International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Adaptive Dragonfly Optimization (Ado) Feature Selection Model and Distributed Bayesian Matrix Decomposition for Big Data Analytics

[1]Mrs M.Vijetha, [2]Dr G.Maria Priscilla,

Submitted: 06/02/2024 **Revised**: 14/03/2024 **Accepted**: 20/03/2024

Abstract: Matrix decompositions are fundamental methods for extracting knowledge from large data sets produced by contemporary applications. Processing extremely large amounts of data using single machines are still inefficient or impractical. Distributed matrix decompositions are necessary and practical tools for big data analytics where high dimensionalities and complexities of large datasets hinder the data mining processes. Current approaches consume more execution time making it imperative to reduce dataset feature counts in processing. This work presents a novel wrapper feature selection method utilising Adaptive Dragonfly Optimisation (ADO) algorithm for making the search space more appropriate for feature selections. ADO was used to transform continuous vector search spaces into their binary representations. Distributed Bayesian Matrix Decomposition (DBMD) model is presented for clustering and mining voluminous data. This work specifically uses, 1) accelerated gradient descent, 2) alternate direction method of multipliers (ADMM), and 3) statistical inferences to model distributed computing. These algorithms' theoretical convergence behaviours are examined where tests reveal that the suggested algorithms perform better or on par with two common distributed approaches. The methods also scale up effectively to large data sets. Clustering performances are assessed using the metrics of precision, recall, F-measure, and Rand Index (RI), which are better suited for imbalanced classes.

***Index Terms**: Distributed algorithm, Bayesian matrix decomposition, clustering, data mining, feature selection, AdaptiveDragonfly Optimization (ADO), and big data.*

1. Introduction

With the introduction of 5G technologies, enormous amounts of data are being produced extremely fast. This enormous volume of data is referred to as Big Data. Data analytics are extremely challenging due to characteristics like enormous volumes, multivalued data, high velocities, and broad diversities in data. Furthermore, it is not an easy process to extract useful information from such large amounts of data [1]. Clustering algorithms are vital techniques in data mining crucial to analyze large data.

The amount of digitally saved data has increased dramatically over the past 20 years, and growing interest in investigating and creating cutting-edge big data processing and data mining approaches have also increased in paralell. Dealing with huge volumes of high-dimensional data typically requires the adoption of suitable low-dimensional representations [2] as they can effectively disclose hidden structural information in high-dimensional data while simplifying computations and improve learning.

Matrix factorizations are popular dimensionality reduction techniques as they can learn low-dimensional representationsof high-dimensional data. Nonnegative matrix factorization (NMF) [3, 4], singular value decomposition (SVD) [5, 6], QR decomposition (QRD) [7], Linear Discriminant Analysis (LDA) [7], Principal Component Analysis (PCA) [8, and Independent Component Analysis (ICA) [9] are some more well-known instances of canonical approaches.

Standard matrix decomposition techniques, however, have a tendency to overfit the noisy, missing-value observed matrix. Overfitting can be minimised by the enforced regularizers, although careful parameter tweaking is necessary [10,11]. The majority of those algorithms often operate on the assumption that the data matrices have a similar basis matrix, which allows the approaches to conduct an integrative analysis. The analysis of low noise data may be impacted by the high noise data perspective, as not all current approaches take into account the heterogeneity of the noise [11,12]. In order to tackle this issue, Zhang and Zhang [13] utilised the Bayesian methodology and suggested the implementation of Bayesian joint matrix decomposition (BJMD). Their research demonstrated that by considering heterogeneous noises, clustering efficiencies could be enhanced. Yet there is still a paucity of theoretical analysis. Thus, it is crucial to create effective matrix decomposition techniques in distributed systems.

Numerous distributed matrix decomposition techniques have been created by researchers, such as distributed NMF [15], distributed Bayesian probabilistic matrix

[1]Part Time Research Scholar, Department of Computer Science, Sri Ramakrishna College of Arts and Science, Coimbatore.
Email: vijethachandran1983@gmail.com
[2]Head-Department of Computer Science, Sri Ramakrishna College of Arts and Science, Coimbatore

factorization (BPMF) [14], and so on. The partition strategy of the distributed data should be taken into consideration while developing effective distributed algorithms for matrix decomposition techniques, and an appropriate optimisation approach should subsequently be adopted. Few research, nonetheless, have examined various optimisation techniques and clarified how they vary. The fact that so few approaches address the variability of noise among dispersed data is even more concerning.

One of the most popular methods for reducing the high dimensionality of large-scale data is feature selection (FS), which builds an effective clustering by selecting a small subset of relevant and related features and removing redundant and unrelated features. In order to improve classification performance, feature selection techniques seek to remove characteristics that are noisy, redundant, or irrelevant [16, 17]. Nevertheless, conventional techniques are not sufficiently scalable to handle datasets containing millions of occurrences and get good outcomes within a restricted timeframe. In order for the results of the data mining techniques used over the reduced dataset to be as near as possible—or even better—to the results produced using all attributes, the feature selection procedure aims to get a minimum set of attributes. However, when working with really big issues, an excessive increase in the individual size may restrict their usefulness since they cannot deliver a pre-processed dataset in a fair amount of time. There are no methods for dealing with the feature space using evolutionary big data models in the literature as of yet [19].Evolutionary approaches are among the methods now in use that have shown effective for feature selection [18].

In order to lower the overall amount of features in the dataset, a novel wrapper feature selection via ADO approach is suggested in this study. For massive data mining and clustering, a DBMD model that expands on the Bayesian joint matrix decomposition (BJMD) is developed. In particular, three approaches—namely, the Accelerated Gradient Descent (AGD), the ADMM, and the Communication-Efficient Accurate Statistical Estimation (CEASE)—are presented in order to implement distributed computing. Their convergence behaviour is then examined in the context of distributed computing. Their research demonstrated that taking heterogeneous noise into account improves clustering efficiency.

2. Literature Review

Chavoshinejad et al [3] proposed Self-Supervised Semi-Supervised Nonnegative Matrix Factorization (S^4NMF) in semi-supervised clustering settings for effectiveness. The schema derived consensus results directly using regularisations for similarities/ dissimilarities from ensembled NMF. These data were iteratively sent back to the suggested model for enhancing semi-supervised learning and generate unique clusters. The optimisation issue with well-specified objective functionswere defined in the suggested iterative approach. Additionally, the convergence of the suggested optimisation method is examined by the theoretical and empirical investigations. Extensive tests are undertaken on common benchmark datasets to illustrate the efficacy of the proposed approach in semi-supervised clustering.

