Journal of Intelligent & Fuzzy Systems 45 (2023) 2625–2640
DOI:10.3233/JIFS-231389
IOS Press

A Heuristic Approach to Improve the Data Processing in Big Data using Enhanced Salp Swarm Algorithm (ESSA) and MK-means Algorithm

M.R. Sundarakumar[a,*], D. Salangai Nayagi[b], V. Vinodhini[c], S. VinayagaPriya[d], M. Marimuthu[c], Shajahan Basheer[e], D. Santhakumar[f] and A. Johny Renoald[g]
[a]*School of Computing Science and Engineering, Galgotias University, Greater Noida, Uttar Pradesh, India*
[b]*Department of CSE, New Horizon College of Engineering, Bengaluru, India*
[c]*Department of CSE, Sona College of Technology, Salem, Tamilnadu, India*
[d]*Department of ECE, St. Josephs College of Engineering, Chennai, India*
[e]*Department of Computer science and Engineering, Jain University, Kanakapura, Bengaluru*
[f]*Department of CSE, CK College of Engineering and Technology, Cuddalore, Tamilnadu, India*
[g]*Department of EEE, Erode Sengunthar Engineering College, Erode, Tamilnadu, India*

Abstract. Improving data processing in big data is a delicate procedure in our current digital era due to the massive amounts of data created by humans and machines in daily life. Handling this data, creating a repository for storage, and retrieving photos from internet platforms is a difficult issue for businesses and industries. Currently, clusters have been constructed for many types of data, such as text, documents, audio, and video files, but the extraction time and accuracy during data processing remain stressful. Hadoop Distributed File System (HDFS) is a system that provides a large storage area in big data for managing large datasets, although the accuracy level is not as high as desired. Furthermore, query optimization was used to produce low latency and high throughput outcomes. To address these concerns, this study proposes a novel technique for query optimization termed the Enhanced Salp Swarm Algorithm (ESSA) in conjunction with the Modified K-Means Algorithm (MKM) for cluster construction. The process is separated into two stages: data collection and organization, followed by data extraction from the repository. Finally, numerous experiments with assessments were carried out, and the outcomes were compared. This strategy provides a more efficient method for enhancing data processing speed in a big data environment while maintaining an accuracy level of 98% while processing large amounts of data.

Keywords: Hadoop distributed file system, latency, throughput, query optimization, hash algorithms clustering

*Corresponding author. M.R. Sundarakumar, School of Computing Science and Engineering, Galgotias University, Greater Noida, Uttar Pradesh, India. E-mail: sundar.infotech@gmail.com.

ISSN 1064-1246/$35.00

Table for Abbreviation and symbol

Terms & symbols used	Description
HDFS	Hadoop Distributed File System
ESSA	Enhanced Salp Swarm Algorithm
RDBMS	Relational database management systems
SHA	Secure Hash Algorithm
SSA	Salt Swarm Algorithm
ACO	Any Colony Optimization
GA	Genetic Algorithm
PCA	Principle Component Analysis
BGA	Binary Genetic Algorithm
BBCS	Binary Binomial Cuckoo Search
bGWO	Binary Grey Wolf Optimizer
BCSO	Binary Competitive Swarm Optimizer
BCSA	Binary Crow Search Algorithm
CH	Cluster Head
ELD	Economic Load Dispatch
HC	Hill Climbing
PSO	Particle Swarm Optimization
GSA	Gravitational Search Algorithm
GWE	Grey Wolf Enhancement
MWOA-SPD	Modified Whale Optimization Algorithm for Spam Profile Detection
WOA	Whale Optimization Algorithm
SAABC	Self-Adaptive Artificial Bee Colony
CS	Cuckoo Search
GSA	Gravitational Search Algorithm
BOA	Butterfly Optimization Algorithm
ABC	Artificial Bee Colony
MKM	Modified K-Means algorithm
MB	MegaBytes
FCM	Fuzzy C-Means
ms	Milli seconds
$\cup$	Union
/	Division
$\sum$	Sum
∂	Data points
Δ	Data points
$\in$	Implies

1. Introduction

Because of its massive volume and diverse nature, cloud storage has several restrictions over a vast network. As a result, the created data will be kept in a vast cluster network equipped with a massive repository system known as big data processing. In certain cluster networks, processing time and accuracy are not up to expectations. Despite the employment of sophisticated and contemporary approaches in large data processing, extraction efficiency is depleted at network nodes. The HDFS system enables large amounts of data to be stored in a scale-out memory storage system that does not support shared architecture. The map-reduce approach is used to handle data from a repository as well as user needs. Since cluster networks have restricted parameters for extracting data, all internet apps communicate their data bi-directionally from the user to the system [1–5]. Several sectors and businesses rely on big data to manage their real-world application data repositories and make decisions based on it for their company plans with analytics. Strong processing engines are required to extract reliable data in the shortest amount of time possible for consumers [32–39]. In large data processing, answering queries to obtain data from repositories was critical [6–8].

Yet, the multiple queries designed to process data in a system may result in a delay in the accuracy of their extraction. To address this issue, query optimization is a method that plays an important role in data processing at each cluster node. To reduce network conflicts during transmission, cluster nodes often store the same categories of data [9–11,40]. Relational database management systems (RDBMS) provide the fundamental structure to obtain original data; however, as the volume is large, more queries are required to obtain the data. Delay and throughput characteristics are influenced at that moment [12–14,43]. Several sophisticated technologies for large data processing, such as Hadoop, Spark, and Flume, are available on the market. Yet, the programming created for retrieving data from repositories was a key idea when developing programmes as queries.

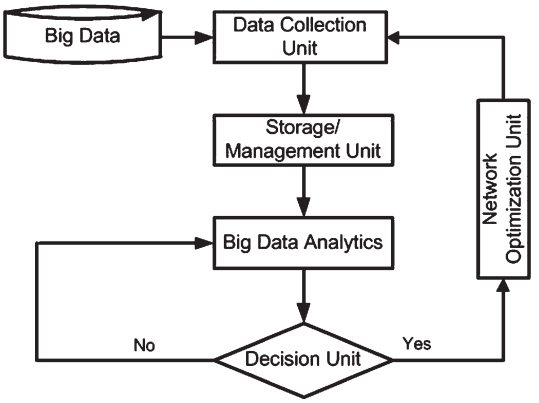

Fig. 1. Big data processing units.

Therefore, various languages, including Java, C#, and Python, must be utilized in the field to build the map reduction query as well as other extraction queries [15–17, 42]. Figure 1 describes the units working for big data processing on larger networks.

1.1. HDFS-Map reduce techniques

The phases of data processing must begin with the collection of large amounts of data from various resources on the internet and other sources. Collected data is saved in the HDFS structured file system, which is replicated on their client nodes. It is again preprocessed with its characteristics, and SHA (Secure Hash Algorithm) hash values are created [18, 41, 45–50]. Feature extractions are detected via the network based on their metadata and are then processed during transmission [51–56]. Clusters are constructed in accordance with the categories of data held in the network; the arrangements were made using map-reduce techniques on the bigger network. The same method must be repeated using network queries and an optimization technique such as the salp swarm algorithm to select the most accurate data from the multi-objective solutions. In general, in a distributed environment, databases and their values are spread among cluster nodes with replication factors on the HDFS-Map-Reduce framework, resulting in inaccurate data being transmitted over the network.

1.2. Salp swarm algorithm (SSA)

A large-scale repository houses databases and many sorts of datasets. Queries assist users in retrieving information from these databases as needed. Longer queries, on the other hand, will cause the processor to extract the data with high latency and sluggish processing speed. To address this issue, query optimization has developed many techniques.

The Salt Swarm Algorithm (SSA) [19–26] is one of the most popular algorithms for getting good results from a large number of solutions in a meta-heuristic fashion. The main aspect of SSA is feature selection using the closed frequent item set, which will generate entropy for each dataset assigned in different places [57–62]. First, the entire dataset is obtained and randomly assigned access points. The data set is partitioned into subsets, and each place in the repository is allocated to all subsets. Every time, it computes the fitness value for each salp and updates it as a chaotic map on the sequence as needed. These updates often alter the positions of all stations visible on the network map, after which the assigned stations send their information to the central controller [27–29].

