Cognitive Intelligent Decisions for Big Data and Cloud Computing in Industrial Applications using Trifold Algorithms

Shitharth Selvarajan[1], Hariprasath Manoharan[2], Rakan A. Alsowail[3] , Achyut Shankar[4] , Saravanan Pandiaraj[5] , Carsten Maple[6], Wattana Viriyasitavat[7]

[1] School of Built Environment, Engineering and Computing, Leeds Beckett University, LS1 3HE Leeds, U.K.
[3]Department of Electronics and Communication Engineering, Panimalar Engineering College, Poonamallee, Chennai, India
[3] Computer Skills, Self-Development Skills Department, Deanship of Common First Year, King Saud University, 11362, Riyadh, Saudi Arabia
[4] WMG, University of Warwick, United Kingdom
[5] Computer Skills, Self-Development Skills Department, Deanship of Common First Year, King Saud University, 11362, Riyadh, Saudi Arabia
[6] Secure Cyber Systems Research Group (SCSRG), WMG, University of Warwick, Coventry, UK
[7]Chulalongkorn Business School, Faculty of commerce and accountancy, Chulalongkorn University, Thailand;

[*]Corresponding author – s.selvarajan@leedsbeckett.ac.uk

Abstract

In contemporary real-time applications, diminutive devices are increasingly employing a greater portion of the spectrum to transmit data despite the relatively small size of said data. The demand for big data in cloud storage networks is on the rise, as cognitive networks can enable intelligent decision-making with minimal spectrum utilization. The introduction of cognitive networks has facilitated the provision of a novel channel that enables the allocation of low power resources while minimizing path loss. The proposed method involves the integration of three algorithms to examine the process of big data. Whenever big data applications are examined then distance measurement, decisions mechanism and learning techniques from past data is much importance thus algorithms are chosen according to the requirements of big data and cloud storage networks. Further the effect of integration process is examined with three case studies that considers low resource, path loss and weight functions where optimized outcome is achieved in all defined case studies as compared to existing approach.

Keywords: Big data; Cloud storage; Cognitive channel; Intelligent decisions

1. Introduction

In the context of burgeoning industrial advancements, it is imperative to conduct a thorough analysis of each data set that is transmitted through discrete channels. The increasing volume of data across various industrial sectors necessitates the processing of diverse data types to their respective destinations. The primary requirement for various forms of data is the ability to compare current and past operational states, and to identify and rectify any changes that may be present. The proposed approach involves generating big data representations that can be applied to real-time industrial applications, which are critical for making significant decisions. It is imperative to store data according to their designated types as big data is being processed. Therefore, the utilization of cloud computing storage methodology is employed to accommodate larger quantities of data with enhanced security measures. The outcome of employing this storage methodology is to make informed decisions by effectively utilizing all available data and resources. In order to enhance the efficacy of the necessary measures, it is imperative to transform them into an intelligent format. To achieve this, a cognitive intelligent technique has been incorporated into the designed cognitive channel. In the context of business product development, it is imperative to make informed decisions by conducting a comparative analysis of current and past stored values. In such instances, the suggested approach confers significant benefits by enabling the development of a novel methodology that exhibits superior operational efficacy. The combination of big data and cloud computing techniques can be considered a time-saving process.

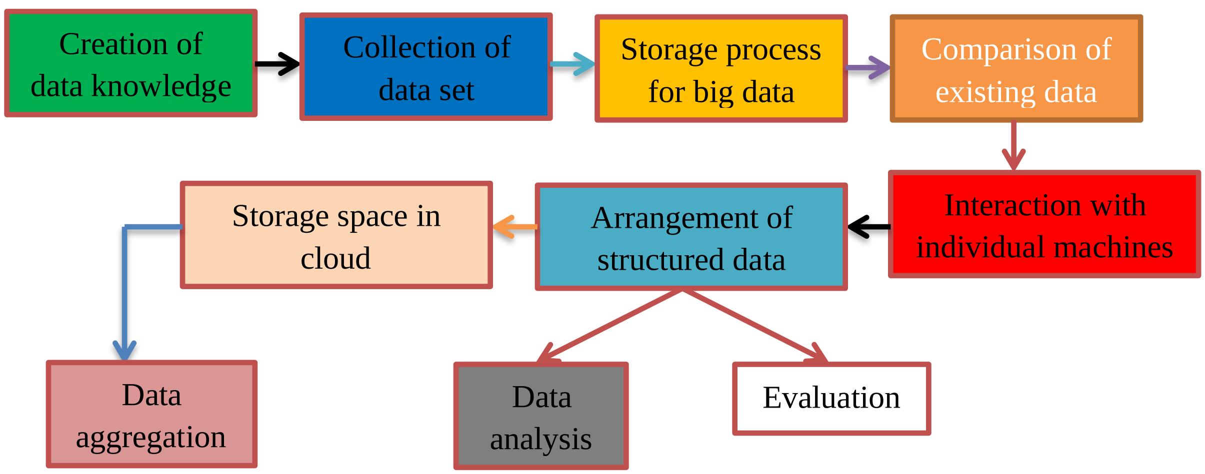

Figure 1 Block diagram of big data representations

However, it is imperative that a structured pattern be defined promptly to ensure the efficacy of this process. The block diagram depicted in Figure 1 illustrates the process of big data and cloud computing, incorporating a cognitive intelligent decision making system. The block diagram representations for big data are constructed in Figure 1 by generating data knowledge for diverse types of data. After the creation of knowledge representations, the data set is collected in a manner that is individualized, with a focus on representing structured data formats. Upon collection of a dataset, it is necessary to eliminate duplicate values that are present throughout the entire dataset in a manner that ensures appropriate allocation of storage space. Therefore, a comparison is made between the current and previous state values, and only the system with a low error factor is permitted to interact with the cloud storage device. Following the comparison of various values, all valid data is systematically organized and allocated uniform storage space. Following this, the data that has been stored will undergo analysis, aggregation, and evaluation in order to facilitate the making of intelligent decisions based on cognitive processes.

1.1 Background and related works

This section delves into the relevant literature pertaining to the utilization of big data and cloud computing for the purpose of making informed decisions. The literature review offers insight into the procedural aspects of implementing big data and its storage methodology through cloud networks. Therefore, many contemporary patterns have been identified and the limitations associated with parametric configuration have been addressed through various learning methodologies. The fundamental requirements for implementing data flows, as outlined in [1], are associated with a cost-effective model. This allows for the reuse of all big data processed across multiple applications, utilizing various deployment models. Furthermore, it has been observed that when resources are reused, identical outputs are generated, leading to decreased processing efficiency and increased storage complexity. Furthermore, the dynamic mode of big data operations involves multiple migration activities, which facilitates the estimation of unique psychological features [2]. One of the primary challenges encountered during cloud computing migration is the limited utilization ratio, which restricts the extent to which certain segments can be leveraged. Consequently, other segments may only be utilized at a discounted rate. To enhance the efficacy of big data storage in conjunction with cognitive intelligent decision-making systems, a decentralized offloading technique is implemented through various algorithms [3]. The computational offloading process necessitates the big data to operate in a collaborative manner by utilizing wireless and multiple computing resources, thereby accomplishing an equitable allocation policy. However, the challenge of distributing resources equally among all data in the network persists as a daunting task that is considerably challenging to accomplish.

