Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.029533
Article

Tech Science Press

A Novel Approach to Design Distribution Preserving Framework for Big Data

Mini Prince[1,*] and P. M. Joe Prathap[2]

[1]Department of Information Technology, St. Peter's College of Engineering and Technology, Chennai, 600054, Tamilnadu, India
[2]Department of Information Technology, R.M.D Engineering College, Chennai, 601206, Tamilnadu, India
*Corresponding Author: Mini Prince. Email: miniprince171@gmail.com
Received: 05 March 2022; Accepted: 13 April 2022

Abstract: In several fields like financial dealing, industry, business, medicine, *et cetera*, Big Data (BD) has been utilized extensively, which is nothing but a collection of a huge amount of data. However, it is highly complicated along with time-consuming to process a massive amount of data. Thus, to design the Distribution Preserving Framework for BD, a novel methodology has been proposed utilizing Manhattan Distance (MD)-centered Partition Around Medoid (MD–PAM) along with Conjugate Gradient Artificial Neural Network (CG-ANN), which undergoes various steps to reduce the complications of BD. Firstly, the data are processed in the pre-processing phase by mitigating the data repetition utilizing the map-reduce function; subsequently, the missing data are handled by substituting or by ignoring the missed values. After that, the data are transmuted into a normalized form. Next, to enhance the classification performance, the data's dimensionalities are minimized by employing Gaussian Kernel (GK)-Fisher Discriminant Analysis (GK-FDA). Afterwards, the processed data is submitted to the partitioning phase after transmuting it into a structured format. In the partition phase, by utilizing the MD-PAM, the data are partitioned along with grouped into a cluster. Lastly, by employing CG-ANN, the data are classified in the classification phase so that the needed data can be effortlessly retrieved by the user. To analogize the outcomes of the CG-ANN with the prevailing methodologies, the NSL-KDD openly accessible datasets are utilized. The experiential outcomes displayed that an efficient result along with a reduced computation cost was shown by the proposed CG-ANN. The proposed work outperforms well in terms of accuracy, sensitivity and specificity than the existing systems.

Keywords: Big data; artificial neural network; fisher discriminant analysis; distribution preserving framework; manhattan distance

1 Introduction

Big Data (BD) is a massive collection of data [1], which is being used in a variety of application domains such as financial trading, business, education, medical, and so on [2]. Big Data is also referred to as "big data analytics." This BD is extremely beneficial to both individuals and businesses; nevertheless, the exploitation of this BD is a difficult task since the investigation, evaluation, and data retrieval are quite complex, require a large amount of memory, and are time-consuming [3,4]. Furthermore, the data size is large, and the data structure is intricate; as a result, a larger calculation cost is required for the implementation of BD [5]. For an effective BD Analysis (BDA), the following steps must be completed: pre-processing, data structuring, partitioning phase, and data preservation phase [6–8]. In the first place, by applying the map-reduce function, the data are processed in the pre-processing phase, which reduces repetition by a factor of seven [7]. Following that, the data that was missing from the dataset is substituted or ignored [8] depending on the circumstances. Following that, the data is transformed into a normalised format. Following that, in order to improve the performance, the dimensionalities of the data are reduced [9]. Reference [10] After then, the pre-processed data is transformed into a structured format. Then, during the partitioning step, the data that is being structured is partitioned and grouped into clusters in addition to being categorised. Once this phase is completed, the clusters are sent into the classification phase, where they are processed by the classifier, allowing the user to have immediate access to the data [11,12]. Despite this, a greater correlation error is observed in BD [13], which can be attributed to the unstructured data supplied. Many techniques, such Naive Bayes [14], Gaussian Discriminant Analysis [15], and others, are proposed to address the aforementioned concerns. Other neural networks are also proposed to overcome the aforementioned issues. Despite the fact that these approaches lessen the likelihood of such issues occurring, they have some disadvantages [16]. And the drawbacks are that it is not capable of resolving aberrant situations in BD Exploration (BDE) [17], and that in order to acquire the best classification outcomes, a large number of essential criteria must be considered [18]. Certain commonly used classification approaches are not appropriate for huge datasets because, when the target classes overlap, the performance of the technique is poor [19]. The learning approaches do not address the BD issues, particularly when the material is presented in a sequential manner [20,21].

1.1 Contribution of the Research Work

A distribution preserving framework for BD is proposed utilizing the MD-PAM along with CG–ANN to conquer such difficulties. The paper's remaining part is structure as: the related works regarding the proposed methodology are surveyed in Section 2; the proposed model of distribution Preserving Framework for BD using MD-PAM and CG–ANN is explicated in Section 3; the results together with discussion is demonstrated in Section 4. Lastly, the paper is winded up with future scope in Section 5.

2 Literature Survey

Michele Ianni and colleagues [21] pioneered the application of BDE approach. It was through this system, which operated in an unsupervised manner, that higher-quality data clusters, which were clustered around their centroids, were made available to users. The results revealed that the performance of the proposed methodology outperformed the other techniques in terms of increased accuracy and scalability, respectively. However, when the dataset had a significant amount of noise, the performance would suffer. Hernandeza et al. [22] introduced a hybrid neural architecture that combined morphological neurons with perceptrons in an integrated fashion. There were several other types of neural networks before the Morphological-Linear Neural Network (MLNN), which was the first architecture to integrate an output layer of classical perceptrons coupled with a hidden layer of morphological neurons.

