International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Application of Machine Learning Algorithm for Query Processing in Distributed Database System

Nkanyi Nwadiogo Ginika[1], Onyekaba Ogechukwu[1], Chieme Gabriel Chukwujekwu[1], Chigbo Chukwuebuka Ikechukwu[1]
[1]Department of Computer Science, Anambra State Polytechnic Mgbakwu, Mgbagwu, Anambra State, Nigeria

***Abstract*—*This paper has presented an application of machine learning algorithm for query processing in distributed database (DDB) system. A database of a security agency was studied and a database structure was developed to manage the system. The developed database structure was modeled for distributed database management in handling information on crimes and other related security matters. A programme to compute query for proper and optimal processing using machine learning algorithm was developed. The machine learning algorithm used in this paper is the Iterative Dichotomizer 3 (ID3). The analysis and test results obtained shows that the developed system improved the query time of the distributed database system.*

Keywords*— *Algorithm, DDB, ID3, Machine-learning, Query.

I. Introduction

The combination of two different technologies used for data processing is known as distributed database (DDB). That is a database system and a computer networks. The main element of a database is the data which is basically the collection of facts about something. A DDB system is actually a collection of multiple, logically interrelated databases spread over a computer network. Data retrieval from different sites in a DDB is known as distributed query processing.

Database users in query processing generally specify what data are required rather than specifying the particular method or procedure to retrieve the required data. Hence, an essential aspect of query processing is query optimization. During query optimization, the database system optimizer finds a good way to execute the queries.

In distributed environment, query process is more complex and difficult compare to centralized environment because large numbers of parameters affect the performance of distributed queries. There may be fragmentation and/or replication, and looking at the fact that many sites are to be accessed, the response time of query may become very high

The cost model, larger set of queries, optimization cost and optimization interval are problems associated with distributed query optimization. In distribute query optimization, the objective is to execute queries efficiently in order to minimize the response time and the total cost of communication associated with query. Hence, it will be logical to consider the potential benefits in relation to their costs or time complexity. There are three major activities in the processing of DDB system. These are: database fragmentation; allocation of the fragmented database to the different sites using complex mechanism; and the execution of task.

An effective database fragmentation improves its performance. It is no doubt that fragmenting database results to increase in the complexity of design. However, it significantly impact performance and manageability. Many possible execution strategies are usually available in query system. The act of choosing a strategy the s suitable for processing a query is known as query optimization. This is expressed using a high-level language, for example, Structured Query Language (SQL) in relational data model.

The transformation of a high-level query into an equivalent lower-level query (relational algebra), is the main function of relational query processor. The transformation must achieve both correctness and efficiency. The lower-level query implements the execution strategy for the given query. The fact that data is geographically distributed in distributed relational database system, the processing of a distributed query consists of the following phases: local processing phase, reduction phase, and final processing phase.

In local processing phase, the following takes place: selections and projections. The reduction phase employs a sequence of reducers (i.e. semi-joins and joins) to reduce the size of the relations. For the final processing phase, all results relations are sent to the assembly site where the final result of the query is constructed. In order to process a distributed query for searching all relations directly to all the assembly site, an optimizer is used. Such an optimizer is the machine learning algorithms of Iterative Dichotomizer 3 (ID3) which functions to minimize the time and overall costs required for the applications to run in the network.

A. *Statement of the Problem*

When designing an optimizer for a distributed database (DDB), certain problems are encountered.

a) Problem related to the search space size: it is observed that the search space size in a centralized system becomes huge even moderately sized queries. The existence of relations copies at multiple sites and the number of execution sites contribute in making the size of the search space to be large.
b) Problem of identification: in order to be able to determine the minimum possible total execution time of the plan, it is necessary to identify inter-operator parallelism within a given execution plan.
c) Problem of updating database information: in a situation where data are partitioned at different location, there is

problem of updating database information associated with DDB.

B. Objectives of the Study

The main objective of this paper is to apply query processing in distributed database systems using intelligent method. Other specific objectives are:

I. Developing a programme to compute a query using an Iterative Dichotomizer 3 (ID3) machine learning algorithm.
II. Designing a DDB system for security agency
III. Developing rules and pattern using ID3 algorithm to minimize execution cost of queries to an acceptable minimum time in a DDB environment.

C. Significance of the Study

The application of DDB system has been tremendous. It has gain acceptance in different sectors such as the banking industries where financial transactions can be carried out at any platform within banking institutions. Also in telecommunication, where communication are done within different communication network. In this paper an intelligent query processing system is developed for DDB system for security agencies.

