Reinforced Experience Generator based on Query Nature and Data Bulking

Mazhar Hameed
Department of Computer Science & Engineering
HITEC University
Taxila, Pakistan
Mazharhameed91@gmail.com

Hiba Khalid
Department of Informatics
University Libre De Bruxelles
Brussels, Belgium
Hb.khalid92@gmail.com

Usman Qamar
Department of Computer & Software Engineering
College of Electrical & mechanical Engineering, NUST
Rawalpindi, Pakistan
usmanq@ceme.edu.pk

***Abstract*—The technology advancement has given a more enlightened perspective on developing recent solutions. The advancement of imitating the human brain has achieved some milestones that are providing promising results. However, the technicalities and reliance between different computing entities still remains a concern. This research provides a module based framework that can create a brain or experience imitation for itself using the reinforcement learning agents. The research has established a framework design that can connect and validate user requirements and map it according to functionality with reasonable data retrieval and performance measures are observed. Data is a concern for modern technologies, so is the customer requirements. This research combines the two widest entities of research and brings about a design framework that allows the agent to communicate using knowledge from data and instruction or queries in the forms of user requests or business requirements. The experience generator than enhances the next performance and lessens the cost, expense and improves the overall performance of two widely separated modules with reinforcement learning and experience or knowledge base.**

Keywords—Big data; Online Analytical Processing (OLAP); Online Transactional Processing (OLTP); contextual query; scientific data management

I. INTRODUCTION

This term "big data" has been a part of active research and development for a long time now. Big data and its development have faced many misconceptions with relevance to its computation, the size and variety available. The concepts are however much clear now after significant research in the field. Big data is a representative necessity of the business and functional processing. The most important characteristic for any business at the current moment is management of big data.

Another important aspect of relevance is "data analysis". Data Analysis has been around for decades however due to the changes in nature and structure of data the prospects have changed and increases in complexity over time and space. Data Analysis has transformed from traditional methodologies to unconventional processing. Data Analytics in the current scientific and corporate development community is based on various tools that help in analyzing critical business data. The analysis further supports the decision-making process [11]. The tools for analytics have also been evolving with new techniques and algorithms. The most commonly associated term with data analysis is "predictive analysis". Predictive analysis presents a broader perspective to the commonly used data analysis. Some of the most recent achievements in the area of predictive algorithms include the Deep learning, reinforcement learning and quantum computation. These algorithms capture the human behavior and action policy in computational representation to provide more accurate results. Google and other tech giants have been successful in applying the technologies with ground breaking results and mechanisms. The concept of providing computers with insight is not an old concept however, the successful interventions near to band to intelligence just appeared a decade ago.

In relevance and connection to big data, predictive analytics a common concept is "distributed data systems" and "Cloud Systems". Most of the processing systems now-a-days require distribution of workload and processing. Thus systems are rarely on a single processing set. The complex systems in current scenarios tend to be in a more complex situation with the use of multiple systems of distribution and cloud connections. However, the query and search requests can request information in parallel from locally or heterogeneously distributed systems. To resolve the problem of over querying and minute projection based parallel behavior, the nature of the query is taken into account. There are two concepts that fall closely with each other in this prospect. The first concept is the use of nature of query i.e. the type of response that is required by an entity. The second most common aspect is the nature of data that is under the querying process. The second prospects provide insight into a much deeper concept on the storage and retrieval properties of the data.

The type of data can lead to very complex and simplified query processing techniques. If the type of the data is known, then the structure required to store and process it becomes more viable. Traditionally in big data systems the data is categorized as the following three categories

a) Structured Data: This contains the highest level of "form" in a database. The most common representation is the traditional databases. The information in structured data is organized and can easily be predicted.

b) Unstructured Data: This is a much more complex representation of the traditional data. The unstructured data is not in the form of a particular model. It also does not have any

pre-definition of organization and relationship development in the collection of a set of unstructured data.

c) Semi-Structured Data: As the name suggests it has properties linking from both unstructured and structured data. This type of data does not necessarily have a model for data representation but a simpler more generic annotation structure for representation and identification may be present.

Each type is representative and requires a different storage and retrieval mechanism for optimal searching and query optimization. In relevance to distributed systems that are based on processing of querying and searching big data. The problems not only exhibit the storage and retrieval capacities. It also extends it to the measurement of complexity of the problem under discussion. This is regarded as the "nature of query". If the nature of data is known and nature of query can be calculated the processing power and storage structures can be optimized to provide more optimal solutions in distributed environments.

Big data is typically distributed data. Thus, single node processing is not a concept to be integrated with big data. In subject and practicality of the nature of the work distributed systems function optimally with big data provided the nature of query and data is known. This can be understood on a conceptual level by understanding the types of queries that can be requested as a part of any system. These are the generic representations of the types of queries.

a) Navigational Queries: Navigational queries tend to provide connections to a next possible search result or requirement based on an input form the user. The most common explanation of the query is defined as the "Known Intent" i.e. the intent of result as well as the intent of data placement is known.

b) Transactional Queries (OLTP): These are the most commonly and most traditional forms of database queries. OLTP stands for Online Transactional Processing. These queries typically answer to most traditional queries typically data entry, retrieval from a relational database system.

c) Informational Queries: These queries are not single content or single intent based. These queries are typically generated and executed on graph systems. These can be representative data graphs, knowledge graphs or topological data representation graphs.

d) Analytical Queries (OLAP): OLAP is defined as Online Analytical Processing. This is concentrated towards answering more complex data queries. The computational complexity is higher for these queries and are constructive multidimensional analytical structures.

