from 78% to 95%, which represents the highest values obtained across the techniques under consideration. For all datasets, DT has the lowest recall values; while, for the anti-correlated (Figure 14(d1)), correlated (Figure 14(d2)), and independent (Figure 14(d3)) datasets, RF performed better in terms of recall than LR,  $k$ -NN, and DT; but lower recall values than  $k$ -NN at dimensions 15 and 17 of the NBA dataset (Figure 14(d4)).

The results of the F1-score, a harmonic mean of recall and precision, indicate that CRM performed better than the other techniques, which include LR,  $k$ -NN, RF, and DT. Moreover,

DT has the lowest F1-scores across the synthetic and NBA datasets. Meanwhile, LR has better F1-score values compared to  $k$ -NN, RF, and DT with the following exceptions: anti-correlated dataset (Figure 14(e1)) with 10 and 15 dimensions, independent dataset (Figure 14(e3)) with 15 dimensions, and NBA dataset (Figure 14(e4)) with 15 and 17 dimensions.

Consequently, the CRM method performed better than other prediction techniques in terms of execution time, RMSE, precision, recall, and F1-score, with different number of dimensions,  $|d|$ .

### 3) PERFORMANCE WITH VARYING UNCERTAINTY DISTRIBUTION

It is important to study the effect of uncertainty distribution in a dataset on the performance of LR,  $k$ -NN, RF, DT, and CRM. The parameter settings of the synthetic dataset which include anti-correlated, correlated, and independent are as follows: the uncertainty distribution is varied between 10% and 90%, the number of objects,  $|n|$ , is fixed at 1M, and the number of dimensions,  $|d|$ , is set to 10. In the meantime, the uncertainty distribution for the NBA dataset is varied from

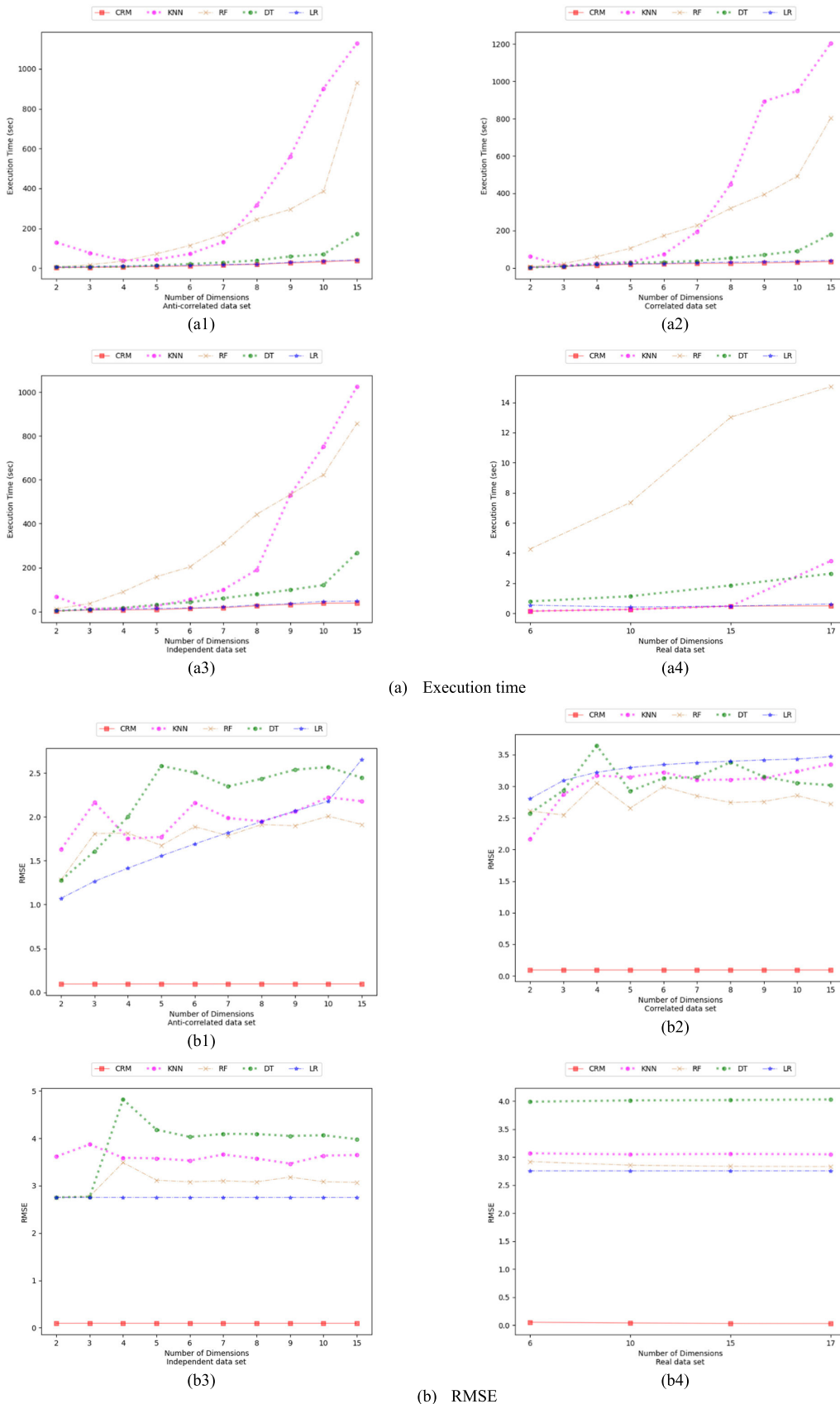


FIGURE 14. The performance of LR, k-NN, RF, DT, and CRM with varying number of dimensions,  $|d|$ .



