Experimental investigation to identify galaxy clusters using sparse matrix clustering algorithm

Alok Rai (raialok06@gmail.com)
Indian Institute of Information Technology, Prayagraj
Snigdha Sen
Indian Institute of Information Technology, Prayagraj
Pavan Chakraborty
Indian Institute of Information Technology, Prayagraj

Research Article

Keywords: 3D Sparse Matrix, HDF5, Redshift, Clustering

Posted Date: May 12th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1637295/v1

Experimental investigation to identify galaxy clusters using sparse matrix clustering algorithm

Alok Kumar Rai[1][0000−0002−3313−8474], Snigdha Sen[1,2][0000−0001−7909−5193], and Pavan Chakraborty[1][0000−0002−9260−1131]

[1] Indian Institute of Information Technology, Prayagraj, India
[2] Global Academy of Technology, Bangalore, India

Abstract. Recent advances in astronomy have shifted observational astronomy toward data-driven astronomy, with an exponential increase in data associated with celestial objects.If we can handle this massive amount of quickly changing data, we will be able to detect galaxy clusters, revealing a wealth of vital information about the evolution of the universe.In this paper, we have proposed a novel clustering technique that can handle massive amounts of rapidly changing data in real time.In this clustering technique we have used 3D sparse matrix where each cell of the 3D matrix can be used to locate/identify a neighbor against the central galactic coordinate.In our investigation, we also used HDF5 to store and organize the discovered galaxy clusters so that we didn't have to re-run our algorithm every time a new coordinate was encountered.Following preparation, the astronomical data containing the galactic coordinates RA, DEC, and radial-distance obtained from redshift is input into our developed algorithm to identify, store, and depict the galaxy clusters.

Keywords: 3D Sparse Matrix · HDF5 · Redshift · Clustering

1 Introduction

Galaxy clusters, also known as clusters of galaxies, are the biggest gravitationally bonded formations in the universe, and typically a galaxy cluster contains 2 to 50 galaxies [11]. These clusters are mostly made up of galaxies, dark energy, and hot gases. Because these clusters change slowly, they preserve information about their structure and genesis so astrophysicists are interested in discovering galaxy groups and clusters so that they can research and analyze the evolution and genesis of these types of systems and learn more about their activities. When the gases in these clusters are heated to millions of Kelvins, they emit high-energy radiation that can be studied by X-ray telescopes [15]. The X-ray generated by heated gases is one of the most dependable methods for detecting galaxy clusters, but with constantly expanding data and the need to automate the clustering process, more experimentation is being carried using machine learning techniques.

The machine learning algorithms can be broadly divided into supervised and unsupervised. In the supervised method, we train our model using train data

and then compare the results with the test data to find whether we are getting the required result or not whereas in the unsupervised method we don't have any prior information about the expected outcomes. As there is no prior information on whether a particular galaxy belongs to a cluster or not unsupervised methods are used for detecting a galaxy cluster. K-means is a common unsupervised machine learning approach that is frequently used to detect clusters. However, K-means alone is insufficient for discovering galaxy clusters since we need prior knowledge about the number of clusters and it cannot deal with noisy data or outliers. Typically, galaxy clusters have diameters ranging from one to five Mpc [16], but under K-means, every galaxy will be a member of at least one cluster, even if it is very far away from the other members of the cluster.
We have proposed a novel technique for finding galaxy clusters using a sparse matrix in this study. The HDF5 file system is used to handle and store the processed data. HDF5 file systems are gaining popularity because they can handle heterogeneous and complicated data while also providing rapid I/O and easy sharing. Because we process the data in real-time, our approach is appropriate for dealing with huge and complex datasets.

2 Related Work

Modern observatory instruments and machine learning methods have accelerated the study of celestial objects. These objects contain a vast amount of information, and their study can disclose a great deal about the genesis of the cosmos. Many academic and scientific studies have recently been conducted in these fields.Initially, groups and clusters were discovered using observatory instruments, but as more data on galaxies becomes available, machine learning and real-time techniques for detecting galaxy clusters become necessary. Here, we look various developments in this field.

