A Method for Processing Top-k Continuous Query on Uncertain Data Stream in Sliding Window Model

RAJA AZHAN SYAH RAJA WAHAB, SITI NURULAIN MOHD RUM, HAMIDAH IBRAHIM, FATIMAH SIDI, ISKANDAR ISHAK
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
43400 UPM Serdang
Selangor Darul Ehsan
MALAYSIA

Abstract: The data stream is a series of data generated at sequential time from different sources. Processing such data is very important in many contemporary applications such as sensor networks, RFID technology, mobile computing and many more. The huge amount data generated and frequent changes in a short time makes the conventional processing methods insufficient. The Sliding Window Model (SWM) was introduced by Datar et. al to handle this problem. Avoiding multiple scans of the whole data sets, optimizing memory usage, and processing only the most recent tuple are the main challenges. The number of possible world instances grows exponentially in uncertain data and it is highly difficult to comprehend what it takes to meet Top-k query processing in the shortest amount of time. Following the generation of rules and the probability theory of this model, a framework was anticipated to sustain top-k processing algorithm over the SWM approach until the candidates expired. Based on the literature review study, none of the existing work have been made to tackle the issue arises from the top-k query processing of the possible world instance of the uncertain data streams within the SWM. The major issue resulted from these scenarios need to be addressed especially in the computation redundancy area that contributed to the increases of computational cost within the SWM. Therefore, the main objective of this research work is to propose the top-k query processing methods over uncertain data streams in SWM utilizing the score and the Possible World (PW) setting. In this study, a novel expiration and object indexing method is introduced to address the computational redundancy issues. We believed the proposed method can reduce computational costs and by managing insertion and exit policy on the right tuple candidates within a specified window frame. This research work will contribute to the area of computational query processing.

Key-Words: Top-k, Possible World, Uncertain

Received: December 17, 2020. Revised: April 20, 2021. Accepted: May 4, 2021. Published: May 15, 2021.

1 Introduction

Over the last decade, query processing over data streams has drawn major research interest in many emerging applications. The processing of data streams is challenging due to the various reasons: (i) the stream data objects occur online, (ii) the application lack of management control of the order of the arriving tuple, either within a data stream or across data streams, (iii) the data streams are inherently unbound in scale, and (iv) the data stream object will be discarded after it has been processed. The characteristics of the tuple can differ over time and continuously generated, and the greater size of the window can contribute to the overestimation of the required size of the window that inevitably results in the undesired and unexpected returned tuples [1]. This can also add to the volatility of the likelihood production of tuples for both scenarios.

Many studies have been conducted to develop models for describing the uncertain data streams [2] in the world of semantics that can be utilized as a sequence of events of a relationship such as the relational data model [3, 4] semi-structured data model [5, 6] stream data model [1, 5], multidimensional data model [6] and others. The predominant semantics of these models are all based on the possible world [7, 8] and the theory of a possible world is an important concept in the uncertain data research area. A variety of instances of the possible world are formed by uncertain data tuples. The framework of the possible world is governed by the standard generation rules, for an

example the exemption from inter-company tuples representing the same scenario of world entity. The possible world environment can be expressed as all possible tuple's combination created in the sliding window model[9]. Tuples inside the SWM can be created with several possible worlds for a given timestamp, which can increase the number of tuples exponentially inside the sliding window as the window size increases. Continuous processing of sliding window queries is generally required, in which users should be alerted when there are changes in the query's result to ensure the newest sliding window will always keep updated [10]. Sliding windows is a typical data stream model designed to increase the performance of the data stream processing by abstracting the data logically [11]. Sliding window also owns a set of optimization approaches to reduce the repetitive computing induced by crossing events between adjacent windows. [12]. Intuitively, due to the uncertainty of the existence of such a tuple, the amount of tuple that may be present in the sliding window will exceed the size of the window specified by the user [5]. Handling the high level of top-k query processing over uncertain streams have been developed by many researchers such as [13-15], but few have been made within the area of count-based and time-based sliding window model.

In many fields, including web searching, data sensor management, tracked data, data extraction and multimedia management, ranking queries approach which cannot be met introduced the combination of these two techniques was introduced in answering the key questions [16]. The importance of top-k queries has resulted in many developments of algorithms to answer a number of variants of the first-k objects [17]. Studies performed on top-k and skyline algorithms contributed to inconsistencies in the search results in certain data [18]. There are certain criteria for many modern applications cannot be met by these two questions. The top-k dominating approaches by [18, 19] introduced the combination of these two techniques was introduced to address these problems. Top-k on certain data, primarily depends on the scoring method, however, computation on top-k query of uncertain data require complex processing due to the need of scoring calculation and probability generation for each tuple[2]. The combination of these two factors can create a variety of uncertain top-k query semantics such as U-Topk [20-22], probabilistic threshold top-k (PT-k) [23-25], eScore Rank [25], global Top-k[25], probabilistic top-k dominating (PTD) [26-28], probabilistic threshold top-k Simplified (PTkS) [27, 29] , uncertain top-(k,l) range (UTR) [29] [35] and etc.