As the points on the map are altered, the sequence of salp positions is automatically updated to construct the chain. This chain process has been continuously improved in larger network cluster nodes, so that if any node failure or single-point failure occurs, it will be immediately rectified and addressed by the salp swarm algorithm [30, 31].

Eventually, the threshold value that fulfils the constraints of the dataset that has been obtained will offer the best subset from the many subsets. If not, the procedure is repeated until the exact subset is obtained. Initially the dataset taken for the research has loaded in this approach and the salp positions are selected randomly chosen for creating the population. Then each salp position is set to feature subset for classification. Calculate the fitness of each salp and it is updated the sequences of the chaotic maps. Once done all salp positions are updated else this is working iteratively until it has done. Figure 2 denotes the steps in feature selection for SSA.

2. Related works

[1] suggested Quantized Salp Swarm Algorithm for feature selection and given various strategic methods to extract the features from the data sets. However, the limited datasets can be accessed from the whole data set of repository it consumes more time for data processing. A Feature Selection Using New Version of V-Shaped Transfer Function for Salp Swarm Algorithm in Sentiment Analysis method to extract the data from the repositories [2], which is taken from

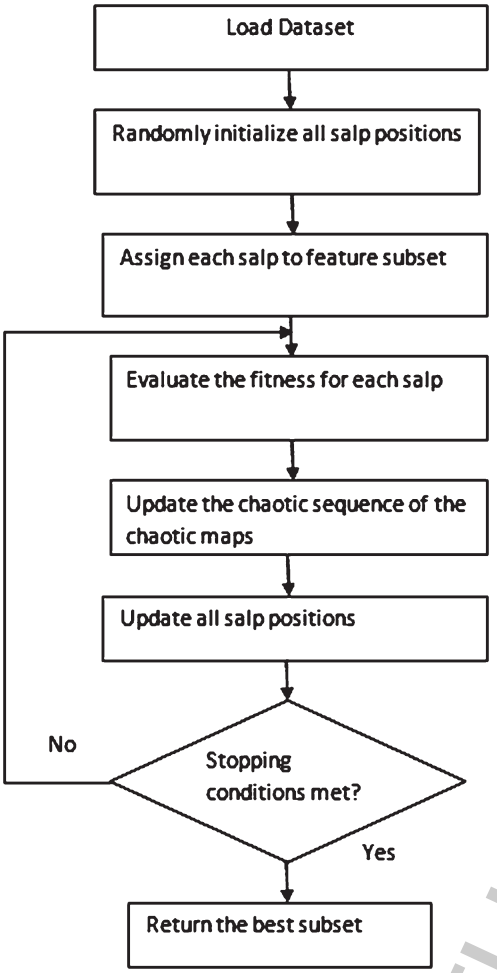

Fig. 2. Feature selection steps in SSA.

the different sources. This kind of functions is used to develop the speed of the data proceeding at huge repositories. But accuracy levels are remains not sufficient during online assessments.

[2] developed Hyper-heuristic salp swarm optimization of multi-kernel support vector machines for big data classification for feature classification techniques and it is used for deriving exact data from the larger datasets on the clusters. [3] suggested Improved k-means clustering algorithm for big data based on distributed Smartphone-neural engine processor to implement the cluster creation methods for developing the data processing speed effectively. [4] suggested an ACOGA algorithm with HDFS map reduction strategy to increase query optimization in big data systems. The query optimization problem and its solutions must be addressed using Any Colony Optimization and Genetic Algorithm concepts. Clusters are formed using K-Means methods, while individual clusters are formed using Normalized K-Means clustering for all forms of data. For security reasons, the SHA 512 algorithm is utilised to produce hash values for generated Metadata. It yielded reliable data processing values in a large data environment, and query optimization shows user-required data within the time frame specified.

[5] offered an examination of composite websites and their contents from online sources. The adaptive environment generates web ranking optimization principles with minor adjustments to the Salp Swarm Algorithm. It deals with multi-objective solutions to fitness outfitting issues that arise in vast networks. It is accomplished through the use of Principle Component Analysis in image processing and Fuzzy C-Means algorithms in cloud storage ideas. The changes made in SSA will affect the solutions corrected on pictures collected from its cloud clusters.

[6] and [7] offered complicated multi-modal solutions in a bigger network with several benchmarks used in data processing. Random opposition-based learning along with diverse leadership strategies must be used to properly develop the overall process. In addition, simulated annealing is performed on the same processing as the SSA technique. The CEC-2015 standard is used to classify multimodal challenges and their solutions.

[8] presented the Binary Swarm Algorithm (BSSA) for feature selection with hybrid data transfer. The processing system addressed issues such as low system performance and high power consumption. To forecast the precise pictures from the repository, PCA and rapid independent component analysis are utilised. This paper's approaches for query optimization include the Binary Genetic Algorithm, Binary Binomial Cuckoo Search, Binary Grey Wolf Optimizer, Binary Competitive Swarm Optimizer, and Binary Crow Search Algorithm.

[9] developed a levy battle to enhance SSA in feature selection and high dimensionality situations. Using SSA principles, query optimization is performed using a met-heuristic technique. Despite this method cannot manage large amounts of data, SSA was enhanced with iSSA by adding new features to the previous one. The entire system runs on huge cluster nodes to efficiently analyse data utilizing query optimization for reliable results. To prevent less secure routing concerns and energy consumption issues, [10] presented Node Replacement Based Energy Optimization Using Enhanced Salp Swarm Algorithm. It also handles node failure in network

clusters; if any node fails instantly, it is replaced by another node/neighbor node. Cluster Head activities use the NS2 simulator to monitor and control all nodes in the cluster. It produces accurate results by combining the findings of improved SSA with nodes connected via a network. E2SA aids in predicting the precise outcomes of the CH's many solutions.

[10] presented mutation strategies for low convergence rate problems and falling sub-optimal solutions. These mutations use different combinations of relationships as an input parameter and are tested against 23 common benchmark issues utilizing statistics and convergence curves. Gaussian mutations are examined, and subset searching is balanced with the mutation outcomes achieved by the benchmark systems. For mutations, Gaussian Cauchy and levy battle schemes are utilised, and only SSA can predict the correct result from these.

[11] A suggested a unique feature selection strategy for some redundant and irrelevant characteristics that appeared in distinct clusters. Sets are subdivided into subsets in this strategy to separate the features detected in the system. Finally, the projected characteristics are balanced based on the subsets found on the network system using SSA. Though it is difficult to forecast the correct one from the different solutions, it must be built and enhanced using the FS-SSA approach using various attributes contained in the data sets.

[12] A recommended a massive data text clustering approach that successfully forecast the Meta-heuristic optimization technique on fitness difficulties and clustering issues. Typically, building clusters on the network would not cause any problems; nevertheless, the factors evaluated for such clusters while designing a multi-heuristic method make things hard in certain instances. To execute a multi-heuristic optimization technique, this algorithm offered classification and partition clustering.

[13] A proposed a hybrid SSA for economic load dispatch problems on non-convex economic scenarios during population-based load dispatch concerns. The above-mentioned problem is solved using SSA and HC approaches as a single point-based algorithm. This approach predicts ELD issue solutions using a hybrid of SSA and HC. [14] and [15] proposed a Denary SSA for objective issues including feature selection and multiple values. With this approach, solutions/results are always available in various objective methods, making it difficult to select the best one. Nevertheless, denary SSA separates all of the results gathered from diverse resources into single solutions from numerous values based on their characteristics.

[16] presented SSA-based selection strategies for predicting precise answers from repositories. Its study and implementations are focused on hybrid engineering difficulties that emerged in network clusters. Their novel technique of Proportional Scheme-HSSA enables to anticipate the exact one from the numerous items using the selection scheme utilised on it. Hybrid concepts aid in the division of solutions into numerous parts and the selection of the best one among them. [17] developed different SSA algorithms based on local and swarm articles. The meta-heuristic technique is the best strategy for selecting the best solution from the network's various solutions. Moreover, the hill-climbing technique is utilised to estimate the optimum solution among them, however accuracy was lacking in several areas. Query optimization was performed using GA properties found in the populations.