Each type of query invokes a different set of data that can be further conceptualized as well. The types of data for groups of queries can also vary depending upon the gravity and situation the query relies in. similarly there are concepts and queries that can be completely disjoint such as predictive analytical queries. These queries are best processed in non-traditional systems, i.e. no-relational systems. If categorization can be understood the storage and query processing can be enhanced for distributed big data systems. The rest of the paper is organized as follows: Section II presents the reason of research development as Problem statement. Section III of the research paper gathers the literature from existing work and presents a summarization of relevant concepts as Literature Review. Section IV follows the research development as Methodology and presents the research artifact as its primary contribution with scenario development and explanation. Section V concludes the literary and experimental research conducted as its research conclusion. Section VI focuses on explaining the concepts that integrate and originate as Future Work for the research disseminated.

II. Problem Statement

The most anticipated future representation of queries is to create a conceivable intelligence for the queries. This means that queries should be executed based on their type and the type of data under consideration. The classifications of data can never be completely separated since queries require all four sets of their types in a probability of one or more queries at any given time. Thus no data can be completely regarded as only for the use of OLTP or OLAP. A categorical conceptualization can however be generated to support the concepts that arise as problems in distributed big data computations.

The problem at hand is the use of intelligence i.e. machine learning algorithms in flux with the database management, warehousing and storing techniques to identify the probable and most efficient manner of manipulation. Data insights are not typically handed by data on its own. Thus, a computationally complex identification is required by the system to mutually bind and understand the nature under representation as well as the requested data nature.

The working and execution of the above mentioned concepts is not typically simple or easy to perform. Thus, in order to apply the generics in common corporate and execution databases the real-time conjunction has to be stipulated. The real time conjunction indicates the following:

a) The nature of the data is to be understood and known prior to preliminary prediction or classification strategy.

b) The detection of query type has to be determined at the run time and a categorical classification has to be presented.

c) The computational expense of representation has to be subjected before query is answered.

d) The runtime decision support mechanism for handling and answering queries based on the

i. type of the query itself;
ii. type of the data required to answer the query; and
iii. the structure most generically typical and suitable for answering the combined set (i, ii).

The problem under discussion is to understand the complexity of data storage based on its nature it can be processed for. This is a problem of both distributed systems and big data. In order to understand the data and how it should be stored. An important question needs to be answered beforehand i.e. how can this data be queried or searched for. The conceptualization of the process is not simple and the querying nature and querying goals can easily identify the

possible utilization of data and its nature. Thus, it can result in effective computational storage structures. Another important contributing factor includes the prior data intake knowledge i.e. the data at the time of acquisition is based on what classification and what is its nature at the time it was acquired. This can be simplified and represented as follows:

a) Types of queries required to answer a particular question.

b) The actual data collection format

c) The optimal data representation format

d) Nature of query

e) Query Goals

f) Storage Computational Structures

The research has been designed and conducted in order to analyze the prospects of storing data based on the nature of queries and its representational format. The process of formulating a computationally efficient algorithm is based on factors of data types, storage, structure, need, goals of query and conceptualization of data. The system should effectively answer with results that are based on the preliminary analytics of query and data nature types. A representative data and computation framework needs to be devised in order to support the mechanics as well as functional behavior of the system.

III. Literature Review

Query localization [1] and query representation have been a matter of discussion in scientific community for more than two decades. Queries differ from each other based on their location or their operation location. This is identified as the query questioning domain which is represented in the form of the users. The query processing becomes intensive as well as expensive in prospects of answering the query location questions such as imprecise query knowledge. The location or identification of a costly user request can cause system initialization and processing problems. Google's successful reinforcement Atari learning game [2] was the first deep learning deep model developed by the organization with highest performance and accuracy results. This research focused on developing policies using high level sensory input systems. The training as performed on 2600 Atari games and function proximity and Q-variance was deployed as a part of methodology to achieve the optimal results. Another important aspect of developing technologies that are efficient and intelligent involves the concept of independent AI [8]. This is also regarded as the general purpose AI. Independent AI focuses on imitation learning, model based and framework based learning to enhance the capacities of computers when in interaction with the human entities or human experiences. Increasing the experience of computers is similar to basic experience games for children. Thus, many scientific aspirations have been focused on changing the way programs are developed. It induces more human like techniques for teaching computers. This is the basic essence of reinforcement learning technique [12], [16]. The reward systems and allocation of rewards has been a point of debate. Some scientists have developed systems that function promisingly on immediate rewards [1]. Some other are more focused on the knowledge and experience building using delayed rewards [13]. Use of delayed rewards is also a very commonly attribute that has been derived from human learning behavior. Learning is not only constituted for systems. Data learning [6] in spaces and large scale multi-perspective system data bases [7] require the process of learning and computational power for better management, results, predictive analytics and analysis [5]. Data discovery and understanding of data on its own is not a simple process. This has to be constituted for since currently everything is directly or indirectly related with data. A more efficient system of creating useful data that has its own understanding and power is the next step towards general purpose systems and general purpose artificial intelligence systems.