II. Review of Related Works

In this section, the previous works in literature are reviewed. With the development of computer network and database technology, distributed database is more and more widely used. And with the expanding application, data queries are increasingly complex, the efficiency requests are increasingly high, so query processing is a key issue of the distributed database system. In a distributed database environment, data stored at different sites connected through network. A distributed database management systems (DDBMS) support creation and maintenance of distributed database.

According to [1], the implementation of distributed database were hindered with lots of challenges perused in past researches, few of such include unreliable network technology, high cost of Computers, and insecurity among users and also some of the challenges encountered include; problems of distribution of resources, search and updating of resources. Kunal et al. [2] maintained that distributed database management systems support creation and maintenance of distributed database, and all database must be able to respond to requests for information from the user that is, process queries by following the processing algorithm approach. Distributed Database system does not only make data access faster, but also a single-point of failure is less likely to occur, and local control of data for users is provided [3].

Database consists of two or more data files located at different sites on a computer network. Because the database is distributed, different users can access it without interfering with one another. Collections of data (e.g. in a database) can be distributed across multiple physical locations. According to [4], a distributed database can reside on network servers on the Internet, on corporate intranets or extranets, or on other company networks. Replication and distribution of databases improve database performance at end-user worksites.

However, to ensure that the distributive databases are up to date and current, there are two processes: replication and duplication. Replication involves using specialized software that looks for changes in the distributive database. Once the changes have been identified, the replication process makes all the databases look the same. The replication process can be very complex and time consuming depending on the size and number of the distributive databases. This process can also require a lot of time and computer resources. Duplication on the other hand is not as complicated. It basically identifies one database as a master and then duplicates that database. Also according to [5] stated that distributed database system is cluster of distributed computers that are coupled with one another with the help of some communication media (like twisted pair, coaxial cable, fiber optics, satellite etc.) on which a database is allocated and placed. It is obvious that a query may have different equivalent transformation that lead to different resource consumption.

Some researchers have presented paper on distributed database query processing and optimization. In [6] a multi-colony ant algorithm for optimizing join queries in distributed database systems. It proposed a multi-colony ant algorithm for optimizing join queries in a distributed environment where relations can be replicated but not fragmented. Four types of ants collaborate to create an execution plan in the proposed algorithm. Hence, there are four ant colonies in each iteration. Each type of ant makes an important decision to find the optimal plan. In order to evaluate the quality of the generated plan, two cost models are used; one based on the total time and the other on the response time. In the work of [7] on dynamic programming solution for query optimization in homogeneous distributed databases, a “Multiple Query Optimization” (MQO) is deployed. This was used so as to reduce the execution cost of a group of queries by performing common tasks only once, whereas traditional query optimization considers single query at a time an optimal dynamic programming method for such high dimensional queries has the big disadvantage of its exponential order and thus we are interested in semi-optimal but faster approaches. In their work, [8] presented query execution and maintenance costs in a dynamic distributed federated database, and stated that the cost of query evaluation in a dynamic distributed federated databases (DDFD) depends on the topology connecting the database nodes together. Different topologies provided opportunities to adopt a variety of query optimization strategies and topology also influences the efficiency of these strategies. It described a number of strategies to optimize join queries and then derive cost estimation formulae. The costs of maintaining these topologies were also formulated and compared.

A typical distributed databases overview is presented in Fig. 1. The environment that exist in a distributed database environment are either homogenous or heterogeneous, this depends on the nature of database in their respective node. When the same DBMS is at each node (homogenous), and potential different DBMS at each node is (heterogeneous).

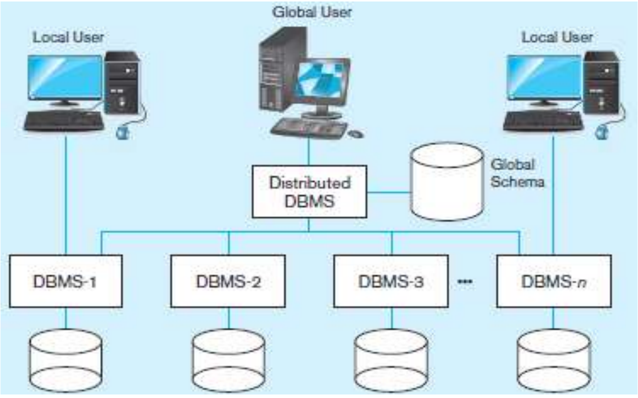


Fig. 1. Heterogeneous distributed databases.