[18] presented a mix of SSA and Particle Swarm Optimization to avoid the optimal exploration and exploitation for each function throughout the network search process. Numerous benchmarks, including CEC 2005 and CEC 2017, have been used to evaluate the outcomes of the hybrid algorithm created. [19] developed improved SSA for high dimensionality and multidimensionality difficulties encountered during data transmission. Orthogonal learning is used to break down local optimal solutions, whereas quadratic interpolation is used to improve the accuracy of global optimum solutions. It was improved using SSA to give remedies to the aforementioned difficulty.

[20] A submitted a big data analytics feature selection methodology to increase data processing speed and time in a bigger network system. In data processing, the K-Means approach is used for clustering, while the Cross-validation K-fold method is used to pick mutation schemes. Grey Wolf Enhancement The query optimization approach methodologies utilised in big data analytics are GWO and PSO, as well as a hybrid Gravitational Search Algorithm (GSA). [21] presented sigmoid SSA for multi-objective optimization on the network using linear and non-linear functions. Perturbation-SSA is a newly suggested approach for developing the sigmoid decreasing function and providing answers to multi-objective network challenges.

[22] A recommended a hybrid optimization approach for delivering solutions for VM (Virtual Machine) deployment in cloud data centres. To give

solutions, their SLA (Service Level Agreement) includes First Fit, Virtual Machine Placement Ant Colony System, and Enhanced Best Fit Decreasing techniques. As discrete multi-objective and chaotic functions, the Sine-Cosine Algorithm and SSA decrease power consumption and wastage of resources. [23] presented the Quantum inspired binary chaotic salp swarm method (QBCSSA) for work scheduling and VM placement difficulties. It also has a multi-objective fitness function for assessing particle fitness. In this method, the quantum-inspired binary chaotic salp swarm algorithm BCSSA is applied to find solutions for the aforementioned problems.

[24] developed a quantum computing technique for dealing with network system complexity, nonlinearity, restrictions, and modelling challenges. Quantum-based SSA is utilised to explore answers to the aforementioned challenge on a bigger network.

Quantum computing is a mechanism for transferring data that uses light sources. The pace between nodes is extremely fast, and no data is lost. [25] suggested a Reliable Enhanced Whale Optimization Algorithm for recognizing the dynamic nature of spammers in networks. Hybrid method The Modified Whale Optimization Algorithm for Spam Profile Detection and hybridizing the Whale Optimization Algorithm for data processing are also explored in length in this work. [26] A recommended a memetic SSA for plant disease detection. However, they are not always provided valid data. Hence, the memetic salp swarm optimization method is utilized to forecast network difficulties in order to obtain accurate results across clusters.

[27] suggested a Map-Reduce System for Big Data Clustering Utilizing the Moth-Flame Bat Optimization and Sparse Fuzzy C-Means approach to solve clustering challenges. It also includes a sparse optimizer for the creation of multi-objective solutions. During data processing, the Sparse Fuzzy C-Means method delivers exact solutions from the numerous solutions generated on the repository. In big data analytics, the aforementioned techniques are utilised for query optimization and to increase data processing efficiency.

[28] A presented a technique for node location in WSNs employing non-GPS architecture for data access in bigger networks. This paper discusses and works out Self-Adaptive Artificial Bee Colony, Genetic Algorithm, Cuckoo Search, Gravitational Search Algorithm, Butterfly Optimization Algorithm, Particle Swarm Optimization, and Artificial Bee Colony algorithms with accurate data transmission results. [29–31] suggested the methods to store the huge data sets in the horizontal memory setup and extracted with data mining algorithms. Hadoop and SPARK are the tools used to succeed this work with lot of features like size, compression method, map reduce, in memory analytics etc. Schedulers are used in map reduce process to re arrange the tasks based on the Compression Elastic Index Search and Map Reduce Based Genetic Algorithm. These methods are improved the data processing speed in big data analytics.

2.1. Research gap

This study proposes increasing the data processing speed in big data analytics to solve challenges with query optimization while pulling data from repositories. Once the data has been gathered from the repository, it must be divided into smaller snippets using data pre-processing.

SSA is a technique for populating best-match results from pre-processed data and determining the best response to a given query. Nonetheless, this must be accomplished by picking the necessary matches and extracting the salps from the entire set of matches. Once a match is detected, it will generate clusters based on the results, but with restricted solutions. This needs to be grouped together based on the properties of the data picked as salps. Though cluster formation is a common task in big data analytics, obtaining the most precise matches from the whole solution is not. ESSA is an upgraded variant of SSA that finds acceptable matches based on the closest values of their attributes, such as distance, length, size, type, and so on. Once the selection is complete, this cluster is generated using the modified K-means method, which changes the Euclidean distance values at each focal point. Finally, the best match result arrives with high accuracy and in less time as a result of the user's inquiry.

3. Proposed methodology

Using two steps, the suggested query optimization approach explains the mechanisms behind data processing enhancement in big data analytics. Query optimization does not take place directly in a data processing unit. Yet, collecting data from many real-time sources is critical from the beginning. In a big data context, the HDFS file system and map-reduce

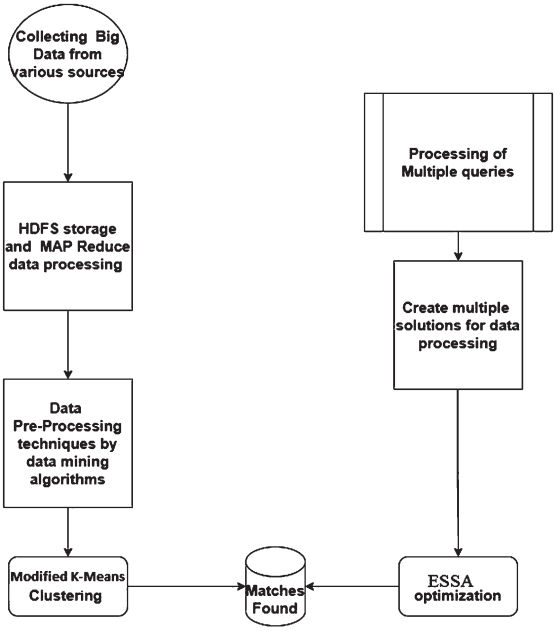

Fig. 3. Proposed methodology architecture.

algorithms are used to gather and store data. The Enhanced Normalized K-Means technique was used to cluster data generated on the network in order to get closed frequent item set values by calculating the Euclidean distance between two nodes. Finally, the parameters called support, confidence, and entropy have been calculated based on Euclidean distances, and the remaining will be considered for the assessment match during optimization. Once a match is detected, it will generate clusters based on the results, but with restricted solutions. This needs to be grouped together based on the properties of the data picked as salps. Though cluster formation is a common task in big data analytics, obtaining the most precise matches from the whole solution is not. ESSA is an upgraded variant of SSA that finds acceptable matches based on the closest values of their attributes, such as distance, length, size, type, and so on. Once the selection is complete, this cluster is generated using the modified K-means method, which changes the Euclidean distance values at each focal point. Finally, the best match result arrives with high accuracy and in less time as a result of the user's inquiry Fig. 3 will describe the block diagram of proposed methodology.

As shown in the diagram above, collected big data can be preprocessed using data mining techniques to remove duplicates and unnecessary data from the input stream. The data must then be saved in HDFS with the metadata hash values computed using the SHA technique. The obtained data's feature selection may be processed using the discovery of a close set of frequent item sets from the repository, as well as their support and confidence values. The results will be used as evaluations, and the same procedure may be repeated on the other end utilizing query optimization. In this case, the ESSA method is employed as a meta-heuristic strategy to obtain precise answers from various objective solutions.

Ultimately, the matches detected in the evaluation record section concerning the system's solutions are used for big data output analytics. The ESSA approach increased the speed of data processing in a big data environment over larger cluster nodes, and their accuracy improved as well.

