International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT23903140

Comparative Studies on Intelligent Swarming Network (iSWAN) Geno-Generative Algorithm and Top-K Query Processing Algorithm

Nlerum Promise Anebo*, Obasi Emmanuela C. M
Computer Science and Informatics Department, Federal University Otuoke, Bayelsa State, Nigeria

ARTICLEINFO

Article History:

Accepted: 10 June 2023
Published: 24 June 2023

Publication Issue
Volume 9, Issue 3
May-June-2023

Page Number
581-594

ABSTRACT

This paper proposed an enhanced Top-k query processing in a real time distributed database system. The system employs a Particle Swarm Optimizer (PSO) based Geno-Generative iSWAN Model technique that enhances and allows multi-task concurrent query processing in a real time co-simulation data acquisition platform and as part of refinement to an existing Top-k query processing Technique. In this paper, the proposed system is compared for efficiency with the Top-K Query Algorithm, which is emerging as an alternative to more conventional technique for real time query processing in distributed databases. Dynamic simulations were performed with a real time small testbed comprising of physical and non-physical devices to test and evaluate the performance and efficiency of the two systems. Considering the estimated and expected temperatures, the result of simulation study proves that the Intelligent Swarming Network (iSWAN) Geno-Generative Model is more preferred over Top-K Query Algorithm due to its 70% accuracy over the Top-K Model, which reported a lower accuracy level of 40%.
Keywords: Geno-generative, database Intelligent Swarming Network (iSWAN) Geno-Generative Model, sensor, swarm, Top-k

I. INTRODUCTION

The Geno-Generative Model for improving query processing in information management systems is a recent approach that promises better query plans. A good query plan is needed for an improved performance of big data in real world environments. Queried data in real-time industrial environments is typically streaming and unstructured in nature. The data is typically obtained sequentially over time. In most parts of the world, data is conventionally obtained by manual entries; though some technical data is obtained using specialized metering, but this form of entry is rarely automated. The conversion of useful data into information is a task mostly done by trained experts. Information is only useful if it can be converted into diverse beneficial forms, and in a timely manner. Speech synthesis from text, audio-visual systemic converters, and sensor-aware acquisitioned systems can result in very robust information systems [1]. The ultimate aim of any information system is to obtain relevant messages or codes from noisy or contaminated data distributions. The origins of information theory could be traced to the works of Shannon [2] and has deep probabilistic roots. Geno--Generative models is inspired by k-Nearest Neighbor (KNN), and two core machine learning disciplines– Genetic Algorithms and Generative Models [3]. While Genetic Algorithms is a biological motivated model based on human genetics and evolution, generative models are basically statistically driven models used to probabilistically define a data generating process which may be stochastic or not. The study involves querying of large and unstructured database system in an organization. It is use in an ambiguous query and also incorporating preferences. In particular, this research work seeks to proffer a more optimal solution to the Top-k query response approach proposed in [4] by using the concept of intelligent swarming particle networking (iSWAN) in a distributed database context. It also compared the efficiency of Top-K Query Algorithm, which is emerging as an alternative to more conventional technique for real time query processing in distributed databases with an Intelligent Swarming Network (iSWAN) Geno-Generative Algorithm.

II. REVIEW OF RELATED LITERATURES

In statistical classification, including machine learning, two main approaches are called the generative approach and the discriminative approach. These compute classifiers by different approaches, differing in the degree of statistical modelling. Terminology is inconsistent, but three major types can be distinguished, following [5]. A discriminative model is a model of the conditional probability of the target Y, given an observation X. Classifiers computed without using a probability model are also referred to, loosely as "discriminative". The distinction between these last two classes is not consistently made; Jebara refers to these three classes as generative learning, conditional learning, and discriminative learning [6]. He distinguished two classes, calling them generative classifiers (joint distribution) and discriminative classifiers (conditional distribution or no distribution. The education sector in Nigeria has experienced a huge growth especially in the area of available data. Furthermore, the emergence of social networking has continued to boost the growth of data usage in Tertiary Institutions in Nigeria. However, this trend of growth comes with the problem of big data management. The study also views big data as the joining of data management concepts that Tertiary Institutions store, organize, manage and manipulate large amount of datasets and still be efficient in speed, so as to gain the right insights. Data Mining can also be described as the process of extracting vital information from voluminous data [7]. Big data technologies have great impacts on scientific discoveries and value creation. Structured (numerical) and unstructured (textual) are two main types of data forms in big data[8]. Intelligent Swarming Networks (iSWANs) present an alternative generative model solution that mimics the functions of group behavior found in many natural life systems e.g. social insects and animals. The idea behind the iSWAN approach is to use the principle of evolution basing on exploitation and exploration to model the solution. In this thesis, the iSWAN approach based on Particle Swarm Optimizer (PSO) is described and applied in finding an optimal Geno-Generative Query processing solution. PSO is a very useful alternative strategy for solving optimization problems. Idea of swarm particles was first introduced by Kennedy and Eberhart[9]. The randomization step is typically influenced by generative updates of a weighted velocity vector in conjunction with randomized position states – the states being a sum of the difference between random previous best position states with previous (initial) states at both local and global levels[10].