Wang et al [20] proposed generalized deep learning clustering (GDLC) algorithm based on NMF. First, a nonlinear constrained Nonnegative Matrix Factorization (NNMF) method was built with learning rates as guides to accomplish sequential updates of matrixcomponents. Gradients corresponding to element updates are transformed into weights and generalised biases by feeding into non-linear activation functionsfor understanding complex inference processes and develop GDLC algorithm. The results show that GDLC functions well from the experimental findings of eight datasets.

Zhang et al [21] proposed DBMD for mining and clustering big data.Three approaches demonstrated distributed computing: 1) statistical inferences, 2) ADMM, and 3) AGD. Zhang et al. [22] created three methods: MSC IAS (Multi-view Subspace Clustering with Intactness-Aware Similarity), LMSC (Latent Multi-view Subspace Clustering), and SwMC (Self-weighted Multiview Clustering). The three most important measurements are accuracy (ACC), normalised mutual information (NMI), and purity. Tensor decomposition approaches based on constraint selections have been developed to extract more comprehensive latent components. Several tensor models were also investigated in trials ofapplicationlinked to supervised learning and dimensionality reductions.

Wang et al [23]exploited tensor training decompositions and for learning data's low-dimensional representations using high-dimensional tensors' stable representations. Support Vector Machine (SVM)and tensor train decompositions were combined SVM based on lowly ranked tensor decompositions on voluminous data. The proposed method resulted in significant processing savings while maintaining high classification performances. Duan et al. [24] proposed an effective extreme learning machine (SELM) based on the Spark framework. It was made up of three simultaneous subalgorithms for categorising enormous amounts of data. Hidden layer output matrix computation procedures were achieved Through rational divisions of data sets,.The costs were reduce by saving intermediaty results in distributed memories and caches and using diagonal matrices as

broadcast variables for job copies. These moves improved SELM's capacity for learning.

Xie et al [25]explored feature space clustering as a solution to the low efficiency that K-means clustering of large data sets produced. In contrast to conventional techniques, the algorithm ensured accuracy of clusters both before and after reducing dimension, accelerated K-means when distance functions and clustering centres met specific requirements, matched exactly in both preprocessing and clustering stages, and increased accuracy and efficiency. Results from experiments have shown that the suggested strategy is successful.

Chen et al [26] suggested deep matrix factorizationsof large data sets with non-negative constraints that attempted to generate deeper interpretabilitiesof representations. Deep architectures are designed specifically for pattern mining where supervisor networks reduce noises in inputs and learndeep representations of interpretability. Deep matrix factorization architectures learn from constructedintreprabilitylosses, which constitutes symmetric, opposition, and non-negative constraint losses. This ensures knowledge transmissions from supervisor networks to learning networks, enhancing deep representative durability. The work's experiments on benchmarked datasets indicated advantages of deep matrix factorization techniques.

Tang and Feng [27] suggested a unique strong clustering for robust local-coordinate NMF with adaptive graph (RLNMFAG). It lessens the disruption caused by noise (outliers) in space exploration and data restoration. Furthermore, orthogonal regularisation terms are added to models, guaranteeing factor matrix's orthogonality and improving capacity of discriminantions. The resulting model is then solved using an effective method, and its convergence is examined from both theoretical and experimental perspectives. Experimental findings on benchmark databases and random synthetic data sets show that the proposed technique performed better than robust NMF methods in terms of resilience, discriminanations, and learning of spatial structures.

3. Proposed Methodology

A novel wrapper feature selection via ADO approach is suggested in this study in order to lower the overall amount of features in the dataset,. To make the search space more appropriate for the feature selection issue, the ADO technique was used to transform the continuous vector search space into a binary one. The DBMD model is presented for clustering and massive data mining. Large-scale studies confirm the theoretical findings, and practical tests demonstrate the applicability and efficacy of suggested techniques.

2.1 ADO algorithm

One type of evolutionary algorithm is ADO. The concept developed from the way dragonflies behaved, both dynamically and statically. A static swarm of dragonflies is defined as having characteristics intended for hunting prey inside a limited area. Swarm DF's dynamic behaviour is utilised to describe the exploring stage. This lays the groundwork for ADO. Five characteristics of dragonflies are presented in order to create the mathematical model for representing the flies' movements in a dataset: separation, alignment, cohesiveness, feeding, and adversary. The separation, alignment, cohesion, food, and enemy properties of an i sample dragonflies in a dataset are represented by the variables Se_i, Al_i, Co_i, af_i, and Ee_i. In the given locale, the distance between neighbouring DF is required to maximise the feature search space and prevent collision. Let i be the number of samples with 'n' neighbours in a dataset. Divorce The location of the k_th neighbouring DF is indicated by x_k, and Se_i of an individual with x being the current feature position of DF.

$$Se_i = \sum_{k=1}^{n} x - x_k \tag{1}$$

Matching velocities of features with other DFswithin same localities are based on alignmentsAl_i as represented in Equation (2) and where v_k represents velocities of kthneighboring DF.

$$Al_i = \frac{\sum_{k=1}^{n} V_k}{n} \tag{2}$$

All individuals of DF are inclined to move in directions of mass centers of neighboringflies. The cohesion features $Co_i$of DF can be determined using Equation (3),

$$Co_i = \frac{\sum_{k=1}^{n} V_k}{n} \tag{3}$$

Given that food is necessary for survival, all the characteristics in a dataset have a tendency to migrate in that direction. Equation (4) is used to obtain the attraction for food Af_i feature at position x_{food}.

$$Af_i = x_{food} - x \tag{4}$$

DatasetSamples tend to move away from the enemy. The enemy feature Ee_i at locationsx_e can be computed using equation(5).

$$Ee_i = x_e + x \tag{5}$$

The behavior of DF in a dataset is influenced by the combining all the five attributes. The updated location of

the individual DF is calculated by step Δx_i wasdenoted in equation (6).

$$x_i = x_i + \Delta x_i \tag{6}$$

$$\Delta x_i = aw_i \Delta x_i + a.Se_i + b.Al_i + c.Co_i + d.Af_i + e.Ee_i \tag{7}$$

where the food component is represented by d and the adversary factor by e. The inertial weight is denoted by aw_i, while the separation, alignment, and cohesiveness weights are represented by a, b, and c, respectively. The properties of an individual dragonfly in a cluster that are utilised to express separation, alignment, cohesiveness, food, and enemy are represented by Se_i, Al_i, Co_i, af_i, and Ee_i. By utilising various parameter values, the DF's exploratory and exploitative behaviours may be varied. In the ith dimension, the weight vector aw_i may be computed as follows:

$$aw_i = x + \frac{p \times (p+1)}{2} \times D \tag{8}$$

where awstands for weight vectors, $p \in \left\{1, \ldots, \left\lceil \frac{H'}{2} \right\rceil \right\}$ which is dependent on sub-interval weight vectors t, D implies selections from $d_{1,i}$ or $d_{2,i}$.H' represents weight vector counts generated from feature ranges in i^{th} samples. The overall flow of the proposed ADO algorithm is illustrated in the figure 1.