IV. Methodology

Large databases have many problems associated with them. In concern with the recent development and service requirements of business intelligence is a necessary part of any system. The traditional systems were capable of providing needed results using simpler databases and even more simple services. The prospects have however changed. The size as well as requirements from a business or task has potentially changed. The knowledge and understanding of executing algorithms have become a necessary entity for most of the working algorithms and functions. The underlying understanding however indicates that all learning process still require experience like humans. Humans have learnt over the past tricks and gained knowledge by using experience as guidance. In computer science the same concept has been initialized to combat the machine intelligence.

In this section we will explain the methodologies and studies undertaken to observer and develop an Experience Based Regulator Framework in support of developing an intelligent query efficient system. The first objective of the methodology is to establish a reliable system of identification of query and its representation. The second objective of the methodology is to present the data variability and use in accordance with the types of queries. The third outcome of the methodology is to understand and develop and Experience Regulator Framework. Finally, the methodology will conclude with a research problem in case of non-distributed systems that require multiple communication channels for coherence and data predictions.

"Queries" are the fundamental questions that we ask computers to answer in order to achieve or accomplish a goal. These queries have nature that can be understood and exploited in order to provide reflection on how the computer systems can answer better in regards to the asked questions. Nature of the query cannot be easily determined since queries are in the form of representational text. Thus, in order for identification for a type of query and its nature a specialized mining system has to be designed for understanding it's working. The working set for identification of query is based on the feature set for its positive and negative cases. Queries can be interpreted in the following broad categories:

a) Text based user queries i.e. business requirements [9].

b) Traditional computer language queries i.e. queries written in a specific language.

For answering text based business queries [9] many systems for data mining have been developed that can classify and extract meaning form user text, keyword based searches or specific business requirements for developing an algorithm and application. Similarly, many system query recognizers have also been developed that identify queries in various different database languages and provide high performance insights. The research under discussion utilizes both of the concepts and represents a Query module with the combination of both the systems. For simplicity and referencing the systems have been classified as follows:

a) System[A]: BR (Business Requirements)

b) System[B]: PDQ (Programmable Database specific queries)

Note: For simplicity of the system application the PDQ have been restricted to only two database languages SQL and NoSQL. The business Requirements have however been conducted without any restrictions as per system utilization of developed research [9].

The system for query identification and development requires a spectrum a broad categorization for mapping scenarios or inputs as per instructions. The query based system model set has been set similar for both Systems [A], [B]. Following types have been identified for classification and band matching in case of "Query Framework representation":

- Navigational Queries
- Transactional Queries
- Informational Queries
- Analytical Queries
- Responsive Queries
- Unknown Queries
- Invalid Queries

TABLE I. QUERY TYPES

No.	Query type	System [A]	System [B]
1	Navigational Query [NQ]	✓	✓
2	Transactional Query [TQ]	✓	✓
3	Informational Query [IFQ]	✓	✓
4	Analytical Query [AQ]	✓	✓
5	Responsive Query [RQ]	✓	✓
6	Unknown Query [UQ]	✓	✓
7	Invalid Query [INQ]	✓	✓

The mapping of each query and identification is performed on specific rule sets. If the query is business requirement it falls into a category of [A]. If query is presented in a database language it is feed into another classification system called [B]. The first identification is performed using the text classification and mining methodology developed for identifying the business requirements [9]. The second classification is designed as a rule set mechanism for identifying the query language for SQL and NoSQL languages. For Example, if the text has terms from the SQL language such as 'SELECT', 'FROM', 'AS', etc. and operators like '*' and combinations like 'SELECT *' the identification is performed based on relevance to language constraints and available learning techniques.

In case of System [B] a complete framework as a separate research contribution has been established by us. The idea behind the utilization and training is based on semi-supervised learning. The system [B] is developed using the Naïve Bayes algorithm in construct and functionality of semi-supervised tagging. The system was input the language syntax with initial tagging. A 'reference tag' was also substituted in the primary training set to establish the difference between the syntax of SQL and NoSQL. The occurrences and priority frequency was also established for each language tagged as [i,A] for the SQL and [k,B] for the NoSQL. Some of the variables used for development of this system have been identified below with their purpose and use.

a) Main Keywords [A], [B]: These keywords were identified as a set of syntax that included the main tags as select, from, while etc. that are commonly used in the SQL and NoSQL language for operation. This variable constituted towards the identification variation.

b) Priority keywords [A], [B]: These Keywords were included when the transaction was changed in order due to specific keywords or ending session highlights.

c) Following Keywords [A], [B]: This is a mapping equation and a mapping path that represented the keywords that immediately followed each other such as SELECT * FROM XYZ. Thus for this sentence SELECT is followed by FROM. Representation is S1➔ F1.

d) Linking possibilities [A], [B]: This included the joins and complex database query representations where more than one operation was co-linked.

e) Frequency of Occurrence: This was a count program tabulation for the total occurrences of each identifier in both [A] and [B].

f) Incorrect Keywords: This was a bool representation for identifying the garbage keywords such as 'selecting'. This based on root of origin is corrected to 'SELECT'. But if a sentence appeared as CHOOSE ALL FROM XYZ. It will disregard the sentence.