Typically, the operations that follow are Newtonian and may be described by a velocity update calculation as in (1):

$$vel_{ij}(new) = w * vel_{ij}(old) + c_1 rand_1(pbest_{ij}(old)) - pos_{ij}(old) + c_2 rand_2(pbest_{ij}(old)) - pos_{ij}(old) \quad (1)$$

New positions are updated by adding the velocity updates obtained in (1) to its

old position as in (2):

$$pos_{ij}(new) = pos_{ij}(old) + vel_{ij}(new) \quad (2)$$

where,

$rand_1$, $rand_2$ = random number between 0 and 1

w = inertia weight

c_1 = coefficient of self-recognition

c_2 = social coefficient

c_1, c_2 = 2.

Kangseok in 2010[11], adopted the architecture for scalable, distributed database system built on multicore servers. The author presented performance results for shape similarity queries, which is extremely, time intensive with traditional architectures. According to the work; many scientific fields routinely generate huge datasets. In many cases, these datasets are not static but rapidly grow in size. Patrick in [12] proposed a query processing based on compressed intermediates. The author presented a balanced query processing based on compressed intermediates to improve query performance. Furthermore, he provided an overview of the important research challenges on the way to the objectives of the research. The author did a good job; however, he could not implement the discussed issue to a model for proper clarification and understanding.

III. MATERIALS AND METHODS

In the quest to developing an effective query system application, the authors employed an iterative object oriented design software engineering methodology known as Rational Unified Process (RUP). Rational Unified Process is a software Engineering process that provides a disciplined approach to assigning tasks and responsibilities within a development organization. It is an iterative software development process framework created by the Rational Software Corporation. The RUP aims at ensuring the production of high-quality software that meets the needs of its end users. A RUP activity creates and maintains models and emphasizes the development and maintenance of models-semantically [13]. RUP is a guide on to effectively use the Unified Modelling Language (UML)[14].

3.1 TOP-K QUERY PROCESSING ALGORITHM

Top-*k* query processing algorithms are applied in various fields, such as pervasive systems, contet-aware systems, e-commerce, and web search engine [4]. The existing system as proposed in Hyeong-Jin et al (2022)[4] is based on a centralized Collaborative Top-k query filtering for wireless sensor networks (see Figure 1).

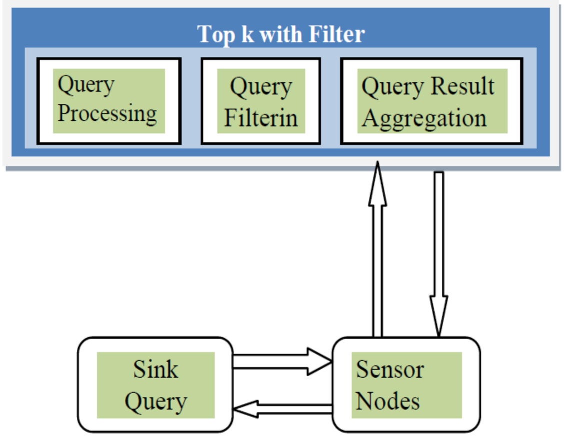

Figure 1: Top-K Query Processing Architecture

The model comprises child nodes and a parent node and may occur in two-topologies; The Top-k query with median-filtering is evaluated by the authors. In particular, a key objective of the Top-k query technique used in industrial real time big databases such as in WSNs is finding the k-sensor nodes with highest sensing values. Thus, from the point of view of the QO Encoding problem described earlier, the query based data system is recast as a Top-k query distributed data [15] search system that seeks to optimally find the sensor nodes transmitting at critically peak reference values.

Some key instances are finding the highest temperature values to detect a fire outbreak in a production plant; finding the highest-pressure points in a gas pipeline that may lead to explosion.

Topology 1: - Top-k query with median-filtering

In this case, the nodes are conducted using the following modules as follows:

i. Sensor Child (Base) Node module
ii. Sink Query Parent (Server) Node Module
iii. Top-k values.

The Sensor Child (Base) Node Module, Sink Query Parent (Server) Node Module and the Top-k values Module are the key component of the Architectural concept.

Sensor Child (Base) Nodes module:

In this node, the following query functions are accomplished:

1. Child node(s) compute the median value of the top-k query response readings.
2. Child node(s) send the computed median value readings to their respective parent node and wait for a filter value from parent.
3. Parent node receives median values of the top-k query readings from all child nodes
   i. Parent computes the median of the received median values.
   ii. Parent sends its computed median to child nodes as filter value.
4. Child node(s) receive filter value from their associated parent and returns all the values greater than or equal to the filter value back to parent node.

Sink Query Parent (Server) Node Module

1. Parent(s) sorts the median sequence (received median values from child node) in decreasing order.
2. Parent(s) calculate a median value from the median sequence.
3. Parent(s) broadcast (transmit) the computed median value to each child nodes – to be used as filter value.
4. Parent(s) identify the top-k readings based on the child node readings received and their own readings.

In general, the Parent does some sort of screening to obtain a useful statistic of the queried sensor readings.

Top-k Values Module

This module stores the screened out (filtered) max-values and the corresponding matching sensor with the aforementioned max-value(s).

3.2 INTELLIGENT SWARMING NETWORK (iSWAN) GENO-GENERATIVE ALGORITHM

The proposed Intelligent Swarming Network (iSWAN) Geno-Generative Model provides a justification for overcoming the drawbacks of the existing Top-k Query Technique while accounting for recent developments in the state-of-the-art. With respect to optimization of Top-k queries, the proposed model uses a generative approach to evolve candidates' solutions to the Top-k query problem. Figure 2 describes the new systems' modules and the objective function formulation.