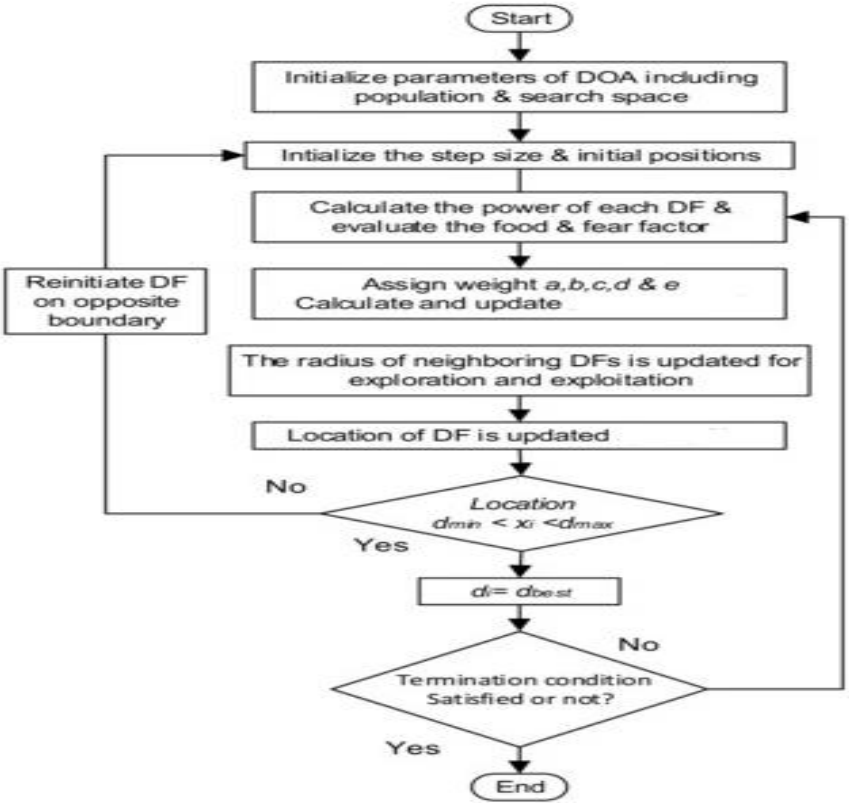

FIG 1. FLOW DIAGRAM OF THE PROPOSED ADO ALGORITHM

The pseudo code for the application of ADO algorithm is discussed in algorithm 1.

Algorithm 1. ADO algorithm

Initilization of DF population $x_i (i = 1, \ldots, n)$

Initilizing step valuesx_i

While end condition is not met

Determine classification accuracies of features by DF

Update food sources and enemies

Update values of a,b,c,d and e

Calculatre Se_i, Al_i, Co_i, af_i, and Ee_iusing equations (1-5)

Update nearest feature accuracies

If DF has atleast one neighbour

Update feature positionvectors using equation (6)

Update velcoity vectorsusing equation (7)

Else

Update feature position vectorswith adaptive weights using equation (8)

End if

Examine and correct new feature positions based on attribute boundaries

End while

3.1. DBMD model

Assumedata matrices are represented by$X \in R^{m\times n}$, where m implies feature counts, n stands for sample counts and $n \gg m$. Processing X on single machines is complex due to large values of n and X is split and stored on C machines $\{X_C\}_{c=1}^{C}$ based on columns swhere $X_C \in R^{m\times n_c}$ and $\sum_{c=1}^{C} n_c = n$. Assuming $\{X_C\}_{c=1}^{C}$ based on [13]sharing same basis matrix W can be generated using:

$$X_c = WH_c + E_c \tag{9}$$

where $W \in R^{m\times r}, H_c \in R^{r\times n_c}$ and E_c represents IID Gaussiannoises, $(e_{ij})_c \sim N(0, \sigma_c^2)$. Moreover, zero-mean Laplaceis placed onW for enforcing sparsity.

$$w_{ik} \sim p(w_{ik}|0, \lambda_0) \tag{10}$$

where density functions are,

$$p(y|\mu, \lambda) = \frac{1}{2\lambda}\exp\left(-\frac{|y-\mu|}{\lambda}\right) \tag{11}$$

Which are proportional to complete likelihoodnessbased on Bayes theorem and written as:

$$p(W, H_1, \ldots., H_C, \sigma_1^2, \ldots., \sigma_C^2, X_1, \ldots, X_C; \lambda_0, \alpha_0) = p(W; \lambda)\prod_{c=1}^{C} p(X_c|W, H_c, \sigma_c^2)\, p(H_c; \alpha_0) \tag{12}$$

Optimization problem as follows,

$$\min_{W,H_c} \sum_{c=1}^{C} \frac{1}{2}\|X_c - WH_c\|_F^2 + \lambda\|W\|_1 - \sum_{c=1}^{C}\sum_{k,j} \alpha_k \ln(h_{kj})_c \;\; s.t \sum_{k=1}^{r}(h_{kj})_c = 1, (h_{kj})_c > 0 \tag{13}$$

where L_1-norm regularizerensures W's sparsity. Dirichlet prior results in 3rd term and regularizes coefficient matrices $\{H_c\}_{c=1}^{C}$ of Dirichlet prior. Hence, third term enforces$(h_{.j})_c$on prior and reduces overfit risks.Generally, W and $H_c$are alternatively updated.