Banchhor et al. [23] developed an approach for BD classification that is based on the Correlative Naive Bayes classifier and the Map Reduce Model, with the Cuckoo–Grey wolf classifier serving as the basis for the methodology (CGCNBMRM). The Cuckoo–Grey Wolf-based Optimization (CGWO) was designed through the effective incorporation of the Cuckoo Search (CS) Algorithm into the Grey Wolf Optimizer (GWO); in addition, the optimal selection of model parameters optimised the CNB model. The experience outcome shown that the approach attained accuracy, sensitivity, and specificity of 80.7 percent, 84.5 percent, and 76.9 percent, respectively, using this approach. BDE regarding data relevance was made easier by Ada

Bagozi and colleagues [24], who built an IDEA as a Service (Interactive Data Exploration As-a-Service) to aid in the BDE process. In this scheme, various methodologies had been presented, including: (a) an incremental clustering approach, which provided a summarised representation of the amassed data streams; (b) for BDE, a multi-dimensional organisation of summarised data regarding various analysis dimensions; and (c) data relevance analysis methodologies, which concentrated on relevant data at the time of BDE. The MOANOFS (Multi-Objective Automated Negotiation centred Online Feature Selection) system, developed by Fatma BenSaid and colleagues [25], investigated the current enhancements of online machine learning methodologies in conjunction with a conflict resolution scheme for the improvement of classification performance in ultra-higher dimensional databases. Selcuk Aslan and colleagues [26] developed an improved Artificial Bee Colony (ABC) method that took into account the characteristics of BD optimization issues, as well as a variant known as genetic BD ABC.

2.1 Identified Drawbacks from Existing Systems

The outcomes displayed that a better outcome was attained by the scheme than the other BD optimization methodologies. Nevertheless, a poor accuracy rate was achieved by the system over the unstructured data. The existing systems consumes more time was consumed to process data; in addition, it wasn't sensitive towards attacked data. The system was not efficient towards the correlation errors.

3 Proposed Methodology

The structured and unstructured forms data are included in the BD, which is a collection of a massive amount of data [27]. Consequently, it is a complicated task for the user to retrieve along with to recognize the appropriate data as of the larger amount of data [28]. Thus, to design the distribution preserving framework for BD, a methodology has been proposed utilizing MD-PAM and CG–ANN. Firstly, from the NSL-KDD dataset, the data are amassed and are given to the pre-processing step. In the pre-processing step, to enhance the clustering performance, the data is processed by mitigating data redundancy, dealing with the missing data, converting the data into a machine-understandable format, along with minimizing the data's dimensionalities by employing GK–FDA. Next, the data being processed is transmuted into a structured format, which is then inputted to the partitioning phase. In the partitioning phase, by utilizing the MD-PAM, the cluster formation is performed; thus, the data with minimal distance along with minimum dissimilarities is placed in a single cluster. After that, the preserving phase is executed following the completion of data clustering. In this, the clustered data is classified by giving to the CG–ANN classifier. Thus, the data desired by the users is offered by the classification process over a big transactional database in an effortless way. Fig. 1 exhibits the proposed methodology's architecture.

3.1 Input Source

In the proposed system, the openly accessible NSL-KDD dataset is utilized as the input source. A massive amount of data is included in the NSL-KDD dataset, which is a classified dataset. The data values of the attacked along with non-attacked data are contained in it. Here, 80% and 20% of data are provided for the training and testing phase, respectively.

3.2 Preprocessing

In this phase, the original input data is transmuted into a supportive along with effectual format. (i) Redundant Data Removal utilizing Hadoop Distributed File System (HDFS), (ii) Missing data (iii) Data Transformation, and (iv) Dimensionality Reduction are the '4' steps included in this phase. The pre-processing function is given as,

$$\Im = \Im_p\{\mathrm{x}_\mathrm{d}^\mathrm{t}\} \tag{1}$$

where, the pre-processing function's output is specified as $\Im$, the input data is signified as χ_d^t and the pre-processing function is denoted as $\Im_p$, which is formulated as,

$$\Im_p = \{\Im_{rd}, \Im_{md}, \Im_{dt}, \Im_{dr}\} \tag{2}$$

where, the redundant data removal function is indicated as $\Im_{rd}$, the missing data function is specified as $\Im_{md}$, the data transformation function is proffered as $\Im_{dt}$, along with the dimensionality reduction function is denoted as $\Im_{dr}$.

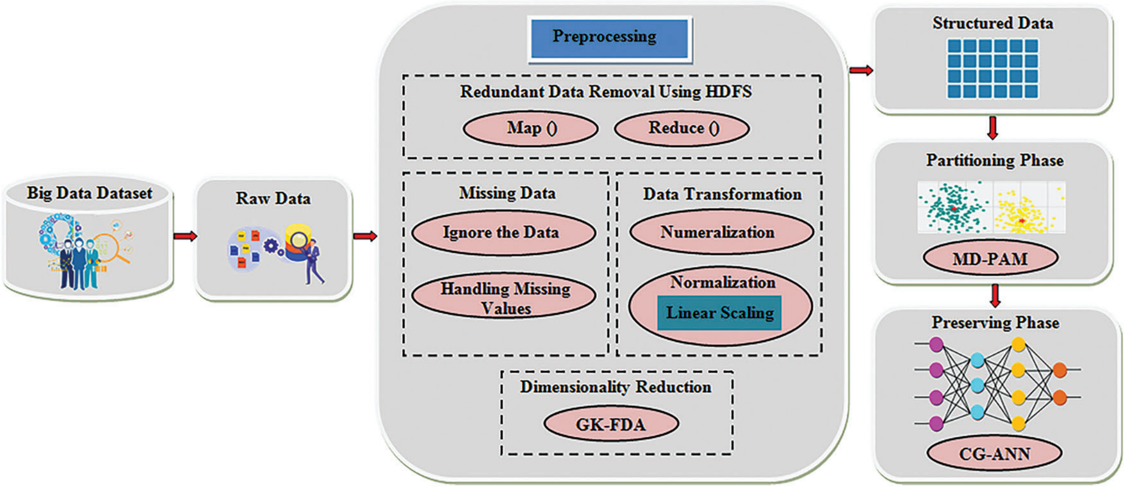


Figure 1: The architecture of the proposed work

3.2.1 Redundant Data Removal Using HDFS

Eliminating the duplicate data together with avoiding the repetition of identical data amassed in various locations are the procedures done here. Thus, it significantly mitigated the training along with execution time. The HDFS is utilized to remove the redundant data efficiently. Map as well reduce are the '2' major tasks included in the HDFS. Fig. 2 displays the map-reduce function's general structure.