Jeremy Kepner et al. [12] proposed an automated approach for finding galaxy clusters in imaging and redshift galaxy surveys that make use of galaxy positions, magnitudes, and photometric or spectroscopic redshifts if available.Michael D. Gladders et al. [10] suggested a novel approach for cluster detection based on the observation that all rich clusters appear to contain a red sequence of early-type galaxies, which acts as a direct indicator of over-density.Ball et al. [6] initially discussed the necessity for astronomical data mining and then presented different machine learning and data mining methodologies for improved data analysis.I.H.Li et al. [13] developed an algorithm by integrating the friends-of-friends algorithm [8] and the photometric redshift probability densities to find galaxy groupings in photometric redshift data sets.Z Wen et al. [20] attempted to detect galaxy clusters using photometric redshifts ranging from 0.05 to 0.6 from the Sloan Digital Sky Survey Data Release 6. (SDSS DR6). They are detecting the galaxies cluster by selecting a galaxy at a specific z and then searching for all galaxies at a radius of 0.5 MPC and with a redshift gap of z $0.04(1 + z)$. To prevent identifying a cluster several times, they analysed only one cluster candidate within a radius of 1 Mpc and a redshift gap of 0.1.G. I. Perren et al. [14] pre-

sented the Automated Star Cluster Analysis package (ASteCA), a set of tools designed to fully automate the conventional tests used to establish the basic properties of stellar clusters. P. H. Barchi et al. [7] have used both supervised and unsupervised machine learning algorithms to improve the classifications of galaxies into spiral and elliptical with morphological parameters.Selim et al. [17] introduced a novel technique in which 2D and 3D Kmeans, as well as normal distribution features, were employed to predict the most likely galaxies of Virgo clusters and the Virgo center.Sen et al. [18] have described a number of machine learning and big data tools that can be used to handle and process massive amounts of astronomical data in their review paper.Snigdha et al. [19] proposed a neural network model for interpreting redshift data of celestial objects, and two different ways to train the model were proposed: one utilizing Lipschitz-based adaptive learning rate in a single node/machine, and the other using a multinode clustered environment.

The following is a breakdown of the manuscript's structure. In section 2, we covered numerous relevant research on discovering galactic groups, and in section 3, we presented the various attributes/datasets that we employed in our experiment. The proposed methodology was presented in section 4, and the findings and description were described in section 5. Finally we conclude with prospective future enhancements.

3 Description of Dataset

Various sky surveys, such as SDSS [5], KIDS [4], and COSMOS [2], are actively working and producing massive amounts of data that are used by many researchers and academic scholars to carry out experiments and projects.For our experiment we have used dataset available with SDSS and COSMOS sky surveys.

Using the sdss casjob server [1], we downloaded our first dataset from the Sloan Digital Sky Survey (SDSS). SDSS casjob provides a user interface through which users may enter SQL queries to extract the required galaxies data.After downloading the required dataset in CSV format we have applied techniques to remove and process missing and NULL values.

We have downloaded our second dataset from the COSMOS 2015 sky survey. The requested dataset was downloaded from the COSMOS sky survey using Table Access Protocol (TAP), which is used to access generic table data. TAP is an astronomical data query language comparable to SQL that is commonly used to extract required astronomical data from tables. After getting the necessary data in Fits format, we have converted the fits file to csv using the astroquery package [9] .After downloading both datasets, we merged them to form a single dataset with a wide range of RA, DEC, and Redshift values.
Below are the steps required to convert fits to csv:

Algorithm 1 Steps required to convert fits to csv

Data: Galactic coordinates in fits format.
Result: Galactic coordinates in csv format.
Steps:
1. Install the astroquery and astropy python packages and import the required libraries such as fits,table and TapPlus.
2. Read the fits file using below command.
table.Table.read(filename,format='fits').
3. Select the required columns and store the required information in the form of csv file using below command:
tablename.write('fiename',format='csv',overwrite=True).

Below table contains the required information about each attribute used in our experiment:

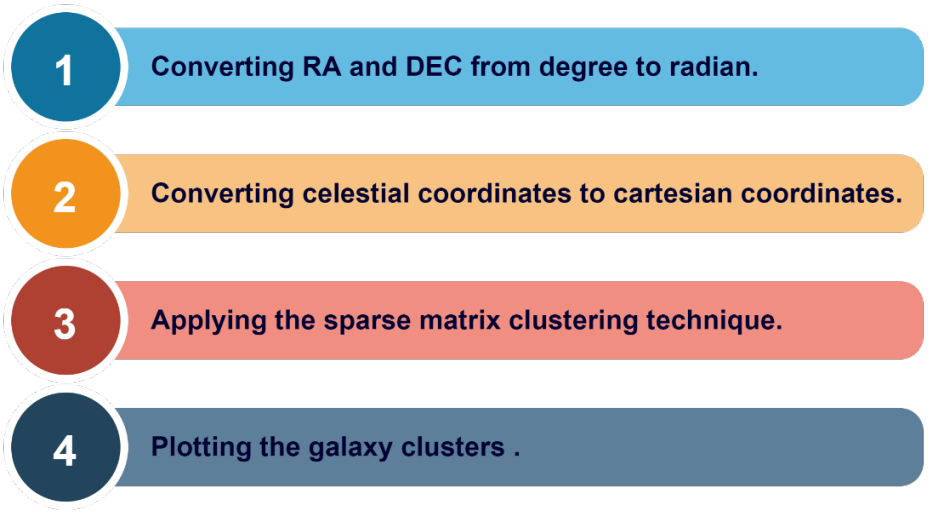
Table 1 Attributes used and their descriptions.