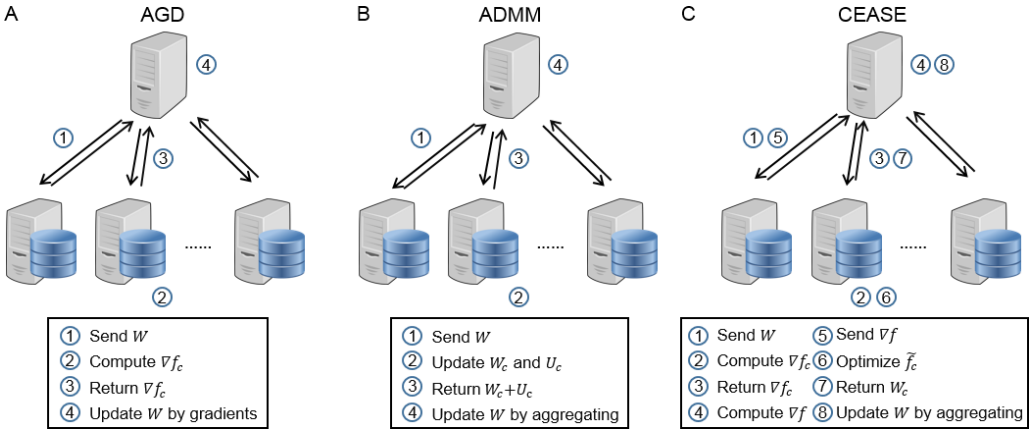

FIG 2. ILLUSTRATION OF UPDATING W

AGD optimizes W on central processors. W_c is updated on the nodes by ADMM and CEASE, and subsequently results are ahhregatedon central processors. Figure 2A summarises AGD, Figure 2B summarises ADMM, and Figure 2C summarises CEASE.

AGD: By storing corresponding H_c on c^{th} nodes and W on central processors, maximum posterior algorithm in [13] solves equation (13) by updating W and

$\{H_c\}_{c=1}^{C}$ alternatively. Specifically, W is updated with other parameters fixed,

$$\min_{W} \frac{1}{2}\|X_c - WH_c\|_F^2 + \lambda\|W\|_1 \tag{14}$$

Let $\mathbf{f_c(W)} = \frac{1}{2}\|X_c - WH_c\|_F^2$ and $g(W) = \lambda\|W\|_1$. The quadratic loss terms and the non-smooth L_1-norm regularizer g(W) make up objective functionsin equation (14). Consequently, fast iterative shrinkage-thresholding method (FISTA) may solve it effectively. The AGD algorithm FISTA has a quadratic rate of convergence. Specifically, two sequences $\{Y^k\}$ and $\{W^k\}$ are constructed, and alternatively update them,

$$W^k = \arg\min_{W} g(W) + \frac{L}{2}\left\| W - \left(Y^k - \frac{1}{L}\nabla\tilde{f}_c(Y^k)\right)\right\|_F^2 \tag{15}$$

$$Y^{k+1} = W^k + \frac{v_k - 1}{v_{k+1}}\left(W^k - W^{k-1}\right) \tag{16}$$

where $L_f = \left\|\sum_{c=1}^{C} H_c H_c^T\right\|_2 > 0$ 0 implies Lipschitz constants of $\sum_c \nabla f_c(W)$. W^kincludes approximations of minimized proximal functions. Equation (15) depicts closed-form solutions,

$$W^k = S_{\lambda/L} + \left(Y^k - \frac{1}{L}\sum_{c} \nabla f_c(Y^k)\right) \tag{17}$$

$$S_{\lambda/L}(X) = \text{sign}(X) \circ \max\left(|X| - \frac{\lambda}{L}, 0\right) \tag{18}$$

$S_{\lambda/L}(X)$ represents soft thresholding operators, and $\circ$ implies Hadamard products. Y^{k+1} saves search points constructed by linearcombinations of two recent approximate solutions (W^{k-1} and W^k). v_{t+1} , the coefficient combinations were designed using:

$$v_{k+1} = \frac{1 + \sqrt{4v_k^2 + 1}}{2} \tag{19}$$

Current Y^k is broadcastedto all nodesduring cycles for computing their gradients. Subsequently, computed gradients are gathered and W^k updatedbefore sending them to central processors. W^k and Y^k are updated iteratively until convergence. $\{H_c\}_{c=1}^{C}$ aresimple updates. The following issues can be fixed by using the same method once Broadcast W is sent out by central processors to nodes [13]:

$$\min_{H_c} \frac{1}{2}\|X_c - WH_c\|_F^2 - \sum_{c=1}^{C}\sum_{k,j} \alpha_k \ln\left(h_{kj}\right)_c \text{ s.t.} \sum_{k=1}^{r}\left(h_{kj}\right)_c = 1, \left(h_{kj}\right)_c > 0 \tag{20}$$

with respect to H_c in parallel.There is no data transmission necessary to solve equation (20). As a result, it is identical to the single machine algorithm. One issue, on updating W in a distributed system is that it might take a long time to broadcast W to nodes and pntaingradients. In a distributed system, internode communication can occur at a substantially slower speed than intranode processing [28]. As a result, data transfer is frequently the distributed algorithm's bottleneck, therefore updating W needs careful thought.

ADMM: ADMM implementations update W. Note that W in equation (22) are global variables and optimization problem can be formulated as:

$$\min_{W,W_c} \sum_{c=1}^{C} \frac{1}{2}\|X_c - W_c H_c\|_F^2 + \lambda\|W\|_1 \text{ s.t. } W_c = W \tag{21}$$

where W represents consensus variables and their augmentedLagrangian is ,

$$L_\rho(W, W_c, U_c) = \sum_{c=1}^{C} \frac{1}{2}\|X_c - W_c H_c\|_F^2 + \lambda\|W\|_1 + \sum_{c=1}^{C}\langle U_c, W - W_c\rangle + \frac{1}{2}\rho\sum_{c=1}^{C}\|X_c - W_c\|_F^2 \tag{22}$$

where $\rho > 0$implies parameters used for penalties, and $U_c$represents corresponding dual variables. W stands for global variables stored on central processors, and nodes locally store W_c, U_c, H_c. ADMM at $(k + 1)^{th}$ iterations encompass steps given below,

$$W_c^{k+1} = \arg\min_{W_c} L_\rho\left(W^k, W_c, U_c^k\right) \tag{23}$$

$$W^{k+1} = \arg\min_{W} L_\rho\left(W, W_c^{k+1}, U_c^k\right) \tag{24}$$

$$U_c^{k+1} = U_c^k + \rho\left(W^{k+1} - W_c^{k+1}\right) \tag{25}$$

Both equation (23) and equation (24) have the closed-form solutions,

$$W_c^{k+1} = \left(\frac{X_c H_c^T + U_c^k}{\rho} + W^k\right)\left(I_r + \frac{H_c H_c^T}{\rho}\right)^{-1} \quad (26)$$