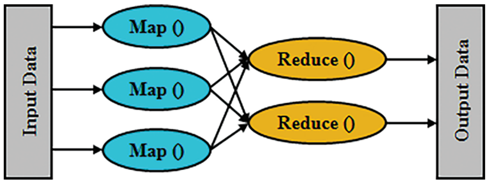


Figure 2: The general structure of the map-reduce function

(a) Map Stage

Firstly, the input data in the form of a file or directory is amassed in the HDFS. The input files are transferred to the mapper line by line. In the mapper, the input data is broken into smaller sets of data after being processed. The mapping function is given as,

$$\rho_{map}(g_1, \delta_1) = \ list(g_2, \delta_2) \tag{3}$$

where, ρ_{map} specifies the mapping function, which is implemented in parallel to every pair (g_1, δ_1) in the dataset; in addition, creates the list of pairs (g_2, δ_2).

(b) Reduce Stage

The data are moved to the Reduce stage after breaking it into smaller data. The data obtained as of the mapper is processed by the reducer. A new set of outputs is produced after processing the data. These data are amassed in the HDFS. The mathematical formulation for the reduce stage is,

$$R_{reduce}(g_1, list\ (\delta_2)) \rightarrow list\ ((g_3, \delta_3)) \tag{4}$$

Here, the reducing function for the pair $(g_1, list\ (\delta_2)$ is signified as $\mathrm{R}_{\mathrm{reduce}}$ and transmutes the pair into another list of (g_3, δ_3) airs.

3.2.2 Missing Data

In this phase, the data is checked whether missed or not. The data being missed is substituted or rearranged. (i) Ignore the data, and (ii) Handling of missing values are the '2' steps included in this phase. The missing data phase is mathematically given as,

$$\Im_{md} = \left\{h_1^{id}, h_2^{hm}\right\} \tag{5}$$

where, the missing data function is denoted as $\Im_{md}$, the ignore data function and handling missing value function are represented as $\mathrm{h}_1^{\mathrm{id}}$ and $\mathrm{h}_2^{\mathrm{hm}}$, respectively.

(a) Ignore the Data

When larger numbers of data are missed or numerous values are missed then, the whole data or row is ignored. It is expressed as,

$$\lambdabar_1^{op} = \ \mathrm{h}_1^{\mathrm{id}}\left[\mathrm{x}_\mathrm{d}^\mathrm{t}\right] \tag{6}$$

where, the ignoring data function's outcome is specified as $\lambdabar_1^{op}$.

(b) Handling Missing Values

Handling of missing values is opposite to that of the data ignoring step. Here, the missing values are substituted by random variables or average or mean regarding the missing attribute. It expressed as,

$$\lambdabar_2^{op} = \ \mathrm{h}_2^{\mathrm{hm}}\left[\mathrm{x}_\mathrm{d}^\mathrm{t}\right] \tag{7}$$

where, the outcome of handling of the missing value function is exhibited as $\lambdabar_2^{op}$.

3.2.3 Data Transformation

The procedure of transforming the data format into another needed format is termed data transformation. In this, the transformed data format is highly suitable along with machine-understandable. The '2' steps included in this phase are numeralization together with normalization, which are expressed mathematically as,

$$\Im_{dt} = \left\{\lambda^{\mathrm{n}}, \lambda^{\mathrm{t}}\right\} \tag{8}$$

where, the data transformation function is signified as $\Im_{dt}$, the numeralization function is denoted as λ^{n} and the normalization function is indicated as λ^{t}.

(a) Numeralization

The string values or characters that are existed in the dataset are transmuted into a numerical format in this process. It is given as,

$$\varnothing_n = \lambda^n \left[x_d^t\right] \tag{9}$$

where, the numeralization function's outcome is specified as $Ø_n$.

(b) Normalization

In this, the data values are adjusted into a particular range betwixt 0 to 1 or −1 to 1utilizing the minimum along with a maximum of feature values to scale the data. Highly effectual access to data is provided by this approach. The data are normalized by employing Linear Scaling methodology in the proposed work. The linear scale normalization is expressed as,

$$\lambda = \frac{x_t^d - \min\left(x_t^d\right)}{\max\left(x_t^d\right) - \min\left(x_t^d\right)} \tag{10}$$

where, the original value is indicated as χ_d^t and the normalized value is represented as λ.To rescale the ranges betwixt the arbitrary set of values α and β, the rescaling function is formulated as,

$$\lambda = \alpha + \frac{\left[x_t^d - \min\left(x_t^d\right)\right](\beta - \alpha)}{\max\left(x_t^d\right) - \min\left(x_t^d\right)} \tag{11}$$

where, the min-max values are signified as α and β.

3.2.4 Dimensionality Reduction

The data is converted as of a higher-dimensional space into a lower-dimensional space in the dimensionality reduction; thus, the lower-dimensional data is calculated effortlessly [29]. The GK-FDA is utilized to mitigate the data's dimensionalities. The GK approach is hybrid to the FDA algorithm with an intention to alleviate the complications in the FDA since it is unstable under several conditions. Therefore, the reduction accuracy is enhanced considerably by the GK-FDA. Following the algorithmic steps included in GK-FDA.

Step 1: FDA is an efficient dimensionality reduction methodology that performs as maximization of various classes. In this, the data are constructed into a $m \times n$ matrix format. It is expressed as,

$$\lambda = \begin{bmatrix} \lambda_{1x1} & \lambda_{1x2} & \lambda_{1xn} \\ \lambda_{2x1} & \lambda_{2x2} & \lambda_{2xn} \\ \lambda_{mx1} & \lambda_{mx2} & \lambda_{mxn} \end{bmatrix} \tag{12}$$

where, m^{th} rows specify the data, and the n^{th} columns indicate the specific features.