NOTE: The system can only identify the same root origin words and correct them. This is the limitation of the system. Even though Choose in meaning is similar to select and can be represented as a synonym. The current application does not have a support for mapping synonyms antonyms and appropriate keyword mapping such as '' can be equal to words as 'ALL' but the research under discussion does not have this feature at the moment.*

Similarly, many other variables and factors concluded the functionality basis for this module. The business requirements were extracted and mined according to text based mining techniques and a classification for future datasets was also constructed using the Naïve Bayes algorithm after necessary keyword generation, stop word elimination etc. for the identification of the type of query separate variables were used for [A] & [B]. A separate knowledge for understanding the requirement and query nature was designed it is called the

"The decision Library". The decision library was based on keyword representation and probability of resulting in more than a simple transaction. For each type identified in Table 1, a probabilistic measurement was calculated that generated a conclusion of whether a query falls into any of the mentioned criteria's. For example, if the query was Analytical for business requirement it would be classified based on a question of prediction, analysis or other keywords that indicate a deeper insight is required. For DB queries a representation was calculated which indicates and finds out if the query answers an analytical question using the business requirement as its mapper function and its syntax and utilization of relevant keywords.

The second module of the research under discussion was establishment and analysis of big data modularization. This module was focused on understanding and analyzing the nature of data presented as a process. The data was collected on stock market analysis from the publically available from benchmark datasets.

A bulk of data was considered i.e. a singleton set of data was not considered. The Bulk consisted of three types of data:

a) Structured: In the form of relational tables for the stock exchange. All tables and relational information was utilized for these.

b) Graph based data: This dataset was also utilized to formulate a simpler query performance metric

c) Semi-Structured data: This was collected from graph data that had some simpler or specific information available but not completely available so it can provide direct insight into the solution.

d) Irrelevant Data: This data was used to measure out precision and accuracy of the predictive systems. It consisted of movies, songs etc. anything could be included as bulk irrelevant data to provide test bed for algorithm efficiency. No prior tagging was performed. The algorithm and module had to identify the domain of incoming data as stock exchange relevancy or not.

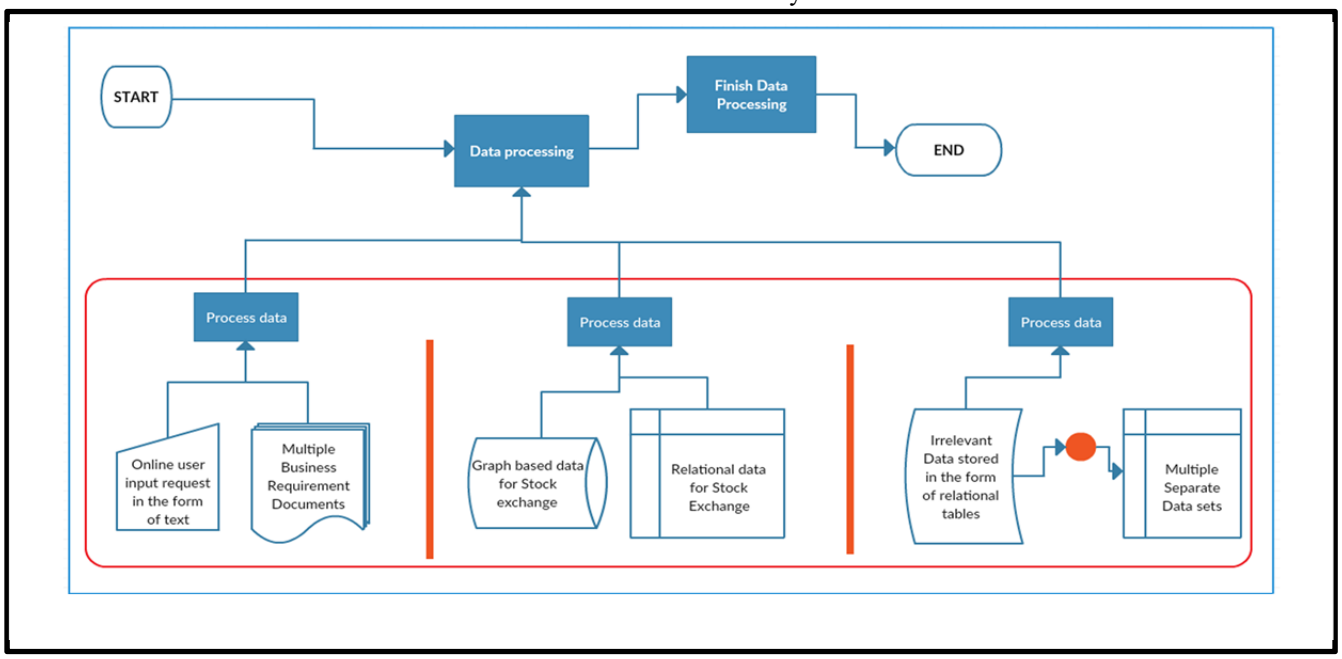

Fig. 1. Data bulking representation.