3.1. Collection of big data

Initially, datasets have collected from various sources such as hospital datasets, employee datasets, YouTube viewer's datasets, etc. Each dataset has its features and parameters for consideration of processing from the repository. The data gathered from different places are written as,

$$K_s = \{K_1, K_2 \ldots\ldots K_n\} \quad (1)$$

Where K_s describe the data set K_n denotes the number of data gathered.

3.2. Data preprocessing

The SHA algorithm handles the preprocessing idea of data acquired from the repository by transforming the hash values of the Meta data characteristics. With these hash values, the map reduction approach will act on the HDFS storage system to keep the data sequences.

3.2.1. Creating hash values

The input bits are computed as a mixture of 8,16,32,64,256,512 depending on the hash value required and the size of the data collection. The data value will be kept in blocks, and each block will consist of 64 bit combinations from the inputs, therefore 128 bits means 2 blocks and 512 bits means 8 blocks. Based on the input data size, a buffer is employed in the hash value construction, and many bits are automatically taken for temporary storage as a buffer. Earlier works initialized the buffer with preset values, however here the buffer executes the data automatically. The generated hash values are denoted as in

each block as follows,

$$U_{A(K)} = \left\{A_{(K1)}, A_{(K2)} \ldots..A_{(Kn)}\right\} \quad (2)$$

where $U_{A(K)}$ is finding the value of the hash function of all data and A (K_n) represents the Hash value of data.

3.2.2. HDFS

Hadoop Distributed File System (HDFS) was created by the Apache Foundation as a framework for commodity hardware and software to store massive files on a repository. Their scale-out approach was created with big size varied format files and used for real-world applications. The Hadoop framework controls the speed and method of data flow, which is then processed using Map Reduce techniques. It is a programming paradigm that is written in either Java or Python and is used by a wide range of sectors and businesses. It generates documents based on key-value pairs from the input formats' metadata. For data processing, HDFS uses two phases: map and reduce. That functions are acted as an instances and will be denoted as follows,

$$A_{(Kn)} = \left[N_f, S_f\right] \quad (3)$$

for all hash value data in the blocks will be accessed in mapping N_f function and reducer in S_f function respectively.

3.2.3. Mapper () function

The primary node serves as the master node, while the remaining nodes serve as data nodes. The master node will assign the job to the data node while the incoming input data stream is initially transformed to hash values. The SHA function is used for this conversion, and the Mapper function divides the input data into little chunks based on key-value pairings. A number of Mappers will be formed to break up the original data before being reduced to a tiny size in comparison to the input size. This function is denoted as,

$$N_f = map\left(A_{(Kn)}\right) \quad (4)$$

where M_f is the output of the mapper function and map () denotes created Mapper.

3.2.4. Reducer () function

The Mapper output is gathered as a partition from the Mapper function and reduced to a small amount of output depending on the Metadata of the input file size.

The number of Mapper generated during the map () function is not identical to the number of Mapper created during the reduce () function. Because the reduce () function is used to merge all of the Mapper outputs to provide consolidated output. It is denoted as the mathematical equation

$$S_f = reduce\left(N_f\right) \quad (5)$$

For all Mapper output S_f is denoted as reducer function output and reduce () function provides the overall output.

3.2.5. Feature extraction

In the first step, data is preprocessed, and the output data is saved in HDFS for analytics and feature extraction using the following criteria. Frequent Closed Item Set refers to the amount of times a data item set appears again. Moreover, Support and Confidence are further parameters examined in order to determine the proportion of transactions that occurred, with confidence having a link with the number of transactions and item sets values. The functions mentioned above were represented by the equations below.

Frequent Closed Set

$$[T_{(cf)}] = \left\{t_{(c1)}, t_{(c2)}, t_{(c3)} \ldots\ldots\ldots t_{(cn)}\right\} \quad (6)$$

Support

$$S_t = P\left(M \cup N\right) \quad (7)$$

where M and N are the item sets

Confidence

$$Q_e = P\left(N/M\right) = \frac{P\left(M \cup N\right)}{P\left(M\right)} = \left[\frac{St}{P\left(M\right)}\right] \quad (8)$$

3.2.6. Entropy values

The entropy value is determined by adding the support and confidence probability values. The numerous data sets have been recognized as the frequent closed item set, which will be chosen at random for taking probability contribution. These entropy values' minimum and maximum values, as well as their differences, will be utilized to create clusters with the Enhanced K-Means method to order the data sequence to the next level.

$$\text{Entropy}\left(S_t\right) = \sum P\left(S_t\right) \log 2 \left\{P\left(S_t\right)\right\} \quad (9)$$

$$\text{Entropy}\left(Q_e\right) = -\sum_{e=0}^{G} P\left(Q_e\right) \log 2 \left\{P\left(Q_e\right)\right\} \quad (10)$$

where $P(S_t)$ and $P(Q_e)$ are the probability value of the support and confidence from the item set selected randomly. Now, it has both min and max values as a result of the frequent closed item set from the repository. Next step is to create a cluster based on the probability values and min, max values using Enhanced K-Means algorithm.

3.2.7. *Modified K-means algorithm*

For cluster formation using prior output values from frequent closed item sets, their support and confidence values are combined with probability contribution factors. In general, the lowest and maximum values of that probability have been considered while performing the normalization procedure to choose the precise number from the output. After the output has been received, it will be loaded into the cluster and prepared for data processing. The key problem in the system was obtaining the probability distributions values to rely on the size and locating various frequent item sets. If the number of occurrences is more, it has several outputs, however if there are few matches discovered as output, the likelihood is very low. So the input file size is also a factor to consider the speed of the data processing in a big data environment. The normalization concept utilized in this approach from the output values followed by,

$$W_\sigma = \frac{A - A_{min}}{A_{max} - A_{min}} \tag{11}$$

Normalized values such as minimum and maximum have been taken from the item sets of data and used to create clusters using the aforementioned algorithm. This number of clusters would have been constructed and their centered beginning values determined using the well-known K-Means technique. This technique is used to effectively build clusters by grouping similar types of format data. The ideal value of the multiple objectives function must be established, and the distance between centered, normalized values must be computed as Euclidean distance. The cluster with the fewest distance values produces the best results. The number of data $A = 1$, 2, 3.... N must be considered while determining normalized values, which will be utilized to compute the Euclidean distance between the distinct clusters centric. That expression followed by,

$$U_{dist} = \sqrt{\sum_{i=1}^{A} (\partial_i - \Delta_i)^2} \tag{12}$$

where ∂, Δ are the data points of two different normalized values which has taken from the input data and U_{dist} will be calculated using the equation number 12. The same procedure is used to determine all data points and cluster centricity. More different points have been considered, and their associated centric values in each cluster were determined using the Equation 12. Also, the final output must be calculated from all of the normalized numbers and distinct locations. The updated K-Means Algorithm is explained in pseudo code below.

Modified K-Means Algorithm – Pseudo Code

Input Given: *Entropy Values from the frequent closed item set*
Output Values: $O = \{O_1, O_2, O_3 \ldots O_n\}$

```
Begin
    Function Modified K-Means Algorithm
        Initialize a, b(minimum and maximum number of iteration)
        Find Normalization values from item set # all data
        Set Initial value O <= ∂, Δ
        for each data do
            min (dist((∂i − Δi)) # between the cluster centroid and data point
        end
        While a<b
            for each i ∈ {1, 2 ... ... ... n} Iteratively check the cluster center
            end
            for each A∞
                min (dist (∂i − Δi))# for various cluster center and data points
            end
            Set a = a + 1
        End While
End
```

3.2.8. Enhanced SSA

The Enhanced-Salp swarm algorithm operates similarly to a salp dissolved in our body, propelled by the water force of the intake particles. Salp swarm concept is also a chain model since it is always functioning like a chain. It is divided into two sections: header and followers. The external water force causes the header to flow inside the body, where it may successfully achieve its objective. The Enhanced-Salp Swarm Algorithm treated the whole salp population as the population, and the best outcome salp is the fitness of everyone after the swarm. Many variables are expressed as particles as n, which acts as a population, and the salp's location is announced as e. To get the order/sequence from this salp chain T it has to get the positions of all salp and their followers movement stored in a two dimensional array.