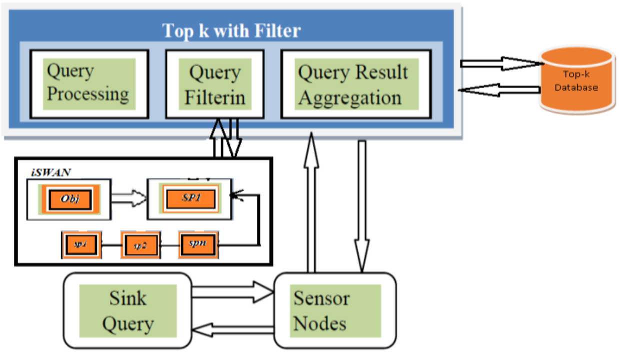

Figure 2 : iSWAN Geno-Generative Architecture

iSWAN Geno-Generative algorithm performs optimization of the Top-k query values by carrying out swarming particle optimization which leads to a more precise and optimal value. It uses the following primary modules as described in the existing Top-K query system, with the exception of Sensor child nodes' modules, which use an optimization sub-module to perform optimal query filtering in a real time distributed data processing system.

i. Sink Query Module

This module requires the development of the Query-Order formulation (QOF) program which accounts for the Query-Order (QO), the QO Representation (QOR) and the QO Encoding (QOE). The emphasis of the QOF is to allow for structured distributed query processing in real time databases. Thus, with the QOF, the aspect of data aggregation, representation and encoding is fully captured in a dynamic context, which is pervasive in nature

Query-Order (QO):

In real time big data transmission systems, the query order (QO) is typically a function of the sensor transmission rates in a distributed network of acquisition sensors (SAN). It defines the number of sensors (or sensor points) that must be activated in addition to some specified transmission rates. Thus sensors transmit data packets to the base station based on the QO (see Figure 3). The labels S1 to SN as shown in Figure 3, are sensor nodes, the dotted arrows are base station queries and the boldened arrows the responses generated from the sensors at uncertain orders.

Query-Order Representation (QOR):

The QO representation defines the following key structures (Sarode & Nandhini, 2018):

1. The number of sensors using a particular transmission channel
2. The data transmission capacity or rate

The overall representation is based on the eqns (3 - 4):

$$S_i = \lfloor z_l \rfloor, \quad 0 \le l \le N_L - 1, \quad S_i \in S \qquad (3)$$

$$t_{ij} = \| z_l - S_i \| \qquad (4)$$

Where,

$\lfloor z_l \rfloor$ = queried sensor label

N_L =number of possible sensor states

L = integer index label, at state N

i = integer index label, at state S

Query-Order Encoding

In order to aggregate sensor queried data from two or more channels, an encoding scheme is needed. In this paper, the encoding scheme is attained by forming sensor solution variables using the scheme in (Figure 3). This scheme assumes a representative variable $S_i t_j$, and constrained by $t_j \ge t_{ij}$,

where:

$S_i t_j$ = a solution variable representing *ith* sensor transmitting *j* amounts of data packets

t_j = the amount of data packets transmitted by Sensor *i* at time *j* for a given channel

t_{ij} = the amount of data packets transmitted by Sensor *i* from time *i* to *j* for a given channel

In (Sarode & Nandhini, 2018), the channel is typically defined as 1/4th of its capacity but need not be strictly so in practice – as in reality, the capacities will vary sparsely due to environmental constraints, threshold etc.

Thus, the Authors proposed to describe the channel capacity by a sparsity criterion rule as follows in Algorithm 1.

Algorithm 1: Query-Order Encoding
Step 1: Initialize Sensor *S*, Sparse Sensor *S'*, instance *i*, threshold, *th*, Vj
Step 2: Normalize S: S ← {0, 1}
Step 3: For each Sensor, at instance *i*
Step 4: If $S_{(i)} > th$
$S'_{(i)} = $ Si
End
Step 5: Obtain the cardinality in S: $n(S) \leftarrow S_{count}$
Step 6: Set Capacity as C_i: $S_{count} \leftarrow C_i$
s.t.:
$\sum_{j=1,\, i=1}^{C_j, S_i} V_j^{(i)} > T_i$ //Channel capacity data availability constraint
where,
T_i = the data to be transmitted to the base station by *ith* sensor
V_j = the data to be transmitted by the *ith* sensor through *jth* channel

ii. Sensor Nodes module

This is the child modules that represent real world physical parameter black boxes that can support query retrieval, response value generation and data parsing. In this node, the following query functions are specifically\accomplished:

1. Child node(s) compute the median value of the top-k query response readings.
2. The computed median value readings of Child node(s) are captured by iSWAN agents.
3. Agent node(s) receives median values of the top-k query readings from all child nodes by swarming particles
   a. Agent node(s) perform adaptive filtering by computing the optimized median index of the received median values using group behavior and an objective (cost) function.
   b. The iSWAN system reports the generated optimal (fitted) median of each child node.

iii. iSWAN module

This represents the intelligent swarming network used in the solution of the Top-k response filter values. The sub-modules *sp1, sp2 … spn* represent swarming agents in the solution search space that scans the sensor nodes via the Top-k with filter module interface. The main optimal solutions are carried basing on the model equations (2) and (3). The solution points are obtained using the swarming particle optimizer algorithm as shown in Algorithm 2 (Sumathi & Surekha, 2010).