Fig. 1 indicates a simplified representation of data collection in the conducted research it is called the "Data Pool". There are three types of data bulk that have been created and generated for conducting experimentation of this research. The first bulk corresponds to the processing of data in relational perspective. This is mostly used for transactional queries and in some cases of navigational queries. Thus the set representation for the first bulk of data pool is as follows:

$$Bulk(SET1) = TQ \pm C[NQ] \quad (1)$$

The relationship exhibited in (1) identifies that C[NQ] is conditional Navigational Queries. It means that all navigational queries do not directly correspond to transactional or relational tables. Some queries may be representative but not all. Similarly, the two other bulks of data represent information according to their domain and needs. In case of analytical business requirements, the first bulk set can identify if a requirement is analytical based on keyword propagation, probability and predictive learning using Naïve Bayes [10]. The equation changes when analytical scenarios or analytical requirements are fed into the system.

$$Bulk(SET\ 1_1) = AQ \pm IFQ \quad (2)$$

Thus, each module has its representation based on the data it is handling and the query it is answering as a part of the process. Each module and data query has a representation scenario equation similar to (1) and (2). The equations can adapt based on the type of the query. This is called rule based equation system.

NOTE: The system is not efficient enough to generate rules based on knowledge and priorities. A rule engine is established

that takes input from the query type tables and data tables to follow up and provide representational equations for classifications. An intelligent equation rule system can be developed but it is beyond the scope of the research under discussion.

The third module is the main module/ framework for the developed and conducted research. This is based on the development of an efficient "Experience Calculator". The main function of the experience calculator is to create a Conscious for the system. In artificial intelligence the computing systems have been designed to overcome the lacking of the system that function on rules that are not capable of performing operations or functions outside the description of their tasks or goal achievement with criterion. The idea behind developing an experience calculator is to establish the sense of awareness in the system. The idea has been understood and extracted as a part of Google's applied Deep learning context [2]. This is also based on the reward generation systems [6]. The research thus combines the attributes form studied research in the field of distributed query based decision systems. The scenario that has been selected for experimentation is based on the stock market predictive analysis. The domain data sets, business requirements, graph systems, rule based systems all fall into the category of stock market analysis. The problem scenario sample has been presented below for further understanding of the system. This scenario is a simple case that has been considered. A total of 500 scenarios were constructed to identify and test the validity of the system functionality.

Example Scenario:

A business company titled "Aurei Builders" is looking into investing in a different market share. The current analytical procedures and market leading share is represented by the "Aurei Builders". Mr. Jack Shepherd has earned the legacy of this firm and is looking to expand the company horizons beyond building and contracting shares. The company partners have decided to expand in the business of telecommunication services. The stock exchange for telecommunications and competitive edge compartment is very short in band as well as capacity. The partners however want to set up an analysis period of trial 6 months observing the market shares and concluding in either procurement of service providence in telecommunication as a part of business expansion and dimension liberation.

In the above scenario the stock market can be analyzed for both transactional queries and analytical queries. This scenario also binds in the business requirements that can be taken as input in the form of text. The parts of system that require simple review of existing information on market leaders and future competitors fall into the Transaction query section with data bulking in the relational table departments. This information can however be more useful in graph form for analytical purposes. Thus a graphical sub-representation of market analysis can provide connections and predictions based on the analytical query and graph based databases. The experience calculator uses input from both data bulking and query processor to build its knowledge and experience using a reward system. The first step will be the identification of the query that has been set as input to the system. If the input is completely textual in the form of business requirement the System [A] is set to execution. If system [A] is set to execution, then an equivalent equation and conclusion for representation is generated as a part of the process which is then mapped onto the bulking data system. If the conclusion forms System [A] for supposition was AQ, i.e. analytical Query. This indicates that provides a decision to the system to work on graphical representation or graph databases and calculate results. In parallel if the system output was AQ+ TQ then a parallel process is generated for transactions and two conclusive results are produced that serve one part each for the requested question or query. For example, if the query question was: "Calculate the total sales for XYZ in 2015 for sim cards and find out the total sales that can appear based on customer responses. This query has both TQ and AQ requirements. Thus BULK SET equation will perform the both operations and provide a conclusive result. After all necessary calculations and result generation a "User response" is taken in the form of a visual. A simple question is asked: Was this what you were looking for? The response has three options:

a) Yes
b) No
c) Somewhat relevant.

This response is sent to the Experience Calculator. This user response is kept separate from the algorithm accuracy and predictive performance. The learning capabilities of an algorithm are measured using the standard formulas and techniques for efficiency, accuracy, performance and precision. However, the user response is also calculated in this manner to initiate reward for the calculating agent. Taking a direct user response increase the prospects of future learning and teaches the agent to find and calculate more based on the experiences gathered. The reward calculation for agent has been performed on the following guidelines.

Environment: For the agent building up an Experience Knowledge or Experience Calculator. The environment can be deterministic or non-deterministic i.e. the rewarded agent does not have to calculate a future prediction of the reward. The predictive algorithms are for the other modules. In this reward system the agent has to find out for the states of determinism or non-deterministic. For simplicity and more definitive results the experience is calculated in cases of deterministic cases with a reward that is associated and generated for it.

Reward Function: This function determines based on success and failure for the agent to build up knowledge or experience system. The reward function id typically applied in reinforcement learning environments [4]. Similarly, here the agent learns both from the algorithms predictive analytics, efficiency rate, accuracy rate, classification precision arte and the user experience collected at the end of a service or question. The main outcome or goal for the reward agent is to increase its rewards. Thus, like in human behavior based on good predictions and right result calculations the rewards will increase. There can be different kinds of rewards. For the research under discussion only three types of rewards have been dedicated the agent.

a) The scalar health rewards: these rewards are awarded as health points to the agent when a correct prediction is made by the predictive algorithm Q-learning and Naïve Bayes prediction from both modules [14], [15].

b) The scalar boost rewards: these rewards are awardee to agent based on user satisfaction results gathered at the end of a result or service that has been provided to the user in the form of an answer to the requested query.

c) The goal achievement rewards: this is an award that is set as a goal for this agent. This functions as a set achievement. If the agent has collected 10 health rewards and 10 boost rewards a major goal reward is added to the list. Now, if all rewards were attained without any negative feedback or result the increment for goal reward is incremented by power of 2.