This study provides an analysis of big data and cloud computing networks in the context of intelligent decision-making processes, wherein cognitive nodes generate large volumes of data [4]. Evidently, cognitive decisions are executed in consideration of distinct resource allocation systems, whereby the service role is predicated on a platform-based work model. The reference roles in cloud services have a significant impact on the paradigm shift, resulting in scalable analysis within the designated network. In the field of big data analysis, the scalability of a system is a crucial factor. When dealing with large and varying data sets, processing them using identical segments can prove to be challenging. Thus, it is recommended to represent the data set with a unique size at the initial stage. This approach can significantly reduce the complexity of the subsequent processing steps. In the aforementioned case, an analysis was conducted on the performance of cognitive networks in order to make informed decisions and attain self-sustaining data operations through the use of statistical values [5]. Given the ability of cloud computing networks to maintain functionality across diverse operating conditions, it is imperative that they be allocated equivalent spectrum bandwidth for optimal performance. However, it should be noted that data transmitted in a specific sequence through a cloud network using the same spectrum bandwidth may not necessarily result in intelligent decision-making. Furthermore, certain real-time observations necessitate the use of wide area networks due to the extensive spectrum ranges that the transmitted large data must encompass [6]. When transmitting data over long distances, it is necessary for the receiver to choose an appropriate signaling rate for the transmission, as the actual rate may be lower than anticipated. It is imperative to analyze the signaling rate of big data without any feature extraction, as the dynamic characteristics of such data cannot be altered even if it is clustered with multiple classification values.

The process of feature extraction involves the mapping of parameters onto representations of big data, and the measurement of index values through learning modes [7]. The storage of data in low complexity states with time-varying data properties becomes essential as big data facilitates specific data indexing. During the aforementioned examination process, many cognitive systems are unable to make informed decisions due to the fluctuation of data over time. Moreover, in the realm of big data operations, it is highly probable that the primary user may surpass decision threshold values for data limit if cognitive intelligent decisions are implemented [8]. When the parametric values of power and voltage are significantly elevated, it is plausible that external noise factors may affect the transmission of large data through a specific channel. If the channels are employed with appropriate functionalities, it is feasible to utilize the complete spectrum without any interference. On the other hand, decisions regarding scheduling tasks can also be applied to representations of large data sets, which necessitate a suitable time frame for transmission, as stated in reference [9]. The creation of a schedule is contingent upon the examination of other cognitive nodes within the entire network, as individual time periods are provided. When examining individual nodes in relation to time periods, the sequence of data can facilitate appropriate delivery and storage. However, if the sequence of the entire data is altered, achieving both storage and delivery becomes significantly more challenging. Hence, a technique for integrating data across diverse industrial stations within manufacturing systems can be offered [10]. In the context of manufacturing systems, it is imperative to ensure that cloud units are provided with consistent data, and in the event that duplicate data is detected, it must be promptly eliminated from the system. The abstraction of data units leads to an improvement in the response of data functions and integration procedures. Table 1 presents a comparative analysis between the proposed methodology and prior research in the field.

Table 1 Existing vs Proposed

Reference	**Methods/Algorithm**	**Objectives**			
		A	**B**	**C**	**D**
[11]	**Deep reinforcement learning with computational offloading**	✓		✓	
[12]	**Fading channel representations**	✓	✓		
[13]	Cognitive channel sensing with reinforcement algorithm	✓		✓	
[14]	Square law integrated energy detection for big data analysis		✓	✓	
[15]	Explainable artificial intelligence for clustered big data	✓			✓
[16]	Multitask neural network for industrial applications			✓	✓
[17]	Black box artificial intelligence for interpretability analysis	✓	✓		
[18]	Prototype design for big data with sensing techniques		✓	✓	
Proposed	**Trifold algorithm for big data and cloud computing with intelligent decisions**	✓	✓	✓	✓
A: Minimization of power; B: Minimization of resources; C: Path loss reduction; D: Individual weight functions for big data					

1.2 Research gap and motivation

As stated in the background and related literature, numerous methods have been developed to ensure the effective operation of big data and cloud computing technologies. However, the incorporation of multi-objective optimization within established network configurations has not been addressed in any of the current models. Despite the availability of appropriate protocols for managing large datasets, certain astute decisions are not executed within the designated timeframe, resulting in diminished efficacy of storage infrastructures. At a later point in time, even if determinations have been made, certain data is replicated during each service interval, which is regarded as a significant limitation in all established approaches. In addition, the utilization of big data approximations is implemented within the specified cognitive channel, resulting in a reduction of input power across all big data representations.

Therefore, in order to address the aforementioned limitations, the proposed approach incorporates multi-objective parametric representations that encompass a cognitive channel framework. The cognitive framework facilitates the making of intelligent decisions during the transmission phase of big data, thereby ensuring the implementation of a distinctive storage technique. The system model incorporates a trifold algorithm comprising K-nearest neighbor, decision tree, and deep Q-learning algorithm in conjunction with parametric evaluations. This algorithm is designed to store all big data representations of industrial processes in cloud computing systems.

1.3 Major contributions

The major objective of proposed method on big data and cloud computing with cognitive based intelligent decisions is to provide a unique framework for data representations. Therefore the parametric objective functions for determined big data functions are as follows.

- To develop a unique cognitive channel that reduces the power consumed from big data and storage networks.
- To provide low resource for transmitting big data thus reducing cognitive spectrum allocation for allocated time periods.
- To minimize the path loss and weight functions of every data that is present in the system before storing as trifold algorithms are integrated.

1.4 Paper organization

The rest of the section in the paper is organized as follows: Section 2 provides mathematical representations of big data and cloud computing networks for taking intelligent decisions. Section 3 integrates a trifold algorithm with defined system model using pseudo code implementation. Section 4 focuses on experimental outcomes with different scenarios that are based on individual parameters. Finally, Section 5 concludes the proposed method with directions on future work.

2. Proposed system model

The proposed methodology involves the development of a system model utilizing a cognitive intelligent framework. This approach is necessary to effectively manage spectrum utilization in various industries that transmit large volumes of data for storage in the cloud. The system model being presented comprises several parameters that require thorough examination prior to the integration of the intelligent framework into the design, thereby ensuring optimal utilization of the framework. Therefore, the cognitive framework's channel model can be precisely defined through the utilization of Equation (1) in the subsequent manner.

$$cognitive_{channel} = min \sum_{i=1}^{n} \frac{P_1+..+P_i}{P_j} + G_n(i) \tag{1}$$

Where,

$P_1+..+P_i$ indicates power of cognitive source of each data

P_j denotes total power of input cognitive systems

$G_n(i)$ represents the gain of transmitting data

The first equation elucidates the crucial importance of individual data power in large-scale industrial applications of big data. In such contexts, the absence of even a single data point within a given time period can significantly complicate the monitoring process. Therefore, the average input power can be expressed in the following manner.

$$P_{input} = min \sum_{i=1}^{n} \frac{\delta_i I_{in}}{\sigma_i} \tag{2}$$

Where,

δ_i denotes modification of power in transmitted data

I_{in} represents interference that is present in big data

σ_i indicates the mean value of transmitted data

Equation (2) stipulates that in order to establish appropriate links for cloud storage, it is imperative that total input functions with data exhibit minimal interference, even in the presence of high data volumes. Furthermore, the utilization of fog devices enables the complete implementation of cloud storage, incorporating channel functions. The transmission capacity can be mathematically represented by Equation (3).