$$W^{k+1} = S_{\lambda/C_\rho}\left(\bar{W}_c^{k+1} - \frac{\bar{U}_c^k}{\rho}\right) \quad (27)$$

where soft thresholding operators S_{λ/C_ρ} with parameters$\lambda/C_\rho$are defined. $W_c$get optimised locally on nodes parallelly and doesnot needto be transmitted. Parallel local optimisation is also applied to U_c. $\bar{W}_c^{k+1} - \frac{\bar{U}_c^k}{\rho}$ is all that is required, along with a gentle thresholding procedure. $W_c^k - U_c^k$is calculated concurrently on each node computer, and the results are aggregated on the central node. After obtaining the updated W, apply the thresholding operator and broadcast it to every node. It should be noted that just a matrix of size m×rhas to be collected and broadcast at each iteration. As a result, there was a noticeable decrease in the data communication burden.

Efficient Distributed Statistical Inference: Because of its efficacy, CEASE is presented to design an efficient distributed statistical process. Assume that W is W_k at iteration k. As per the CEASE system, every node computer calculates,

$$W_c^k = \arg\min_W \tilde{f}_c(W) \quad (28)$$

$$\tilde{f}_c(W) = f_c(W) - \langle \nabla f_c(W^k) - \nabla f(W^k), W\rangle + \frac{\gamma}{2}\|W - W^k\|_F^2 + g(W) \quad (29)$$

where $\gamma \geq 0$represents parameters of proximal points algorithm. $f_c(W) - \langle \nabla f_c(W^k) - \nabla f(W^k), W\rangle$ signifies gradient-enhanced loss (GEL) functions where local data losses$X_c$are enhanced by global gradients$\nabla f(W_k)$. Global gradients adaptively enhancesimilarities of f_c and accelerateconvergences. The solutions are closed ones when $\lambda = 0$, while $\lambda > 0$equation (28) encompasses non-smooth L_1-norm regularizer g(W) and remaining smooth terms which are sums of quadratic loss terms and linear terms which can also be effectively solved by FISTA. Optimizations of Wk c does not require any data communications. The central processors collect W_c^k and aggregate by averaging $W^{k+1} = \frac{1}{C}\sum_{c=1}^{C} W_c^k$.

4. Experimental Results

The proposed strategies for clustering datasets are contrasted with Scalable-NMF and distributed k-means and where experiments of synthetic data were conducted on 2GHz Intel Xeon E5-2683 v3 CPU desktop with GTX 1080 GPU card, and 16GB RAM, while real-world data was tested on a small Spark cluster comprising of eight workstations (one central processor and nodes) where every system had a 16GB memory with an Intel i7 CPU.

DATABASE: In total, there are three datasets. Specifically, UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets.php) downloaded RCV1. Data on forest cover types is called CoverType. There are seven classes in all, and two classes account for 85% of the occurrences. There are 23 courses in KDD-99.Eleven classes were kept, while those that happened less than 100 times were eliminated. 97% of cases are accounted for by the three largest classes, which are 57%, 21%, and 19%, respectively. SUSY has only two classes and is a physical dataset.

METRICS: Pairand instance-level measures can be calculated within cluster-level calculation framework with performance metrics of precision, recall, f-measure, and RI .

F-measures are harmonic means of Cluster Recall (CR) and Cluster Precision (CP) values and depicted in equations (30-32),

$$CR = \frac{|P \cap T|}{|T|} \quad (30)$$

$$CP = \frac{|P \cap T|}{|P|} \quad (31)$$

$$CF = \frac{2 \times CR \times CP}{(CR + CP)} \quad (32)$$

Where, P implies sets of predicted clusters, while T stands for sets of truth clusters. The numerator $|P \cap T|$ counts predicted clusters that contain all and only instances belonging to same truth clusters. CR signifies ratio of numerator over truth cluster counts ($|T|$). CPstands for ratio of this numerator over predicted cluster counts ($|P|$).

RI as measures of percentages of correct decisions made by algorithms. They can be computed using equation (33),

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \quad (33)$$

where TP implies true positive counts, TN stands for true negative counts, FP implies false positive counts, and FN signifies false negative counts.

TABLE 1. SUMMARY OF THE DATASETS

Dataset	No.of instances	No.of features	No.of classes
Covertype	~540,000	54	7
KDD-99	~4,900,000	41	11
SUSY	~50,00,000	18	2

TABLE 2. CLUSTERING PERFORMANCE ON REAL-WORLD DATASETS

Covertype				
Clustering methods	**Precision (%)**	**Recall (%)**	**F-Measure (%)**	**RI (%)**
NNMF	72.15	75.10	73.59	76.72
S^{4}NMF	75.62	78.91	77.23	79.05
RLNMFAG	80.25	82.41	81.32	82.19
DBMD-AGD	83.40	92.73	91.03	81.56
DBMD-ADMM	90.92	92.99	91.94	81.63
DBMD- CEASE	91.76	93.50	92.62	82.69
KDD-99				
Clustering methods	**Precision (%)**	**Recall (%)**	**F-Measure (%)**	**RI (%)**
NNMF	74.37	76.21	75.28	78.09
S^{4}NMF	76.15	80.65	78.33	80.25
RLNMFAG	79.24	82.54	78.86	81.61
DBMD-AGD	83.48	86.69	79.54	81.49
DBMD-ADMM	84.44	87.19	80.64	81.73
DBMD- CEASE	86.47	88.21	81.65	82.45
SUSY				
Clustering methods	**Precision (%)**	**Recall (%)**	**F-Measure (%)**	**RI (%)**
NNMF	77.23	75.77	76.49	75.66
S^{4}NMF	79.45	77.81	78.62	77.25
RLNMFAG	81.56	80.23	80.89	80.34
DBMD-AGD	82.21	80.79	80.97	81.29
DBMD-ADMM	85.86	85.96	85.91	81.62
DBMD- CEASE	87.68	87.75	87.72	81.74

Suggested clustering methodachieves competitive/superior performances compared others in terms of values for precision, recall, F-measure, and RI on three datasets (Table 2).