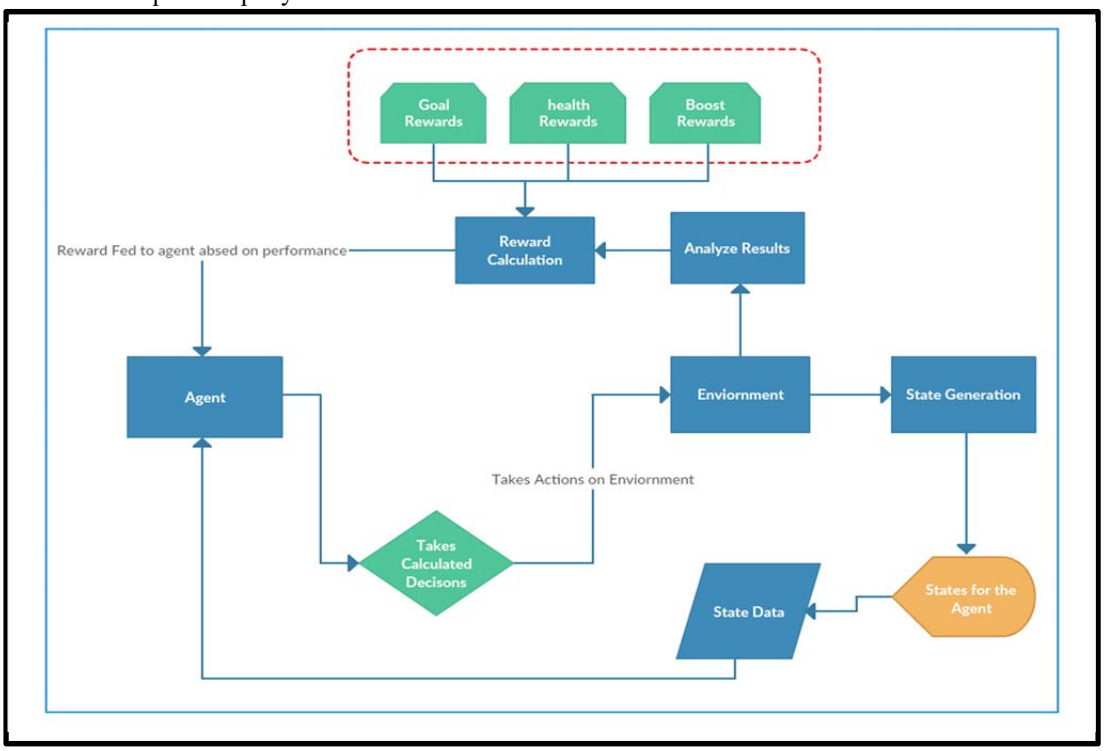

Fig. 2. Reward based agent system and flow of operation.

Fig. 2 demonstrates the working functionality for the reward calculating agent. It also identifies the role of three different rewards and the role of environment from the two calculation and prediction modules Query framework and Data bulking framework.

The fourth module of this research was conducted to study the aspects of a case when data separation is not a possibility. This means that data cannot be identified or separated for optimal storage and retrieval purposes. Also, the data does not have definitive boundaries of separation on the basis of identification. The information co-dependency is higher than information independency in this scenario. The research under discussion studied the necessary causes and variables that can potentially affect the performance of a system based on the placement and arrangement of data. Thus, for the system algorithm and methodology developed to function in scenarios of non-distributed data environments another framework design has been concluded to provide better and more efficient performance measures. The framework designed has been described below in Fig. 3. The image represents a change of direction as well as change of data manipulation, prediction and reward experience calculation. The first step in the process is to initialize the query process. In this scenario the only input that can be attained is the user input or the business requirements. No database language can be restricted or added to collection since the data representation and formatting is one single entity. The query identifiers however remain similar to the case_1 problem in case of data bulking. The queries in text are identified using text mining and training using Naïve Bayes algorithm to provide necessary classification and prediction into the categories of TQ, IFQ, INQ, etc. The query processor then inputs the trained analysis and predictions into the Mapper AI module. This module has two sets of inputs. The first input is in the form or tagged requested queries with predictions and classifications. The second input is a collection of data. The data needs to be scrolled and iterated according to the process needs to answer the customer queries. Finally, the mapper applies reinforcement and reward based learning using iterations over a couple to determine the answers. The mapper API sends its analysis and calculations into the knowledge development center or the experience calculator and displays the answer to customer queries. No customer feedback is generated for this case. The main functionality in this module is for the mapper AI API. This API functions on many variables and classification equation however some of them have been highlighted below to build understanding for the scenario.

NOTE: The domain of data has been limited to only stock exchange for this module to maintain and observe simplicity.

Query Complexity C_Q: This variable is an output of the function or program that takes query classification as input and generates its complexity based upon the number of iterations, row searches and tables it requires for referencing the answer.