III. System Design and Method

The main conceptual problems encountered when designing an optimizer for a distributed database is highlighted. First the size of the search space, and secondly the identification of the inter-operator parallelism opportunities within a given plan in order to be able to determine the minimum possible total execution time of the plan accurately using the proposed system.

A. System Analysis and Design

A multi-step process that is based on data structure, software architecture, Procedural details (algorithms etc) and interface between the modules. Also before coding begins, the design process translates the requirements into the representations of the software that can be assessed for quality [9]. It suffices to simply put system analysis and design as the entire process taken to analyze a system, define problems, state specification for the system, and provide the most realizable solution to a given problem.

The step-by-step procedure employed in structuring, planning, and controlling the process of information system development is known as system development methodology (SMD). Many of such framework has be developed over the years, with each having its known strength and weakness. For instance, Object Oriented Analysis and Design Methodology (OOADM) in the analysis of intelligent methods of query application in a distributed database system were adopted for Nigeria Security Agency.

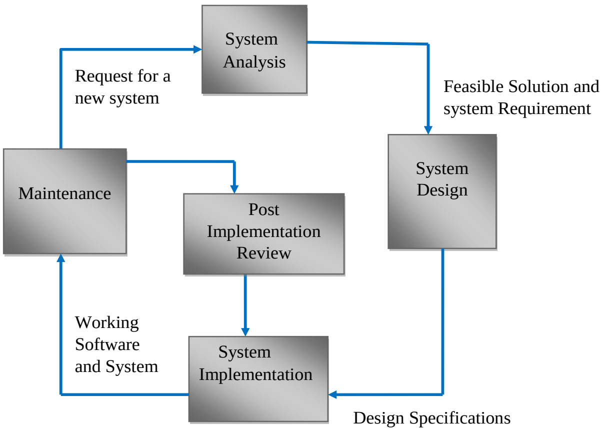


Fig. 2. Basic steps of system design using OOADM.

The process of defining problem in terms of real world object with which the system must interact, and the software objects employed in exploring various alternative solutions, is known as object oriented analysis (OOA). Also, the process of defining components, interfaces, objects, classes, attributes, and operations that will satisfy the requirements is known as object oriented design (OOD). Fig. 2 shows the basic steps of system design using OOADM.

This methodology feature Unified Modelling Language (UML): Case diagram, active diagram and the architectural design of the system.

B. Duties of Law Enforcement Agency UML Diagram

UML diagram for law enforcement agency duties show the public order and safety activities that include agency duties and administration and the operation of law courts and prisons. The duties include traffic regulation, maintenance of law and order, and provision of equipment and supplies for security work (such as vehicles, aircraft, vessels and docks). This dataset also includes the statistics generated in the process of the administration and operation of civil and criminal courts, tribunals and judicial system. Also incorporated are the statistics generated by the rendering of judgment and interpretation of the law including arbitration of civil actions, prison administration and provision of correctional services (incarceration and rehabilitation services, e.g. jails). Since official crime statistics cover only those reported to the law enforcement agency, these data only represent a proportion of offences committed. The UML diagram for security agency duties is shown in Fig.3 represents the functionalities of the system.

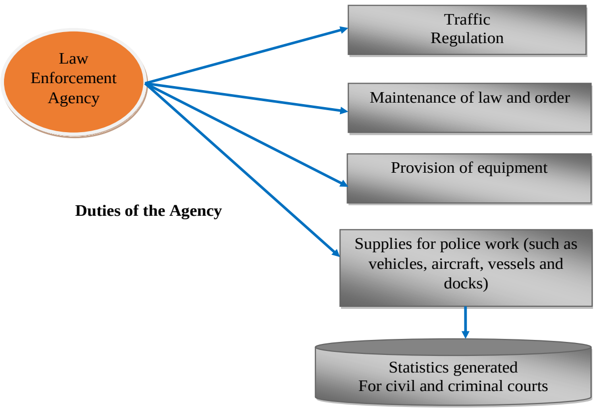

Law Enforcement Agency
Traffic Regulation
Maintenance of law and order
Provision of equipment
Duties of the Agency
Supplies for police work (such as vehicles, aircraft, vessels and docks)
Statistics generated For civil and criminal courts

Fig. 3. UML diagram of a law enforcement agency.