Step 2: To enhance the dimensionality reduction accuracies, the input values are altered by employing GK. The mathematical formulation for GK is provided as,

$$f(\lambda) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\lambda - \mu)^2}{2\sigma^2}} \tag{13}$$

where, the input values are represented as λ, the standard deviation is specified as σ, and the mean value is signified as μ.

Step 3: The total scatter matrix δ_t for the m^{th} row of the matrix with the row vector λ_m, is expressed as,

$$\delta_t = \sum_{n=1}^{N} (\lambda_{\mathrm{m}} - \lambda_{\mathrm{mean}})(\lambda_{\mathrm{m}} - \lambda_{\mathrm{mean}})^{\mathrm{T}} \tag{14}$$

where, the total mean vector is proffered as λ_{mean}.

Step 4: Λ_j is regarded as the set of class outcomes from the row vectors λ_m. It specifies the scatter δ_m within the matrix Λ for the class j and it is mathematically formulated as,

$$\delta_m = \sum_{\lambda_m=1}^{N} (\lambda_{\mathrm{m}} - \lambda_{\mathrm{l.mean}})(\lambda_{\mathrm{m}} - \lambda_{\mathrm{l.mean}})^{\mathrm{T}} \tag{15}$$

where, the mean vector for the event outcome of the class l is illustrated as $\lambda_{l.mean}$, and the transpose of matrix is illustrated as T.

Step 5: If the total number of classes is L and that are differentiated for the complete data set, the within-class scatter δ_w for the matrix is expressed as,

$$\delta_w = \sum_{l=1}^{L} \delta_l \tag{16}$$

Step 6: The between-class scatter δ_b for the matrix Λ is gauged as,

$$\delta_b = \sum_{l=1}^{L} n_l (\lambda_{\mathrm{l.mean}} - \lambda_{\mathrm{mean}})(\lambda_{\mathrm{l.mean}} - \lambda_{\mathrm{mean}})^{\mathrm{T}} \tag{17}$$

where, the between-class scatter matrix is specified as δ_b, the total observations within the class l is denoted as η_l.

Step 7: By considering the summation of the between-class scatter δ_b and within-class scatter δ_w for the matrix, the total scatter matrix δ_m is formulated as,

$$\delta_m = \delta_b + \delta_w \tag{18}$$

Step 8: The first FDA vector ϖ_1 is signified as,

$$\omega_1 = max \frac{\omega_1^T \delta_b \omega_1}{\omega_1^T \delta_w \omega_1} \tag{19}$$

Step 9: The second FDA vector ϖ_y maximizes the scatter betwixt the L classes; however, minimizes the scatter within each class. The generalized eigen value expression's eigenvectors ϖ_y is elucidated by the FDA vectors.

$$\delta_b \omega_y = e_l \delta_w \omega_y \tag{20}$$

where, the extent of overall separation among the L classes is signified as e_l.

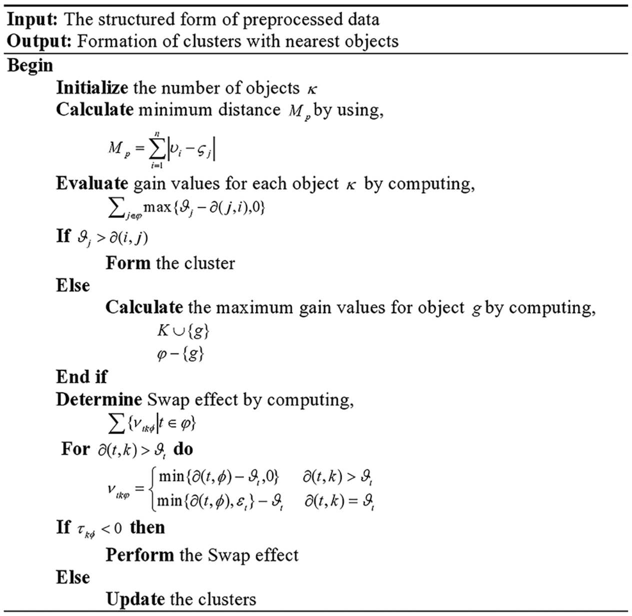
3.3 Structured Data

Lastly, the pre-processed data is transmuted into a structured format. The data's reduced normalized form is mentioned as the data's structured format. Consequently, the classification accuracy is enhanced by this structured data.

3.4 Partitioning Phase

The structured data is submitted to the partitioning phase. In this phase, regarding the data's characteristics along with similarities, the data are partitioned and arranged into different groups or clusters [30].

Therefore, a significant impact is possessed by the classification function on the fast processing of larger data [31]. The PAM is utilized to attain the formation of the finest cluster. The PAM is nothing but a kind of K-Medoid Algorithm, which is a clustering technique pertinent to the k-means clustering algorithm. In k-medoids clustering, every single cluster is signified by one of the data points in the cluster. These points are termed cluster medoids. K-medoid is a strong alternative to k-means clustering. The K-medoid-based PAM algorithm utilized medoids as cluster centers rather than means that are used in k-means and so they are lesser sensitive to noise along with outliers than the k-means. By employing MD, the minimum distance is gauged in the PAM. Consequently, the clustering accuracy is enhanced by the MD-PAM. The Fig. 3 illustrates the pseudo-code for the MD-PAM. The following steps explicate the procedures in the cluster formation.



```
Input: The structured form of preprocessed data
Output: Formation of clusters with nearest objects
Begin
    Initialize the number of objects $\kappa$
    Calculate minimum distance $M_p$ by using,
        $M_p = \sum_{i=1}^{n} |\upsilon_i - \varsigma_j|$
    Evaluate gain values for each object $\kappa$ by computing,
        $\sum_{j\in\varphi} \max\{\vartheta_j - \partial(j,i),0\}$
    If $\vartheta_j > \partial(i,j)$
        Form the cluster
    Else
        Calculate the maximum gain values for object $g$ by computing,
            $K \cup \{g\}$
            $\varphi - \{g\}$
    End if
    Determine Swap effect by computing,
        $\sum \{v_{tk\phi} | t \in \varphi\}$
    For $\partial(t,k) > \vartheta_t$ do
        $v_{tk\varphi} = \begin{cases} \min\{\partial(t,\phi) - \vartheta_t,0\} & \partial(t,k) > \vartheta_t \\ \min\{\partial(t,\phi),\varepsilon_t\} - \vartheta_t & \partial(t,k) = \vartheta_t \end{cases}$
    If $\tau_{k\phi} < 0$ then
        Perform the Swap effect
    Else
        Update the clusters
```