Algorithm 2: iSWAN Main solution Process
Step 1: Initialize the size of the swarm particles, n
Step 2: Randomly initialize swarm particle position, *x* and velocities, *v*
While stopping criterion is false **do**
Step 3: t = t+1 //Increment iteration counter
Step 4: Compute initial fitness value of each particle

$$x* = \arg\min_{t-1}^{n}\begin{pmatrix} f\left(x*(t-1)\right), f\left(x_1(t)\right), \\ f\left(x_2(t)\right), \dots f\left(x_t(t)\right), \dots f\left(x_n(t)\right) \end{pmatrix};$$

for *i*=1 to n
Step 5:

$$x_t^{\#}(t) = \arg\min_{t-1}^{n}\left(f\left(x_t^{\#}(t-1)\right), f\left(x_t(t)\right)\right)$$

for *j* = 1 to Dimension
Step 6: Update the jth dimension of x_t and v_t // $x_t = pos$, $v_t = vel$
Execute (2)
Execute(3)

End *for*
End *for*
End While

iv. Top-k Query Filter Module

This represents the topmost structure of the system and serves as the container module that accommodates the rest processing functions or modules. Thus, it represents the main functional class or module in the OOP paradigm. The following key sub-functions are performed in this module: Query Processing (QP), Query Filtering (QF) and Query Result Aggregation (QRA). In QP, the queries (sensor data calls) are first initiated by generating a subset of QF and QRA commands. The QF commands follows from the iSWAN and works in tandem with the QRA for each sensor data call. Specifically, the QF sub-module does the job of extracting the queries with the top or maximum reading considering the various sensor nodes. On the other hand, the QRA sub-module performs the specific function of sensor data aggregation of all queries in a streaming data matrix or store-of-value. This is achieved by using relevant function data calls in near real time. All query results are stored in the Top-k database; retrievals are also performed on this database.

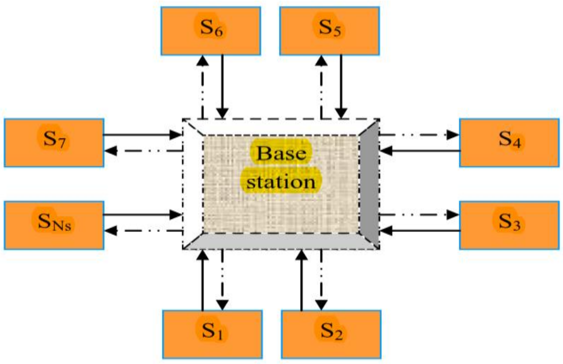

Figure 3 : Architectural view of a typical data aggrgation model formed from the QO (Source: Sarode & Nandhini, 2018).

Objective Function Formulation

The objective function plays a critical role in the design of any optimization system. It defines the fitness or cost criterion that needs to be minimized in order to attain the required solution value (points or positions). In this research study, the Median Error Difference (MED) of the pre-computed medians between the child nodes and their combined iSWAN agent median state is considered as objective function. This difference is computed as follows in Algorithm 3:

Algorithm 3: iSWAN Filter Query Objective Function

Step 1: Initialize Query Optimization Parameters – setting the lower and upper bounds, population size and number of generations

Step 2: Compute Median Error Difference (MED) Calculation:

$$\eta = sum\left(\kappa_{agent(ref)} - \kappa^{i}_{child}\right) \qquad (5)$$

where,

$\kappa_{agent(ref)}$ = agent-based median used as reference

κ_{child} = set of child median values at time step i

We determine if a data transmission is successfully and correctly (maximally) terminated in a sensor node using eqn(3.4) as:

$$T_1 = \min(\eta) \qquad (6)$$

T_1 = a fitness criterion (parameter) for top-k query data transmission validation i.e. if the absolute difference between received child nodes data transmission and a reference parent node standard meets the expected least minimum levels, it is selected; otherwise it is not selected. This is done in an evolutionary manner based on swarming particles.

System Parameters, Input and Output Specifications

The motivating system parameters for actualizing the geno-generative solution are as described in Table 1. These parameters are used to evolve the solution space by varying all possible indices of sensor values while searching for the best possible top-k query candidate sensor. Several of the parameters are fixed while two (the number of generations and the population size) change as the simulation progresses. These parameters were defined based on related studies and found suitable in this context. The simulation process in conjunction with these parameters is done dynamically using the Matlab-SIMULINK which gives an insight into the real time capabilities of the proposed system.

Matlab-SIMULINK simulation interface showing sensor signal captures and query aggregation is as shown in figure 4 and MongoDB Compass GUI Server Interface after first launching in figure 5. The input and output specification are illustrated in tables 2 and 3 respectively. Under the attribute field we have some of the input and output data while under the attribute type float is used as data type. Table 2 shows the database design for the proposed model, which contains four attributes - field name, data type, data range and the corresponding unit.