$$T_c = max \sum_{i=1}^{n} C_i w_i \tag{3}$$

Where,

C_i indicates individual cognitive channel capacity

w_i denotes total allocated load to each cognitive nodes

According to Equation (3), it is imperative in industrial settings to augment the load of individual nodes to ensure efficient transmission of data without any spectrum wastage. The allocation of resources plays a crucial role in determining spectrum utilization in cognitive intelligent systems. To represent the total number of allocated resources, Equation (4) is employed.

$$R_i = min \sum_{i=1}^{n} \frac{w_i}{R_1 + .. + R_n} \tag{4}$$

Where,

$R_1 + .. + R_n$ denotes total number of data resources that are provided to cloud

According to Equation (4), the optimal allocation of resources in service functions can lead to a reduction in total workload. Therefore, the calculation of resource demand can be achieved by utilizing individual weight functions, as denoted in Equation (5).

$$D_r(i) = min \sum_{i=1}^{n} (\alpha_1 + .. + \alpha_i) \times (R_i - ut_i) \tag{5}$$

Where,

$\alpha_1 + .. + \alpha_i$ represents weight function of individual data

ut_i denotes cognitive data utility functions

According to Equation (5), the weight functions of each data must be minimized in order to maintain a low total utility function. Thus, the expression for the data detection threshold is denoted by Equation (6) in the following manner.

$$threshold_{data} = min \sum_{i=1}^{n} \rho_{in} - G_n \tag{6}$$

Where,

ρ_{in} denotes detection device threshold value

According to Equation (6), the minimization of data threshold values is necessary in order to ensure the optimal utilization of spectrum for demand purposes. Furthermore, it is imperative to minimize the total path loss by reducing the delay in spectrum utilization, as specified in Equation (7).

$$L_p = min \sum_{i=1}^{n} \frac{dist_1 + .. + dist_i}{dist_t} \tag{7}$$

Where,

$dist_1 + .. + dist_i$ denotes distance of individual data

$dist_t$ indicates total distance

According to Equation (7), the provision of separate distance measurements in cloud networks can result in complete reduction of path loss. Since a cognitive channel model is designed for industrial applications the parametric changes for processing big data must be represented with composite objective functions. Hence Equations (8) and (9) is formulated as follows.

$$f_1(x) = min \sum_{i=1}^{n} D_r, R_i, L_p \tag{8}$$

$$f_2(x) = max \sum_{i=1}^{n} T_c \tag{9}$$

The combined case for integrating with trifold algorithms for analysing data mapping procedures can be formulated using Equation (10) as follows.

$$obj_t = f_1(x) + f_2(x) \tag{10}$$

The optimization objective is to minimize demand, resources, and path loss while maximizing big data transmission capacity, based on a min-max criterion. The attainment of objective functions necessitates their integration into an optimization algorithm. Consequently, a trifold algorithm has been selected and incorporated to facilitate the realization of optimal outcomes.

3. Methods/Optimization algorithms

The optimization algorithm is implemented in a three-fold manner in this section, whereby the proposed system model is integrated into each step's functions. Furthermore, all three algorithms incorporate the parameters that facilitate the system's operation. The optimization of spectrum utilization is a crucial aspect in cognitive radio, with the aim of efficiently transmitting and storing the vast amount of available data within the network. Hence, the K-nearest algorithm, utilized for distance measurement, has been selected. Given that data is partitioned for multiple decisions, it is imperative to introduce certain data branches, thereby necessitating the use of a decision tree. The integration of triple data authentication has become a standard practice for ensuring proper authentication of big data stored in the cloud. The integration of the proposed model and optimization algorithms is imperative due to the ability to define an approximation function with a collection of data maps [15-17]. This results in the transmission of data following cognitive actions and a dedicated path, thereby enabling the mapping of both input and output functions. If optimization algorithms are not incorporated into the proposed model, the accuracy and speed of operation of cognitive intelligent systems during the big data training phase may be compromised. Therefore, effective routing mechanisms are utilized to manage all issues related to big data and cloud storage. Furthermore, in the process of optimization, a clearly defined dataset that is oriented towards the future is established, ensuring that each datum originates from a predetermined feasible point. The following is a comprehensive exposition of trifold algorithms.

3.1 *K*-nearest neighbor

A data proximity procedure is necessary in cognitive intelligent big data handling networks for industries, thus the K-nearest algorithm has been incorporated into the system model. In industrial settings where the goal is to monitor the behavior of each device through the transmission of large amounts of data, it is necessary to group individual data points [19]. This approach enables the utilization of storage techniques that can effectively separate the data while maintaining a high level of privacy. The utilization of big data for cognitive purposes in various industries is commonly categorized as either classification or regression. To achieve this, the K-

Nearest Neighbor (KNN) algorithm is employed to assign identical data points to the nearest cognitive nodes. Subsequently, the cognitive intelligent system proceeds to eliminate the given information and subsequently establishes a mapping of alternative information to facilitate further data processing, without any intervening disruptions. If any issues arise with data classification following the utilization of the spectrum, it is possible to establish a set of labels to aid in spectrum identification. The classification challenges of big data in cloud storage, as stated above, are applicable solely to discrete scenarios. In contrast, regression can be appropriately defined for continuous cases through adequate training procedures. The major advantage of KNN in big data and cloud storage applications is that assumption of data is not made and decisions are made only during action period. Hence most of the time that is allocated for data training is saved where new data points are established at high security. The proposed method incorporates cognitive intelligent decision-making during each transmission period, as opposed to neglecting training errors for data. The calculation of the distance for each data node can be performed by utilizing Equation (11) in the following manner.

$$distance_{KNN} = max \sum_{i=1}^{n} (u_i - v_i)^2 \quad (11)$$

Where,

u_i, v_i indicates individual data points

According to Equation (11), it is necessary for all data points to exhibit distinct differences, thereby enabling the establishment of new data points with equivalent distances. Equation (12) outlines two necessary conditions for determining points in the context of big data.

$$\vartheta_i = \begin{cases} distance_{KNN} = 0 \; for \; u = v \\ distance_{KNN} = 1 \; for \; u \neq v \end{cases} \quad (12)$$

The constraints specified in Equation (11) imply that the introduction of new data points is precluded when the distances between existing points are equalized. In the absence of equalization of points, it is possible to elevate data points without any cognitive intervention. Equation (13) provides a functional representation of KNN.