C. Data Collection UML Diagram

Unified Modelling Language (UML) diagram of data collection consists of data generation, station level, divisional headquarters, command headquarters, force headquarters, and zonal headquarters. Fig. 4 shows a diagram of UML for data collection.

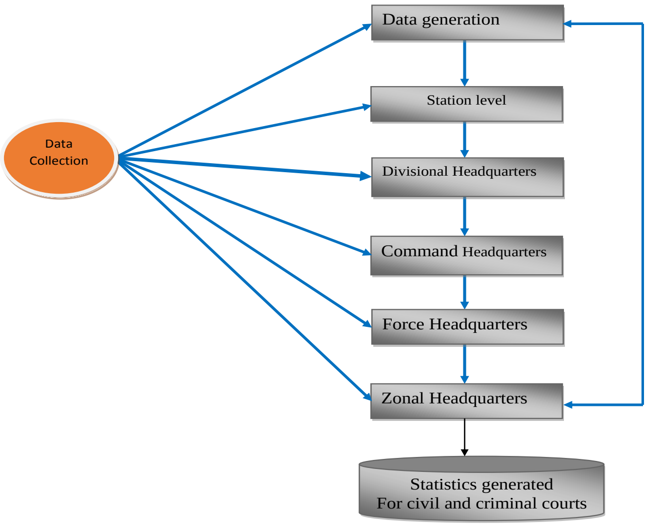


Fig. 4. A typical UML diagram for data collection.

D. *Design Architecture*

A typical intelligent decision query application for distributed database systems for security agency is presented in Fig. 5.

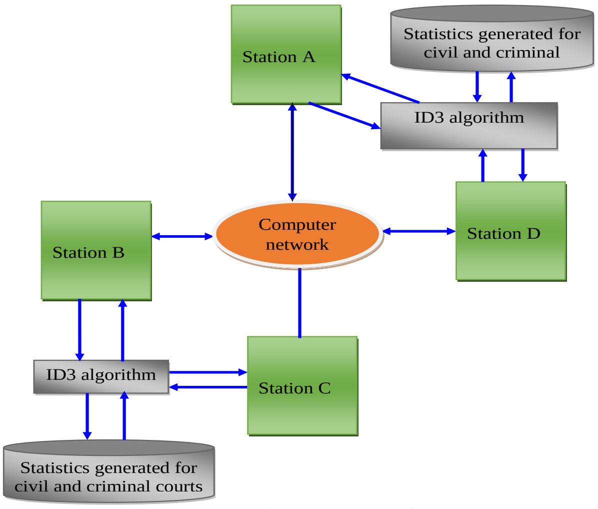

Station A
Statistics generated for civil and criminal
ID3 algorithm
Station B
Computer network
Station D
ID3 algorithm
Station C
Statistics generated for civil and criminal courts

Fig. 5. Design architecture of a DDB system for security agency.

E. *Optimization of Query in Distributed Database*

As a result of the low cost but high performance of personal computer (PC), hardware and high speed Local Area Network (LAN)/Wide Area Network (WAN) technologies, distributed database (DDB) systems has become attractive research where query optimization is greatly considered. Efficient processing is required by queries on DDB, and this ensures optimal query processing strategies. Fig. 6 shows a high level model for the designed distributed query optimization system.

A query optimizer is seen as the most essential element of a given database management system.

F. *Algorithm Development*

The algorithm developed in this paper is based on three decision generation using Iterative Dichotomizer 3 (ID3). The ID3 was chosen due to the fact that it makes simple and efficient tree with the smallest depth. It uses two concepts for generating a tree from top to down. These are Entropy and information gain.

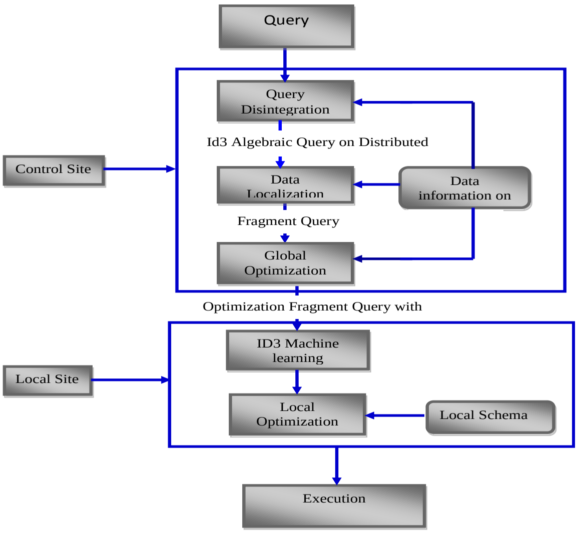


Fig. 6. A high level model for the designed distributed query optimization system.