TABLE 1 : Geno-Generative System Parameters

Parameter	Default Used Value
Population size	5
Maximum number of iterations (Generations)	50
Constriction coefficient	2.05
Inertia weight damping ratio	1

TABLE 2 : Input Specification

Field	Type	Range	Unit
Temperature	Float	20 – 40	°C
Noise Level (Pressure)	Float	10 – 400	dB
Light Intensity	Float	1000–2500	Lumens

Table 3 : Output Specification

Field	Type
Median Temperature	Float
Median Noise	Float
Light Intensity	Float

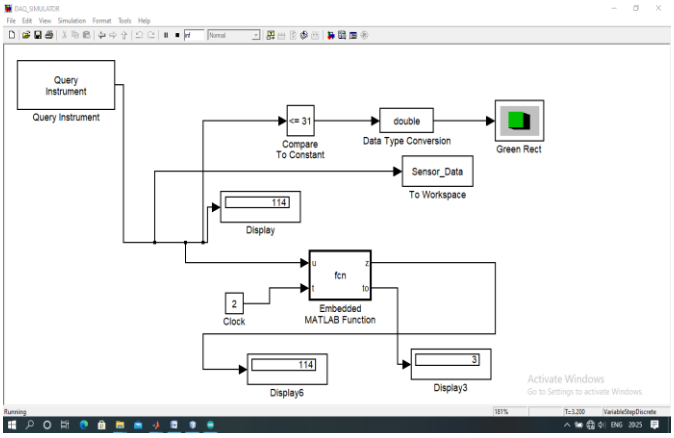

Figure 4: Matlab-SIMULINK MATLAB running simulation program showing sensor signal captures and query aggregation

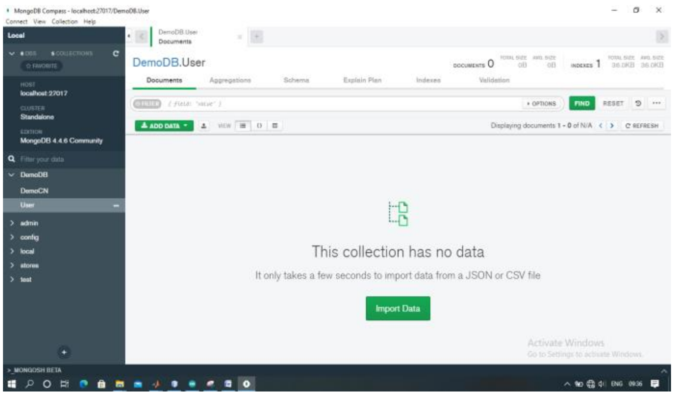

Figure 5 : After first launching

IV. RESULTS AND DISCUSSION

In this paper work, the authors used a two-stage approach in conducting a co-simulation of experiments with a laboratory-type testbed. In the first stage, the query response of the testbed is evaluated on the basis of the existing Top-K Query system model described earlier in previous section (see Tables 1 to 3). The second stage employs the proposed iSWAN technique for evaluating the testbed. In particular, the second stage describes a dual-tier boundary constraint for each considered sensor; a Dynamic Lower Bound (DLB) constraint and a Static Lower Bound (SLB) constraint (see Table 4 to 6).

TABLE 4 : Top-k Query Response for child sensor node 1

loc_id	Tkv (ºC)
1	27
7	32
19	23
25	26
28	38
34	20
49	36

TABLE 5:
Top-k Query Response for child sensor node 2

loc_id	Tkv (ºC)
2	28
8	32
17	20
20	23
26	26
29	37
32	20
35	21
50	36

TABLE 6:
Top-k Query Response for child sensor node 3

loc_id	Tkv (ºC)
3	24
9	35
18	21
21	23
24	20
27	29
30	36
33	21

This experiment applies the Top-k query approach based on median filtering. The simulation results (Top-k query temperature values) using testbed as shown in Tables 1 to 3 for the child sensor nodes 1, 2 and 3 respectively, shows the filtered response locator id (loc_id) and the corresponding Top-k query values (Tkv). The loc_id represents the positions of the Tkv in the child node query list.

TABLE 7 : Top-k Query Response for iswan Geno-Generative System with SLB

Trial No.	Cost	Sensor 1 (ºC)	Sensor 2 (ºC)	Sensor 3 (ºC)
1	25.0000	25.0000	0.0000	25.0000
2	20.9995	0.0000	21.0005	21.0000
3	18.9995	0.0000	19.0005	19.0000
4	15.4890	15.5072	0.0038	15.5000
5	23.0000	23.0000	23.0000	0.0000
6	19.0000	19.0000	0.0000	19.0000
7	19.9990	20.0000	20.0006	0.0005
8	19.9958	0.0000	20.0042	20.0000
9	19.9994	0.0000	19.9997	20.0000
10	23.0000	23.0000	23.0000	0.0000
11	32.9985	0.0000	33.0000	32.9992
12	38.9771	0.0000	39.0229	39.0000
13	21.0000	21.0000	0.0000	21.0000
14	27.9998	0.0000	27.9999	28.0000
15	16.0000	16.0000	0.0000	16.0000
16	16.9986	17.0014	17.0000	0.0000
17	22.4986	22.5014	22.5000	0.0000
18	41.9784	41.9892	0.0000	42.0000
19	25.9998	0.0000	25.9999	26.0000
20	23.9993	0.0000	24.0007	24.0000