The U-*k*Ranks and U-top*k* are among the earliest studies for handling the top-k query processing on uncertain data [20]. In U-kRanks, the i^{th} record from a list of k record will be returned by the query result and have the highest probability of being ranked in all possible worlds at the i^{th} position that led to the multiple records return. Whereby in U-Topk, the query results merely consist of k tuples, which in many situations it is not sufficient due to these reasons: firstly, the chance of outcome is very limited, and it is difficult for users to consider, secondly, it abandons the relationships between the tuples and the corresponding entities, so it does not fully represent the actual state of the tracked entities. Zhang [29] defined tuples as the sum of the probabilities of all possible worlds whose best solution consisted of tuples and subsequently returned the k tuples to the highest Global-Topk probability. It can be concluded from these finding, none of the above research works have been made to tackle the issue arises from the top-k query processing of the possible world instance of the uncertain data streams within the SWM. The major issue resulted from these scenarios need to be addressed especially in the computation redundancy area that contributed to the increases of computational cost within the SWM. In this study, the tuple-level uncertainty for all independent items will be considered. This research work will also involve the development of an efficient top-k query techniques over the proposed SWM.

2 Problem Formulation

In the real world practice, processing of top-k continuous queries through a sliding window model (SWM) is an essential area for managing the uncertain data stream [30]. Typically, the uncertain data object is often modelled using probability methods, and the possibility of the system malfunction is high due to its complexity making the application that process the data model much more difficult. Processing data streaming is challenging due to number of reasons.

The first reason is the system has no control over the order of objects arrival, either within or through data streams. Second, the overall size of data streams is unlimited, and the third reason is when extracting objects from streams, others will be removed or erased from data streams and cannot easily be recovered through archiving. Thus, it cannot be retrieved conveniently unless it is directly stored in

memory, which is usually limited relative to the size of the data streams. While top-k computation for uncertain data streams over SWM has been extensively studied by [13, 14, 31-34]. However, few studies have been conducted on the development on the general framework comprising SWM with possible world computation for the Top-k. The possible world model is a common approach to represent uncertain data streams. The cost of maintaining a set of objects, however, is very high, that in the worst-case scenario, it has linear complexity in window size. These complexities often fail to meet for the real-time requirements in real-world applications. In this section, the exploration task of the relevant sources concerning the research problem is given in Table 1.

Table 1. A snapshot example of uncertain data stream

Id	Time	Time2 (min)	Speed (km)	Prob.
R1	09:05:00 AM	5	80	0.3
R2	09:10:00 AM	10	65	0.4
R3	09:10:00 AM	10	45	0.5
R4	09:15:00 AM	15	30	1
R5	09:20:00 AM	20	50	0.8
R6	09:20:00 AM	20	25	0.2
R7	09:25:00 AM	25	40	0.5
R8	09:26:00 AM	26	55	0.6
R9	09:26:00 AM	26	78	0.4
R10	09:30:00 AM	30	90	0.8

Considering the application for vehicle tracking snapshot mentioned in Table 1 above, the tracker is used at the control point to detect vehicle speed. At the tracking point, three motion trackers are mounted. Speed calculations may be inaccurate or may only be accurate with certain probability due to different factors. The uncertainty is shown in from 09:01:00 AM to 09:30:00 AM. In many modern systems, multiple continuous queries can be carried out simultaneously on the same data stream under varying circumstances across the TIME FRAME and SLIDE parameters. These parameters can be used for the SWM's setting. Using Table 1 as the example, a user wants to know the top-3 or top-2 readings speed for the last 20 minutes. The results will be sent for every 4 minutes (SLIDE-depend on SWM approach) during the last 20 minutes (TIME FRAME) as represented by the query of SWM with tuple insertion and exit policy.