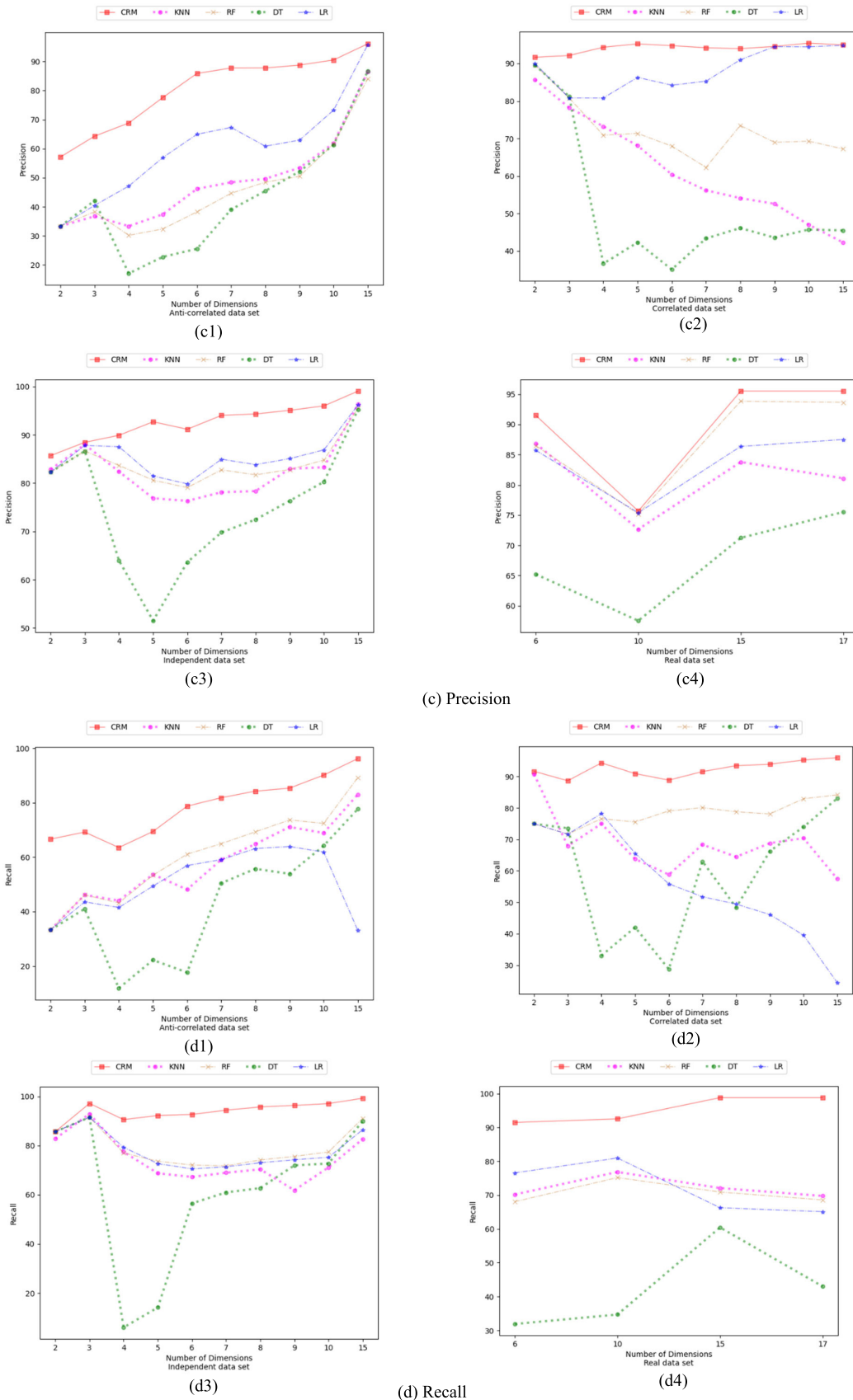


FIGURE 14. (Continued.) The performance of LR, k-NN, RF, DT, and CRM with varying number of dimensions,  $|d|$ .

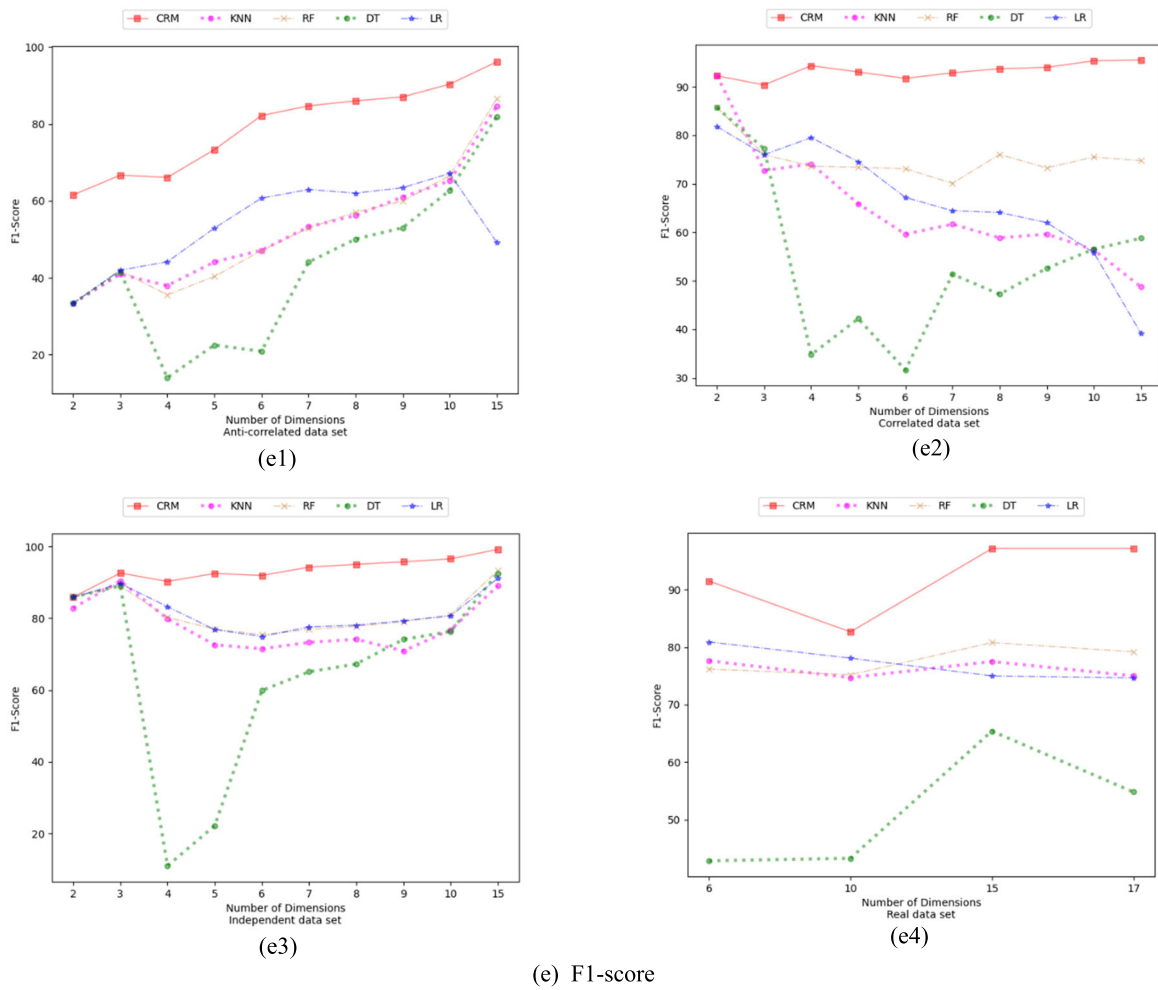


FIGURE 14. (Continued.) The performance of LR, k-NN, RF, DT, and CRM with varying number of dimensions,  $|d|$ .