Customer Question $Cust_q$: This is a reference and bool table that changes its value when a query has been answered. This has been designed to avoid repetitive queries with different words etc.

Data Independence D_{in}: This calculates the independence of data section under consideration. This is again a class that has multiple functions finding out the dependency between different variables, tables and even cross examination.

Data Co-Dependence D_{CO-IN}: This is the child class for the dependency class in the program. This identifies the different tables or relations that are functionally dependent for certain common queries that are asked by the domain customers of stock market.

Traversal Cost $Cost_{traversal}$: This calculates the cost of traverse and how it can be minimized if similar in between products or joins to be calculated for more than once. This will store the byproducts as a separate entry for not retrieving the complete databases every time a customer query is asked for same purpose. This will be updated after historical information has been added i.e. there has been a change in the data values. Specific data sets and time slots allow for data migration from real time tables to regular tables. After that byproducts and costs are updated for every trial. This is provided as a function on user interface.

There are many variables that are a part of a class called 'regulars'. These are the most requested operations in the domain of stock market. Such variables have been separated since regular update, retrieval is associated with them. Two of them have been listed below from 125.

Profit Margin $Profit_m$: This is based on the customer request for the profit calculations for a particular day or stock. This has the same formula for calculating profit but different sessions and requests can be handles at the same moment. For current developed research 5 separate profits and profit margins can be calculated at any given time 't'.

Market shift Calculation $Shift_c$: This variable also represents many in one function called 'Market shift'. The primary objective is to answer the request of customer on where the market is leading. This strategically calculates and provides insights into the system for customer query answering and result generation.

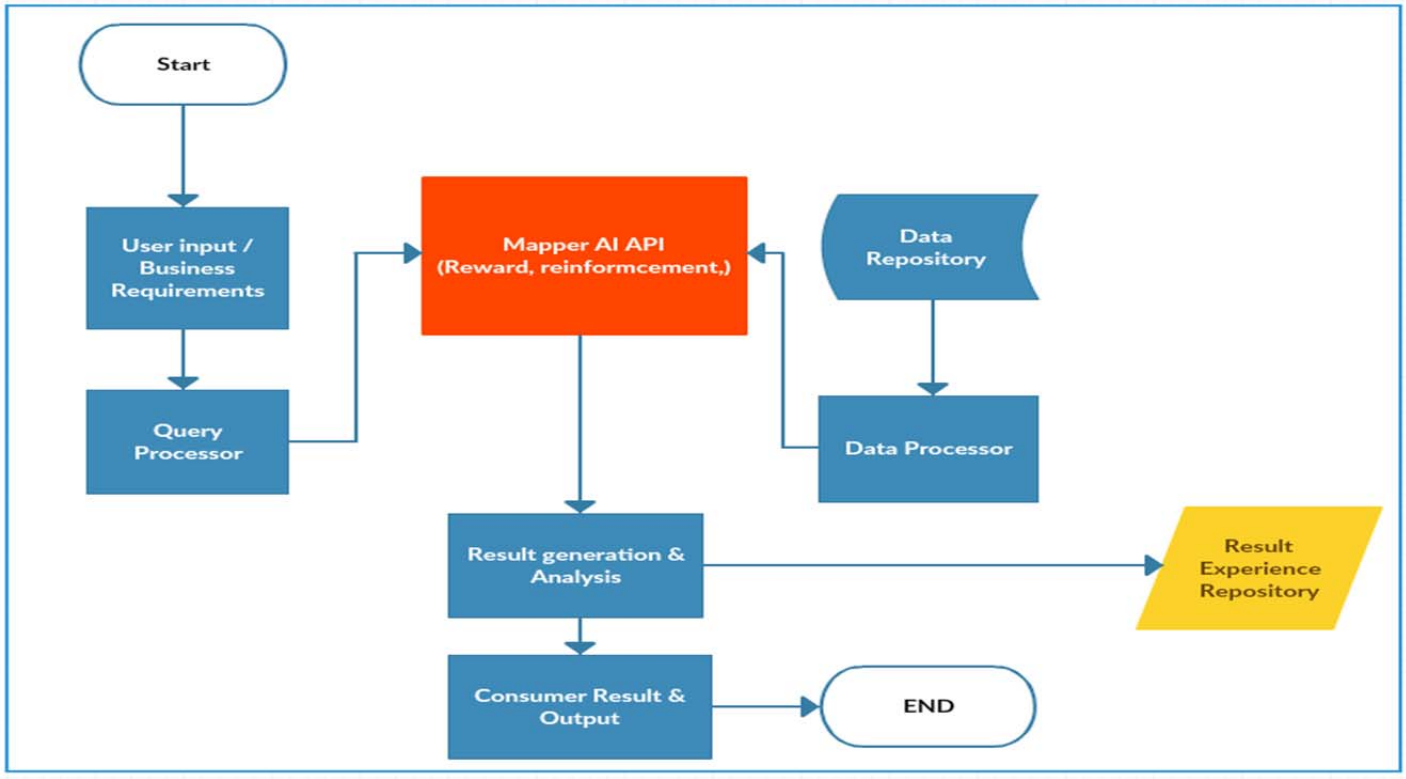

Fig. 3. An overview of expereince generation.