Fig. 1 and Fig. 2 present the detailed explanation on how it works from two different approaches is that it is only providing may lead to the resources exhaustion be removed from the window in each of the formed windows. Fig. 1 illustrates the example of count based SWM with the tuple insertion and exit method. In the count based SWM, the count window is dimensioned by the number of events within it. A count window has no fixed time that it will be valid. It retains the specified number of rows. Once the window is full, each new row that arrives or each new bucket that is created in the window displaces the oldest row or bucket, which is then removed from the window. For an example (see Fig. 1), the last tuple is expected to be expired from the SWM and the overlapping computation of top-k tuple candidates occurred in the window frame. For example, in W=1, the computation takes place on R6, R5, R4, R3, R2, R1 in sliding window as there is no tuple inside. Six tuples are inserted into the window frame. In W=2, the R7 is inserted and computation takes place on R7, R6, R5, R4, R3, R2 where R1 exits due to the number of tuples within the window has reach the maximum size. During this process, computation redundancy occurred on R6, R5, R4, R3 and R2. In W=3, the computation takes place on R8, R7, R6, R5, R4, R3, and R2 exit from the SWM. During this process, computation redundancy occurred on R7, R6, R5, R4, and R3.

The drawback of this approach, it is only providing the mechanism on how to compute the candidates in general basis and no details on the efficiency of the top-k query processing of the possible world computation. Since the number of window size is fixed, the number of tuples that can be processed within the window is limited and problem arises if the number of tuples that need to be processed is large. Overlapping computation will be happening within the window frame since the number of tuples that can be retained at one time within the window frame should be equal to the window size, for example if the size of window is 6, and if the last tuple within the window exit, the new tuple will be inserted into window frames, the last five tuples within SWM need to be recomputed together with the possible world and this may lead to the recourses hunger. If the window size it too big, more PW need to be generated for processing the top-k tuples.

Fig. 1. SWM of Count-based approach with tuple insertion and exit method.

Fig. 2 illustrates the time-based SWM with tuple entry and exit method. A time-based SWM keeps number of tuples within the window frame depending on how many candidates arrived within the specific interval time. Tuples that have been in the window frame longer than the specified interval will be exit from the window. The expiration happened, regardless whether new rows arrived or not. For an example (See Fig. 2), in this example, the window slide for every 4 minutes and the specified interval time to process the tuples is 20 minutes, In W=1, the computation takes place on R6, R5, R4, R3, R2, R1 within the sliding window for 20 minutes since there are no tuple inside the frame. As the time moving, the window frame will be shifted by 4 minutes and the interval time remained 20 minutes, for example in W=2, as R1 has been the longest within the time frame, it will be left out as the time for window frame is moving. R2, R4 and R5 will be recomputed. The re-computation also happened for R4 and R5 in W=3. R2 will be out from the SWM and two new arriving tuples, R7 and R8 will be within the window frame for processing.

If the proposed algorithm utilizes the pruning or top k dominate when a new tuple is inserted, the algorithm must determine whether the new tuple belongs to the dominated region or not. If yes, then It is likely to affect the possible world computation for every query and massive updates the top-k query processing. When the top-k tuple expired, the computation from scratch is performed if the current tuple does not have a higher score and probability than the expired tuple.

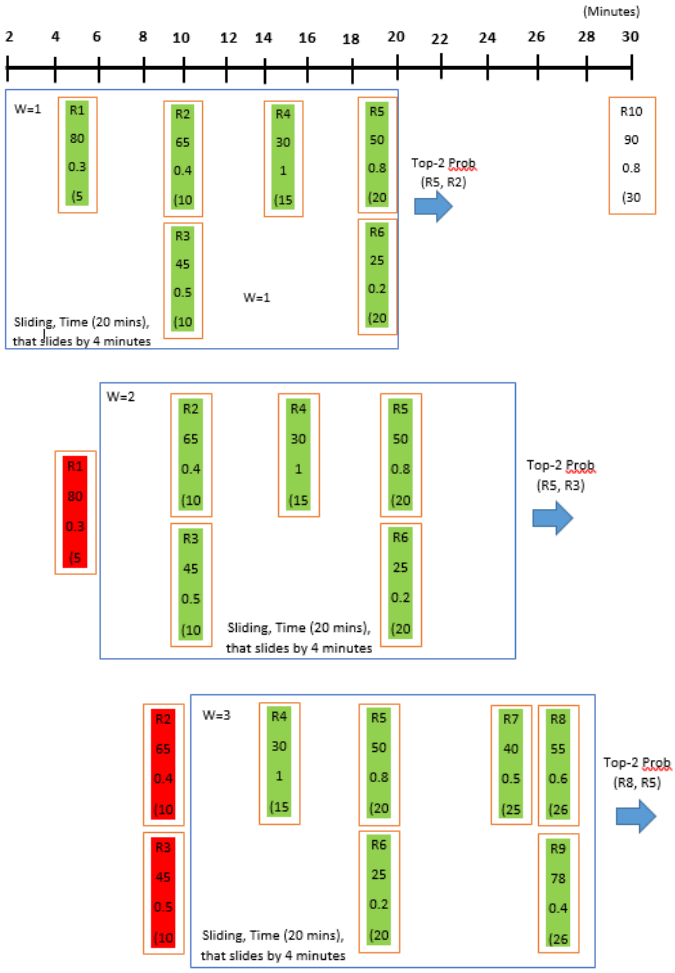

Fig. 2. SWM of Time-based approach with tuple insertion and exit method.