10% to 90% with 21,961 number of objects, and 17 number of dimensions. Figures 15 (a) – (e) present the performance of LR,  $k$ -NN, RF, DT, and CRM with regard to (i) execution time, (ii) RMSE, (iii) precision (P), (iv) recall (R), and (v) F1-score (F), respectively.

Figures 15 (a1), (a2), (a3), and (a4) show that the CRM and LR techniques have the lowest execution time, with nearly equal execution time and a slight increase as the uncertainty distribution is increased. This is then followed by DT, RF, and  $k$ -NN for all datasets (figures 15 (a1) – (a3)) except for the NBA dataset (Figure 15(a4)) in which RF is worse than  $k$ -NN.

Meanwhile, figures 15 (b1), (b2), (b3), and (b4) display the results of LR,  $k$ -NN, RF, DT, and CRM in respect to RMSE. CRM exhibits the least RMSE values for all datasets, i.e. synthetic and NBA datasets. Figure 15(b3) of the independent dataset and Figure 15(b4) of the NBA dataset, on the other hand, both display a similar trend, with LR outperforming RF, followed by  $k$ -NN and DT. In contrast, RF outperformed DT,  $k$ -NN, and LR for the correlated dataset (Figure 15(b2))

and anti-correlated dataset (Figure 15(b1)) except when the uncertainty distribution is between 10% and 30% in which  $k$ -NN is better than RF. Nonetheless, DT performs the worst in terms of RMSE for all datasets, with the exception of the correlated dataset (Figure 15(b2)), for which LR exhibits the highest RMSE values.

Figures 15 (c), (d), and (e) present the performance of LR,  $k$ -NN, RF, DT, and CRM with regard to precision (P), recall (R), and F1-score (F), respectively. Similar to the previous subsections, the equations (6), (7), and (8) are used to calculate the precision, recall, and F1-score, respectively; while the actual skyline results,  $S_{DC}$ , is compared to the skyline results produced based on each prediction/estimated model, i.e.  $S_{LR}$ ,  $S_{k-NN}$ ,  $S_{RF}$ ,  $S_{DT}$ , and  $S_{CRM}$ . All techniques show a decline in precision as the uncertainty distribution increases. Nevertheless, CRM exhibits the highest precision values among the techniques under consideration for all datasets, i.e. synthetic and NBA; with precision values between 71% – 99%. Nonetheless, for the correlated dataset, LR and CRM have nearly identical precision values

(Figure 15(c2)). However, RF and DT, which perform about equally, have the lowest precision for the anti-correlated dataset (Figure 15(c1)), followed by  $k$ -NN and LR. For the correlated (Figure 15(c2)), independent (Figure 15(c3)), and NBA (Figure 15(c4)) datasets, DT has the lowest precision values, followed by  $k$ -NN, RF, and LR.

On the other hand, the recall results (figures 15 (d1), (d2), (d3), and (d4)) show that, for both datasets—the synthetic and NBA—the fraction of relevant skyline objects retrieved by CRM ranges between 83% and 100%. This is the highest value found across all the techniques that were examined. For both the anti-correlated dataset (Figure 15(d1)) and the NBA dataset (Figure 15(d4)), DT has the lowest recall values; in contrast, LR and  $k$ -NN perform the worst in terms of recall for the correlated (Figure 15(d2)) and independent (Figure 15(d3)) datasets, respectively. However, RF outperformed LR,  $k$ -NN, and DT in terms of recall for both the synthetic and NBA datasets.

The results of the F1-score, a harmonic mean of recall and precision, indicate that as the rate of uncertainty distribution increases, the F1-score for all techniques decreased. Nonetheless, CRM outperformed the other techniques, which include LR,  $k$ -NN, RF, and DT with F1-score values between 77% – 99%. Additionally, DT has the lowest F1-score for both the anti-correlated (Figure 15(e1)) and NBA (Figure 15(e4)) datasets; while  $k$ -NN has the lowest F1-score for the independent dataset (Figure 15(e3)). On the other hand,  $k$ -NN, RF, and LR show comparable F1-score values for the anti-correlated and NBA datasets.

As a result, the CRM method outperformed the other prediction techniques in terms of execution time, RMSE, precision, recall, and F1-score, with varying rate of uncertainty distribution.

#### 4) DISCUSSION

In this subsection, the results of the three analyses conducted with various numbers of objects,  $|n|$ , dimensions,  $|d|$ , and uncertainty distribution as reported in the above subsections are further elaborated.

##### *a: EXECUTION TIME*

Based on the three analyses that have been conducted, the results indicate that there is a slight increase in the execution time of both the CRM and LR, with both having almost the same execution time. While,  $k$ -NN and RF exhibit the highest execution time in most cases. This is primarily because the CRM employs a simple mid-point calculation (see Equation (3) and Equation (4)) whereas the LR is a statistical model which uses the relationships established between dimensions (see Equation (1) and Equation (2)) to estimate/predict the continuous range values. Meanwhile, the  $k$ -NN uses a distance metric to determine the  $k$  nearest neighbours of a given object. The values of these neighbours are then aggregated to determine the object's predicted value. In contrast, the RF aggregates the results of each decision tree and predicts the continuous range values based on the

majority of prediction votes. Evidently, the performance of the technique being examined with regards to execution time is impacted by its complexity.