TABLE II. ANALYSIS OF MODULES BASED ON OUTCOMES

Module	Variability	Flexibility	Correctness
Query-Business Requirements	50%	75%	89%
Query- User Input or files	67%	76%	80%
Query-SQL	45%	50%	90%
Query-NoSQL	60%	78%	94%
Rewards	100%	100%	80%
Mapper AI	64%	77%	78%

TABLE III. OVERALL PERFORMANCE SUMMARY

Type of data	Performance	Accuracy	Data Relevance Metric
Text	90%	93%	80%
Mix data (text + images)	64%	66%	30%
Images	97%	94%	0%
Numbers (Integer, Float, Real)	90%	86%	70%
Search queries	76%	89%	90%

Tables 2 and 3 represent the analysis performed on the individual modules with metrics and data values. The test and training sets for each module were kept separate and demonstrated a decent and viable performance rate in each individual module.

V. CONCLUSION

This research was conducted to understand the need of purposeful systems in the domain of business intelligence. It explored the highly promising results of reinforcement learning and reward systems in cases for business intelligence and predictive analytics. The use of the simplest algorithms for classification such as Naïve Bayes proves to be a highly efficient algorithm for large bulks of text based data. Thus, a cultured input is even better for a learning system such as the one described and developed in this research. The system designed not only presents a solution to multi-perspective business analytics it also looks deep into the development of brain technology for any simple system. Stock Market Prediction is not a recent field. But with historical data and a learning experience can actually impact the decisions on day to day basis. Reward or reinforcement learning creates a powerful and promising solution for predictive analytics [3]. This research has focused on developing and understanding a system that can learn and provide better outcomes for different business.

VI. FUTURE WORK

The research under presentation and discussion has been designed as a four-part process of development. The presented methodology represents the very first conception and developmental structure for the research plan. The future work is associated with the second-part of the ongoing research includes the use of cognitive quantization. Cognitive quantization is to be studied and represented in understanding the efficiency and performance of query build up as well as query residue. Query residue is a new concept that has been generated as a part of the ongoing research. The quantization will use the help of quantum theory to present data in multiple formulations in a given space time "t" for any given complexity and instance. This representation will then be identified as a query process able object in time "t+1" in the space "s+1". Thus, identifying and experimentally proving the concept that one representation for data can be used to answer more than one query in a different data structure and dimension.

ACKNOWLEDGMENT

The authors would like to present their gratitude to Dr. Usman Qamar for his most generous knowledge in the ongoing research and development.

REFERENCES

[1] Punit Pandey, Deepshika Pandey, and Dr. Shirshir Kumar, "Reinforcement Learning by Comparing Immediate Reward," (IJCSIS) International Journal of Computer Science and Information Security, Vol. 8, No. 5, August 2010.

[2] Volodymyr Mnih Koray Kavukcuoglu David Silver Alex Graves Ioannis Antonoglou Daan Wierstra Martin Riedmiller, "Playing Atari with Deep Reinforcement Learning", DeepMind Technologies.

[3] Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In Proceedings of the 12th International Conference on Machine Learning (ICML 1995), pages30–37.

[4] Morgan Kaufmann, 1995J.F. Peters, C. Henry, S. Ramanna, Reinforcement learning with patternbased rewards. in proceding of forth International IASTED Conference. Computational Intelligence (CI 2005) Calgary, Alberta,Canada, 4-6 July 2005, 267-272.

[5] M. J. Cafarella, A. Y. Halevy, D. Z. Wang, E. Wu, and Y. Zhang. Webtables: exploring the power of tables on the web. PVLDB, 1(1):538–549, 2008.

[6] M. Franklin, A. Halevy, and D. Maier. From databases to dataspaces: A new abstraction for information management. SIGMOD Rec., 34(4):27–33, Dec. 2005.

[7] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research, 47:253–279, 2013.

[8] Mazhar Hameed, Usman Qamar, Usman Akram, "Business Intelligence: Self Adapting and prioritizing database algorithm for providing big data insight in domain knowledge and processing of volume based instructions based on scheduled and contextual shifting of data" IEEE, 2016

[9] Marc G Bellemare, Joel Veness, and Michael Bowling. Investigating contingency awareness using atari 2600 games. In AAAI, 2012.

[10] Marc G. Bellemare, Joel Veness, and Michael Bowling. Bayesian learning of recursively factored environments. In Proceedings of the Thirtieth International Conference on Machine Learning (ICML 2013), pages 1211–1219, 2013

[11] I. Konstantinou, E. Angelou, D. Tsoumakos, and N. Koziris. Distributed indexing of web scale datasets for the cloud. In Proceedings of the 2010 Workshop on Massive Data Analytics on the Cloud, MDAC '10, pages 1:1–1:6, 2010.

[12] R.S. Sutton, A.G. Barto, and Reinforcement Learning: An Introduction (Cambridge, MA: The MIT Press, 1998).

[13] C. Watkins, "Learning from Delayed Rewards", PhD thesis, Cambridge University, Cambridge, England, 1989.

[14] Vanden Berghen Frank, Q-Learning, IRIDIA, Universit Libre de Bruxelles.

[15] Tom O'Neill,Leland Aldridge,Harry Glaser, Q-Learning and Collection Agents, Dept. of Computer Science, University of Rochester.

[16] L.P. Kaelbling, M.L. Littman, A.W. Moore, Reinforcement learning: A survey Journal of Artificial Intelligence Research, 4, 1996, 237-285.