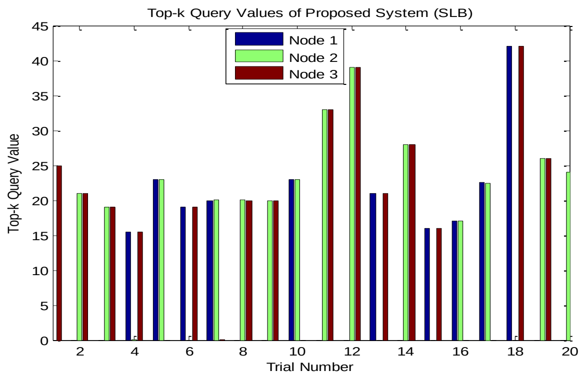

Figure 6 : Response of the iSWAN System under Dynamic Lower Bound

This experiment builds on the principles of optimization following evolutionary computing. It presents the results considering the iSWAN optimization decision variables (DV) constraints at Static Lower Bound (SLB). After the run of 20 trials using the testbed as shown in Table 8, simulation results shows the fitted Top-k query temperature values for the child sensor nodes 1, 2 and 3 respectively; it also shows the fitness function value computed by the PSO search agents at the attained temperature states. Figure 6 shows the comparative graphical query response of each sensor node as obtained by the simulation program.

Table 8: Top-k Query Response for proposed system with DLB

Trial No.	Cost	Sensor 1	Sensor 2	Sensor 3
1	1.5000	18.0000	16.5000	18.0000
2	1.0000	25.0000	27.0000	22.0000
3	0.5000	27.0000	26.0000	24.5000
4	0.5026	0.9068	0.1603	0.7849
5	1.5000	15.5000	13.0000	15.0000
6	0.4015	0.8020	0.7046	0.2057
7	1.0000	19.0000	19.0000	18.0000
8	1.0000	30.0000	29.0000	27.0000
9	0.5000	17.5000	15.0000	16.5000
10	0.7509	0.8268	0.8462	0.0565
11	2.5000	25.0000	23.0000	18.5000
12	0.5573	0.8092	0.8871	0.1741
13	0.5250	0.2893	0.8590	0.9038
14	3.0000	17.0000	17.0000	14.0000
15	1.0000	33.0000	40.5000	47.0000
16	0.5000	45.5000	48.0000	42.5000
17	0.7189	0.7944	0.8685	0.0014
18	0.5300	0.8842	0.9118	0.3267
19	2.5000	19.0000	22.0000	22.5000
20	0.6105	0.9395	0.1728	0.8614

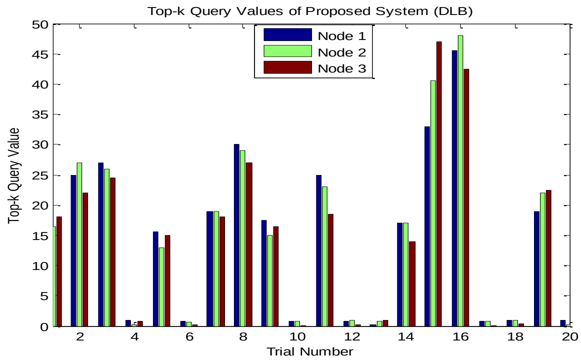

Fig. 7 : Response of the iSWAN System under Dynamic Lower Bound

In this experiment, simulations were performed in order to investigate the nature of using a Dynamic Lower Bound in the iSWAN optimization process. Consequently, the *min* function was used adaptively to extract the lower bounds prior to processing. The results of this experiment for 20 trial runs are as shown in Table 8. Figure 7 shows the comparative graphical query response of each sensor node as obtained by the simulation program.

4.1 COMPARISON EVALUATION OF iSWAN GENO-GENERATIVE MODEL AND TOP-K QUERY PROCESSING MODEL

The comparison evaluation of the two systems were carried after series of runs of simulations as shown in Tables 9 to 11 using the Dynamic Lower Bound (DLB) concept based on the maximum temperature value (T_{max}). For the Top-k model, since multiple values are predicted, the first filtered value is selected as the single best solution variable. The classification accuracies (CA) are reported in Table 12.

Table 9 : Results Comparison of Existing Top-k Query Algorithm and iSWAN Geno-Generative Algorithm Sensor 1

Trial No.	T_{max}(ºC)_existing	T_{max}(ºC)_proposed	T_{max}(ºC)_expected
1	29.0000	30.0000	30.0000
2	17.0000	34.0000	34.0000
3	29.0000	34.0000	34.0000
4	15.0000	34.0000	34.0000
5	36.0000	37.0000	37.0000

6	39.0000	0.7501	37.0000
7	39.0000	39.0000	39.0000
8	36.0000	33.0000	35.0000
9	28.0000	0.0620	36.0000
10	29.0000	40.0000	40.0000

TABLE 10 : Results Comparison of Existing Top-k Query Algorithm and iSWAN Geno-Generative Algorithm Sensor 2

Trial No.	T_{max}(ºC)_existing	T_{max} (ºC)_proposed	T_{max} (ºC)_expected
1	32.0000	34.0000	34.0000
2	16.0000	37.0000	37.0000
3	30.0000	37.0000	37.0000
4	38.0000	38.0000	38.0000
5	35.0000	35.0000	35.0000
6	38.0000	0.9047	35.0000
7	36.0000	36.0000	36.0000
8	36.0000	33.0000	35.0000
9	27.0000	0.5575	33.0000
10	42.0000	42.0000	42.0000

TABLE 11 : Results Comparison of Existing Top-k Query Algorithm and iSWAN Geno-Generative Algorithm Sensor 3