Figure 3: Pseudo-code for the MD–PAM algorithm

Step 1: Initially, by adding the number of objects with minimum distance to all other objects, the optimal medoid K is initialized. The distances betwixt the objects i and j at v_i and ζ_j respectively are measured by the MD formula as,

$$M_p = \sum_{i=1}^{N} |v_i - \delta_i| \tag{21}$$

Step 2: When an object i(iϵA) is regarded as a candidate, then that is appended to K. After that, a total gain v_i is calculated for every single object, which is given as,

$$\gamma_i = \sum_{j\epsilon\varphi} \max\{\rho_j - \partial(i,j), 0\} \tag{22}$$

where, the object of Ø except I is signified as j. In case $\rho_j > \sigma(I,j)$ then the quality of the clustering is ameliorated.

Step 3: After computing the total gain of all objects in φ, the object g that has maximum γ_g is chosen. It is mathematically expressed as,

$$K = K \cup \{g\} \tag{23}$$

$$\varphi = \varphi - \{g\} \tag{24}$$

These steps are repeated until the k objects are chosen.

Step 4: The clustering quality is enhanced in the swap phase by optimizing the set of selected objects; subsequently, they are terminated by considering all swap pairs $(k, \phi) \in K \cup \varphi$ and calculating the effect $\tau_{k\phi}$ on the sum of dissimilarities betwixt objects and their cluster centres by swapping k and ϕ, and then transmitting ϕ from φ to K. $\tau_{k\phi}$ is calculated as,

$$\tau_{k\varphi} = \sum \{v_{tk\varphi} | t \epsilon \varphi\} \tag{25}$$

where, the contribution of every single objects t in φ to swap k and φ is denoted as $v_{tk\phi}$. If $\partial(t,k) > \vartheta_t$ or $\partial(t,k) = \vartheta_t$, then $v_{tk\varphi}$ can be measured as.

$$v_{tk\varphi} = \begin{cases} \min\{\partial(t,\varphi) - \rho_t, 0\}; \ \partial(t,k) > \rho_t \\ \min\{\partial(t,\varphi), \varepsilon_t - \rho_t\}; \ \partial(t,k) = \rho_t \end{cases} \tag{26}$$

Step 5: To estimate whether the swapping is done or not, the pair (k, ϕ) with the minimum $\tau_{k\phi}$ is chosen. If $\tau_{k\phi} < 0$, then, the swapping process is executed along with returned to the swap's beginning phase. Or else, record the medoids.

Step 6: After that, the objects that are nearer to the medoids and with minimum dissimilarities betwixt the objects are grouped into the clusters, which is expressed as,

$$\zeta_p = \{\zeta_1, \zeta_2 \ldots\ldots \zeta_n\} \tag{27}$$

3.5 Preserving Phase

The clusters formed in the partitioning phase are inputted to the preserving phase. Here, by utilizing the CG-ANN, the clustered data are classified into various classes so that the needed data can be accessed by the user effortlessly from a larger dataset. In general, owing to the need for higher memory space, higher computational complexities are faced by the ANN. Thus, to mitigate the higher memory consumption; in addition, to minimize the data's training time, the CG is utilized in the activation function of ANN. Fig. 4 exhibits the ANN's general structure.

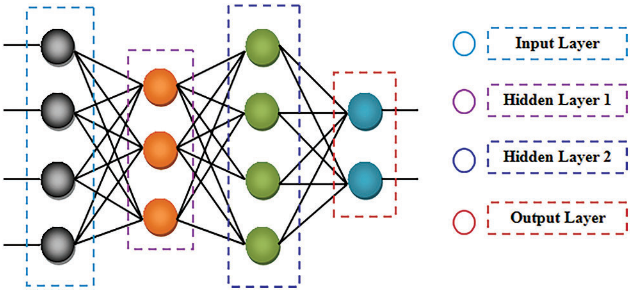


Figure 4: The general structure of the ANN

Following are the procedures included in CG-ANN.

(a) Input Layer

The input data is acquired by the input layer; subsequently, they are transmitted to the hidden layer. The input data and the respective weight values are given as,

$$\zeta_i = \{\zeta_1, \zeta_2 \ldots ..\zeta_n\} \tag{28}$$

$$\omega_i = \{\omega_1, \omega_2 \ldots ..\omega_n\} \tag{29}$$

(b) Hidden Layer

The input layer's output is inputted to the hidden layer. In the hidden layer, input features are trained by aggregating with weight values along with bias value; subsequently, activated by the CG activation function. The classifier's memory usage along with training time is mitigated enormously by the CG activation function, which is formulated as,

$$Z_i = \sum_{i=1}^{N} w_i \tau_i + b_a \tag{30}$$

$$\varepsilon_h^* = \Omega_{cg(\sum_{i=1}^{N} w_i \tau_i + b_a)} \tag{31}$$

where, the weight values are signified as w_i, the bias values that are initialized randomly are illustrated as b_a, and the activation function is signified as Ω_{cg}. Here, to activate the neurons, the CG activation function is utilized and are expressed as,

$$\Omega_{cg}^{(i+1)} = k^{(i+1)} + \Omega^{(i)} * \Delta^{(i)}; \, for \ 1 = 0, 1, 2, \ldots \tag{32}$$

where, the conjugate parameter is specified as $\Delta^{(i)}$.