##### *b: RMSE*

The results of RMSE for the analyses with different numbers of objects,  $|n|$ , dimensions,  $|d|$ , and uncertainty distribution exhibit that the CRM achieved the lowest RMSE values of all the techniques being studied. While in most cases DT shows the highest RMSE values followed by  $k$ -NN, RF, and LR. The LR performs well for all datasets except when there is an increase in the numbers of objects,  $|n|$ , and dimensions,  $|d|$ , in the correlated datasets (Figure 13(b2) and Figure 14(b2), respectively) and uncertainty distribution in anti-correlated and correlated datasets (Figure 15(b1) and Figure 15(b2), respectively). A possible reason for this is that LR assumes a linear relationship between the predictors and the target variable, while anti-correlated and correlated datasets with high dimensionality and uncertainty make it difficult to establish an optimal regression model. Meanwhile, RF demonstrates a moderate RMSE performance among the techniques being examined for all datasets except for correlated dataset in the analyses conducted with varying numbers of objects (Figure 13(b2)), dimensions (Figure 14(b2)), and uncertainty distribution (Figure 15(b2)); in which it gains a better RMSE performance. This is because the correlated dataset provides more consistent patterns, allowing the multiple decision trees within the RF to make more accurate predictions leading to a lower RMSE compared to other datasets. On the other hand,  $k$ -NN shows a low RMSE performance