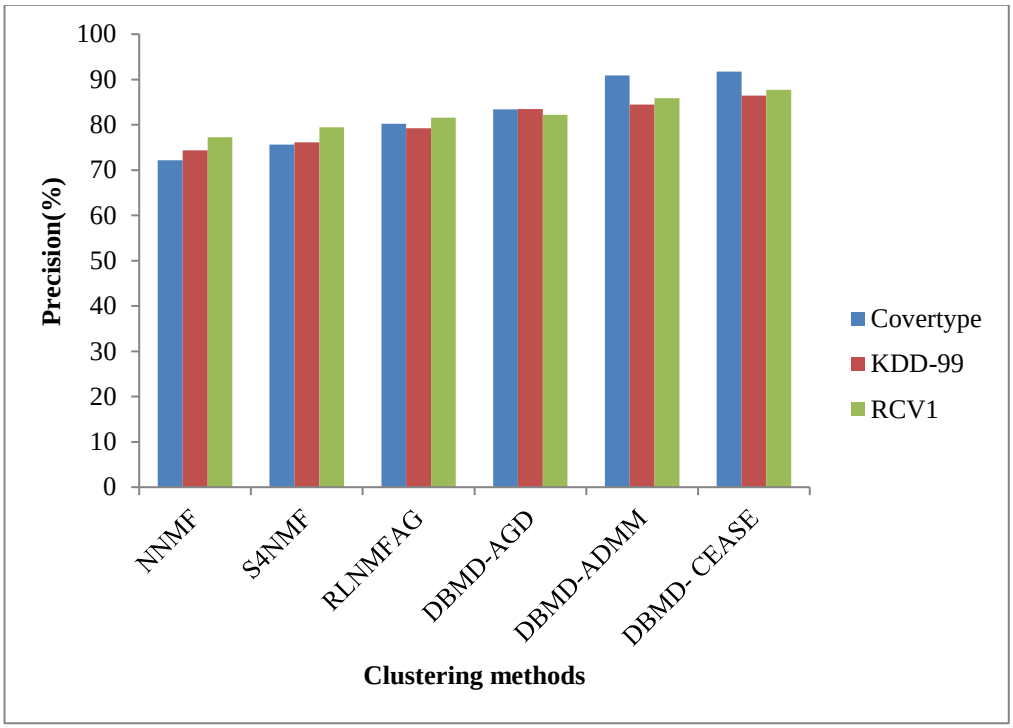

FIG 3. PRECISION RESULTS COMPARISON VS. CLUSTERING METHODS

Figure 3 shows performance comparisons of precision results obtained by clustering methods. The results are measured by various clustering methods and three datasets. From the results it shows that the proposed system has highest precision results of 91.76%, 86.47%, and 87.68% for covertype, KDD-99, and SUSY dataset. Other existing methods like NNMF, S^4NMF, RLNMFAG, DBMD-AGD, and DBMD-ADMM gives lesser precision of 77.23%, 79.45%, 81.56%, 82.21%, and 85.86% for SUSY dataset.

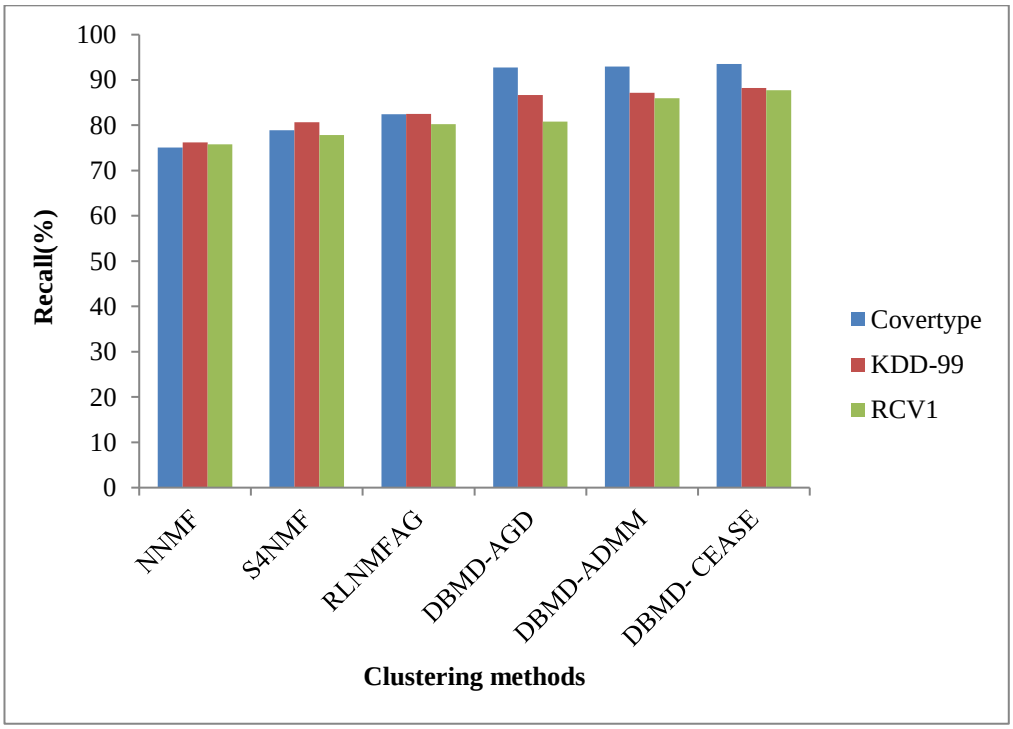

FIG 4. RECALL RESULTS COMPARISON VS. CLUSTERING METHODS

Recall results with respect to clustering methods by three datasets are illustrated in figure 4. It shows that the proposed system has highest results of 93.50%, 88.21%, and 87.75% for covertype, KDD-99, and SUSY dataset. NNMF, S^4NMF, RLNMFAG, DBMD-AGD, and DBMD-ADMM methods provides lesser recall of 75.77%, 77.81%, 80.23%, 80.77%, and 85.96% for SUSY dataset.