(c) Output Layer

The respective outcomes as of the classifier are delivered by the output layer. By adding all the weights of the input signal, the output unit is computed, which is expressed as,

$$\Psi_i = f_s(\sum \varepsilon_h^* w_i + b_a) \tag{33}$$

where, the output unit is signified as Ψ_i, the hidden layers' weights are denoted as w_i, the value of the layer that precedes the output layer is defined as ξ_h^*, together with the softmax function is indicated as f_s. Lastly, by evaluating the target outcome with the actual outcome, the output's loss values are calculated. Thus, the overall loss value is gauged as,

$$\varepsilon r_t^* = \sum_{i=1}^{n} \varepsilon_i - \Psi_i^t \tag{34}$$

where, the error value is specified as εr_t^*, the actual outcome is signified as ξ_i, along with the targeted outcome is denoted as ψ_i. The model provides the appropriate solution if the value of the error $\varepsilon r_t^* = 0$; however, the back propagation is executed by updating the weight values if the error value $\varepsilon r_t^* \neq 0$. Lastly, the data is accessed as of the big transactional dataset effortlessly with the assist of the classification technique.

4 Result and Discussion

In this, regarding several performance metrics, the proposed framework's final outcome is evaluated with the prevailing methodologies. The performance along with the comparative evaluation is performed to confirm the work's efficiency. The proposed model is implemented in the working platform of JAVA and the data are acquired from the NSL-KDD dataset.

4.1 Performance Analysis of Proposed Classification Technique

The performance of the CG-ANN is evaluated with the prevailing ANN, decision tree, Naïve Bayes, along with CNN methodologies regarding performance metrics like accuracy, sensitivity, specificity, precision, recall, together with F-measure to establish the model's efficiency.

The performance appraisal of the CG-ANN with the prevailing ANN, decision tree, naïve bayes, and CNN methodologies regarding accuracy, sensitivity, together with specificity is demonstrated in Tab. 1. The accuracy, sensitivity, and specificity attained by the CG-ANN are 97.23%, 97.34%, and 96.84%, respectively, which are ranged betwixt 96.84%–97.34%. The average rates of accuracy, sensitivity, and specificity acquired by the prevailing ANN, decision tree, naïve Bayes, and CNN methodologies are 93.58%, 93.4%, and 93.7%, correspondingly, which are comparatively lower than the proposed methodology. Thus, an accurate outcome is provided to the user by the proposed methodology.

Table 1: Performance analysis of proposed CG-ANN based on accuracy, sensitivity, and specificity

Performance metrics (%)/Techniques	Accuracy	Sensitivity	Specificity
Proposed CG-ANN	97.23	97.34	96.84
ANN	95.24	95.68	96.23
Decision tree	94.86	94.33	94.28
Navie bayes	92.34	92.78	92.55
CNN	91.88	90.81	91.74

The comparative evaluation of the CG-ANN with the prevailing ANN, decision tree, naïve bayes, and CNN methodologies regarding the accuracy, sensitivity, along with specificity is elucidated in Fig. 5. In accordance with the graphical representation, higher rates of accuracy, sensitivity, and specificity are acquired by the proposed system. However, the rates of accuracy, sensitivity, and specificity attained by the prevailing methodologies are lower than the proposed one. Therefore, the CG-ANN surpasses the existent methodologies and provides better outcomes under complicated situations.

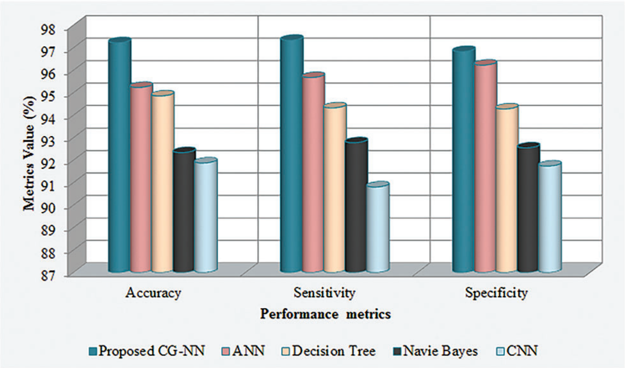


Figure 5: Graphical representation of the proposed CG-ANN based on accuracy, sensitivity, and specificity

The performance metrics like precision, recall, along with F-measure for the CG-ANN and the prevailing ANN, decision tree, naïve bayes, and CNN methodologies are illustrated in Tab. 2. The precision, recall, and F-measure obtained by the proposed model are 96.57%, 97.34%, and 96.84% whereas the prevailing ANN, decision tree, naïve bayes, and CNN methodologies acquires the precision that overall ranges betwixt 91.84%–95.26%, recall rate that overall ranges betwixt 90.81%–95.68%, and F-measure ranges betwixt 91.33%–94.68%. Thus, higher precision, recall, and F-measure values are achieved by the proposed than the prevailing models. Fig. 6 exhibits the graphical representation of Tab. 2.

Table 2: Performance analysis of proposed CG-ANN based on precision, recall, and F-measure

Performance metrics (%)/Techniques	Precision	Recall	F-Measure
Proposed CG-ANN	96.57	97.34	96.84
ANN	95.26	95.68	94.68
Decision tree	94.32	94.33	93.66
Navie bayes	92.84	92.78	93.14
CNN	91.84	90.81	91.33

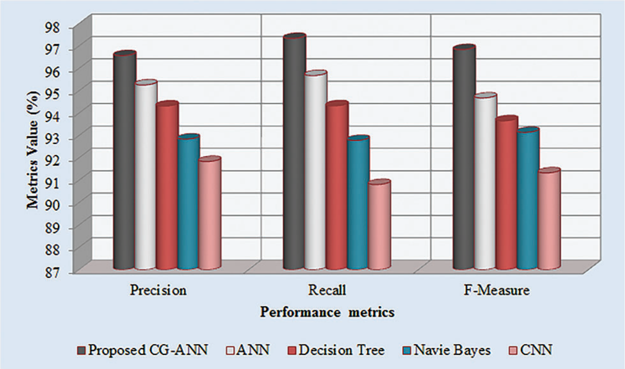


Figure 6: Graphical representation of the proposed CG-ANN based on precision, recall, and F-measure

The precision, recall, along with F-measure rates attained by the CG-ANN and the prevailing ANN, decision tree, naïve Bayes, and CNN methodologies are analogized in Fig. 6. The method should have higher precision, F-measure, together with recall rates to get better efficiency. Accordingly, the precision, recall, and F-measure attained by the CG-ANN are 96.57%, 97.34%, and 96.84%, respectively, which are higher than the prevailing methodologies. Thus, the CG-ANN alleviated the complications along with ameliorates the strength of the methodology.