Obviously, these tuples will be processed using the same uncertain data stream method that includes the same number of objects in each of the formed window. It becomes more complicated when (i) the candidate tuples created from the tuples that already expired, ii) the overlapping computation of the candidate tuples within the window frame and iii) the number of possible world instances increase exponentially with the growth of uncertain data streams. The inefficiency in processing each tuple may lead to the unnecessary top-k computation and caused resources hunger. Intuitively, continuous processing of the top-k query can be carried out using the efficiency of the proposed SWM with the efficient of top-k algorithm that optimizes the scenario of the possible world rules. This can be realized by combining the best score and maximum probability values focusing on the improvement of delta-based and snapshot-based SWM's. Based on these finding, we formulate three research questions that need to be addressed as follows:

a) How to provide an efficient mechanism for processing the top-k query in the sliding window model in the situation where many potential candidates expire and overlapping computation happen?
b) How to improve the computation cost of top-k processing on continuous queries over uncertain data streams?
c) What is the suitable approach for optimizing the continuous query processing for managing the performance issue due to the possible world?

3 Methodology

In this section, we present the developed methodology of this research work. There are six phases needed to be that are manly focused as shown in Fig. 3. In the first phase, further investigation is conducted on the existing top-k query processing algorithms, especially those that are mainly focus on the uncertain data stream. It is also important to examine the concepts related to the continuous processing of top-k queries over uncertain data streams, along with the TIME FRAME and SLIDE parameters of the SWM, etc. There are overlapping computations of possible world candidates created from uncertain tuples before the candidates expired from the sliding window frame, thus the exact top-k processing cannot be derived but instead the high score combination with maximum probability of an object being a top-k rank is computed. Therefore, understanding the concept of optimization theories utilizing the PW with the score and probability computation of an object for the uncertain top-k list is needed. With respect to the finding of the limitations of the literature study provided in section 2, a new model as shown in Fig. 3 is developed to cope with these problems.

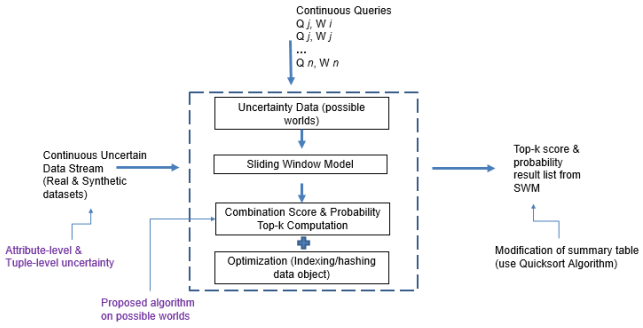

Fig. 3. Methodology for top-k computation for uncertain data stream over sliding window.

The phases in Fig. 3 are further described below: -

To answer research question number two, identifying the uncertainty data and design a sliding window model over uncertain data stream is required. In this phase, a new effective solution to evaluate is developed on top-k continuous query over uncertain data streams, concentrating on the use of SWM efficiency. The principle of SWM is covered in depth together with a well-known strategy of top-k queries which continuously update the results as recent new data comes from the stream. It is a founded strategy to tackle the issue relating to the overlapping computation of candidate's possible world created from uncertain tuples before the candidates exited from the sliding window frame.

However, deciding on the proper tuple candidates of possible world is not a forthright task. We have to visualize two conceivable common SWM methods over uncertain data streams tuple, namely: Count-based SWM as shown in Fig1 and Time-based SWM as shown in Fig 2. The detail analysis of proposed SWM (Delta-based) will be performed with various numbers of continuous top-k queries with different conditions (i.e. TIME FRAME and SLIDE parameters) before a top-k computation is propounded on candidate's possible world scenario. The performance will be analysed on the top-k continuous query over uncertain data streams (See Fig 3), concentrating on the use of proposed SWM efficiency with regards to the reduction in computation of candidate's tuple within sliding window frame response time and dealing with complexity of computation (exponent number possible world) processing time.