$$q_s^1 = T_s + r_1 \left(\left(up_s - low_s\right) r_2 + low_s\right), r_3 \geq 0, \tag{13}$$

$$q_s^1 = T_s + r_1 \left(\left(up_s - low_s\right) r_2 + low_s\right), r_3 < 0, \tag{14}$$

Where Q_s^1 the first salp position and that is will be acted as a leader, then T_s is the position of the chain contains the entropy values from the clusters and up_s, low_s are the upper, lower values taken from the chain randomly. All values are taken at s dimension. Remaining r_1, r_2, r_3 the co efficient of the above equations and that will decide the leader information of all salp. The total value of q_s decides the locations of the salp and its get upgraded when chain values have got used randomly. This co efficient parameter decides the SSA's exploration and exploitation values. They are,

$$r_1 = 2e^{-(4k/K)2} \tag{15}$$

where k is the present location and K is the number of maximum iterations happened in the salp chain. The other parameters r_2, r3 are used generated the exploration and exploitation values at the regular intervals such as [0, 1]. The position of the salp may be goes to positive or negative or infinity. That time the two dimensional matrix has stored all the values according to the signs and positions of the salp from chain. They are

$$q_s^p = \frac{1}{2}(q_s^p + q_s^{p-1}) \tag{16}$$

where q_s^p p is the position of the first salp and the next levels are identified based on the updated position of the salp in the chain with the above Equation 16. The values which are considered as a salp have taken from the clusters such as distance between the cluster centric and data point's position's from the cluster center. This structure is considered as a salp chain and swarm selection have established in this method to implement ESSA. Finally, the big data values have arranged from the output of the ESSA in order to get accurate values as matches or assessment records. The pseudo code of the ESSA to get better solution as a result are followed by,

ESSA Pseudo Code

Initialize Salp population and assign up_s, low_s values (upper and lower)
While # condition satisfied (populations are there)
Calculate every salp fitness and their positions
q=first position of salp
Update the salp positions by Equation (13)
    For each Salp (q_s)
        If ($r3 \geq 0$)
            Update position of Salp by Equation (13)
        Else
            Update position of Salp by Equation (14)
        End
    End
        Change the Salp according to upper and lower bound values
End
Return q

Finally, the ESSA output fitness solutions are regarded as query optimal values for the large network clustering data. This data must be retrieved from HDFS storage using a map-reduce framework and Python programming. The huge amount of data stored in the cluster is utilized in real-world application situations that need low latency and fast data processing. ESSA is developing clusters to transform large, big-data cluster data into valuable, accurate data based on customer demands.

Table 1
Performance report of proposed MKM with FCM, K-Means

Data size in MB	Proposed MKM			K-Means			FCM		
	Accuracy %	Sensitivity (%)	Specificity (%)	Accuracy %	Sensitivity (%)	Specificity (%)	Accuracy %	Sensitivity (%)	Specificity (%)
60	92.22	84.52	83.26	86.16	82.26	82.36	82.36	74.18	72.16
70	93.86	87.46	84.54	87.34	84.14	84.25	83.24	77.36	74.87
80	94.92	88.62	87.93	88.27	86.38	86.96	85.34	80.69	82.54
90	95.82	91.34	92.72	90.54	89.17	88.46	88.19	83.45	83.15
100	96.15	92.84	95.28	92.16	90.24	91.23	90.34	85.63	88.63

4. Experimental setup

The performance of the proposed MKM (Modified K-Means Algorithm) and query optimization algorithm ESSA is investigated using an experimental study in the Python framework with HDFS-Map-Reduce set up Master-Client architecture systems. The primary benefits of this configuration are the ability to save massive amounts of big data in a single repository with multiple clusters connected in a larger network. While accessing data via the HDFS environment, the initial input file size of 60 MB will be raised to 120 MB. To build up this environment, a replication factor of 1 : 3 is required, which means one master node and three data nodes. But this strategy is coupled with more nodes on the centralized master server in order to construct clusters for managing large amounts of data across the network system. The ESSA technique will emphasize query optimization, and accurate results of the new suggested algorithm can offer stochastic output on a regular basis.

4.1. Performance evaluation report of parameters

The performance analysis report will be generated using the parameters utilized in the MKM approach and compared to existing techniques such as Fuzzy C-Means (FCM) and K-Means algorithm performances analysis. The precision, sensitivity, specificity, accuracy, retrieval time, execution time, and memory use of the complete system method are used to provide statistical results for analysis.

Originally, the experimental findings were analyzed using accuracy, sensitivity, and specificity characteristics. All MKM, FCM, and K-Means technologies are compared. The data sizes range between 60 MB and 100 MB. The accuracy of the suggested MKM technique was 92–97% while accessing 100 MB of input file size, while sensitivity and specificity were 90–92 and 91–93, respectively. If the data size is lowered to 100 MB, the numbers in the analysis are Enhanced to 94%, 91%, and 90%. The other techniques FCM and K-Means provided the low percentages of 85 and 86 only in the analysis report with the identical data. As a consequence, the findings clearly show that the suggested MKM technique outperforms the other current approaches in terms of performance. Furthermore, the time required for execution and retrieval findings is included in the analysis report, and the comparisons are tallied. The abovementioned analytical report is described in Table 1.

For determining the completion time of data processing in big data analytics, the retrieval time and execution of all strategies, such as suggested MKM, current FCM, and K-Means algorithm approaches, are studied. When the data size is kept to a minimum of 60MB, the execution time is 118 milliseconds and the retrieval time is 283 milliseconds. The current techniques produced 412 ms and 618 ms, respectively, which are faster than the new strategy. Analyses of this type have been developed for different file sizes. When a fresh strategy to optimization is employed, nearly half of them (50%) of their time is saved. The retrieval and execution timing analysis of all approaches is shown in Table 2.

The memory utilization by the various strategies when performing the process in HDFS and the Map Reduce framework must be analyzed for comparison. When the input file size is 100MB, the FCM approach takes up more kilobytes (7480512) during execution time. The K-Means algorithm used 7049984 KB of memory to run. Ultimately, the new Enhanced K-Means method consumed very little memory during execution. About 6549874 KB of memory was used throughout the execution duration. As a consequence, the total analysis result has provided a conclusion about the methodologies evaluated for comparison. The proposed MKM approach requires relatively little time and memory to execute data from the repository while providing excellent data correctness during the data transmission phase. Table 3

Table 2
Retrieval and execution time analysis of MKM, FCM and K-Means

Data size in MB	Proposed MKM		K-Means		FCM	
	Retrieval time (ms)	Execution time (ms)	Retrieval time (ms)	Execution time (ms)	Retrieval time (ms)	Execution time (ms)
60	283	118	412	124	678	139
70	724	134	1495	148	2084	170
80	1398	180	2984	204	4586	241
90	2418	201	5078	238	6429	279
100	3567	275	7544	307	8796	338

Table 3
Memory occupancy during execution

Data size in MB	Proposed MKM (KB)	K-Means (KB)	FCM (KB)
60	5268,346	5786,647	6257,354
70	5624,145	6037,948	6689,327
80	6017,357	6489,453	7038,982
90	6388,489	6824,247	7268,324
100	6549,874	7049,984	7480,512

describes the memory usage of all the techniques considered for comparison.

4.2. Results and discussion

The proposed MK-Means method characteristics, such as accuracy, sensitivity, specificity, retrieval time, execution time, and CPU memory utilization during execution, were tabulated and compared to current approaches. Additionally, it must be validated using experimental data and considered for the conclusion discussion for visualization.

When it comes to accuracy, the suggested MK-Means methodology outperforms all other available methods. Even as data sizes grow, the accuracy of the outcomes stays effective. By reading input files ranging from 60MB to 100MB in size, about 92–97% of reliable output findings were published. Figure 4 depicts the accuracy details for all strategies for various file size inputs.