for all datasets in the three analyses being conducted. This is mainly due to the choice of an optimal value of  $k$ , the distance metric employed as well as the curse of dimensionality. With high dimensionality, the distances between the data points become insignificant making it challenging to identify the true nearest neighbours. Nonetheless, DT exhibits the highest RMSE values among the techniques being examined for all datasets except for correlated dataset (Figure 13(b2), Figure 14(b2), and (Figure 15(b2)) that shows a slight decrease in RMSE for the three analyses conducted. This is because the inherent structure and predictable patterns within the correlated dataset align more closely with the decision tree's splitting criteria. This alignment reduces overfitting and allows the DT to better capture the underlying relationships between input data leading to a slightly decreased RMSE values.

##### *c: PRECISION (P)*

The results of the analyses conducted with various numbers of objects,  $|n|$ , dimensions,  $|d|$ , and uncertainty distribution show that CRM achieved the highest precision values with values between 60% – 99% among the techniques under consideration for all datasets; while DT shows the lowest performance. This implies that, in comparison to the skyline objects derived using the predicted data of the LR,  $k$ -NN, RF, and DT techniques, the skyline objects retrieved based on

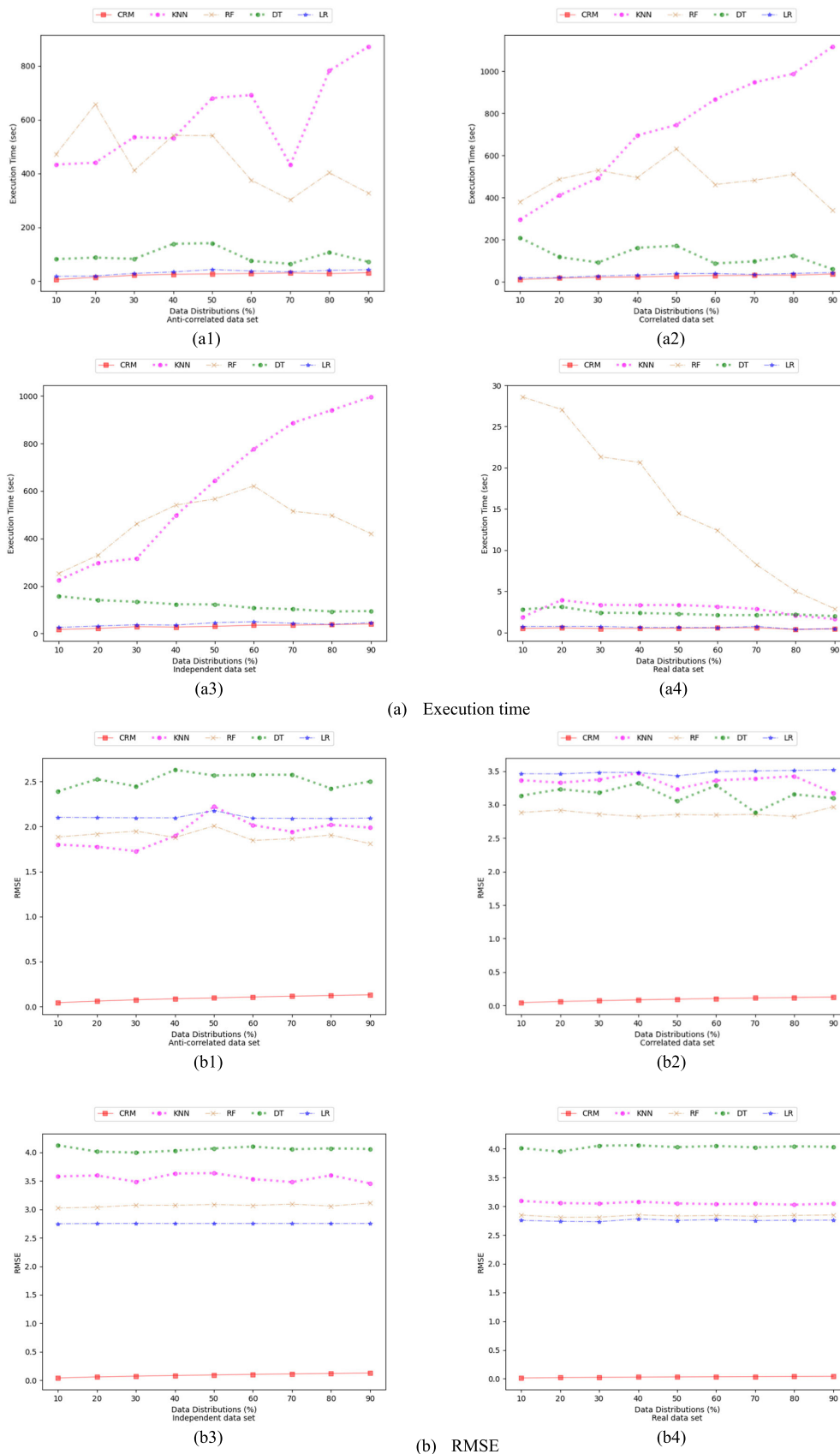
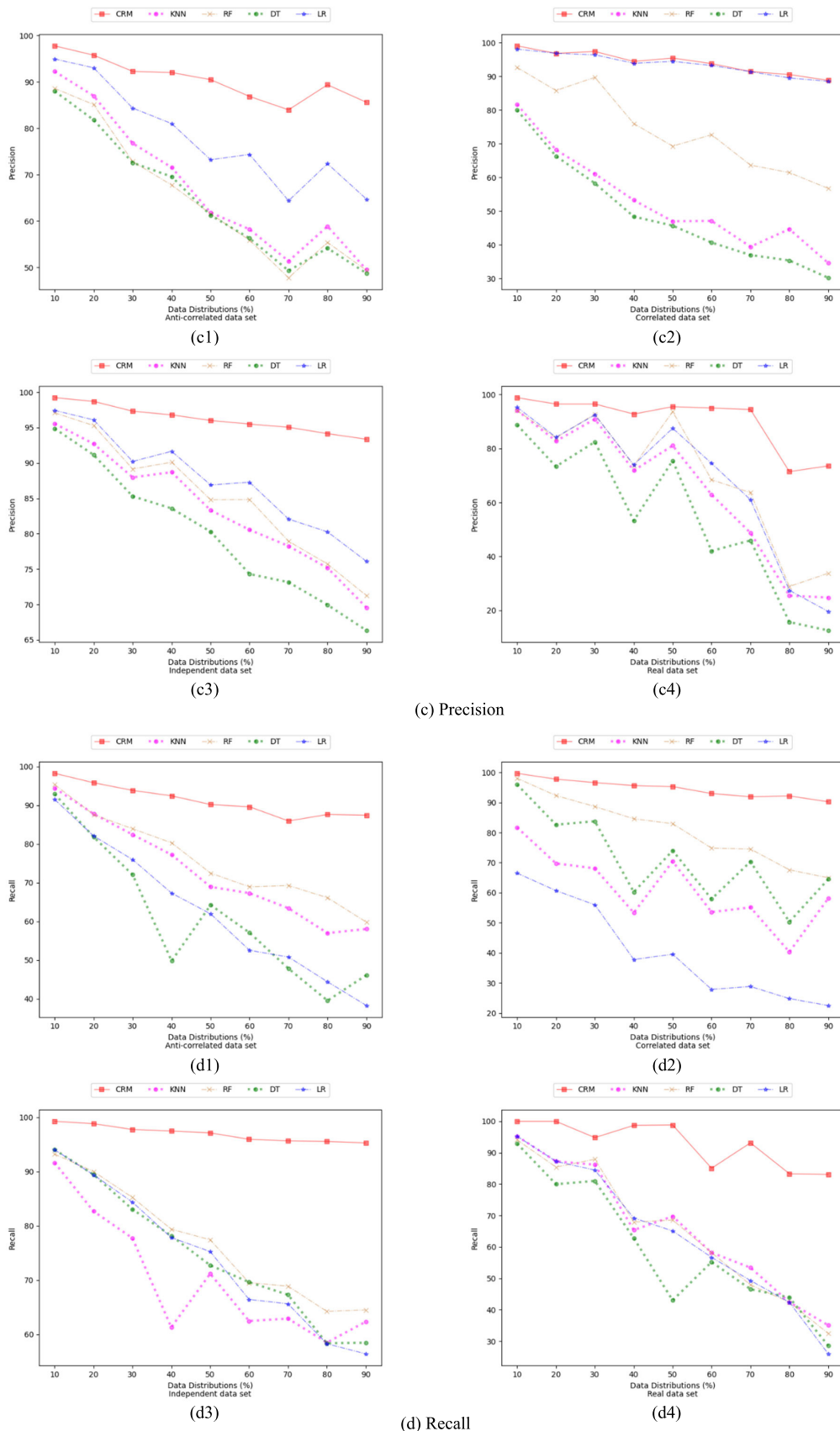