S.no	Attribute Name	Datatype	Description
1	RA	Degree	The Right Accession (Ra) is longitude of the sky similar to that of the earth.
2	DEC	Degree	The Declination (Ra) is latitude of the sky similar to that of the earth.
3	Redshift	Parsec	The displacement of the spectrum of an astronomical object towards longer wavelength (red) is known as Redshift.

4 The Proposed Method

With exponential increase in astronomical data we need a clustering algorithm which can handle and process such a huge amount of data without storing it.We have tried to come with an algorithm which can overcome this problem.We start by preparing and converting polar coordinates to cartesian coordinates, then determining the positional indices of each object. Following the discovery of each object's location indices, we attempted to locate its nearest neighbor using a 3D sparse matrix followed by storing and visualizing the detected galaxy clusters.For storing the identified galaxy clusters we are using HDF5 file system. The steps in our method are depicted in the diagram below:

Fig. 1 Steps required to perform sparse matrix clustering



4.1 Converting Polar coordinates to cartesian coordinates.

RA(right ascension) and DEC(declination) are the longitude and latitude of the sky.RA and DEC along with radial distance R forms the polar coordinates of the object.The polar coordinates are in degree so first we need to convert these coordinates from degree to radians.To convert these polar coordinates (RA,DEC) to radian we have used deg2rad python function present under NumPy library.The radial distance R can be derived from the redshift using the Hubble's law. The Hubble's law can be described as:

$$v = H_0 D \tag{1}$$

where:
v = velocity of the galaxy
H_0 = Hubble's Constant,its value was 67.4 km/s/Mpc in 2018
D = Distance from the earth

Also the velocity of the galaxy can also be defined as product of redshift and velocity of the light i.e

$$v = cz \tag{2}$$

where:
v = velocity of the galaxy
c = speed of light given as 300000 km/sec
z = redshift

Using equation 1 and 2 the radial distance can be derived using the below equation:

$$R = \frac{300000 \times redshift}{Hubble'sConstant} \tag{3}$$

Now we need to convert the polar coordinates of objects to cartesian coordinates.The cartesian form of any polar coordinates can be derived using below equations:

$$X = RadialDistance \times \cos(dec) \times \cos(ra) \tag{4}$$

$$Y = RadialDistance \times \cos(dec) \times \sin(ra) \tag{5}$$

$$Z = RadialDistance \times \sin(dec) \tag{6}$$

4.2 Determining positional index of each object.

Once we have the cartesian form of each object now we need to determine the positional index of each object. For determining the positional index first we will assume a minimum 3D cell (Δx,Δy,Δz) ,and then we will attempt to get the positional index of each object by dividing its cartesian coordinates by the assumed minimum 3D cell.The positional index of each object is calculated using below equations:

$$i = INT(\frac{x}{\Delta x}) \tag{7}$$

$$j = INT(\frac{y}{\Delta y}) \tag{8}$$

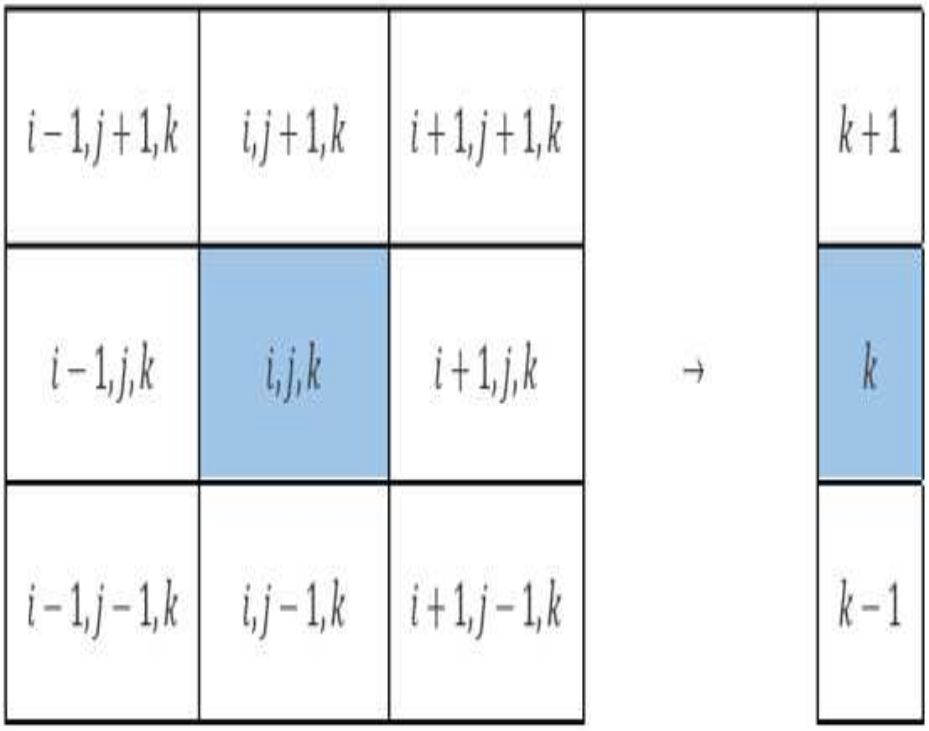
$$k = INT(\frac{z}{\Delta z}) \tag{9}$$

These positional indices will form a 3D sparse matrix of the objects. As we obtain the positional index of the next object, the 27 neighborhoods of the 3D cell , is searched from the already existing to the 3D Cells (positional indices).If the neighbourhood is established then it will become part of the matrix.

4.3 Finding nearest neighbours using 3D sparse matrix.

The major rationale for utilizing a 3D sparse matrix is to restrict the cluster's infinite directions to only 26 neighbors. Each cell in the 3D sparse matrix represents a neighbor to the central object. Initially, we start by taking the smallest coordinate as the central coordinate of the matrix from the available object coordinates and then increase the value of the next central coordinates by three. There can be a maximum of 26 neighbors of the central object and the maximum possible combination of the coordinates can be in range -1 to 1 that's why we are increasing the value of central object coordinates by 3 until we reach the maximum (i,j,k) values.We will try to find the nearest neighbors in the range – 1 to +1 of the central object, and if we find a galaxy in this range, we will store/add it to the matrix and if we do not find any such element, we will add

Fig. 2 3D Sparse Matrix



another matrix and repeat the process until we have processed all the available coordinates.The above diagram represents the 26 neighbourhood of the 3D cell:

Interpretation of the above graph: The graph above depicts a 3D sparse matrix, with each cell representing a neighbour of the central object (i,j,k). Assume we begin with the central coordinate (i,j,k) and have to find all the possible neighbour coordinates corresponding to this central object, so if there is any galaxy with coordinates ranging from (i-1,j-1,k-1) to (i+1,j+1,k+1), they will become one of the neighbours of (i,j,k), otherwise we will increment the value of the central coordinate as (i+3,j+3,k+3) and will continue this process until we reach the maximum value of the available coordinates.

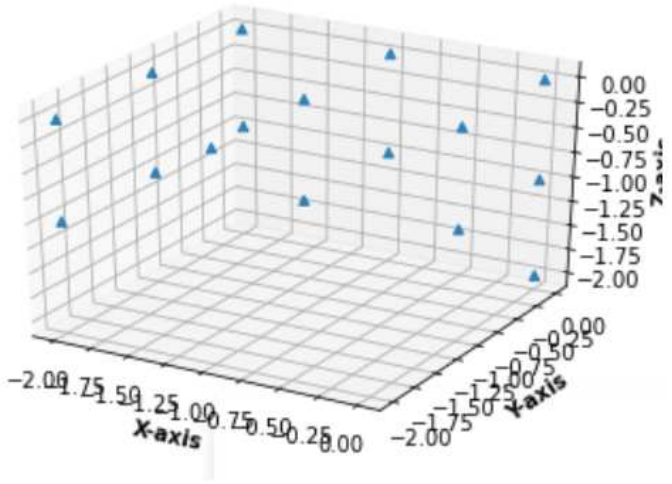
Algorithm for finding the 26 nearest neighbour using 3D sparse matrix: To generate the 26 nearest neighbour matrix, we have created a neighbour function and repeatedly we are generating the 26-neighbour matrix by calling the neighbour function until we have processed all the available coordinates.The below algorithm explains the step required for generating the 3D 26-neighbour matrix:

Algorithm 2 Finding 26 Nearest Neighbour using 3D sparse Matrix

Data: Cartesian Coordinates of galaxies.
Result: 26 Nearest Neighbour using 3D sparse Matrix
Steps:
1. Initialize the start value with minimum possible coordinates (i,j,k) and end value with maximum possible coordinates.
2. Loop until start is less than end by incrementing start value by 3 in each step.
(i) Initialize an empty queue.
(ii) Find all the neighbours which are placed in one of the matrix cell i.e in the range from (i-1,j-1,k-1) to (i+1,j+1,k+1).
(iii) Add the coordinates in the queue.
(iv) While queue is not empty repeat the steps **2.(ii)** and **2.(iii)**.
3. Store and plot the returned matrix.
4. Terminate the algorithm.