Entropy $H(s)$ is the amount of uncertainty in the data set, S , that is, it characterizes the set S .

$$H(S) = -\sum_{x \in X} p(x) \log_2 p(x) \qquad (1)$$

where S is the current (data) set for which entropy is being calculate (it changes every iteration of the ID3 algorithm). X is a set of classes in S. $p(x)$ is the proportion of the number of elements in class x to the number elements in set S. when $H(S) = 0$, the set S is perfectly classified. That is all elements in S are of the same class.

Entropy is computed for each remaining attribute in ID3. To split the set S on this iteration, the attribute with smallest entropy is used. As the entropy gets higher, the potential to improve the classification gets higher.

Information gain IG(A) is the difference in entropy from before to after the set S is on an attribute A. That is to say, how much uncertainty in S was reduced after splitting set S on attribute A (Alex et al).

$$\mathrm{IG(A,S)} = H(S) = \sum_{t \in T} p(t) H(t) \qquad (2)$$

where $H(S)$ is the entropy of set S, T is the subsets created from splitting set S by attribute A such that $S = \bigcup_{t \in T} t$. $p(t)$ is the proportional of the number of elements in t to the number of elements in set S. $H(t)$ is the entropy of subset t.

A summary of the ID3 algorithm steps is:

I. Compute the entropy of every attribute using the data sample S

II. Split the set S into subsets using the attribute for with minimum entropy (or equivalent, with information gain maximum)

III. Create a decision tree node containing that attribute
IV. Recur on subset using remaining attributes.

- ID3 pseudo code

```
function ID3 (I, 0, T) {
/*I is the set of input attributes
*O is the output attribute
*T is a set of training data
**function ID3 returns a decision tree*/
if (T is empty)
{
return a single node with the value “wrong”;
                                    }
if (all records in T have the same value for O) {
Return a single node with that value;
                    }
if (I is empty)
   {
return a single node with the value of the most frequent value of O in T;
          }
/* case where we can't return a single node */
Compute the information gain for each attribute in I relative to T;
let X be the attribute with largest Gain(X, T) of the attributes in I;
Let {x_j| j = 1,2, .., m} be the values of X;
Let {T_j| j = 1,2, .., m} be the subsets of T when T is partitioned according to the value of X;
Return a tree with the root node labelled X and arcs labelled $X_1$, $X_2$, $X_3$....$X_m$, where the arcs go to the trees ID3(I-{X}, O, T_1), ID3(I-{X}, O, T_2)...
ID3 (I-{X}, O, T_m);
}
```

G. Database Design

The database comprises a set of records where each one of them is defined over a set of attributes. Tables 1, 2, 3, 4 and 5 show different database structures for the design.

TABLE 1. Structure of database design.

Field Name	Data Type	Length
Sex	VarChar	5
Offences	Text	25
Age range	Varchar	25

TABLE 2. Structure of main database.

Field Name	Data Type	Length	Description
Information about prison	VarChar	20	
Information about inmate	VarChar	25	
Information about accused persons	VarChar	25	
Database information of security agent	VarChar	28	

TABLE 3. Structure of database for information about prison.

Name	Data Type	Length	Description
Prison identity (ID)	VarChar	12	
Password	VarChar	8	
Prison Name	VarChar	13	
Nationality	VarChar	18	
State	VarChar	12	
City/Location	VarChar	12	
Prison Capacity	Integer	7	

TABLE 4. Structure of database for information about inmate.