Fig. 4 illustrates the proposed study of the delta-based approach of SWM with tuple insertion and exit method. Based on the diagram, the process obviously shows that it can reduce or minimize the overlapping computation on tuple candidates for top-k within the sliding window frame. Delta value can be changed to suit the computation because it does not practice a static set of window size, using eviction policy delta where the number tuple is varied. Delta-based SWM keeps the difference between attribute for exit policy.

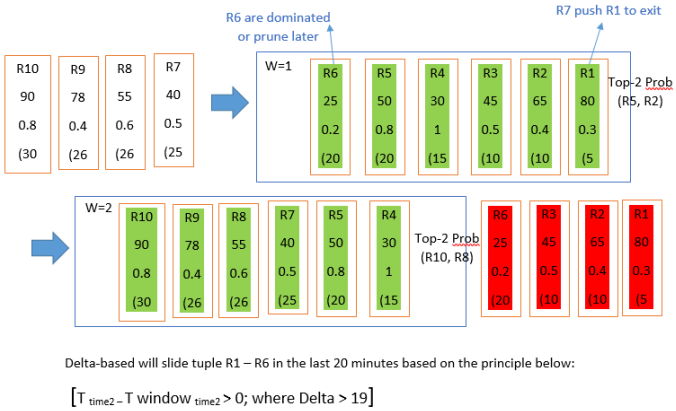

Fig. 4. The proposed SWM of delta-based approach with tuple insertion and exit method.

The next phase is to design the top-k query processing framework to answer the research question number two. During this phase, the scope covered the tuple-level uncertainty with all independent objects. In the context of tuple-level uncertainty, nearly every tuple is mutually independent. It is important to understand the uncertainty data that eventually can reflect the objective of the knowledge or information that needs to be extracted. Thus, this study focuses on the use of the independent uncertainty data model, as well as their corresponding possible worlds.

The next phase is to develop an efficient Top-k algorithm over SWM to meet the second objective and to answer the third research question. The top-k query processing does not consider the tuples score and uncertainty. The uncertain data model and possible world of the semantic model will be first analysed and then the new top-k query semantics of uncertain data streams will be established. Eventually, a top-k query algorithm will be built on uncertain data streams. This algorithm sorts the score of each tuple and selects the k tuples to create the set from the top-k queries. This study focuses on three main Top-k queries for uncertain data with possible world setting. One of them is *Pk*-Top-*k* algorithm as explained below. The proposed computation will be performed to evaluate the correctness of the model for defining the uncertain data objects and the effectiveness of computational of top-k on the tuples, values, and possible worlds over the proposed sliding window model as shown in Table 2.

Table 2. The Calculation Probability of Possible Worlds (proposed SWM Delta-based, W=1)

PW	Calculation Prob.	Prob.	Top-2	Top-3
W1=R1,R2, R4, R5	0.3 X 0.4 X 1 X 0.8	0.096	R1, R2	R1,R2, R5
W2=R1,R2, R4, R6	0.3 X 0.4 X 1 X 0.2	0.024	R1, R2	R1,R2, R4
W3=R1,R3, R4, R5	0.3 X 0.5 X 1 X 0.8	0.12	R1, R5	R1,R5, R3
W4=R1,R3, R4, R6	0.3 X 0.5 X 1 X 0.2	0.03	R1, R3	R1,R3, R4
W5=R1,R4, R5	0.3 X (1-0.4-0.5) X 1 X 0.8	0.024	R1, R5	R1,R4, R5
W6=R1,R4, R6	0.3 X (1-0.4-0.5) X 1 X 0.2	0.006	R1, R4	R1,R4, R6
W7=R2,R4, R5	(1-0.3) X 0.4 X 1 X 0.8	0.224	R2, R5	R2,R4, R5
W8=R2,R4, R6	(1-0.3) X 0.4 X 1 X 0.2	0.056	R2, R4	R2,R4, R6
W9=R3,R4, R5	(1-0.3) X 0.5 X 1 X 0.8	0.28	R5, R3	R5,R3, R4
W10=R3,R4, R6	(1-0.3) X 0.5 X 1 X 0.2	0.07	R3, R4	R3,R4, R6
W11=R4,R5	(1-0.3) X (1-0.4-0.5) X 1 X 0.8	0.056	R5, R4	R5, R4
W12=R4, R6	(1-0.3) X (1-0.4-0.5) X 1 X 0.2	0.014	R4, R6	R4, R6

Possible World

→ R1(0.3;0.7) X R2&R3(0.4;0.5;0.1) X R4(1) X R5&R6(0.8;0.2)