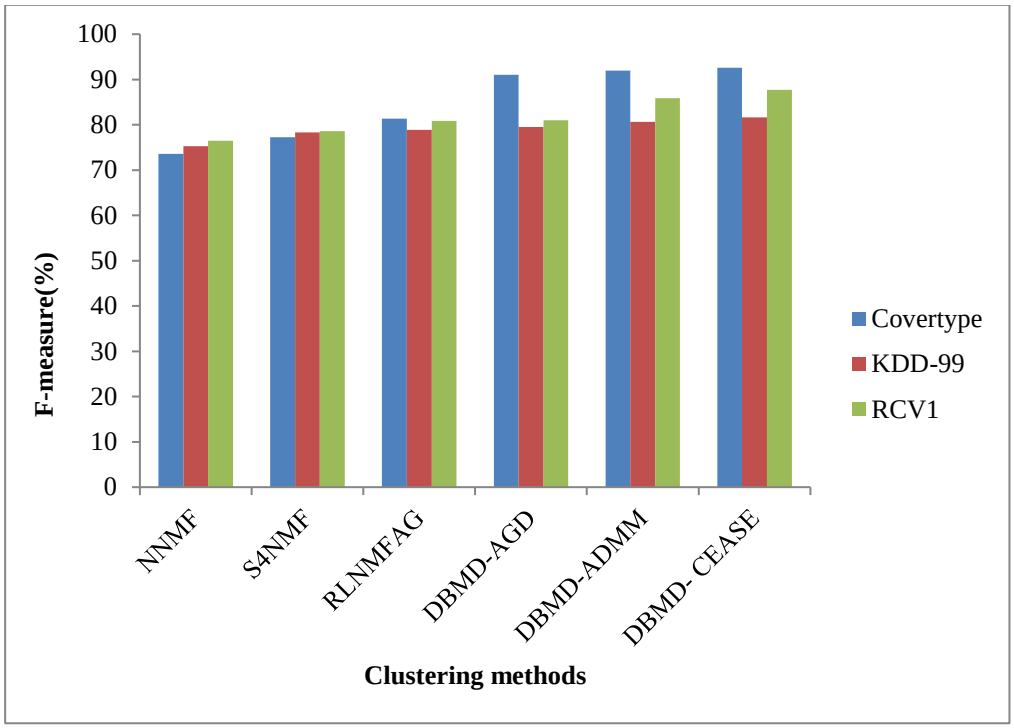

FIG 5. F-MEASURE RESULTS COMPARISON VS. CLUSTERING METHODS

Clustering methods by three datasets among f-measure are illustrated in figure 5. Proposed clustering algorithm has highest results of 92.62%, 81.65%, and 87.72% for covertype, KDD-99, and SUSY dataset. NNMF, S^4NMF, RLNMFAG, DBMD-AGD, and DBMD-ADMM methods provides lesser f-measure of 76.49%, 78.62%, 80.89%, 80.97%, and 85.91% for SUSY dataset.

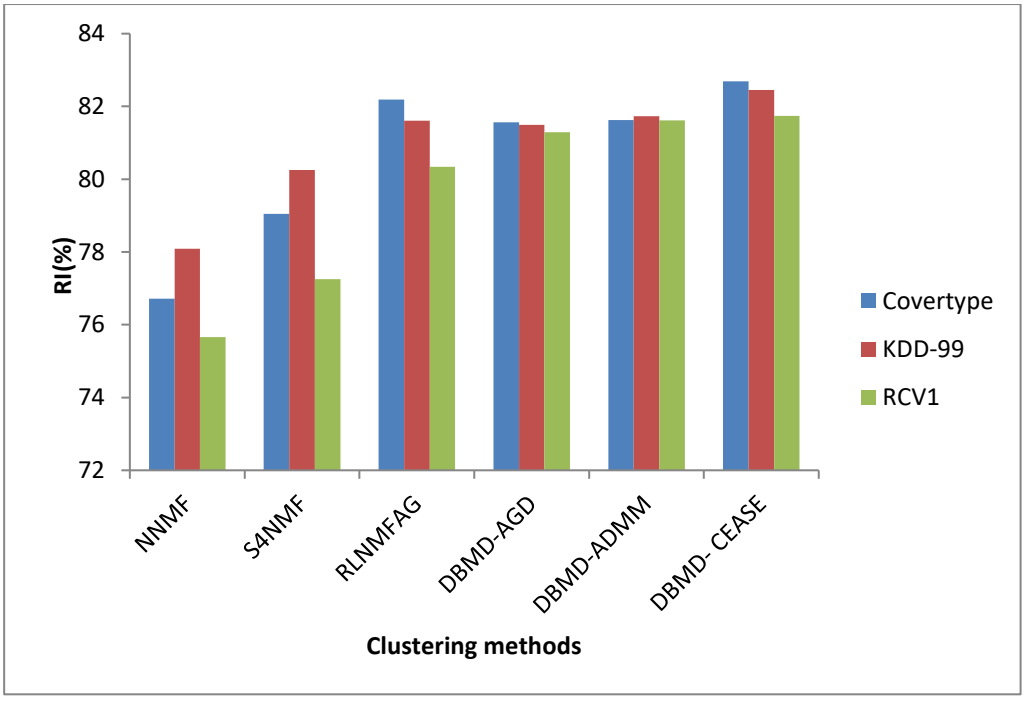

FIG 6. RI RESULTS COMPARISON VS. CLUSTERING METHODS

RI results of clustering methods among three datasets are illustrated in figure 6. Proposed clustering algorithm has highest results of 82.69%, 82.45%, and 81.74% for covertype, KDD-99, and SUSY dataset. NNMF, S^4NMF, RLNMFAG, DBMD-AGD, and DBMD-ADMM methods provides lesser RI of 75.66%, 77.25%, 80.34%, 81.29%, and 81.62% for SUSY dataset.

5. Conclusion and Future Work

An ADO feature selection for large data analytics is suggested in this research. As a pre-processing step to

decrease dataset dimensionality, the ADO method was introduced for the most informative features and classification accuracy. A static swarm of dragonflies is defined as having characteristics intended for hunting prey inside a limited area. Swarm DF's dynamic behaviour is utilised to describe the exploring stage. It is employed to randomly sample and analyse vast amounts of data into subsets. DBMD is then introduced for large data mining and grouping based on the characteristics that have been chosen. To develop distributed computing, three methodologies are introduced: 1) AGD, 2) ADMM, and 3) statistical inference. Numerous tests confirm the theoretical findings of clustering that have been suggested, while practical trials demonstrate the scalability and efficacy of clustering techniques. The accuracy with feature selection (ADO) and precision, recall, f-measure, and recall were used to gauge the effectiveness of the suggested clustering technique. Future research should look at a number of these concerns. First, various algorithms can benefit from the weighted average approach. Second, consider that there is a finite tall-and-skinny transposition of the data matrix. Modern applications often include a thick and tall data matrix, meaning that there are a lot of rows and columns. More research is required to determine this algorithm's convergence rate.