FIGURE 15. The performance of LR, k-NN, RF, DT, and CRM with varying uncertainty distribution (%).



(c) Precision

(d) Recall

FIGURE 15. (Continued.) The performance of LR, k-NN, RF, DT, and CRM with varying uncertainty distribution (%).



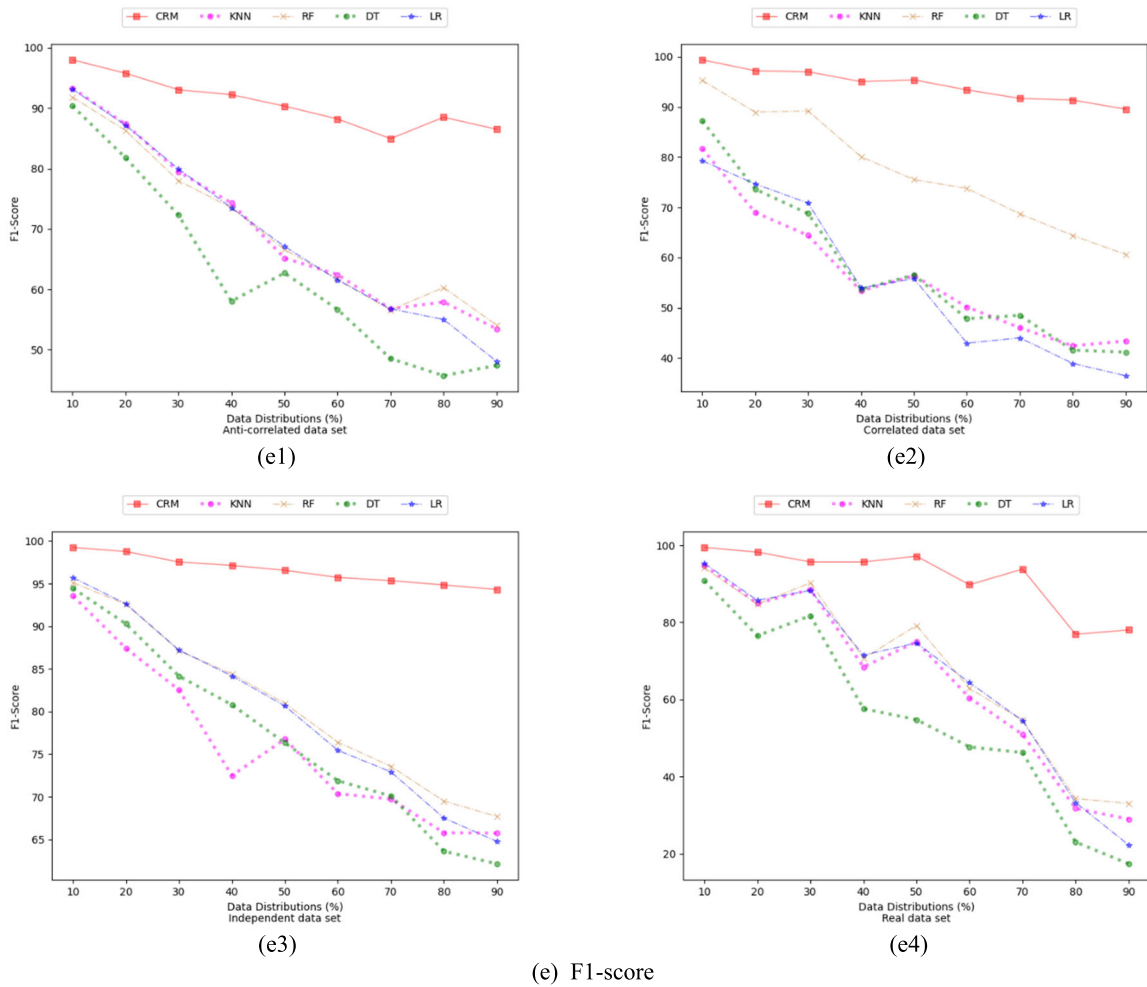


FIGURE 15. (Continued.) The performance of LR, k-NN, RF, DT, and CRM with varying uncertainty distribution (%).

the estimated data of CRM,  $S_{CRM}$ , are more relevant. This is mainly due to the estimated values that CRM generates using the mid-point calculation are always within the range of the given continuous values. If the range of the continuous values is smaller, the estimated values have a higher probability of agreeing with the actual values with a very small percentage of difference. As the LR,  $k$ -NN, RF, and DT rely on the underlying data distribution to construct their prediction models, it is not always possible to ensure that the predicted values will lie within the continuous range values, which undoubtedly affects the skyline results and consequently the precision values.

Nonetheless, all techniques exhibit a slight decrease in precision values as the number of objects and uncertainty distribution increase. While the number of objects increases, the number of uncertain values in the dataset will also increase, to ensure the uncertainty distribution remains fixed at 50%. Apparently, the increase in the number of uncertain values/distribution affects the precision performance of the models owing to more values need to be estimated/predicted. Thus, each technique generates skyline objects based on a

greater number of estimated/predicted values as the number of objects and uncertainty distribution increase. Interestingly, with regard to these analyses, the LR achieved comparable precision to CRM for the anti-correlated (Figure 13(c1)) and correlated (Figure 13(c2)) datasets with 0.5M and 4M objects, respectively, and identical precision values for correlated dataset (Figure 15(c2)). A possible reason for this is the learning ability of LR is improved due to the characteristics of the underlying data distribution.

In contrast, as the number of dimensions increases, all techniques demonstrate an increase in the precision values. As in the previous analyses, an increase in the number of dimensions will result in an increase in the number of uncertain values in the dataset to ensure the 50% uncertainty distribution is maintained. However, an object with higher number of dimensions will have more dimensions with certain values as compared to an object with lower number of dimensions; which leads to higher precision values. Without loss of generality, consider the objects  $o_i(5, *)$ ,  $o_j(5, 9, *, *)$ , and  $o_k(5, 9, 4, *, *, *)$  where  $*$  denotes the uncertain values. These objects have 50% of uncertainty where the objects  $o_i$ ,  $o_j$ , and  $o_k$  have

TABLE 3. The performance results of CRM.

Analyses	Performance Metric	Synthetic			NBA
		Anti-correlated	Correlated	Independent	
Varying number of objects, $ n $	Execution time	17.975	23.250	24.201	0.241
	RMSE	0.096	0.094	0.095	0.031
	Precision	91.217	94.631	97.010	95.235
	Recall	92.462	95.140	97.553	99.740
	F1-score	91.802	94.880	97.280	97.428
Varying number of dimension, $ d $	Execution time	16.750	20.767	18.720	0.350
	RMSE	0.096	0.094	0.095	0.040
	Precision	80.441	94.146	92.654	89.542
	Recall	78.562	92.482	94.152	95.431
	F1-score	79.390	93.329	93.397	92.108
Varying uncertainty distribution (%)	Execution time	23.910	25.214	30.203	0.498
	RMSE	0.094	0.092	0.093	0.030
	Precision	90.467	94.212	96.230	90.504
	Recall	91.229	94.671	96.994	93.006
	F1-score	90.842	94.428	96.610	91.623

one, two, and three certain/uncertain values, respectively. The result of the domination analysis is more accurate when more certain values of an object are available. Nonetheless, for the NBA dataset (Figure 14(c4)) the precision values of all techniques decreased at dimension 10 and increased at dimension 15 and almost stable at dimension 17. This is because the NBA dataset has smaller number of objects as compared to the synthetic dataset. Hence, increasing the number of dimensions results in an increase in the number of certain/uncertain values that enhances the learning ability of the model from the underlying data distribution, leading to more robust predictions.

#### d: RECALL (R)

Based on the analyses with different numbers of objects,  $|n|$ , dimensions,  $|d|$ , and uncertainty distribution, the results indicate that, out of all the techniques considered, CRM achieved the highest recall values, ranging from 78% to 100%, for all datasets; while DT performed the lowest. This suggests that the skyline objects retrieved based on the estimated data of CRM,  $S_{CRM}$ , are largely relevant (most of the true positives are identified) compared to the skyline objects derived using the predicted data of the LR,  $k$ -NN, RF, and DT techniques. The same reasons as mentioned in the precision analysis applied here.

Moreover, all techniques exhibit a slight decrease in recall values as the number of objects and uncertainty distribution increase. To ensure the uncertainty distribution remains fixed at 50% while the number of objects increases, the number of uncertain values in the dataset is also increased. Consequently, increasing the number of uncertain values/distribution affects the recall performance of the techniques due to more values need to be estimated/predicted. Thus, the skyline objects derived by each technique are based on a greater number of estimated/predicted values as the number of objects and uncertainty distribution increase. However, the performance of the techniques being examined is not consistent across the datasets. For instance, DT has the lowest recall values for the anti-correlated dataset (figures 13(d1)

and 15(d1)) and NBA dataset (figures 13(d4) and 15(d4)), whereas LR and  $k$ -NN have the lowest recall values for the correlated dataset (figures 13(d2) and 15(d2)) and independent dataset (figures 13(d3) and 15(d3)), respectively. One explanation for this could be the inherent limitations of each technique in handling specific characteristics of data. DT tends to overfit on the anti-correlated and NBA datasets due to their complex patterns leading to missed general trends and lower recall. Meanwhile, LR struggles with high correlation relationships present in the correlated dataset and  $k$ -NN is affected by the curse of high dimensionality in the independent dataset which result in poor recall for both techniques.