Figure 6 depicts the specificity information of all approaches for various file size inputs. When specificity is considered, the suggested MK-Means strategy outperforms all other known approaches. Even as data amount grows, the specificity of the results remains constant. Specificity values of 91–93% were published when reading input file sizes ranging from 60MB to 100MB. The most essential characteristic of this strategy is the retrieval time of the data processing on the repository that has been done in HDFS. When compared to all existing methodologies, the new MK-Means algorithm

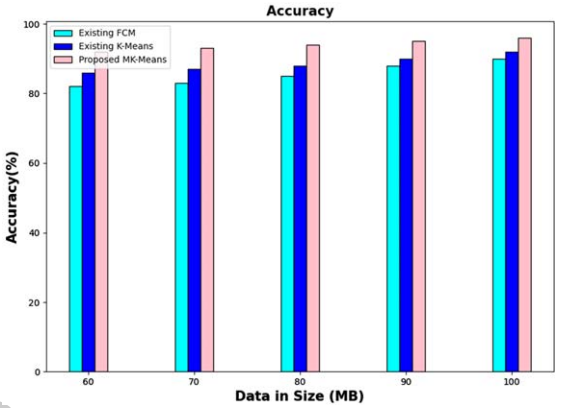

Fig. 4. Accuracy of proposed MK-Means with existing approaches.

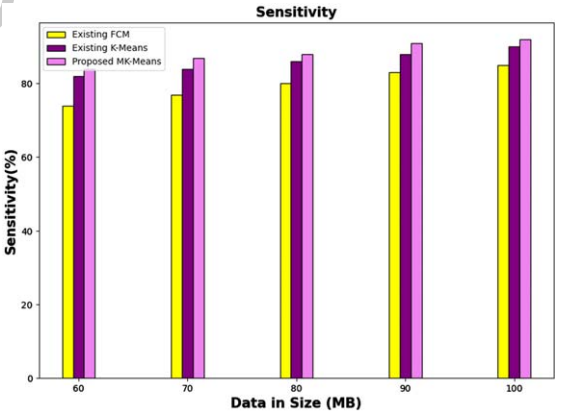

Fig. 5. Sensitivity of proposed MK-Means with existing approaches.

has a far shorter retrieval time. In other words, the ESSA query optimization methodology has boosted the pace of data processing from the HDFS system. Figure 7 represents the retrieval time of all strategies during execution.

At the same time, compared to other strategies, this one has a relatively short execution time. Even if the amount of data has increased, the sensitivity

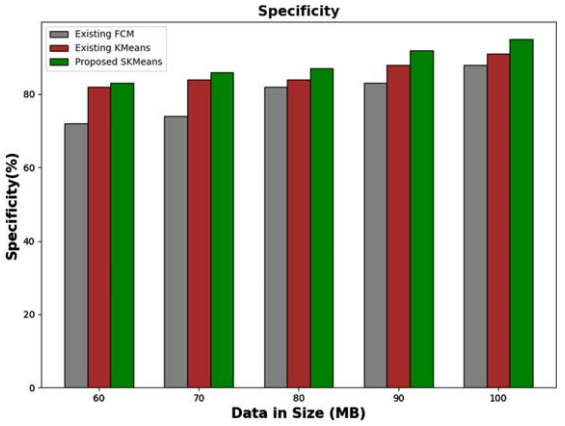

Fig. 6. Specificity of proposed MK-Means with existing approaches.

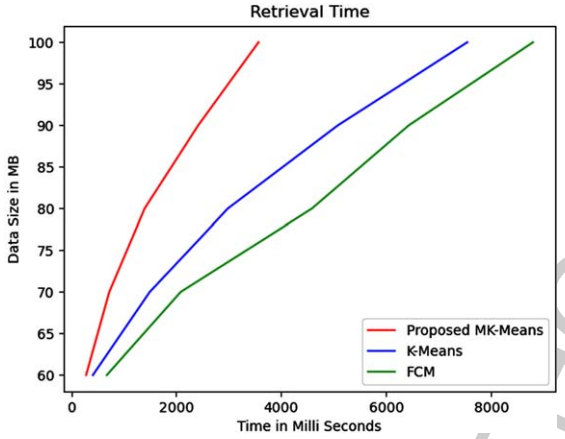

Fig. 7. Retrieval time of proposed MK-Means with existing approaches.

of the results has not. Due to the techniques utilised in the ESSA and MK-Means approaches, the overall completion time of the procedure is quite short when accessing 60MB to 100MB of input file size. The execution time of all techniques ranges between 100 ms and 300 ms. It took more time to execute at first, but then steadily maintained its range for process completion. In a big data setting, reading large files from the HDFS repository becomes more difficult when the speed of recovery is a problem on bigger network cluster nodes. Yet, utilizing this ESSA and MK-Means technique, it may be possible to handle the data reliably and efficiently. Figure 8 depicts the execution times of all strategies during execution.

Finally, the CPU use while doing data processing in HDFS must be calculated in milliseconds and compared to other current ways. Typically, CPU memory

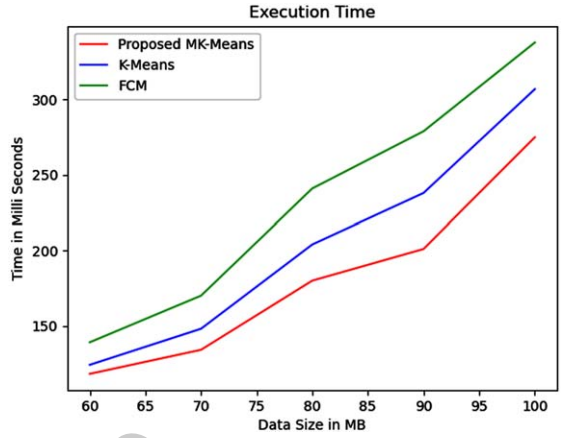

Fig. 8. Execution time of proposed MK-Means with existing approaches.

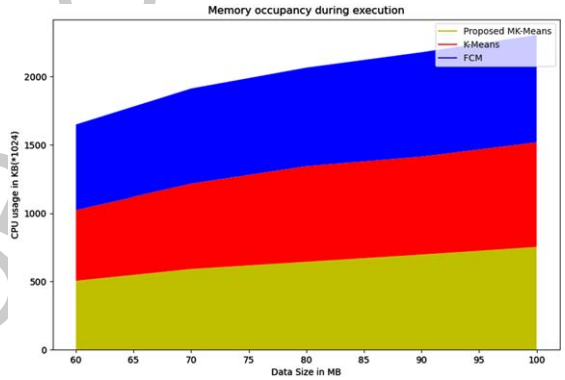

Fig. 9. Memory usage of proposed MK-Means with existing approaches.

utilisation is measured in kilobytes for each job. The count of map and reduce functions in the HDFS method determines the CPU's memory utilisation during execution. According to the results, the suggested MK-means used extremely little memory in KB throughout execution time. Figure 9 shows the memory consumption details.

5. Conclusion

Hadoop's analysis of enormous volumes of data is a research problem for any real-world application due to its speed and data recovery time. The proposed MK-Means and ESSA approaches can produce accurate results in a short period of time. The technique of innovative clustering algorithms, as well as query optimisation methodologies used to enhance

data processing time in a big data context, are comprehensively covered in this work. The preprocessed data is stored in the HDFS system using the Hadoop framework and map-reduce concepts. The proposed approach is divided into two stages: structuring massive amounts of data and optimising queries on bigger network clusters. The input file size range for the recommended technique began at 60 MB and expanded to 100 MB, and performance analysis reports were generated based on the main factors. After accessing 100MB of input files, it obtained 96% accuracy with sensitivity and specificity scores of 90% and 93%, respectively. All of the results were compared to well-known approaches such as K-means and fuzzy C-means. According to the results, the proposed system distributes data processing on cluster nodes with high accuracy and low latency. It also consumes very minimal memory space while executing on active CPUs. As a consequence, the proposed work outperforms current approaches. This approach will be expanded in the future with efficient algorithms based on Machine Learning, Artificial Intelligence and deep learning models to find out the best algorithm or model for reducing data processing time, query time on larger networks.

References

[1] Ajit Kumar Mahapatra, Nibedan Panda and Binod Kumar Pattanayak, Quantized Salp Swarm Algorithm (QSSA) for optimal feature selection, *International Journal of Information Technology* (2023), 1–10.