Name	Data Type	Length	Description
Station Number	VarChar	13	
First Name	VarChar	16	
Last Name	VarChar	18	
Date of Birth	VarChar	8	
Sex	VarChar	5	
Date of Arrival	VarChar	10	
Religion	VarChar	13	
Offence	VarChar	18	
Nationality	VarChar	16	
State	VarChar	12	
City/Location	VarChar	12	
Date of Release	Date/Time	10	

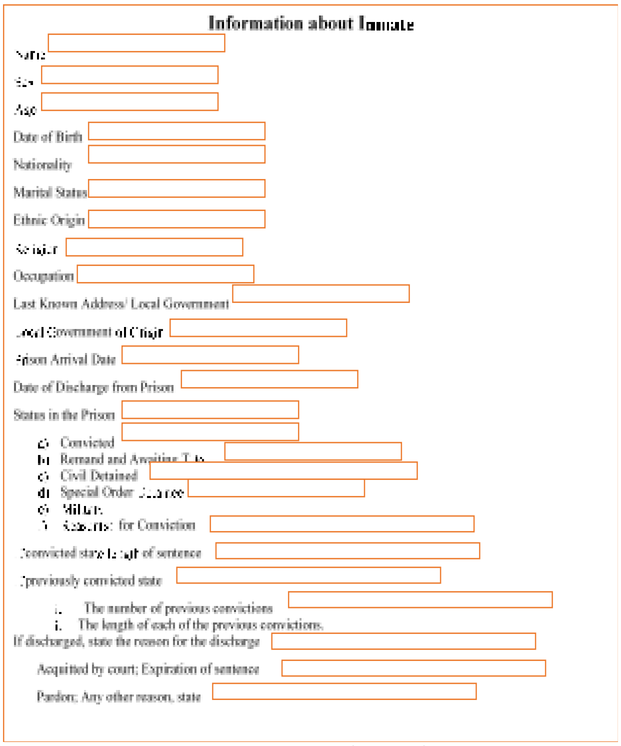
TABLE 5. Database of security agency on accused person.

Name	Data Type	Length	Description
Station Identity	VarChar	13	
Password	VarChar	8	
Name of Station	VarChar	18	
File Number	VarChar	8	
Type of Offence	VarChar	18	
Date Accused	VarChar	10	
Nationality	VarChar	16	
State	VarChar	12	
City/Location	VarChar	12	

H. Specification of Design

The specification of the designed system comprises the input specification and output specifications.

The input specification comprises information for prisoners/inmates, information about prison, security agency database on accused person, and information about accused person. A typical input specification is shown in Fig. 7.



Information about Inmate

Name
Sex
Age
Date of Birth
Nationality
Marital Status
Ethnic Origin
Religion
Occupation
Last Known Address/ Local Government
Local Government of Origin
Prison Arrival Date
Date of Discharge from Prison
Status in the Prison

a) Convicted
b) Remand and Awaiting Trials
c) Civil Detained
d) Special Order Detainee
e) Military
f) Reasons for Conviction

If convicted state length of sentence
If previously convicted state

i. The number of previous convictions
ii. The length of each of the previous convictions.

If discharged, state the reason for the discharge
Acquitted by court; Expiration of sentence
Pardon; Any other reason, state

Fig. 7. A typical input specification for inmate.

The tables which are produced individually and which summarize information about prisons, prisoners, accused or convicted persons in selected attributes such as sex, age, length of imprisonment, type of offence, religion, ethnic or State of origin, etc., can be easily produced using appropriate output processing routines.

I. *Evaluation of Root Nodes Attributes*

The root nodes representing the attributes to use are calculated using information gain. Information gain is measured based on the entropy obtained from information theory. It is used to measure the information gain and gain ratio of the tree. In this paper, the interest is in knowing the entropy, and this leads to knowing the information gain of the output values from training examples. The entropy can be considered as the average number of bits needed to encode an output value. Table 6 shows the data collection for calculating entropy and information gain.

TABLE 6. Collection of data for entropy and information gain calculation.

Attributes	Instances	No. of instances	Gender`	
			M	F
Under 16	Gambling	1	1	0
	Unlawful Assembly	5	3	1
16 -20	Stealing with violence	6	4	2
	Armed Robbery/Murder, Breaking of Burglary	5	5	0
	Gambling	5	6	1
	Unlawful Assembly	8	5	1
21 -25	Cheating	14	11	2
	Impersonation	9	9	1
	Homicide, suicide, infanticide	5	4	1
	Forgery and impersonation	12	12	2
	Stealing with Violence	38	30	4
	Abduction (Kidnapping)	14	18	1
	Armed Robbery/Murder	40	38	2
	Gambling	22	19	3
	Abduction	12	15	4
	Unlawful Assembly	30	22	3
	Corruption, Bribery/Abuse of office	23	11	4
	Frauds by Trustees	2	2	3
	Sedition and Malicious Publication	2	2	8
	Obstructing Security Agency	1	1	0
	Offences Relating to Faith	2	2	0
26 - 50	Cheating	2	2	0
	Homicide, suicide/infanticide	2	2	0
	Stealing with violence	3	3	0
	Armed Robbery/murder	2	2	0
	Adduction (Kidnaping)	2	1	3
	Frauds by Trustees	3	1	0
51 and above	Homicide, Suicide/infanticide	1	1	0
	Armed Robbery/murder	2	2	0
	Corruption, Bribery/Abuse of office	3	1	1
	Abduction (Kidnapping)	1	1	1
		277	236	41

TABLE 7. Calculated information gain.