$$z_i = \sum_{i=1}^{n} f(a_i) \quad (13)$$

Where,

a_i represents neighboring functions

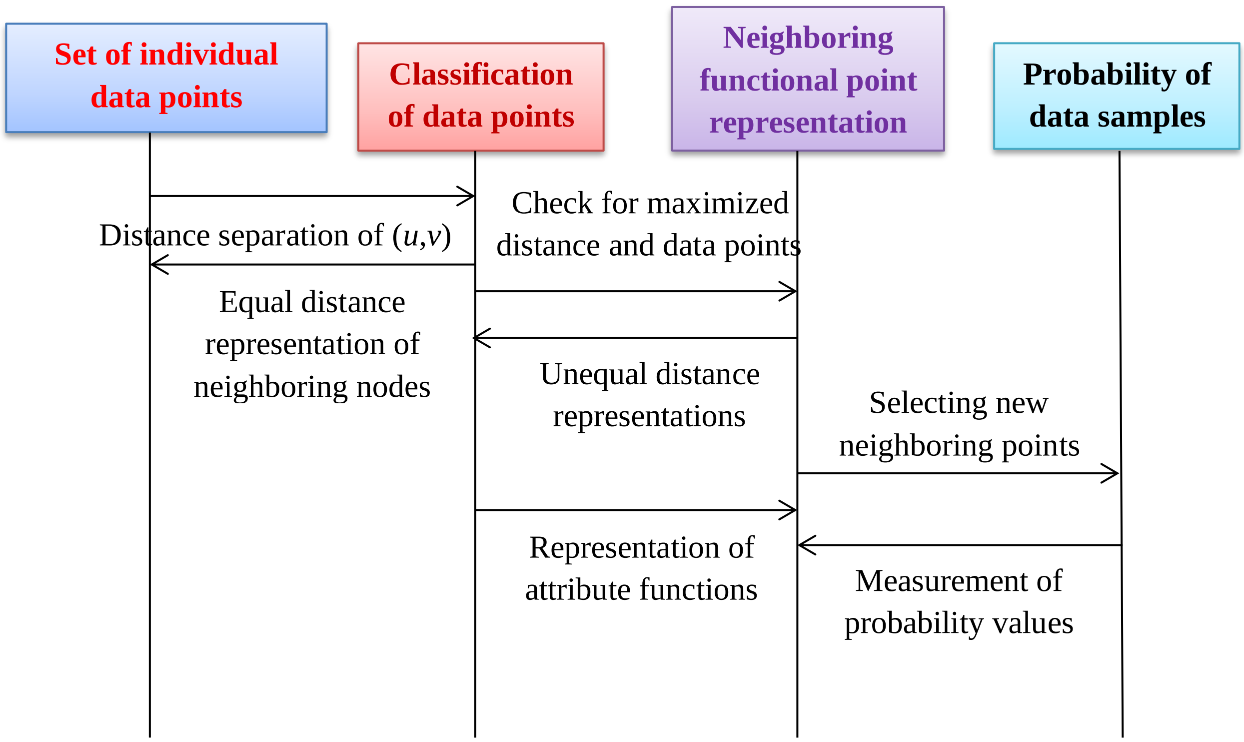

Figure 2 KNN for big data and cognitive intelligent networks

Equation (13) describes that the main function z_i is followed by close neighbors which are indicates with a_i. Hence the sum of overall neighboring sequence can be represented using Equation (14) as follows.

$$SN_i = arg \sum_{i=1}^{n} prob_d(i) \tag{14}$$

Where,

$prob_d(i)$ indicates probability of data samples

The block representations of KNN are illustrated in Figure 2 and the pseudo code for implementation is as follows.

Algorithm 1 *K*-nearest neighbors

Begin PROCEDURE KNN

Given

u_i,v_i: Individual data points

a_i: Total number of neighboring cognitive nodes

for i=1:n **do**

1. $distance_{KNN}$ for measuring the distance of big data separation
2. a_i for choosing appropriate neighboring nodes

end

else

for all i=1:n **do**

3. $prob_d$ for observing the probability of data samples

end

end PROCEDURE

According to Equation (14), the probability of embedding in cognitive space is contingent upon the establishment of data value attributes. It is necessary to decrease the intricacy of adjacent nodes by employing Equation (15) in the following manner.

$$complexity_i = min \sum_{i=1}^{n} s_d(i) \times 2 \tag{15}$$

3.2 Decision tree

Given the increasing volume of data being transmitted and stored in the cloud, it is imperative to adopt a multi-faceted approach to decision-making. Specifically, it is advisable to select individual decision mechanisms based on the branches of each tree segment. The decision tree algorithm, akin to KNN, follows a two-step procedure. The first step involves data classification, while the second step involves the degradation of individual data that is transmitted through multiple paths. The decision tree algorithm employs three distinct data node representations, namely internal, external, and root nodes. In a decision tree, each internal node serves as an independent testing mechanism, ensuring that each data point is evaluated using appropriate measurements [20]. Nevertheless, the determination is not made by any internal node, thereby enabling sub-branches to make the storage decision, as denoted by class labels. The major advantage of decision tree algorithm is that no prior knowledge about individual attributes are needed thus every data will learn using past sample values. In decision tree algorithm only graphical representations of nodes are present therefore it is much easier for external user to provide a particular decision. In the decision tree, two distinct types of gain values are observed, namely purity and impurity of data, which are transmitted instantaneously. Moreover, the intricacies pertaining to error measurements are ascertained in the suggested approach through the utilization of the decision tree algorithm. Equation (16) can be utilized to express the mathematical representation of unadulterated data in cloud storage networks through the implementation of cognitive intelligent systems.

$$Gain_i = max \sum_{i=1}^{n} Ent_{wa} - Ent_f \quad (16)$$

Where,

Ent_{wa} represents entropy of data weights

Ent_f indicates entropy of feature set data

According to Equation (16), in order to optimize the gain of big data representations, it is necessary to carefully analyze the individual data weights within each branch. Equation (17) can be utilized to express the decisions based on weighted averages.

$$Ent_{wa} = P(y_i) \times P(n_i) \quad (17)$$

Where,

y_i, n_i represents decision factor with affirmative and non-affirmative values

If Equation (17) contains a greater number of non-affirmative values, then the weighted average of the output functions will be multiplied by two. The quantification of impurities in large datasets is carried out as a means of mitigating such occurrences.

$$D_{impure} = 1 - min \sum_{i=1}^{n} BS_i \quad (18)$$

Where,

BS_i denotes binary data separations

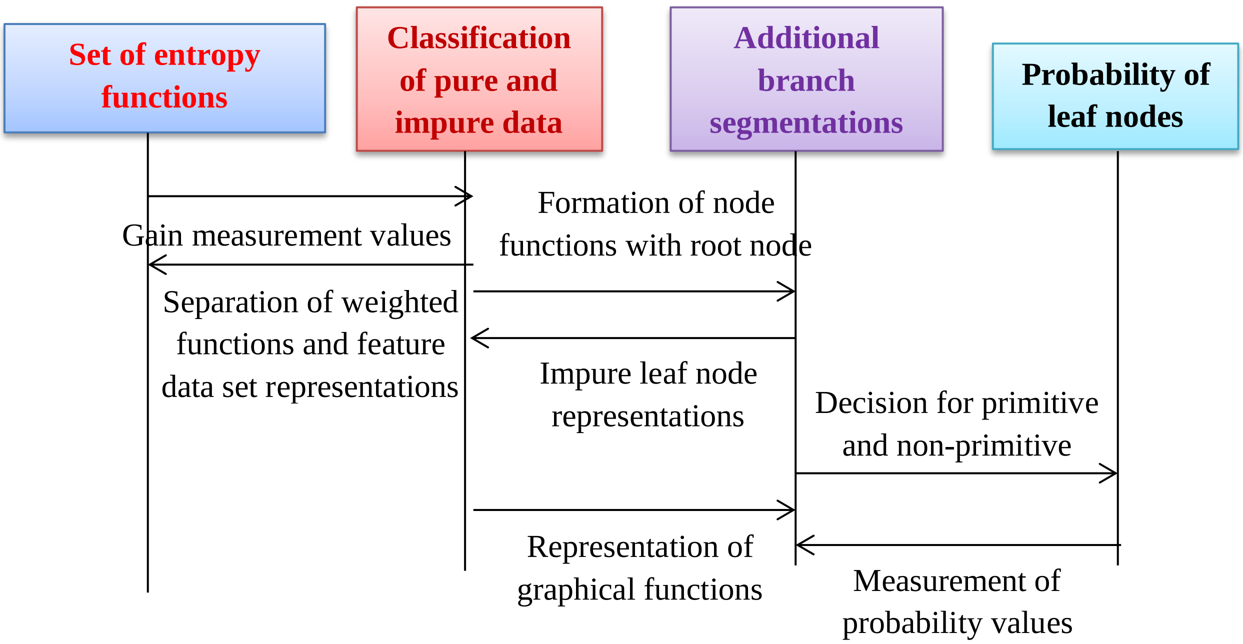

Figure 3 Decision tree for big data and cognitive intelligent networks

According to Equation (18), it is necessary to decrease the distance between binary values, which results in the direct reduction of impure data. Nevertheless, certain terminal nodes that are directly linked to primary branches may encounter some degree of impurity, denoted by Equation (19) in the subsequent manner.

$$L_{impure} = min \sum_{i=1}^{n} \frac{Gain_i}{Ent_{wa}} \tag{19}$$

The pseudo code of decision tree is follows and block representations are illustrated in Figure 3.