→ 2 X 3 X 1 X 2 = **12**

Table 3. SWM Tuple for Top-2 and Top-3 Probability (over proposed SWM Delta-based)

ID	Top-2 Prob. Calculation	Top-2 Prob	Top-3 Prob. Calculation	Top-3 Prob
R1	0.096+0.024+0.12+0.03+0.024+0.006	0.3	0.096+0.024+0.12+0.03+0.024+0.006	0.3
R2	0.096+0.024+0.224+0.056	0.4	0.096+0.024+0.224+0.056+	0.4
R3	0.03+0.28+0.07	0.38	0.12+0.03+0.28+0.07	0.5

R4	0.006+0.056+0.07+0.056+0.014	0.202	0.024+0.03+0.024+0.006+0.224+0.056+0.28+0.07+0.056+0.014	0.784
R5	0.12+0.024+0.224+0.28+0.056	0.704	0.096+0.12+0.024+0.224+0.28+0.056	0.8
R6	0.014	0.014	0.006+0.056+0.07+0.014	0.173

Table 4. Result of Top-2 and Top-3 Probability

Top-2 Prob	Top-3 Prob
R5	R5
R2	R4
	R3

At the point of arrival, the probability method is used to determine if an object will be among the top-k candidate object for a query. There are two possible worlds, one of which is an uncertain tuple, and the other does not contain a tuple [20]. There is only one probability for tuple R4 with value 1 and two probabilities for tuple R1 (with value of 0.3 and 0.7). The probability values of R2 and R3 are less than 1 in all, which implies that there are three cumulative probabilities composed of R2=0.4, R3=0.5 and others=0.1. There are two probabilities for R5 and R6, as their full probabilities are equal to 1. Thus, the possible worlds generated are 2 X 3 X 1 X 2 = 12. For an example, the PW (W1) = {R1, R2, R4, R5) is 0.3 X 0.4 X 1 X 0.8 = 0.096. To respond a query over the last 20 minutes on W=1, with the top 3 or top 2 readings speed, all the possible worlds listed in Table 2 must be reviewed to decide the 3 highest speed readings, each of which has a fair chance of being one of the top 3 readings. In sliding windows, the highest probability of a tuple is the sum of the possibilities of the worlds with the tuple as among the top tuple. For Table 3, P*k*-Top*k* algorithm computes the top-2 and top-3 probability of each tuple. The top-2 probability of R5, for example, represents the number of possible worlds (W3, W5, W7, W9 and W11 possibilities, containing R5 as one of the top-3 rankings. Hence, the Top- 3 probability of R5 is 0.12+0.024+0.224+0.28+0.056 = 0.704. The results of the Top2 query in W = 1 are R5 and R2 (shown in Table 4), with the highest top2 probabilities among them, and the 3rd column of Table 3 displays their sum probabilities with bold fonts.

The next phase is to develop optimization method for SWM for top-k processing to answer the third objective and to answer the third research question. Indexing and query processing methods are interrelated, and this phase will study on how SWM index optimization and dominant relationship method between tuple object can help to improve the query performance. According to the challenging scenario of possible worlds (shown in Table 3), the actual probability of each speed measurement is not the same between top-2 and top-3 candidates. In other words, the sum of the top-3 computation cannot be obtained directly from the results of the top-2 query. Thus, for fast responsive in real-time query processing, multi-dimensional indexing and modification of summary table result are needed. In order to address this issue, we therefore proposed an efficient optimization approach for Top-k queries on uncertain data at this phase. Hence, the indexing of the contents in sliding windows need to be integrate and develop. For an example, to add a new data to the SWM, the indexing structure needs to support a high insertion rates and delete mechanism to handle expired data.

For the final phase, the performance of the proposed model shall be evaluated through the experiments using synthetic data and real data stream. It will be measured using several parameters such as response time, processing time, storage cost and reduction in computation cost of top-k query processing. The evaluation will be executed on two kinds of datasets:

Synthetic dataset:

- Uniform distribution, correlated distribution, and anti-correlated distribution.
- Various numbers of uncertain tuples: 1.2 (billion), 2.5B, 3.75B, 5B and 6.25B.
- More than 6 attributes.
- Each tuple will have a time stamp that reflects streaming data.