[2] Dinar Ajeng Kristiyanti, Imas Sukaesih Sitanggang and Sri Nurdiati, Feature Selection Using New Version of V-Shaped Transfer Function for Salp Swarm Algorithm in Sentiment Analysis, *Computation* **11**(3) (2023), 56.

[3] Issa Mohammed Saeed Ali and D. Hariprasad, Hyper-heuristic salp swarm optimization of multi-kernel support vector machines for big data classification, *International Journal of Information Technology* (2023), 1–13.

[4] Fouad H. Awad and Murtadha M. Hamad, Improved k-means clustering algorithm for big data based on distributed smartphoneneural engine processor, *Electronics* **11**(6) (2022), 883.

[5] Deepak Kumar and Vijay Kumar Jha, An improved query optimization process in big data using ACO-GA algorithm and HDFS map reduce technique, *Distributed and Parallel Databases* **39** (2021), 79–96.

[6] E. Manohar, E. Anandha Banuand D. Shalini Punithavathani, Composite analysis of web pages in adaptive environment through Modified Salp Swarm algorithm to rank the web pages, *Journal of Ambient Intelligence and Humanized Computing* (2022), 1–16.

[7] Divya Bairathi, and Dinesh Gopalani, An improved salp swarm algorithm for complex multi-modal problems, *Soft Computing* **25** (2021), 10441–10465.

[8] Sayar Singh Shekhawat, et al., bSSA: binary salp swarm algorithm with hybrid data transformation for feature selection, *IEEE Access* **9** (2021), 14867–14882.

[9] Kulanthaivel Balakrishnan, R. Dhanalakshmi and Utkarsh Mahadeo Khaire, Improved salp swarm algorithm based on the levy flight for feature selection, *The Journal of Supercomputing* **77**(11) (2021), 12399–12419.

[10] Regin Rajan, et al., Node replacement based energy optimization using enhanced salp swarm algorithm (Es2a) in wireless sensor networks, *Journal of Engineering Science and Technology* **16**(3) (2021), 2487–2501.

[11] Bhaskar Nautiyal, et al., Improved salp swarm algorithm with mutation schemes for solving global optimization and engineering problems, *Engineering with Computers* (2021), 1–23.

[12] Chaokun Yan, et al., A novel feature selection method based on salp swarm algorithm, *2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE)*, IEEE, 2021.

[13] Laith Abualigah, et al., Advances in meta-heuristic optimization algorithms in big data text clustering, *Electronics* **10**(2) (2021), 101.

[14] Mahmud Salem Alkoffash, et al., A non-convex economic load dispatch using hybrid salp swarm algorithm, *Arabian Journal for Science and Engineering* **46**(9) (2021), 8721–8740.

[15] Long Qi and Hui Liu, Feature selection of BOF steelmaking process data based on denary salp swarm algorithm, *Arabian Journal for Science and Engineering* **45** (2020), 10401–10416.

[16] Laith Abualigah, et al., Selection scheme sensitivity for a hybrid Salp Swarm Algorithm: analysis and applications, *Engineering with Computers* **38**(2) (2022), 1149–1175.

[17] Abualigah Laith, et al., Salp swarm algorithm: a comprehensive survey, *Neural Computing & Applications* **32**(15) (2020), 11195–11215.

[18] Narinder Singh, S.B. Singh and Essam H. Houssein, Hybridizing salp swarm algorithm with particle swarm optimization algorithm for recent optimization functions, *Evolutionary Intelligence* (2022), 1–34.

[19] Hongliang Zhang, et al., A multi-strategy enhanced salp swarm algorithm for global optimization, *Engineering with Computers* (2022), 1–27.

[20] Ibrahim M. El-Hasnony, et al., Improved feature selection model for big data analytics, *IEEE Access* **8** (2020), 66989–67004.

[21] Sandeep Kumar, Rajani Kumari and Anand Nayyar, Sigmoidal salp swarm algorithm, *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*. IEEE, 2020.

[22] Sasan Gharehpasha, Mohammad Masdari and Ahmad Jafarian, Power efficient virtual machine placement in cloud data centers with a discrete and chaotic hybrid optimization algorithm, *Cluster Computing* **24** (2021), 1293–1315.

[23] Kaushik Mishra, Rosy Pradhan and Santosh Kumar Majhi, Quantum-inspired binary chaotic salp swarm algorithm (QBCSSA)-based dynamic task scheduling for multiprocessor cloud computing systems, *The Journal of Supercomputing* **77** (2021), 10377–10423.

[24] Fanghao Tian, et al., An improved salp optimization algorithm inspired by quantum computing, *Journal of Physics: Conference Series* **1570**(1), IOP Publishing, 2020.

[25] R. Krithiga and E. Ilavarasan, WITHDRAWN: A Reliable Modified Whale Optimization Algorithm based Approach

for Feature Selection to Classify Twitter Spam Profiles, (2020), 103451.

[26] Sonal Jain and Ramesh Dharavath, Memetic salp swarm optimization algorithm based feature selection approach for crop disease detection system, *Journal of Ambient Intelligence and Humanized Computing* (2021), 1–19.

[27] Vasavi Ravuri and S. Vasundra, Moth-flame optimization-bat optimization: Map-reduce framework for big data clustering using the Moth-flame bat optimization and sparse Fuzzy C-means, *Big Data* **8**(3) (2020), 203–217.

[28] Huthaifa M. Kanoosh, Essam Halim Houssein and Mazen M. Selim, Salp swarm algorithm for node localization in wireless sensor networks, *Journal of Computer Networks and Communications* **2019** (2019).

[29] M.R. Sundara Kumar, et al., Innovation and creativity for data mining using computational statistics, *Methodologies and Applications of Computational Statistics for Machine Intelligence*, IGI Global, (2021), 223–240.

[30] M.R. Sundarakumar, et al., A comprehensive study and review of tuning the performance on database scalability in big data analytics, *Journal of Intelligent & Fuzzy Systems*, Preprint: 1–25.

[31] M.R. Sundarakumar, et al., An Approach in Big Data Analytics to Improve the Velocity of Unstructured Data Using MapReduce, *International Journal of System Dynamics Applications (IJSDA)* **10**(4) (2021), 1–25.

[32] J.R. Albert and A.A. Stonier, Design and development of symmetrical super-lift DC–AC converter using firefly algorithm for solar-photovoltaic applications, *IET Circuits Devices Syst* **14**(3) (2020), 261–269. https://doi.org/10.1049/iet-cds.2018.5292

[33] D. Shunmugham Vanaja, J.R. Albert and A.A. Stonier, An Experimental Investigation on solar PV fed modular STATCOM in WECS using Intelligent controller, *Int Trans Electr Energ Syst* **31**(5) (2021), e12845. https://doi.org/10.1002/2050-7038.12845

[34] Malathi Murugesan, Kalaiselvi Kaliannan, Shankarlal Balraj, Kokila Singaram, Thenmalar Kaliannan and J.R. Albert, A Hybrid Deep Learning Model for Effective Segmentation and Classification of Lung Nodules from CT Images, *Journal of Intelligent and Fuzzy System* **42**(3) (2021), 2667–26791. DOI: 10.3233/JIFS-212189

[35] J.R. Albert, et.al, Investigation on load harmonic reduction through solar-power utilization in intermittent SSFI using particle swarm, genetic, and modified firefly optimization algorithms, *Journal of Intelligent and Fuzzy System* **42**(4) (2022), 4117–4133. DOI:10.3233/JIFS-212559

[36] K. Vanchinathan, K.R. Valluvan, C. Gnanavel, C. Gokul and J.R. Albert, An improved incipient whale optimization algorithm based robust fault detection and diagnosis for sensorless brushless DC motor drive under external disturbances, *Int Trans Electr Energ Syst* **31**(12) (2021), e13251. DOI: 10.1002/2050-7038.13251

[37] Satish Kumar Ramaraju, et al. Design and Experimental Investigation on VL-MLI Intended for Half Height (H-H) Method to Improve Power Quality Using Modified Particle Swarm Optimization (MPSO) Algorithm, 1 Jan. 2022, Vol. **42**(6), pp. 5939–5956. DOI: 10.3233/JIFS-212583