Algorithm 2 Decision tree

Begin PROCEDURE DT

Given

Ent_{wa},Ent_f: Input entropy functions

$P(y_i)$, $P(n_i)$: Probability of primitive and non-primitive nodes

for i=1:n **do**

1. D_{impure} for measuring the impure data
2. L_{impure} for observing additional branch impure segments

end

else

for all i=1:n **do**

3. $Gain_i$ for measuring total information gain

end

end PROCEDURE

3.3 Deep Q-learning

As previously stated, the utilization of reward functions in big data operations enables a user to make intelligent decisions during transmission and reception scenarios. The introduction of the deep Q-learning model aims to assess environmental scenarios and communicate appropriate actions to various stakeholders for the storage of specific data [21]. This Q-learning model incorporates both states and actions to amalgamate data from similar cases. To facilitate the process of making critical cognitive decisions, it is imperative to construct a Q-table that updates both the state and actions upon the transmission of each piece of data to the receiver. At the onset of a data sequence, the determination of the data state is contingent upon discrete agents. In the event of an overload, the storage system is promptly altered. The positive and negative states of a user determine all consequential actions that must be taken. Hence, it can be observed that maximizing the strength of data leads to a positive impact on data transmission, without any alteration in storage values. The initial time period is used to verify the data estimates for both the current and optimal states, which allows for the attainment of expected values in the cognitive decision-making process. The major advantage of deep Q-learning model is that it is completely based on error and trial method therefore every data that is transmitted to cognitive nodes will have some error and it is rectified at later stage. In addition the nature of environment in cognitive intelligent decisions are based on stochastic process more amount of data rewards can be achieved. The mathematical formulation of deep Q-learning algorithms can be expressed as follows.

$$Q(\beta,\gamma) = max \sum_{i=1}^{n} Reward_i(\beta,\gamma) \tag{20}$$

Where,

β,γ indicates reward factors of every data

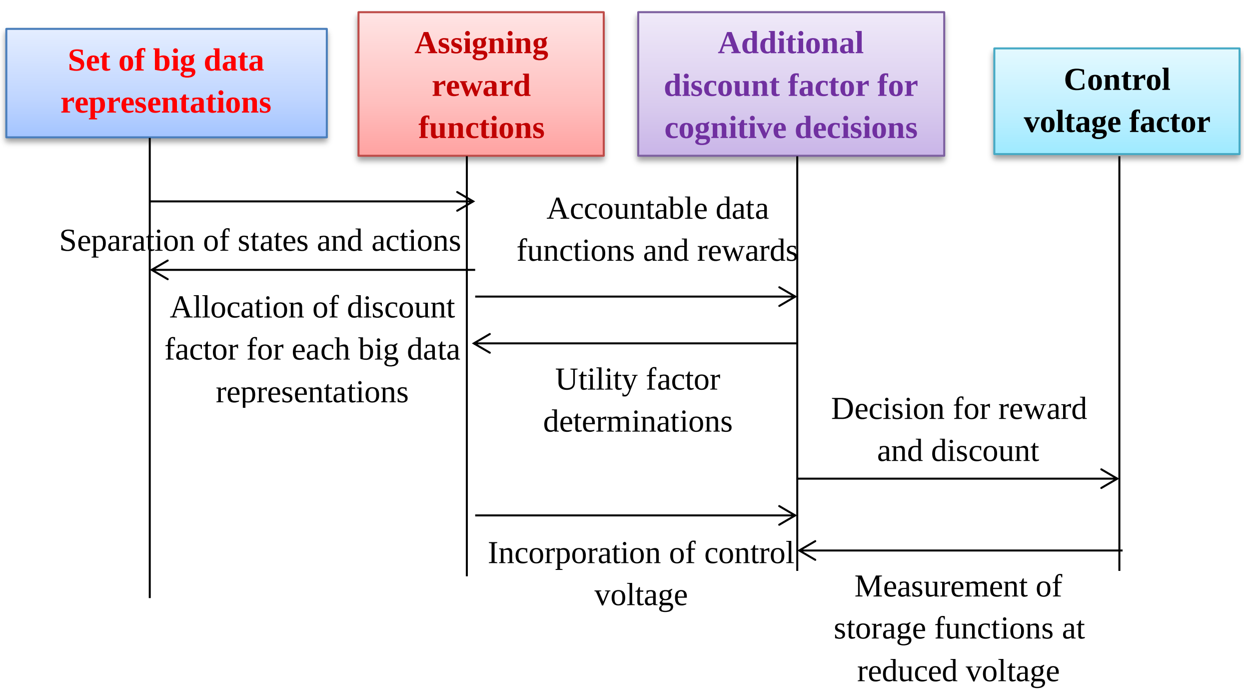

Figure 4 Deep Q-learning for big data and cognitive intelligent networks

The block representation of deep Q-learning is depicted in Figure 4 and the pseudo code representations are as follows.

Algorithm 2 Deep Q-learning

Begin PROCEDURE DQL

Given

β,γ: Big data functions

$Reward_i$: Reward functions for input data representations

for i=1:n **do**

1. ut_i for providing utility data functions
2. dis_f for assigning maximum discount factor

end

else

for all i=1:n **do**

3. $V_c(i)$ for providing maximum control voltage

end

end PROCEDURE

According to Equation (20), it is necessary to maximize the reward functions of each data, which subsequently results in the derivation of a personalized discount factor. Therefore, the discount factor is expressed in the following manner.

$$dis_f = \sum_{i=1}^{n} ut_i(\beta, \gamma) \tag{21}$$

Where,

ut_i describes data utility functions

In order to represent utility functions, it is necessary to provide control voltage as outlined in Equation (22), as follows.

$$ut_i = min \sum_{i=1}^{n} V_c(i) \tag{22}$$

Where,

$V_c(i)$ represents control voltage for data functions

4. Results

This section presents the simulation and description of the real-time examination results for a proposed system model that integrates a trifold algorithm. The aim is to comprehend the impact of cognitive systems in facilitating intelligent decision-making. The proposed methodology establishes a relationship between the impact of the system model and parametric testing, resulting in outcomes that are solely dependent on the cognitive channel that has been developed. The attainment of desired results involves the utilization of a dataset comprising a 1400-bit sequence. This dataset interfaces with seven distinct cloud systems to facilitate optimal storage while ensuring a high level of security. After identifying the dataset, potential strategies for integrating it with the established channel while operating under low power conditions are established. At the outset, it is observed that certain errors are elevated owing to the presence of undefined data sets. Consequently, in subsequent processing stages, all undefined data sets are eliminated. Furthermore, a comparison is conducted with prior state outputs, thereby enabling the determination of the current operational state with minimal impact. The rationale for selecting a 1400-bit sequence is based on the channel's design limitations, which can only accommodate a maximum of 1600 bits. Therefore, to prevent data congestion, the proposed method employs a dataset of this size. The resources allocated are designed to be uniform across all datasets, ensuring equitable decision-making within a specified threshold. Three case studies have been selected to examine the impact of parametric variables based on the defined objective function. The significance of these case studies is presented in Table 2.