Real datasets:

- Real data stream that is extracted from Chicago Trip Taxi Meter and National Basketball Association (NBA)
- Number of uncertain tuples are limited to 1 ~ 2 M (million)

4 Discussion and Conclusion

In uncertain data stream environment, it cannot be assured that its existence would always be present in

specific sliding window. The tuple characteristics can differ over time and lead to constant, and broader windows that may overestimate the window size, which inevitably leads to unnecessary and unwanted outcomes. In other words, retrieving accurate uncertain data objects that satisfy the Top-k query processing is highly complex and challenging. The number of the possible world instances will be an exponential increase with the growth of uncertain data. Solving the issue by decreasing the number of tuples by removing the tuples (not dominate by others) without include it in possible world rules will affect Top-k final result. At that point, based on the generation of result rules and probability theory of this model, a framework was anticipated to retain top-k processing over the SWM approach until the candidates expired. Thus, the study also concerned with the development of efficient techniques for top-k queries over the proposed sliding windows. The uncertainty data and possible world of the semantic model were first analysed and then the new top-k query semantics of uncertain data streams will be established.

Eventually, a top-k query algorithm will be built on uncertain data streams. This algorithm sorts the score of each tuple and selects the k tuples to create the set from the top-k queries. The proposed computation is deployed to evaluate the accuracy of the model for defining uncertain data objects and the effectiveness of computational of top-k on the tuples, values, and possible worlds over the proposed sliding window model. Through the evaluation phase, the performance of the proposed model shall be evaluated through the experiments using synthetic data and real data stream that consists sample of independent of the uncertainty tuple and uncertainty attributes (e.g. Chicago Trip Taxi Meter, National Basketball Association (NBA) and other selected uncertain data stream).

The research methodology deployed is to construct a method to evaluate the effectiveness of computational Top-k on tuples, values, and possible worlds over proposed SWM. Intuitively, the process flow method investigates the improvement of Top-k computational issues through a combination of score and probability (tuple-uncertainty and Attribute/value-uncertainty) possible world setting. When computing, a SWM or object indexing technique is proposed to efficiently reduce the Top-k computational cost and its performance issues. It is expected that at the end of this study, the outcome may demonstrate how streams with value uncertainty are model to prove conditions in theory under which the tuples (possible worlds) become the uncertain data Top-k query result. In order to demonstrate efficiency and effectiveness, comprehensive studies in different settings will be performed using synthetic and real datasets. This work is useful in helping to reduce costs and makes decisions faster and better.

References:

[1] Jin, C., Yi, K., Chen, L., Yu, J. X., and Lin, X.: Sliding window top-k queries on uncertain streams, *Proceedings of the VLDB Endowment,* vol. 1, 2008, pp. 301-312.

[2] Sarma, A. D., Benjelloun, O., Halevy, A., and Widom, J.:Working models for uncertain data, in *22nd International Conference on Data Engineering (ICDE'06)*, 2006, pp. 7-7.

[3] Fuhr, N. and Rölleke, T.: A probabilistic relational algebra for the integration of information retrieval and database systems, *ACM Transactions on Information Systems (TOIS),* vol. 15, 1997, pp. 32-66.

[4] 4. Lakshmanan, L. V., Leone, N., Ross, R., and Subrahmanian, V. S.: Probview: A flexible probabilistic database system, *ACM Transactions on Database Systems (TODS),* vol. 22, 1997, pp. 419-469.

[5] Ré, C., Letchner, J., Balazinksa, M., and Suciu, D.:Event queries on correlated probabilistic streams, in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 715-728.

[6] Burdick, D., Deshpande, P., Jayram, T., Ramakrishnan, R., and Vaithyanathan, S.:OLAP over uncertain and imprecise data, in *VLDB*, 2005, pp. 970-981.

[7] Abiteboul, S., Kanellakis, P., and Grahne, G.: On the representation and querying of sets of possible worlds, *Theoretical computer science,* vol. 78, pp. 159-187, 1991.

[8] Green, T. J.: Models for incomplete and probabilistic information, *Managing and Mining Uncertain Data,* vol. 9, 2009.

[9] Li, L., Wang, H., Li, J., and Gao, H.: A survey of uncertain data management, *Frontiers of Computer Science,* vol. 14, 2020, pp. 162-190.

[10] Xue, Z., Li, R., Zhang, H., Gu, X., and Xu, Z.:DC-Top-k: A Novel Top-k Selecting Algorithm and Its Parallelization, in *2016 45th International Conference on Parallel Processing (ICPP)*, 2016, pp. 370-379.

[11] Chen, B., Lv, Z., Yu, X., and Liu, Y.: Sliding window top-k monitoring over distributed data streams, *Data Science and Engineering,* vol. 2, 2017, pp. 289-300.

[12] Carbone, P., Katsifodimos, A., and Haridi, S.:Stream Window Aggregation Semantics and Optimization, ed, 2019.