On the other hand, as the number of dimensions increases, all techniques demonstrate an increase in the recall values. The same reasons as explained for the precision analysis applied here. Although, DT has the lowest recall values for all datasets; for the anti-correlated (Figure 14(d1)), correlated (Figure 14(d2)), and independent (Figure 14(d3)) datasets, RF performed better than LR,  $k$ -NN, and DT; but lower than  $k$ -NN at dimensions 15 and 17 of the NBA dataset (Figure 14(d4)). This is because RF as an ensemble method effectively captures complex patterns and reduces overfitting leading to better performance in most datasets. However, unlike the synthetic dataset, the NBA dataset has smaller number of objects and the RF is unable to learn well from the underlying data distributions compared to  $k$ -NN even at higher dimensions.

#### e: F1-SCORE

The results of F1-score for the analyses with different numbers of objects,  $|n|$ , dimensions,  $|d|$ , and uncertainty distribution, are influenced by the results of precision and recall as reported above. This is because F1-score is a harmonic mean of recall and precision.

## V. CONCLUSION

The findings of our three main analyses as discussed in Section IV, demonstrate that the CRM method outperformed the other prediction techniques that are LR,  $k$ -NN, RF, and

DT; in terms of execution time, RMSE, precision (P), recall (R), and F1-score (F) for both datasets, i.e. synthetic and NBA, with varying number of objects,  $|n|$ , number of dimensions,  $|d|$ , and uncertainty distribution. Table 3 presents the average result of the CRM method for each analysis and performance metric.

As can be seen from the table, in most cases, the CRM method achieves high values for precision, recall, and F1-score; and low RMSE values (below 0.10). This indicates that the CRM method can accurately estimate the exact values of continuous range values—a type of uncertain data common in database applications that are without a precise description, which make their representation uncertain. The results also suggest that the majority of the skyline objects retrieved by CRM,  $S_{CRM}$ , are relevant.

To validate the capability of the CRM method in handling a collection of continuously generated input data streams and analyse these data streams in close to real-time to offer accurate and fast response; we benchmarked the performance of CRM method against the works by [28] and [29]. As reported by [28], their proposed techniques, *Lazy and Eager* algorithms, can handle very spiky traffic up to  $10^5$  objects per second. Meanwhile, [29] reported that with the arrival rate of stream objects fixed to 1,000 objects/s, the query time taken by their proposed solution does not exceed 0.01s in all cases. Note that although these works deal with streaming data but they do not address the issues of data uncertainty. For this analysis, we used the worst execution time achieved by the CRM method, i.e. the results of execution time with independent dataset as presented in Figure 13(a3). This implies that for other datasets (anti-correlated, correlated, and NBA datasets), a better result than what is shown below can be anticipated. The total number of objects analysed based on the analysis presented in Figure 13(a3) is 15.6M and the total execution time taken to analyse those objects is 169.41 seconds. This means, CRM method can handle up to 92,084 objects per second; almost similar to the work by [28] which achieved  $10^5$  objects per second in very spiky traffic. Meanwhile, CRM method achieved the same result as [29] which takes 0.01s to process 1,000 objects. In conclusion, it is evident that the CRM method produces the most accurate estimates of uncertain values. It can also handle enormous volumes of high-dimensional data and analyse them almost instantly. As a result, the CRM method can be applied in environments with rapid data arrivals and stringent response time requirements, like data streams.

The analyses presented in this paper can be further enhanced to wireless sensor networks where data points are generated and transmitted from sensing devices also known as sensing data. The group of sensors which constitutes the wireless sensor network monitors data points at different sites and transmits these data points to a central site for further analysis. Apparently, the network lifetime is reduced due to energy consumption for transmitting these sensing data. Moreover, sensor data are often noisy, incomplete, or corrupted due to various factors, such as sensor failures,

environmental interference, etc. Hence, it is crucial to perform an in-depth analysis to determine an ideal technique that can handle massive amounts of sensing data while consuming the least amount of energy during transmission and the least amount of RAM and CPU usage in sensor networks.

Statistical data in the form of numbers, vectors, or categories are not always with precise values [30]. These data, also known as fuzzy data (vague data), are commonly found in environmental, biological, medical, sociological, and economic data, as well as data pertaining to quality of life. Analyzing fuzzy data is challenging due its fuzziness characteristic, which require advanced statistical analysis methods. Hence, conducting analyses to identify the optimal statistical method in maintaining fuzzy data is another interesting research direction to be explored.