References

[1] Bhadani, A.K. and Jothimani, D., 2016. Big data: challenges, opportunities, and realities. Effective big data management and opportunities for implementation, pp.1-24.

[2] Oo, M.C.M. and Thein, T., 2022. An efficient predictive analytics system for high dimensional big data. Journal of King Saud University-Computer and Information Sciences, 34(1), pp.1521-1532.

[3] Chavoshinejad, J., Seyedi, S.A., Tab, F.A. and Salahian, N., 2023. Self-supervised semi-supervised nonnegative matrix factorization for data clustering. Pattern Recognition, 137, p.109282.

[4] Liu, T. and Tao, D., 2015. On the performance of manhattan nonnegative matrix factorization. IEEE Transactions on Neural Networks and Learning Systems, 27(9), pp.1851-1863.

[5] Alonso-Betanzos, A. and Bolón-Canedo, V., 2018. Big-data analysis, cluster analysis, and machine-learning approaches. Sex-specific analysis of cardiovascular function, pp.607-626.

[6] Yang, Z., Corander, J. and Oja, E., 2016. Low-rank doubly stochastic matrix decomposition for cluster analysis. The Journal of Machine Learning Research, 17(1), pp.6454-6478.

[7] Wang, S., Lu, J., Gu, X., Du, H. and Yang, J., 2016. Semi-supervised linear discriminant analysis for dimension reduction and classification. Pattern Recognition, 57, pp.179-189.

[8] Kurita, T., 2019. Principal component analysis (PCA). Computer Vision: A Reference Guide, pp.1-4.

[9] Tharwat, A., 2021. Independent component analysis: An introduction. Applied Computing and Informatics, 17(2), pp.222-249.

[10] Nasraoui, O. and N'Cir, C.E.B., 2019. Clustering methods for big data analytics. Techniques, Toolboxes and Applications, 1, pp.91-113.

[11] Jayasri, N.P. and Aruna, R., 2022. Big data analytics in health care by data mining and classification techniques. ICT Express, 8(2), pp.250-257.

[12] Ayesha, S., Hanif, M.K. and Talib, R., 2020. Overview and comparative study of dimensionality reduction techniques for high dimensional data. Information Fusion, 59, pp.44-58.

[13] Zhang C. and S. Zhang, "Bayesian joint matrix decomposition for data integration with heterogeneous noise," IEEE Trans. Pattern Anal. Mach.Intell., pp. 1–14, 2019.

[14] Fonał, K. and Zdunek, R., 2018. Distributed nonnegative matrix factorization with HALS algorithm on apache spark. In Artificial Intelligence and Soft Computing: 17th International Conference, ICAISC 2018, Zakopane, Poland, June 3-7, 2018, Proceedings, Part II 17 (pp. 333-342). Springer International Publishing.

[15] Qin X., P. Blomstedt, E. Lepp¨aaho, P. Parviainen, S. Kaski, J. Davis, E. Fromont, D. Greene, and B. B. Bringmann Xiangju Qin, "Distributed Bayesian matrix factorization with limited communication," Mach. Learn., vol. 108, pp. 1805–1830, 2019.

[16] Lin, K.C., Zhang, K.Y., Huang, Y.H., Hung, J.C. and Yen, N., 2016. Feature selection based on an improved cat swarm optimization algorithm for big data classification. The Journal of Supercomputing, 72, pp.3210-3221.

[17] Devi, S.G. and Sabrigiriraj, M., 2018, Feature selection, online feature selection techniques for big data classification:-a review. In 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), pp. 1-9.

[18] Moslehi, F. and Haeri, A., 2020. An evolutionary computation-based approach for feature

selection. Journal of Ambient Intelligence and Humanized Computing, 11, pp.3757-3769.

[19] Zhang, N., Gupta, A., Chen, Z. and Ong, Y.S., 2021. Evolutionary machine learning with minions: A case study in feature selection. IEEE Transactions on Evolutionary Computation, 26(1), pp.130-144.

[20] Wang, D., Li, T., Deng, P., Zhang, F., Huang, W., Zhang, P. and Liu, J., 2023. A Generalized Deep Learning Clustering Algorithm Based on Non-Negative Matrix Factorization. ACM Transactions on Knowledge Discovery from Data, 17(7), pp.1-20.

[21] Zhang, C., Yang, Y., Zhou, W. and Zhang, S., 2020. Distributed Bayesian Matrix Decomposition for Big Data Mining and Clustering. IEEE Transactions on Knowledge and Data Engineering, 34(8), pp.3701-3713.

[22] Zhang, H., Li, P., Fan, W., Xue, Z. and Meng, F., 2022. Tensor Multi-Clustering Parallel Intelligent Computing Method Based on Tensor Chain Decomposition. Computational Intelligence and Neuroscience, vol.2022, no. 7396185, pp.1-12.

[23] Wang, Y., Zhang, W., Yu, Z., Gu, Z., Liu, H., Cai, Z., Wang, C. and Gao, S., 2017, Support vector machine based on low-rank tensor train decomposition for big data applications. In 2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 850-853.

[24] Duan, M., Li, K., Liao, X. and Li, K., 2017. A parallel multiclassification algorithm for big data using an extreme learning machine. IEEE transactions on neural networks and learning systems, 29(6), pp.2337-2351.

[25] Xie, T., Liu, R. and Wei, Z., 2020. Improvement of the fast clustering algorithm improved by-means in the big data. Applied Mathematics and Nonlinear Sciences, 5(1), pp.1-10.

[26] Chen, Z., Jin, S., Liu, R. and Zhang, J., 2021. A deep non-negative matrix factorization model for big data representation learning. Frontiers in Neurorobotics, 15, pp.1-9.

[27] Tang, J. and Feng, H., 2022. Robust local-coordinate non-negative matrix factorization with adaptive graph for robust clustering. Information Sciences, 610, pp.1058-1077.

[28] Lan, G., Lee, S. and Zhou, Y., 2020. Communication-efficient algorithms for decentralized and stochastic optimization. Mathematical Programming, 180(1-2), pp.237-284.

[29] Jordan M. I., J. D. Lee, and Y. Yang, "Communication-efficient distributed statistical inference," J. Am. Stat. Assoc., vol. 114, no. 526, pp. 668–681, 2019.