[13] Mingyi, D. and Yinju, L.: An effective uncertain data streams top-K query algorithm, *The Open Automation and Control Systems Journal,* vol. 7, 2015.

[14] Dallachiesa, M., Jacques-Silva, G., Gedik, B., Wu, K.-L., and Palpanas, T.: Sliding windows over uncertain data streams, *Knowledge and Information Systems,* vol. 45, 2015, pp. 159-190.

[15] Chen, T., Chen, L., Oezsu, M. T., and Xiao, N.: Optimizing multi-top-k queries over uncertain data streams, *IEEE transactions on knowledge and data engineering,* vol. 25, 2012, pp. 1814-1829.

[16] Khosla, C. and Kakkar, P.: Top-k Query Processing Techniques in Uncertain Databases: A Review, *International Journal of Computer Applications,* vol. 120, 2015.

[17] Shen, Z., Cheema, M. A., Lin, X., Zhang, W., and Wang, H.: A Generic Framework for Top-k Pairs and Top-k Objects Queries over Sliding Windows, *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, 2012, pp. 1349-1366.

[18] Miao, X., Gao, Y., Zheng, B., Chen, G., and Cui, H.: Top-k dominating queries on incomplete data, *IEEE Transactions on Knowledge and Data Engineering,* vol. 28, 2015, pp. 252-266.

[19] Ezatpoor, P., Zhan, J., Wu, J. M.-T., and Chiu, C.: Finding Top-k Dominance on Incomplete Big Data Using MapReduce Framework, *IEEE Access,* vol. 6, 2018, pp. 7872-7887.

[20] Soliman, M. A., Ilyas, I. F., and Chang, K. C.-C.:Top-k query processing in uncertain databases, in *2007 IEEE 23rd International Conference on Data Engineering*, 2007, pp. 896-905.

[21] Soliman, M. A. and Ilyas, I. F.:Ranking with uncertain scores, in *2009 IEEE 25th international conference on data engineering*, 2009, pp. 317-328.

[22] Lian, X. and Chen, L.:Probabilistic ranked queries in uncertain databases, in Proceedings of the 11th international conference on Extending database technology: Advances in database technology, 2008, pp. 511-522.

[23] Hua, M., Pei, J., and Lin, X.: Ranking queries on uncertain data, *The VLDB Journal,* vol. 20, 2011, pp. 129-153.

[24] Hua, M., Pei, J., Zhang, W., and Lin, X.:Efficiently answering probabilistic threshold top-k queries on uncertain data, in *2008 IEEE 24th International Conference on Data Engineering*, 2008, pp. 1403-1405.

[25] Cormode, G., Li, F., and Yi, K.:Semantics of ranking queries for probabilistic data and expected ranks, in *2009 IEEE 25th International Conference on Data Engineering*, 2009, pp. 305-316.

[26] Lian, X. and Chen, L.:Top-k dominating queries in uncertain databases, in Proceedings of the 12th international conference on extending database technology: advances in database technology, 2009, pp. 660-671.

[27] Lian, X. and Chen, L.: Probabilistic top-k dominating queries in uncertain databases, *Information Sciences,* vol. 226, pp. 23-46, 2013.

[28] Lian, X. and Chen, L.: Shooting top-k stars in uncertain databases, *The VLDB journal,* vol. 20, 2011, pp. 819-840.

[29] Xiao, G., Li, K., and Li, K.:Reporting l most favorite objects in uncertain databases with probabilistic reverse top-k queries, in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, 2015, pp. 1592-1599.

[30] Yang, D., Shastri, A., Rundensteiner, E. A., and Ward, M. O.:An optimal strategy for monitoring top-k queries in streaming windows, in *Proceedings of the 14th International Conference on Extending Database Technology*, 2011, pp. 57-68.

[31] Chen, D. and Chen, L.: Sliding-Window Probabilistic Threshold Aggregate Queries on Uncertain Data Streams, *Information Sciences,* vol. 520, 2020, pp. 353-372.

[32] Kar, D. C.: On pruning of data in a sliding window for computing a rank-order element, *IEEE Signal Processing Letters,* vol. 24, 2017, pp. 1005-1009.

[33] Xiao, G., Li, K., Li, K., and Zhou, X.: Efficient top-(k, l) range query processing for uncertain data based on multicore architectures, *Distributed and Parallel Databases,* vol. 33, 2015, pp. 381-413.

[34] Zhang, Z., Wei, X., Xie, X., Pan, H., and Miao, Y.: An efficient optimization approach for top-k queries on uncertain data, *International Journal of cooperative Information systems,* vol. 27, 2018, pp. 1741002.