Trial No.	T_{max} (ºC)_existing	T_{max} (ºC)_proposed	T_{max} (ºC)_expected
1	32.0000	35.0000	35.0000
2	38.0000	38.0000	38.0000
3	31.0000	38.0000	38.0000
4	39.0000	39.0000	39.0000
5	37.0000	37.0000	37.0000
6	26.0000	0.2062	37.0000
7	17.0000	38.0000	38.0000
8	36.0000	33.0000	35.0000
9	29.0000	0.6510	35.0000
10	43.0000	43.0000	43.0000

TABLE 12 : Classification accuracies (CA) of Existing Top-k Query Algorithm and iSWAN Geno-Generative Algorithm at different sensor nodes

Sensor Node	CA(existing) %	CA(proposed) %
1	40	70
2	40	70
3	40	70
Average CA	40	**70**

The existing system results show some variability in top-k queried temperature values as shown in Table 4 - 6. For Sensor Node 1 (Table 4), the Top-k value (TkV) is 38ºC occurring at location id (loc_id) 28 while for Sensor Nodes 2 and 3, the TkV and corresponding loc_id are 37°C:29 and 36°C:30 respectively. Thus, it is immediately obvious that Sensor Node 1 is the critical one followed by Sensor Nodes 2 and 3. Thus, Sensor Node 1 must be given priority with respect to the other nodes.

From the simulation results using Geno-Generative iSWAN optimizer with SLB, the Sensor Node 1 TkV is 41.9892°C occurring at loc_id 18; also the least cost (cost) is obtainable at this solution point and is -41.9784. For the Sensor Nodes 2 and 3 Tkv, loc_id and cost are 39.0229:12: -38.9771 and 42.0000: 18: -41.9784 respectively. From these results, it can be clearly seen that Sensor Node 1 is the critical one; this is followed by Sensor Node 3 and then Sensor Node 2. Priority for conditioning should be given following the aforementioned order. This can also be clearly visualized in Figure 6, while the top-most bars are shared between Sensor nodes 1 and 3 only.

From the simulation results using Geno-Generative iSWAN optimizer with DLB, the Sensor Node 1 TkV is 45.5000°C occurring at loc_id 16; also the corresponding least cost (cost) obtainable at this solution point is -0.5000. For the Sensor Nodes 2 and 3 Tkv, loc_id and cost are 48.0000:16: -0.5000 and 47.0000: 15: -1.0000 respectively. From these results, it can be clearly seen that Sensor Node 2 is the critical one; this is followed by Sensor Node 3 and then Sensor Node 1. In particular, there are sub-optimality issues in the DLB as Sensor node 3 gave best cost as against that of Sensor nodes 1 and 2. Thus, priority for conditioning should be given to Sensor Node 2 followed by Sensor Node 3 and then Sensor Node 1. This can be clearly visualized in Figure 7 with the top-most bars shared between Sensor nodes 2 and 3 in that order.

Comparing the proposed Geno-Generative iSWAN optimizer system with the median based child-parent (client-server) approach, considering the expectation maximum (top-k) queried temperature values and at different sensor nodes.

For the Sensor Node 1, apart from the Trials 6, 8 and 9, the T_{max} for proposed system is equivalent to the expectation; however, in the case of the existing approach, an exact match occurred only at the 7th trial run. In particular, it can be observed that the proposed (iSWAN) system suffered a great deviation from the expectation at the 6th and 9th trials; this great deviation may be attributed to the stalling effect that may occur in the method of swarming particles.

For the Sensor Node 2, apart from the Trials 6, 8 and 9, the T_{max} for proposed (iSWAN) system is equivalent to the expectation. In the case of the existing approach, an exact match occurred at the 4th, 5th, 7th and 10th trial runs. Just as in Sensor Node 1, the proposed system suffered a great deviation from the expectation at the 6th and 9th trials. Thus, the competitive nature of the median based child-parent (client-server) approach is evident.

For the Sensor Node 3, apart from the Trials 6, 8 and 9, the T_{max} for proposed system is equivalent to the expectation. In the case of the existing approach, an exact match occurred at the 2nd, 4th, 5th, and 10th trial runs. Just as in Sensor Node 1, the proposed system suffered a great deviation from the expectation at the 6th and 9th trials. Thus, the competitive nature of the median based child-parent (client-server) approach is also evident. Figures 8 to 10 show graphically the situation of great deviations at sensor nodes 1-3 respectively query processing between Top-k Query processing model, iSWAN Geno-Generative Model and the expected query response values. As can be seen from the Table 12, the proposed system on the average outperformed the existing model with a CA of 70% over that of the existing system which returned a CA of 40% on the average.