4.2 Comparative Analysis of Proposed Technique in Terms of Dimensionality Reduction Time and Clustering Time

The proposed methodology's dimensionality reduction along with clustering time is analogized with several other prevailing methodologies like Linear Discriminant Analysis (LDA), FDA, Principle Component Analysis (PCA), Singular Value Decomposition (SVD), Fuzzy C-Means (FCM), K-Means, Brich, and Mean Shift to illustrate the model's efficiency.

The graphical assessment of the dimensionality reduction time of the GK-FDA with the existent methodologies like FDA, LDA, SVD, and PCA is demonstrated in Fig. 7. To complete the dimensionality reduction process, the time consumed by the GK-FDA is 9652 ms whereas the time utilized by the prevailing FDA, LDA, SVD, and PCA methodologies are 12354, 16874, 20145 and 24157 ms, correspondingly. Thus, it is noticed that the proposed model ameliorates the entire methodology's running time by completing the dimensionality reduction process rapidly. Consequently, the GK-FDA's performance is better than the prevailing methodologies.

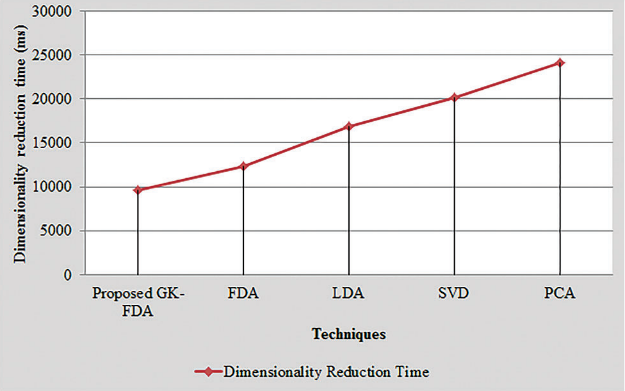


Figure 7: Graphical representation of the proposed GK-FDA based on dimensionality reduction time

The graphical view of clustering time of the MD-PAM with prevailing methodologies like FCM, K-means, Mean shift, along with Brich is depicted in Fig. 8. 17524 ms is needed by the MD-PAM to form cluster efficiently whereas the prevailing-means, FCM, Mean shift, and Brich methodologies required 22168, 26841, 30247 and 33268 ms, respectively. The existent models consume more time for cluster formation than the proposed model; thus degrading the overall system performance. Consequently, the proposed methodology consumes limited time to complete the whole process than the prevailing techniques.

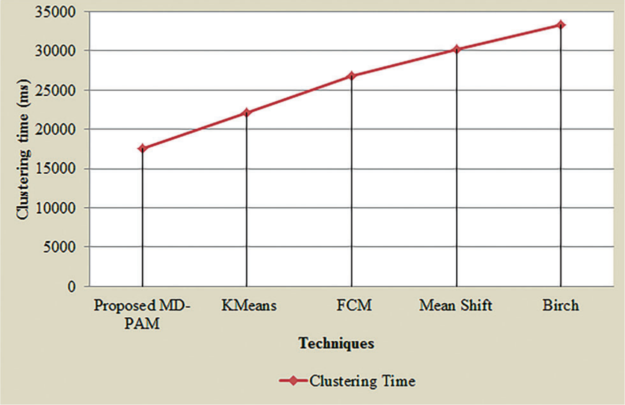


Figure 8: Graphical representation of the proposed MD-PAM based on clustering time

5 Conclusion

The proposed design the distribution preserving framework for BD, a novel technique has been proposed here utilizing MD-PAM along with CG–ANN. To certify the proposed model's efficiency, the proposed along with the prevailing methodologies' performance and comparative analysis are conducted regarding certain performance metrics. Several uncertainties are addressed by this methodology, which in turn provides a promising outcome. Here, the openly accessible datasets are utilized in which the accuracy, sensitivity, and specificity attained by the proposed model are 97.23%, 97.34%, and 96.84%, respectively. On the whole, the proposed methodology has better reliability along with robustness than the prevailing state-of-art methodologies. In the upcoming future, the research will be concentrated on the BD's distribution preserving together with privacy-preserving with enhanced algorithms along with neural networks.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Z. C.Dagdia, "A scalable and distributed dendritic cell algorithm for big data classification," *Swarm and Evolutionary Computation*, vol. 50, pp. 1–47, 2018.

[2] J. Ranjan and C. Foropon, "Big data analytics in building the competitive intelligence of organizations," *International Journal of Information Management*, vol. 56, no. 2, pp. 1–13, 2021.

[3] R. B. Dain, I. J. Miller, N. W. Kwiecien, D. J. Pagliarini, S. Michael *et al.,* "Argonaut a web platform for collaborative multi omic data visualization and exploration," *Patterns*, vol. 1, no. 7, pp. 1–8, 2020.

[4] S. S. Nicolaescu, A. Florea, C. V. Kifor, U. Fiore, N. Cocan *et al.,* "Human capital evaluation in knowledge based organizations based on big data analytics," *Future Generation Computer Systems*, vol. 111, no. 4, pp. 654–667, 2020.

[5] D. G. Rosado, J. Moreno, L. E. Snchez, A. S. Olmo, M. A. Serrano *et al.,* "MARISMA-BiDapattern Integrated risk analysis for big data," *Computers & Security*, vol. 102, no. 102255, pp. 1–25, 2020.

[6] A. Jabbar, P. Akhtar and S. Dani, "Real time big data processing for instantaneous marketing decisions a problematization approach," *Industrial Marketing Management*, vol. 90, no. 2, pp. 558–569, 2020.