H (S)	0.6050
Gain (A_1 ; under 16, S)	0.6000
Gain (A_2 ; 16 - 20, S)	0.5600
Gain (A_3; 21- 25 , S)	0.4000
Gain (A_4 ; 26 -50, S)	0.6049
Gain (A_5; 51 and above , S)	0.5950

Substituting values in Table 6 into (1) and (2), the entropy and the information gain are calculated. Table 7 is the values of the information gain calculated.

The highest information is obtained from attribute with age range 26 -50. It is then taken as the root node of the decision tree drawn in Fig. 8 and 9.

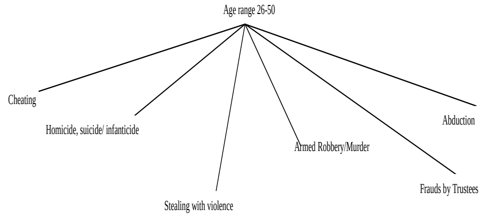


Fig. 8. Decision root node for querry database.

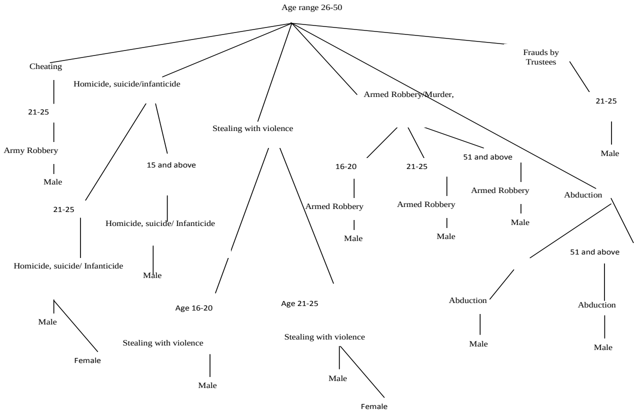


Fig. 9. Sub-decision tree for offence.

J. *Programming Language Employed*

The programming language used in this study is PHP. It was chosen due to its object oriented nature. The development environment is the Net-Beans ID and it was chosen because it has a free PHP and MYSQL-based open-source development environment which is available for most modern platform. Though its configuration is meant for use by developers of Java applications, it has several add-ons made to suit different environments. One of such add-ons is the Net-Beans Mobility Pack, and has been used in the development of this application. It is a well-integrated system for developing applications for MYSQL sever, as well as emulators and on device debugging.

IV. RESULT AND DISCUSION

A. *Result*

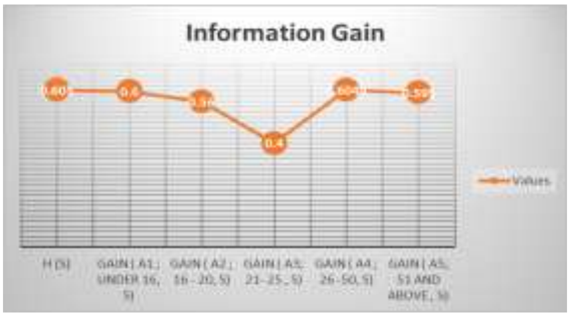


Fig. 10. Informatio gain for determining optimization query root nodes in a DDB using the attributes.

The results obtained from the query time of ID3 and search engine results in a DDB is shown in Table 8.

TABLE 8. ID3 query execution time of ID3 and search engine on offences.