Case study 1: Allocation of power and resources

Case study 2: Path loss representations

Case study 3: Determination of individual weight function

Table 2 Significance of proposed case studies

Case studies	Importance/ Significance
Allocation of power and resources	Demonstrate the data in cognitive channel with identical allocation capacity
Path loss representations	Reduce the data loss in a particular path before storage
Determination of individual weight function	Utilize the data in a proper way before taking intelligent decision

4.1 Discussions

The aforementioned scenarios have been simulated using MATLAB software to facilitate a visual representation for enhanced comprehension. The graphical representations are processed using MATLAB, while the large data sets are processed using cognitive nodes in a network simulator. Subsequently, all values are determined. The comprehensive depiction of each case study is presented below.

Case study 1: Allocation of power and resources

It is imperative to allocate minimal power during the transmission phase for every large dataset that is being transmitted to the recipient. If the allocated power exceeds the necessary amount, it may lead to inefficiency, hence it is crucial to provide an appropriate amount of power to the relevant data segments. Furthermore, it is imperative to appropriately represent additional resources within the cognitive channel to prevent data transmission failure at various stages. Efficient allocation of resources for a given dataset enables informed decision-making within the designated framework. Hence, in the cognitive channel, power allocation is performed for all accessible data, which is segregated by the overall input power, and the gain of input cognitive systems can be evaluated. A higher gain in transmitted data suggests that sufficient resources have been allocated to ensure that each data transmission reaches the receiver without any external interference. Furthermore, the assigned data resources are depicted for each data, thus individual weight functions are also furnished. The simulation results pertaining to the allocation of resources are depicted in Figure 5.

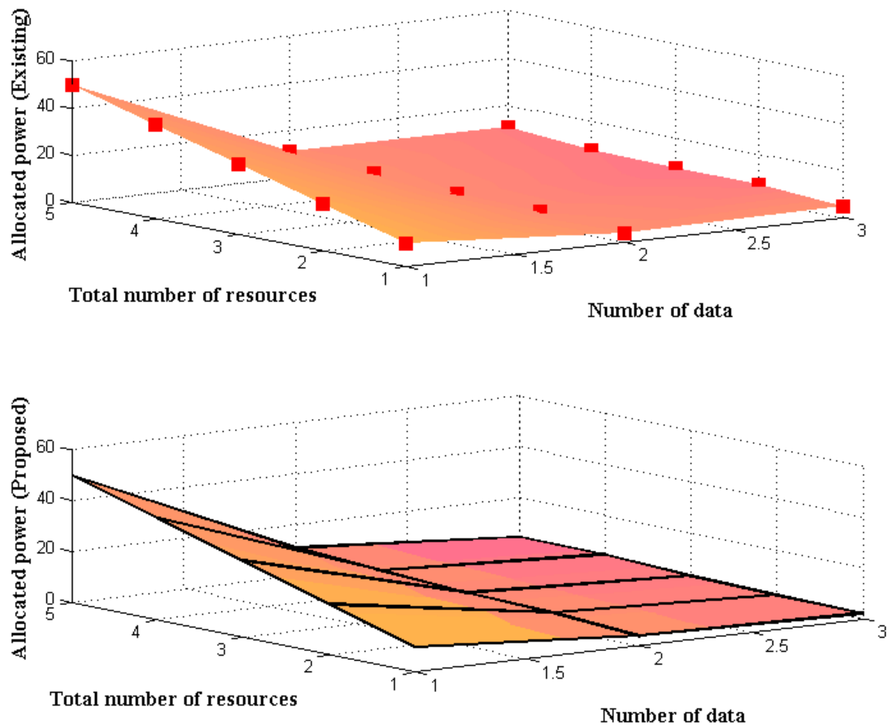

Figure 5 Maximum allocated power for big data transmission

According to the findings presented in Figure 5, it is feasible to minimize the allocation of power and other resources in order to facilitate efficient transmission of large amounts of data. Additionally, the storage of such data can also be carried out using minimal resources. The cognitive channel representations for each data point, specifically 10, 20, 30, 40, and 50, were provided in order to investigate the impact of power allocation. According to the given dataset, the total number of allocated resources will be 4, 6, 7, 9, and 11, respectively. This is because each dataset requires a minimum of one-fourth of the resources for the given data set, as stated in reference [2]. Consequently, the allocation of power is facilitated through the provision of resources in a manner that enables cognitive channels to make informed decisions. The proposed method exhibits an allocated power of 2.14, 2.58, 3.22, 4.67, and 4.94, while the existing method allocates power for each data at transmission state with the same data set and resources, resulting in a maximized range of 4.58, 6.91, 7.02, 8.12, and 10.89. Therefore, by operating at lower power levels, it is possible to gather and retain large amounts of data in a cloud-based platform.

Case study 2: Path loss representations

The present case study delineates the path loss representations for transmitted data with cognitive channel, utilizing individual distance measurements. Consequently, distance values are measured for each datum and subsequently segregated from the total distance value, enabling the determination of individual path losses. Moreover, loss representations indicate that a specific set of data is relocated from one position to another, thereby impacting the current channel path. If a path loss occurs in cognitive intelligent decision making systems, the depletion of the allocated spectrum will ensue. In order to make informed decisions, it is imperative to minimize losses in specified pathways. This can be accomplished through the establishment of appropriate system models that are associated with distance measurements. The simulation results for path loss representations are presented in Figure 6.

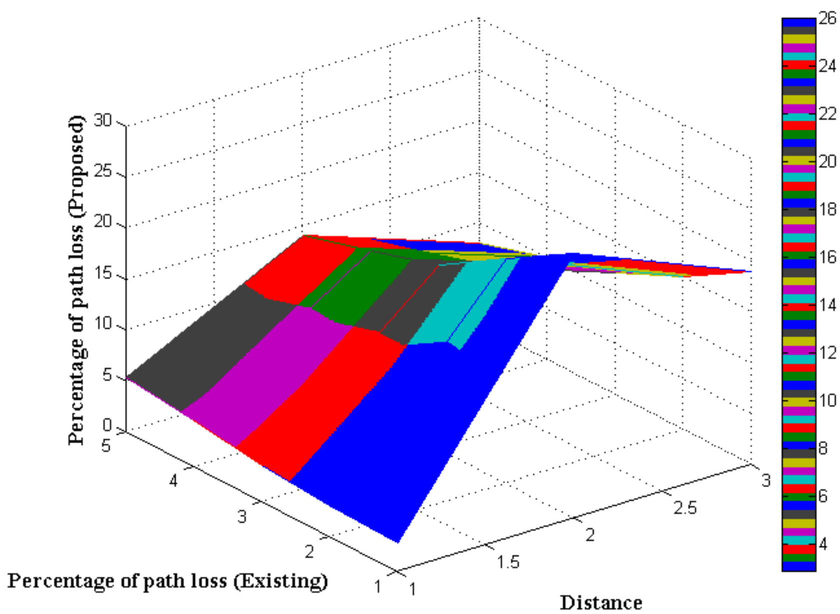

Figure 6 Measurement of path loss with total distance allocation

The findings from Figure 6 demonstrate that the proposed method minimizes path loss representations in comparison to the existing approach, as it effectively maintains the appropriate distance between two data points. To ascertain the path loss distance, the data points are taken into account, specifically 2.88, 3.12, 3.65, 4.57, and 5.33, while ensuring that a minimum distance separation is maintained for each data point. The total distance separation for all data is deemed to be 15. The distance representation outlined above indicates that the proposed method exhibits path loss values of 19, 15, 12, 10, and 8. In contrast, the existing approach maintains path loss values of 26, 22, 18, 16, and 14. Furthermore, it should be noted that the path loss mentioned above is calculated solely prior to data storage, whereas the overall path loss may fluctuate during the transmission phase. Moreover, the reduction in path loss results in the attainment of balanced data types. This ensures stable operations for the entire big data through the use of trifold algorithms.