## 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] M. A. M. Lawal, H. Ibrahim, N. F. M. Sani, and R. Yaakob, "An indexed non-probability skyline query processing framework for uncertain data," in *Proc. Int. Conf. Adv. Mach. Learn. Technol. Appl.* Cham, Switzerland: Springer, 2020, pp. 289–301.
- [2] M. A. Mohamud, H. Ibrahim, F. Sidi, S. N. M. Rum, Z. B. Dzolkhifli, Z. Xiaowei, and M. M. Lawal, "A systematic literature review of skyline query processing over data stream," *IEEE Access*, vol. 11, pp. 72813–72835, 2023.
- [3] N. H. M. Saad, H. Ibrahim, F. Sidi, R. Yaakob, and A. A. Alwan, "Efficient skyline computation on uncertain dimensions," *IEEE Access*, vol. 9, pp. 96975–96994, 2021.
- [4] Z. Dzolkhifli, H. Ibrahim, F. Sidi, L. S. Affendey, S. N. M. Rum, and A. A. Alwan, "A skyline query processing approach over interval uncertain data stream with K-means clustering technique," in *Proc. 11th Int. Conf. Adv. Databases, Knowl., Data Appl. (DBKDA)*, 2019, pp. 51–56.
- [5] F. Mohamed, R. M. Ismail, N. L. Badr, and M. F. Tolba, "Data streams processing techniques," in *Multimedia and Forensics Security*. Cham, Switzerland: Springer, 2017, pp. 279–305.
- [6] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in *Proc. 21st ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst.*, 2002, pp. 1–16.
- [7] E. Tavazzi, S. Daberdaku, R. Vasta, A. Calvo, A. Chiò, and B. Di Camillo, "Exploiting mutual information for the imputation of static and dynamic mixed-type clinical data with an adaptive K-nearest neighbours approach," *BMC Med. Informat. Decis. Making*, vol. 20, pp. 1–23, Aug. 2020.
- [8] J. Ren, S. D. Lee, X. Chen, B. Kao, R. Cheng, and D. Cheung, "Naive Bayes classification of uncertain data," in *Proc. 9th IEEE Int. Conf. Data Mining*, Dec. 2009, pp. 944–949.
- [9] F. Galton, *Natural Inheritance*. New York, NY, USA: Macmillan, 1894.
- [10] G. A. Seber and A. J. Lee, *Linear Regression Analysis*. Hoboken, NJ, USA: Wiley, 2012.
- [11] T. Hanson. *Multiple Regression*. Accessed: Feb. 2, 2024. [Online]. Available: <https://people.stat.sc.edu/hanson/stat704/notes9.pdf>
- [12] N. S. B. Rosli, R. B. Ibrahim, I. Ismail, and K. Bingi, "Application of principal component analysis vs. multiple linear regression in resolving influential factor subject to air booster compressor motor failure," in *Proc. IEEE 4th Int. Symp. Robot. Manuf. Autom. (ROMA)*, Dec. 2018, pp. 1–5.
- [13] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statistician*, vol. 46, no. 3, pp. 175–185, 1992.
- [14] *K-Nearest Neighbor (KNN) Algorithm for Machine Learning*. Accessed: Feb. 2, 2024. [Online]. Available: <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>
- [15] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

- [16] C. Hemanth and S. A. Kumar, "Real estate search and valuation to get best valued sites using linear regression algorithm and compared with random forest algorithm," *Baltic J. Law Politics*, vol. 15, no. 4, pp. 431–438, 2022.
- [17] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. London, U.K.: Chapman & Hall, 1984.
- [18] *Decision Tree in Machine Learning*. Accessed: Feb. 2, 2024. [Online]. Available: <https://www.almabetter.com/bytes/tutorials/data-science/decision-tree>
- [19] E. D. A. L. Neto and F. D. A. T. de Carvalho, "Centre and range method for fitting a linear regression model to symbolic interval data," *Comput. Statist. Data Anal.*, vol. 52, no. 3, pp. 1500–1515, Jan. 2008.
- [20] S. Borzsony, D. Kossmann, and K. Stocker, "The skyline operator," in *Proc. 17th Int. Conf. Data Eng.*, 2001, pp. 421–430.
- [21] J. Liu, X. Li, K. Ren, and J. Song, "Parallelizing uncertain skyline computation against n-of-N data streaming model," *Concurrency Comput., Pract. Exper.*, vol. 31, no. 4, Feb. 2019, Art. no. e4848.
- [22] M. M. Lawal, H. Ibrahim, N. F. M. Sani, and R. Yaakob, "Analyses of indexing techniques on uncertain data with high dimensionality," *IEEE Access*, vol. 8, pp. 74101–74117, 2020.
- [23] H. Chen, N. Wang, X. Du, K. Mei, Y. Zhou, and G. Cai, "Classification prediction of breast cancer based on machine learning," *Comput. Intell. Neurosci.*, vol. 2023, no. 1, Jan. 2023, Art. no. 6530719.
- [24] O. Peretz, M. Koren, and O. Koren, "Naive Bayes classifier—An ensemble procedure for recall and precision enrichment," *Eng. Appl. Artif. Intell.*, vol. 136, Oct. 2024, Art. no. 108972.
- [25] A. Sharma, M. K. Sharma, and R. K. Dwivedi, "Exploratory data analysis and deception detection in news articles on social media using machine learning classifiers," *Ain Shams Eng. J.*, vol. 14, no. 10, Oct. 2023, Art. no. 102166.
- [26] F. Ali, A. Sarwar, F. I. Bakhsh, S. Ahmad, A. A. Shah, and H. Ahmed, "Parameter extraction of photovoltaic models using atomic orbital search algorithm on a decent basis for novel accurate RMSE calculation," *Energy Convers. Manage.*, vol. 277, Feb. 2023, Art. no. 116613.
- [27] J. Li, "Assessing the accuracy of predictive models for numerical data: Not r nor r2, why not? Then what?" *PLoS ONE*, vol. 12, no. 8, Aug. 2017, Art. no. e0183250.
- [28] Y. Tao and D. Papadias, "Maintaining sliding window skylines on data streams," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 3, pp. 377–391, Mar. 2006.
- [29] Z. Huang, S. Sun, and W. Wang, "Efficient mining of skyline objects in subspaces over data streams," *Knowl. Inf. Syst.*, vol. 22, no. 2, pp. 159–183, Feb. 2010.
- [30] R. Viertl, *Statistical Methods for Fuzzy Data*. Hoboken, NJ, USA: Wiley, 2011.



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



**SITI NURULAIN MOHD RUM** received the bachelor's degree from Universiti Teknologi Malaysia (UTM), the Diploma degree from Universiti Teknologi MARA, and the master's and Ph.D. degrees from the University of Malaya (UM). Before joining academia, she spent over 15 years as an IT professional, involved in a number of 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 is currently a Lecturer with the Department of Computer Science and Information Technology, Universiti Putra Malaysia. She has published a number of 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.



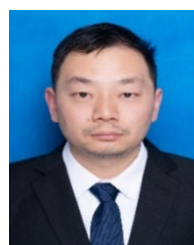
**MUDATHIR AHMED MOHAMUD** (Member, IEEE) is currently pursuing the Ph.D. degree in database systems with the Department of Computer Science, Universiti Putra Malaysia. He is also a Teaching Assistant and a Graduate Research Assistant with the Department of Computer Science, UPM. His research interests include query optimization, preference query evaluation, data science, database integration, data stream, machine learning, data mining, and information systems.



**ZARINA BINTI DZOLKHI FLI** received the bachelor's degree in computer science and the Master of Science degree in database system from Universiti Putra Malaysia. She is currently a Lecturer with the Department of Computer and Networking, Faculty of Computing, Universiti Malaysia Pahang. Her research interests include query processing, data streaming, preference query, and data networking.



**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, 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.



**ZHANG XIAOWEI** 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 mining.

...