Offence	ID3 query time in a DDB × 10^{-4} (second)	Search Engine in a DDB (second)
Offence A	4	6
Offence B	6	11
Offence C	8	13
Offence D	5	10
Offence E	7	10

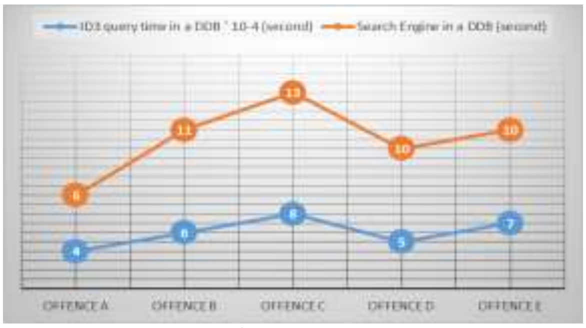


Fig. 11. Graphical comparison of ID3 and search engine query time results in a DDB.

Table 9 shows comparison of the results of the analysis using ID3 rules developed in query DDB and seerch engine in DDB based on age range of crime committed by an inmate/prisoner.

TABLE 9. ID3 and search engine execution based on age range.

Age Range	ID3 query in DDB × 10^{-4} (second)	Search engine query in DDB (second)
Under 16	2	5
16 -20	4	8
Age 21-25	9	10
Age 26-50	3	5
51 and above	5	9

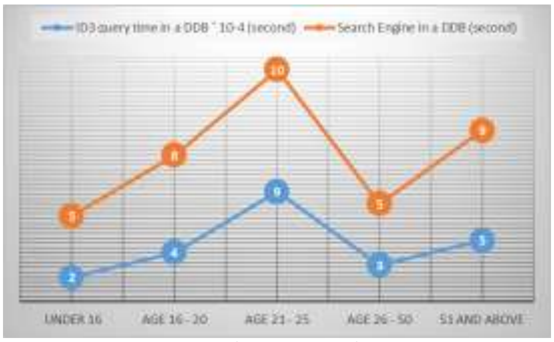


Fig. 12. Graphical comparison of query time of ID3 and search engine with respect to age range.

B. Discussion

Fig. 10 is a graphical representation of the output information gain for the root nodes as calculated and stated in Table 7. In Table 8, the results of ID3 compared to those of search engine with respect to the offence committed is presented. A graphical comparison of the ID3 and search engine query time is plotted in Fig. 11. The plot shows that the developed ID3 algorithm results which follows the set of rules, pattern and decision performed query related offences in the database with lesser time than the search engine in the DDB. Also, Table 9 present the tabulation of ID3 query in distributed database (DDB) and search engine query in DDB based on age range. Fig. 12 shows the comparison plot of Table 9. It can be seen that the ID3 query response time is lesser compare to that of the search engine.

V. CONCLUSION

The complexity of modern database management systems has made the execution time process of the queries to require accurate estimations and prediction for efficient and optimal performance characteristics. This paper has presented a solution for data allocation optimization in DDB system. This was achieved using machine learning ID3 algorithm. The results obtained showed that the developed algorithm gives fast response time to query processing. It is able to give out security related information to security agents.

REFERENCES

[1] S. A. Idowu and S. O. Maitanmi, “Transactions- Distributed database systems: Issues and challenges,” *International Journal of Advances in Computer Science and Communication Engineering (IJACSCE)*, vol. 2 issue I, 2014, ISSN 2347-6788

[2] J. Kunal, P. Viki, and, B. B. Meshram, “Query processing strategies in distributed database,” *Journal of Engineering, Computers & Applied Sciences (JEC&AS)*, vol. 2, no.7, 2013, ISSN No: 2319-5606.

[3] K. T. Anand and M. Tripathi, “A framework of distributed database management systems in the modern enterprise and the uncertainties removal,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, issue 4, 2012, ISSN: 2277 128X.

[4] G. R. Abhijeet and O. Bamnote, “Query processing in distributed database,” *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, issue 2, 2013.

[5] E. Ramez and B. Shamkant, *Fundamentals of Database System*, Fifth Edition, Pearson Education, second Impression, pp894, 2009.

[6] L. Golshanara, M. Seyed, R. R. Taghi and S. Hamed, “A multi-colony ant algorithm for optimizing join queries in distributed database systems,” 2013.

[7] A. Mishra, N. Gunjan, and P. Ashish, “Dynamic programming solution for query optimization in homogeneous distributed databases,” *International Journal of Engineering*, 2012.

[8] P. Stone, et al, "Query Execution and Maintenance Costs in a Dynamic Distributed Federated Database," 2012.

[9] E. O. Osuagwu, “Software Engineering, A pragmatic and Technical Perspective,” Hi- Technology Concepts (WA) Ltd, Owerri, ISBN:978-2747-02-8, pp 326-339, 2008.