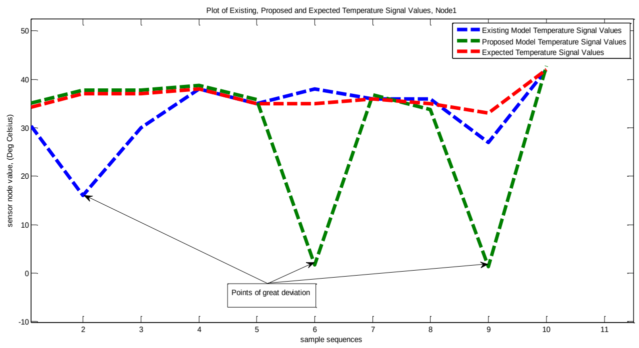

Figure 8 : Comparative results showing the sensor node 1 query responses

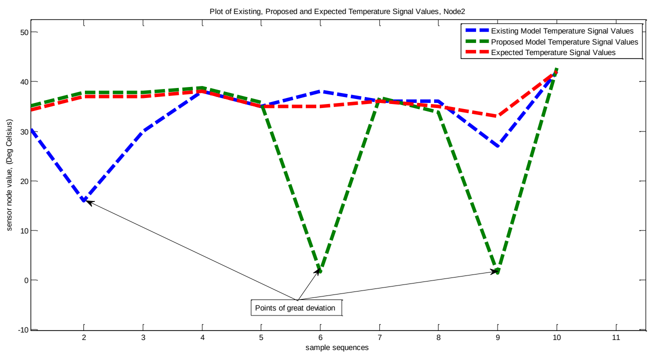

Figure 9 : Comparative results showing the sensor node 2 query responses

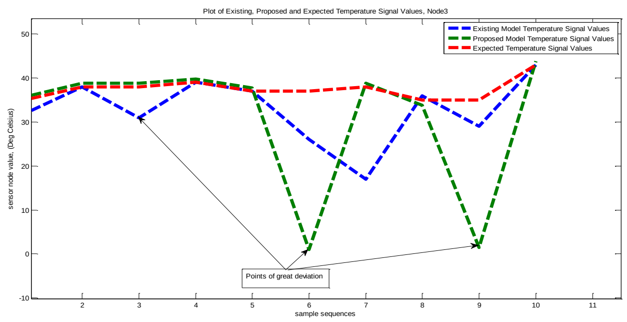

Figure 10 : Comparative results showing the sensor node 3 query responses

V. CONCLUSION AND RECOMMENDATION

This study presented an improved approach to query processing through the application of an iSWAN Generative model and MongoDb. The improved approach depicts evolutionary learning techniques for query processing which is also a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses. The most common evolutionary swarming learning method is based on particle swarms, which is used for efficient global search and optimization. The findings of this study are recommended to database administrators and analysts in real industrial environments, software developers and researchers with keen interest in the study area. This is because data management and request via queries is becoming complex day by day. In other words, the need for an improved query processing using Geno- generative model is highly indispensable.

VI. REFERENCES

[1]. Sharma, K.K, Vishnu S. (2011), Issues in Replicated Data for Distributed Real-time Database Systems, International Journal of Computer Science and Information Technologies (IJCSIT), 2(4), 1364 – 1371.

[2]. Bengio, Y.,Goodfellow, I.J. and Courville, A. (2015). Deep learning. An MIT Press book in preparation. Retrieved from http://www.iro.umontreal.ca/~bengioy/dlbook

[3]. Osegi, N. E., & Enyindah, P. (2015). GOEmbed: A Smart SMS-SQL Database Management System for Low-Cost Microcontrollers. African Journal of Computing & ICT, 8(2), 133-144.

[4]. Kakad, S., Sarode, P., &Bakal, J. W. (2013).

[5]. Analysis and Implementation of Top K Query Response Time Optimization Approach for Reliable Data Communication in Wireless Sensor Network. IJEIT, 3. Inernational Journal of Engineering and Innovative Technology 3(2), 202-211.

[6]. Moghaddam, B., Jebara T & Pentland, A. (2004). The Journal of Machine Learning Research 5, 819-844, 2004

[7]. Jebara T (1996) "Discriminative, Generative and Imitative Learning" PhD Thesis Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning, in partial fulfillment of the requirements for the degree of DOCTOR

OF PHILOSOPHY IN MEDIA ARTS AND SCIENCES
[8]. Bharati M. Ramageri (2010): Data Mining Techniques and Applications, Indian Journal of Computer Science and Engineering 1(4) 301-305
[9]. Acharjya, D.P., and Kauser A. P. (2016). A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools. International Journal of Advanced Computer Science and Applications. 7(2), 511-518.
[10]. Eberhart, R., & Kennedy, J. (1995, November). Particle swarm optimization. In Proceedings of the IEEE international conference on neural networks (Vol. 4, pp. 1942-1948).
[11]. Sumathi & Surekha Paneerselvam(2010). Computational Intelligence Paradigm book. First published, 2010 Imprint CRC Press. Pages 24. eBook ISBN 9780429191565.
[12]. Kangseok K. (2017), Architecture for Scalable, Distributed Database System built on Multicore Servers, International Journal of Computer Application (IJCA), 8(19), 14 – 19.
[13]. Patrick damme (2017). Query processing based on compressed intermediates pages 562–565, 2017.
[14]. Grady, B. and Addison-Wesley (1995) Object Solutions: Managing the Object-Oriented Project. 2(4), 15-23
[15]. Kruchten, P., Capilla, R. and Duenas, J. C. (2009). The Decision view's role in software architecture practice, IEEE software 26(2), 36-42
[16]. Zhang, W.; Liu, S.; Xia, Z. A (2022) A Distributed Privacy-preserving Data Aggregation Scheme for Smart Grid with Fine-grained Access control. J. Inf. Secure. Appl. 2022, 66, 103118.

Cite this article as :

Nlerum Promise Anebo, Obasi Emmanuela C. M , "Comparative Studies on Intelligent Swarming Network (iSWAN) Geno-Generative Algorithm and Top-K Query Processing Algorithm", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 3, pp.581-594, May-June-2023. Available at doi : https://doi.org/10.32628/CSEIT23903140
Journal URL : https://ijsrcseit.com/CSEIT23903140