[38] Logeswaran Thangamuthu, J.R. Albert, Kalaivanan Chinnanan and Banu Gnanavel, Design and development of extract maximum power from single-double diode PV model for different environmental condition using BAT optimization algorithm, *J Intell Fuzzy Syst* **43**(1) (2022), 1091–1102. https://doi.org/10.3233/JIFS-213241

[39] Rajarathinam Palanisamy, Vijayakumar Govindaraj, Saravanan Siddhan and J.R. Albert, Experimental Investigation and Comparative Harmonic Optimization of AMLI Incorporate Modified Genetic Algorithm Using for Power Quality Improvement. *Journal of Intelligent and Fuzzy System* **43**(1) (2022), 1163–1176. DOI: 10.3233/JIFS-212668

[40] J.R. Albert, Design and Investigation of Solar PV Fed Single-Source Voltage-Lift Multilevel Inverter Using Intelligent Controllers. *J Control Autom Electr Syst* **33** (2022), 1537–1562. https://doi.org/10.1007/s40313-021-00892-w

[41] C. Gnanavel, P. Muruganatham, K. Vanchinathan. and J.R. Albert, Experimental Validation and Integration of Solar PV Fed Modular Multilevel Inverter (MMI) and Flywheel Storage System, *2021 IEEE Mysore Sub Section International Conference*, 2021, pp. 147–153, DOI: 10.1109/MysuruCon52639.2021.9641650

[42] J.R. Albert, Stonier Albert Alexander, Vanchinathan Kumarasamy, Testing and Performance Evaluation of Water Pump Irrigation System using Voltage-Lift Multilevel Inverter, *International Journal of Ambient Energy* (2022), 1–14. DOI: 10.1080/01430750.2022.2092773

[43] J.R. Albert, et al. An Advanced Electrical Vehicle Charging Station Using Adaptive Hybrid Particle Swarm Optimization Intended for Renewable Energy System for Simultaneous Distributions, **43**(4) (2022), 4395–4407. DOI: 10.3233/JIFS-220089

[44] J.R. Albert, R. Kannan, S. Karthick, P. Selvan, A. Sivakumar and C. Gnanavel, An Experimental and Investigation on Asymmetric Modular Multilevel Inverter an Approach with Reduced Number of Semiconductor Devices, *J Electrical Systems* **18**(3) (2022), 318–330.

[45] B. Babypriya, J.R. Albert, M. Shyamalagowri and R. Kannan, An Experimental Simulation Testing of Single-diode PV Integrated MPPT Grid-tied Optimized Control Using Grey Wolf Algorithm, 1 Jan. 2022:1–20. DOI: 10.3233/JIFS-213259

[46] Madhumathi Periasamy, Thenmalar Kaliannan, Shobana Selvaraj, Veerasundaram Manickam, Sheela Androse Joseph and J.R. Albert, Various PSO methods investigation in renewable and nonrenewable sources, *International Journal of Power Electronics and Drive Systems* **13**(4) (2022), 2498–2505, DOI: 10.11591/ijpeds.v13.i4.pp2498-2505

[47] J.R. Albert and Dishore Shunmugham Vanaja, Solar Energy Assessment in Various Regions of Indian Sub-continent, Solar Cells – Theory, Materials and Recent Advances, *IntechOpen*, (December 10th 2020). DOI: 10.5772/intechopen.95118

[48] J.R. Albert, Thenmalar Kaliannan, Gopinath Singaram, Fantin Irudaya Raj Edward Sehar, Madhumathi Periasamy and Selvakumar Kuppusamy, A Remote Diagnosis Using Variable Fractional Order with Reinforcement Controller for Solar-MPPT Intelligent System, *Photovoltaic Systems*, pp. 45–64, Publisher: CRC press. https://doi.org/10.1201/9781003202288

[49] J.R. Albert, K. Ramasamy, V. Joseph Michael Jerard, et al. A Symmetric Solar Photovoltaic Inverter to Improve Power Quality Using Digital Pulse-width Modulation Approach, *Wireless Pers Commun* (2023). https://doi.org/10.1007/s11277-023-10372-w

[50] C. Gnanavel, J.R. Albert, S. Saravanan and K. Vanchinathan, An Experimental Investigation of Fuzzy-Based Voltage-Lift Multilevel Inverter Using Solar Photovoltaic Application, *Smart Grids and Green Energy Systems*, pp. 59–74, Wiley publication. https://doi.org/10.1002/9781119872061.ch5

[51] J.R. Albert, K. Premkumar, K. Vanchinathan, A. Nazar Ali, R. Sagayaraj and T.S. Saravanan, Investigation of Super-Lift Multilevel Inverter Using Water Pump Irrigation System, *Smart Grids and Green Energy Systems*, 247, Wiley publication, pp. 247–262. https://doi.org/10.1002/9781119872061.ch16

[52] T. Kaliannan, J.R. Albert, D.M. Begam and P. Madhumathi, Power Quality Improvement in Modular Multilevel Inverter Using for Different Multicarrier PWM, *European Journal of Electrical Engineering and Computer Science* **5**(2) (2021), 19–27. DOI: https://doi.org/10.24018/ejece.2021.5.2.315

[53] J.R. Albert, D. Muhamadha Begam, B. Nishapriya, Micro grid connected solar PV employment using for battery energy storage system, *Journal of Xidian University* **15**(3) (2021), 85–97. https://doi.org/10.37896/jxu15.3/010

[54] M. Dhivya and J.R. Albert, Investigation on Super Lift DC/AC Inverters Using Photovoltaic Energy for AC Component Application, *International Journal of Engineering and Computer Science* **5**(11) (2016), 18832–18837. https://ijecs.in/index.php/ijecs/article/view/2872

[55] M. Dhivya, J.R. Albert, Fuzzy Grammar Based Hybrid Split-Capacitors and Split Inductors Applied In Positive Output Luo-Converters, *International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET)* **3**(1) (2017), 327–332. DOI: 10.32628/IJSRSET173174

[56] J.R. Albert, V. Hemalatha, R. Punitha, M. Sasikala and M. Sasikala, Solar Roadways-The Future Rebuilding Infrastructure and Economy, *International Journal of Electrical and Electronics Research* **4**(2) (2016), 14–19. https://researchpublish.com/journal-details/IJEER

[57] A. Johny Renoald, M.S. Keerthana, Design and Implementation of Super-Lift Multilevel Inverter using Renewable Photovoltaic Energy for AC Module Application, *International Journal of Science Technology & Engineering* **2**(11) (2016), 617–624.

[58] A. Johny Renoald, M. Dhivya, Analysis on Super Lift Multilevel DC/AC Inverters using Photovoltaic Energy with AC Module Application, *International Journal for Scientific Research & Development* **5**(2) (2017), 479–481. http://ijsrd.com/Article.php?manuscript=IJSRDV5I20496

[59] Johny Renoald Albert, M. Saranya, S. Shobana, Drone based system for cleaning the environment, *International Journal of Innovative Research in Science, Engineering and Technology* **9**(3) (2020), 1141–1145. DOI: 10.15680/IJIRSET.2019.0903138

[60] A. Johny Renoald, M. Saranya, S. Shobana, R. Nivethitha, Patients Physical Condition Care Framework by Utilizing Web of Things (IoT), *Journal of Control and Instrumentation Engineering* **6**(1) (2020), 23–29. http://doi.org/10.5281/zenodo.3773021

[61] A. Johny Renoald, S. JayaPradha, P. Kanimozhi, V. Monisha, D. Pavithra, Design and Development of Hand Gesture Voice Conversion System Using for Dump and Deaf People, *Journal of Controller and Converters* **5**(1) (2020), 28–31. https:// doi.org0.46610/JOCC.2020.v05i01.0040.46610/JOCC.2020.v05i01.004 /1

[62] K. Santhiya, M. Devimuppudathi, D. Santhosh Kumar and A. Johny Renold, Real Time Speed Control of Three Phase Induction Motor by Using Lab View with Fuzzy Logic, *Journal on Science Engineering and Technology* **5**(2) (2018), 21–27.