Case study 3: Determination of individual weight function

In order to efficiently store transmitted data, it is necessary to determine a weight function for each data set received. This weight function serves to reduce the demand for data retrieval. Therefore, in this particular case study, weight functions are computed for each data point and corresponding utility functions are subsequently aggregated. To determine suitable weight functions, individual data is incorporated and replicated using available resources. Therefore, based on the available resources, the level of utilization is decreased. This occurs when a low weight factor is employed, allowing for uninterrupted transmission of all data to the receiver. Furthermore, in cases where weight functions exhibit significantly lower values for a specific dataset, it may be possible to optimize utility functions up to a certain degree. The utility functions require that all data transmitted and stored in a given block must be fully utilized without incurring any penalty. Provided that complete utility factors are utilized effectively, it is possible to furnish reward functions aimed at minimizing supplementary weight functions. The simulation output of individual weight functions allocated to each data is presented in Figure 7.

The findings from Figure 7 demonstrate that the proposed method minimizes individual weight functions in comparison to the existing approach. This suggests that all transmitted data is effectively utilized. The utility of real-time outcomes data has been demonstrated through values of 56, 64, 70, 78, and 85, with the remaining data being utilized in a partial manner. The utilization function mentioned above involves the examination of individual weight functions. In the proposed method, these weight functions are found to be 821, 702, 589, 357, and 129. On the other hand, the existing method yields individual weight functions of 1278, 1156, 1028, 887, and 543. Therefore, by utilizing weight functions that are allocated appropriately, it is possible to transmit the complete dataset with high reward functions. The reward factors are allocated to each data based on comprehensive resource utilization. Consequently, not all transmitted data is taken into account during the intelligent decision-making process. Only cognitive intelligent decision-making processes consider low data weight functions with high utilization factors.

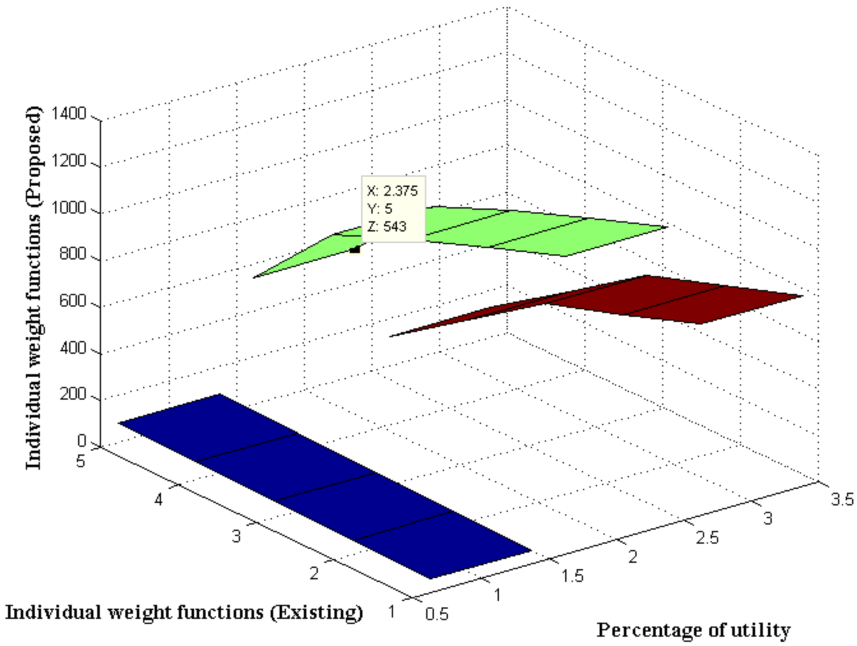

Figure 7 Data weight functions and utility values

4.2 Performance analysis

To analyse the best and worst case scenario on incorporating cognitive computations it is necessary to perform big data analysis with respect to input size. Since in industrial applications more amount of data is observed and stored in cloud the complexity of data will be much higher and in the proposed method scaling metrics is used for determining all three algorithmic cases. Further complete limiting behaviour of industrial applications is achieved in order to prevent the data from moving towards infinite solutions thus reduced complexities can be observed in this case. Moreover both upper and lower bounds for complete data is maintained thus a feasible point is determined at this case thereby boundary limitations of big data is also reduced. As a result of the boundary limitations robustness of all three algorithms are determined with best epoch which is indicated in Figure 8 where group of iterations is combined.

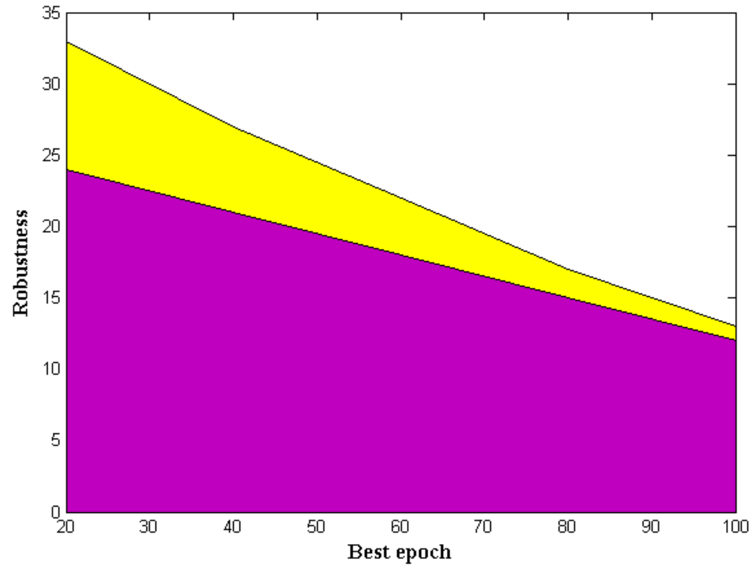

Figure 8 Comparison of robustness for proposed and existing approaches

From Figure 8 it is obvious that robustness of proposed approach is reduced using deep Q-learning as compared to traditional optimizations that considers big data for industrial applications. The major reason for reductions in robustness is that individual cloud platforms are used for proper storage thereby with the help of cognitions it is possible to protect data in an individual way. To verify the outcome of robustness best epoch with 20 step variations are considered and reduced threshold for robustness conditions are observed. At initial state the robustness for considered data is reduced from 9,6,4,2 and 1% in case of proposed method whereas in existing approach [2] the robustness is observed to be 24,21,18,15 and 12% respectively.