[7] A. Ekambaram, A. O. Sorensen, H. B. Berg and N. O. E. Olsson, "The role of big data and knowledge management in improving projects and project based organizations," *Procedia Computer Science*, vol. 138, no. 3, pp. 851–858, 2018.

[8] P. Lu and S. Nie, "The strength distribution and combined duration prediction of online collective actions big data analysis and BP neural networks," *Physica a Statistical Mechanics and its Applications*, vol. 535, no. 3, pp. 1–16, 2019.

[9] H. Wang, W. Wang, L. Cui, H. Sun, J. Zhao *et al.,* "A hybrid multi objective firefly algorithm for big data optimization," *Applied Soft Computing*, vol. 69, no. 1, pp. 806–815, 2017.

[10] C. A. Marino and M. Marufuzzaman, "A microgrid energy management system based on chance-constrained stochastic optimization and big data analytics," *Computers & Industrial Engineering*, vol. 143, pp. 1–14, 2020.

[11] J. Song, Z. Ma, R. Thomas and G. Yu, "Energy efficiency optimization in big data processing platform by improving resources utilization," *Sustainable Computing Informatics and Systems*, vol. 21, pp. 1–13, 2018.

[12] C. B. Gonzalez, J. G. Nieto, A. J. Nebro, J. A. Cordero, J. J. Durillo *et al.,* "JMetalSP a framework for dynamic multi objective big data optimization," *Applied Soft Computing*, vol. 69, no. 4, pp. 737–748, 2017.

[13] N. Krishnaraj, B. Sivakumar, R. Kuppusamy, Y. Teekaraman and A. R. Thelkar, "Design of automated deep learning based fusion model for copy-move image forgery detection," *Computational Intelligence and Neuroscience*, vol. 2022, no. 8501738, pp. 1–13, 2022.

[14] N. Sun, B. Sun, J. D. Lin and M. Y. C. Wu, "Lossless pruned naive bayes for big data classifications," *Big Data Research*, vol. 14, pp. 27–36, 2018.

[15] Y. Zhuo and Z. Ge, "Gaussian discriminative analysis aided GAN for imbalanced big data augmentation and fault classification," *Journal of Process Control*, vol. 92, no. 11, pp. 271–287, 2020.

[16] Y. Abdi and M. R. F. Derakhshi, "Hybrid multi objective evolutionary algorithm based on search manager framework for big data optimization problems," *Applied Soft Computing*, vol. 87, no. 5, pp. 1–18, 2020.

[17] N. Krishnaraj, M. Elhoseny, E. Laxmi Lydia, K. Shankar and O. ALDabbas, "An efficient radix trie-based semantic visual indexing model for large-scale image retrieval in cloud environment," *Software: Practice and Experience*, vol. 51, no. 3, pp. 489–502, 2021.

[18] D. Prabakaran and S. Ramachandran, "Multi-factor authentication for secured financial transactions in cloud environment," *Computers, Materials & Continua*, vol. 70, no. 1, pp. 1781–1798, 2022.

[19] B. Zhang, X. Wang and Z. Zheng, "The optimization for recurring queries in big data analysis system with Mapreduce," *Future Generation Computer Systems*, vol. 87, no. 1, pp. 549–556, 2017.

[20] A. Ahmad, M. Khan, A. Paul, S. Din, M. M. Rathore *et al.,* "Toward modeling and optimization of features selection in big data based social Internet of Things," *Future Generation Computer Systems*, vol. 82, no. 2, pp. 715–726, 2017.

[21] M. Ianni, E. Masciari, G. M. Mazzeo, M. Mezzanzanica and C. Zaniolo, "Fast and effective big data exploration by clustering," *Future Generation Computer Systems*, vol. 102, pp. 84–94, 2020.

[22] G. Hernandez, E. Zamora, H. Sossa, G. Tellez and F. Furlan, "Hybrid neural networks for big data classification," *Neurocomputing*, vol. 390, no. 1, pp. 327–340, 2019.

[23] C. Banchhor and N. Srinivasu, "Integrating cuckoo search grey wolf optimization and correlative naive bayes classifier with map reduce model for big data classification," *Data & Knowledge Engineering*, vol. 127, no. 11, pp. 1–14, 2020.

[24] A. Bagozi, D. Bianchini, V. De Antonellis, M. Garda and A. Marini, "A relevance based approach for big data exploration," *Future Generation Computer Systems*, vol. 101, no. 1, pp. 51–69, 2019.

[25] F. BenSaid and A. M. Alimi, "Online feature selection system for big data classification based on multi objective automated negotiation," *Pattern Recognition*, vol. 110, no. 1, pp. 1–12, 2021.

[26] S. Aslan and D. Karaboga, "A genetic artificial bee colony algorithm for signal reconstruction based big data optimization," *Applied Soft Computing*, vol. 88, no. 1, pp. 1–19, 2020.

[27] K. S. Prakash and P. M. Joe Prathap, "Bitmap indexing a suitable approach for data warehouse design," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, no. 2, pp. 680–683, 2015.

[28] K. S. Prakash and P. M. Joe Prathap, "Tracking pointer and look ahead matching strategy to evaluate iceberg driven query," *Journal of Computer Sciences*, vol. 13, no. 3, pp. 55–67, 2017.

[29] K. S. Prakash and P. M. Joe Prathap, "Efficient execution of data warehouse query using look ahead matching algorithm," in *Int. Conf. on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, Pune, India, pp. 9–10, 2016.

[30] K. S. Prakash and P. M. Joe Prathap, "Priority and probability based model to evaluate aggregate function used in iceberg query," *International Journal of Applied Engineering Research*, vol. 12, no. 17, pp. 6542–6552, 2017.

[31] K. S. Prakash and P. M. Joe Prathap, "Evaluating aggregate functions of iceberg query using priority based bitmap indexing strategy," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 6, pp. 3745–3752, 2017.