5. Conclusions and future work

The significance of large-scale data and cloud-based storage networks lies in their ability to offer distinct identification of all ongoing activities across diverse industrial applications. The advent of big data has enabled the monitoring of timely activities without reliance on external factors. The proposed methodology involves the analysis of big data and cloud storage networks, wherein cognitive intelligent decisions are utilized to determine the transmission and reception of data. Additionally, the system model has been formulated to examine the various parameters of big data that are evident in the cognitive channels produced, while simultaneously minimizing resource allocation. The proposed method offers a primary advantage in the form of cloud networks, which enable the transmission and storage of monitored big data with minimized power and other resources. The level of path loss in big data is typically elevated, thus necessitating the integration of a trifold algorithm that comprises of shortest distance mapping, branch-based decisions, and deep Q-learning. The challenge of spectrum utilization for big data persists when cognitive networks make intelligent decisions. To address this issue, a proposed method incorporates a short distance mapping procedure to solve the spectrum utilization problem. Furthermore, a crucial feature of the proposed approach is the consideration of individual weight factors for each data point to facilitate maximum transmission distance. This implies that specific data networks can be assigned additional weights, while some pre-defined data may be excluded. The present study utilizes a system model and integrated trifold algorithm to investigate three case studies that demonstrate the efficacy of big data and cloud storage systems in conjunction with cognitive networks to make intelligent decisions. The results indicate a reduction in resource utilization by 60 percent when compared to existing methods. Furthermore, the proposed system exhibits a reduction in path loss by 75% in comparison to the current approach, resulting in a stable overall optimized performance of approximately 80%. In the future, it is possible to expand the proposed work by incorporating a greater number of parametric estimations utilizing advanced integration algorithms to ensure the provision of secure features.

5.1 Policy implications

If the proposed method is implemented in real time then industrialists will have the benefits of observing various data where time boundary decisions can be made. Further complete behaviour can be analysed with cognitive actions thereby solving all inconsistencies and missing data at high accuracy rates. Moreover, the identical industrial products can be observed by sorting entire data thus utilizing trifold algorithms for data cleaning functionalities.

Ethical Statements:

Funding: “This research received no external funding”.

Institutional Review Board Statement: “Not applicable”.

Informed Consent Statement: “Not applicable.”

Data Availability Statement: Data will be made available based on request.

Acknowledgments: The authors of this study extend their appreciation to the Researchers Supporting Project number (RSPD2023R544), King Saud University, Riyadh, Saudi Arabia."

References

[1] Chaturvedi S, Tyagi S, Simmhan Y. Cost-Effective Sharing of Streaming Dataflows for IoT Applications. IEEE Trans Cloud Comput 2021;9:1391–407. https://doi.org/10.1109/TCC.2019.2921371.

[2] Chen Y, Qiu Z. Cloud Network and Mathematical Model Calculation Scheme for Dynamic Big Data. IEEE Access 2020;8:137322–9. https://doi.org/10.1109/ACCESS.2020.3009675.

[3] Josilo S, Dan G. Joint Management of Wireless and Computing Resources for Computation Offloading in Mobile Edge Clouds. IEEE Trans Cloud Comput 2021;9:1507–20. https://doi.org/10.1109/TCC.2019.2923768.

[4] Berisha B, Mëziu E, Shabani I. Big data analytics in Cloud computing: an overview. J Cloud Comput 2022;11. https://doi.org/10.1186/s13677-022-00301-w.

[5] Kozić N, Blagojević V, Ivaniš P. Performance analysis of underlay cognitive radio system with self-sustainable relay and statistical csi. Sensors 2021;21. https://doi.org/10.3390/s21113727.

[6] Takyi K, Bagga A. Real-time application clustering in wide area networks. Comput Electr Eng 2020;85:1–15. https://doi.org/10.1016/j.compeleceng.2020.106691.

[7] Lemes AL, Guimarães DA, Masselli YMC. System-to-distribution parameter mapping for the Gini index detector test statistic via artificial neural networks. Comput Electr Eng 2020;85:106692. https://doi.org/10.1016/j.compeleceng.2020.106692.

[8] Gulzar W, Waqas A, Dilpazir H, Khan A, Alam A, Mahmood H. Power Control for Cognitive Radio Networks: A Game Theoretic Approach. Wirel Pers Commun 2022;123:745–59. https://doi.org/10.1007/s11277-021-09156-x.

[9] He J. Decision Scheduling for Cloud Computing Tasks Relying on Solving Large Linear Systems of Equations. Comput Intell Neurosci 2022;2022. https://doi.org/10.1155/2022/3411959.

[10] Chen X. Big Data Integration Method of Mathematical Modeling and Manufacturing System Based on Fog Calculation. Math Probl Eng 2021;2021. https://doi.org/10.1155/2021/9987714.

[11] Xu S, Guo C. Computation offloading in a cognitive vehicular networks with vehicular cloud computing and remote cloud computing. Sensors (Switzerland) 2020;20:1–28. https://doi.org/10.3390/s20236820.
[12] Altrad O, Muhaidat S. A new mathematical analysis of the probability of detection in cognitive radio over fading channels. Eurasip J Wirel Commun Netw 2013;2013:1–11. https://doi.org/10.1186/1687-1499-2013-159.
[13] Yau KLA, Goh HG, Chieng D, Kwong KH. Application of reinforcement learning to wireless sensor networks: models and algorithms. Computing 2015;97:1045–75. https://doi.org/10.1007/s00607-014-0438-1.
[14] Lorincz J, Ramljak I, Begusic D. Algorithm for evaluating energy detection spectrum sensing performance of cognitive radio MIMO-OFDM systems. Sensors 2021;21:1–22. https://doi.org/10.3390/s21206881.
[15] Hassija V, Chamola V, Mahapatra A, Singal A, Goel D, Huang K, et al. Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence. Cognit Comput [Internet]. 2024;16:45–74. Available from: https://doi.org/10.1007/s12559-023-10179-82.
[16] Foggia P, Greco A, Roberto A, Saggese A, Vento M. Identity, Gender, Age, and Emotion Recognition from Speaker Voice with Multi-task Deep Networks for Cognitive Robotics. Cognit Comput. 2024
[17] La V, Vincenzo G, Marco M, Giancarlo P. An eXplainable Artificial Intelligence Methodology on Big Data Architecture. 2024
[18] Chatterjee S, Roy A, Roy SK, Misra S, Bhogal MS, Daga R. Big-Sensor-Cloud Infrastructure: A Holistic Prototype for Provisioning Sensors-as-a-Service. IEEE Trans Cloud Comput 2021;9:1323–34. https://doi.org/10.1109/TCC.2019.2908820.
[19] Almagrabi H, Alshareef AM, Manoharan H, Mujlid H, Yafoz A, Selvarajan S. Empirical Compression Features of Mobile Computing and Data Applications Using Deep Neural Networks 2022;2022.
[20] Hasanin T, Manoharan H, Alterazi HA, Srivastava G, Selvarajan S, Lin JCW. Mathematical approach of fiber optics for renewable energy sources using general adversarial networks. Front Ecol Evol 2023;11. https://doi.org/10.3389/fevo.2023.1132678.
[21] Khalil U, Ahmad A, Abdel-Aty AH, Elhoseny M, El-Soud MWA, Zeshan F. Identification of trusted IoT devices for secure delegation. Comput Electr Eng 2021;90:106988. https://doi.org/10.1016/j.compeleceng.2021.106988.