After running the above algorithm we will be able to find the nearest neighbours of the corresponding coordinates.Below diagram depicts the nearest neighbour of the coordinate (-1,-1,-1).

Fig. 3 26 Nearest Neighbour using 3D sparse matrix

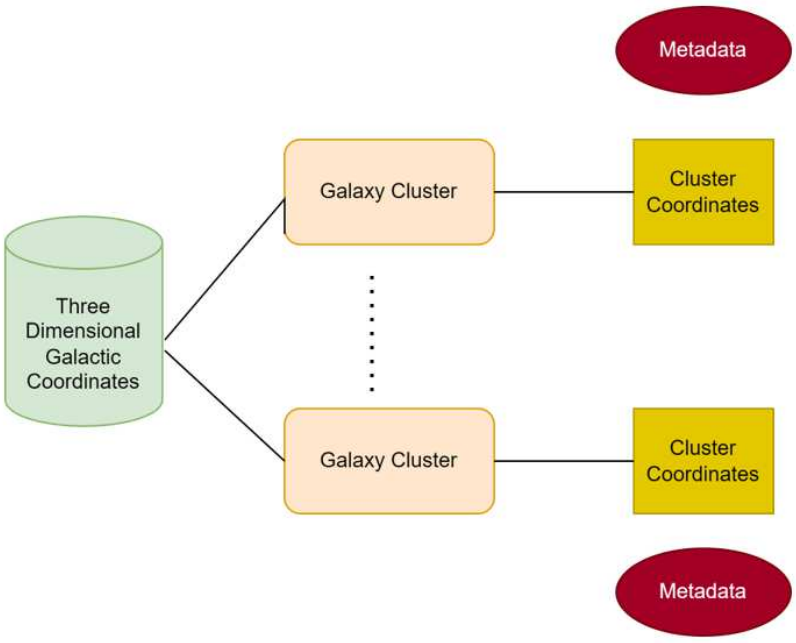


4.4 Using HDF5 for storing the galaxy clusters

With the development of new observatory instruments and telescopes, there is an exponential rise in astronomical data. Traditional storage systems are insufficient to hold such huge volumes of data, necessitating the development of a new storage system capable of handling such massive amounts of data.To overcome these

above mentioned issues we have used HDF5 file system [3] in our study.HDF5 file system organizes the data in the form of files and directories.The directories are known as "Groups" and the files are known as "Datasets" in HDF5. These file systems are becoming very popular in the field of astronomy due to its ability to handle ,process and manage huge amount of heterogeneous and complex data.For our experiment we are storing the three dimensional galactic coordinates in hdf5 file system and then applying the 26-Nearest neighbour algorithm to find the galaxy clusters. For example we are staring with the initial coordinates having the values (0,0,0) and its nearest neighbour coordinates are (1,1,1) ,(-1,1,0) and (1,0,0) . For the above coordinates first we will create a hdf5 file names as "coordinates" and then under this file system we will create a group as "Group 0" and under this group we will create a dataset which will contain all the above mentioned neighbour coordinates. Below block diagram explains how we are organizing and storing the galaxy clusters:

Fig. 4 HDF5 for organizing three dimensional coordinates.



Datasets are comparable to Numpy arrays in how they work.The detected galaxy clusters which are stored as dataset in HDF5 can be easily accessed and manipulated like numpy arrays. So, if we want to check what all coordinates are kept under a specific galaxy group, we can simply open the file that contains the

coordinates and retrieve the information we need. The command to access the discovered galaxy cluster coordinates is as follows:

```
with h5py.File('filename', 'r') as f:
    data=f['Group Number/Cluster/Coordinates']
```

5 Experimental Setup and libraries used

We used Google Colaboratory on a single machine with 16 GB RAM to conduct our experiment. Below are the python libraries used as part of our experiment:
Pandas : We have used pandas to read and remove null and missing values from our csv file.
NumPy : Numpy python library has been used to convert the polar coordinates of galaxies present in degree to radian.
h5py : The h5py package offers a Python interface to the HDF5 file system. We used the HDF5 properties to store and organize observed galaxy clusters using this package.
Matplotlib : To visualise the identified galaxy clusters we have python matplotlib ibrary.

6 Results and Discussion

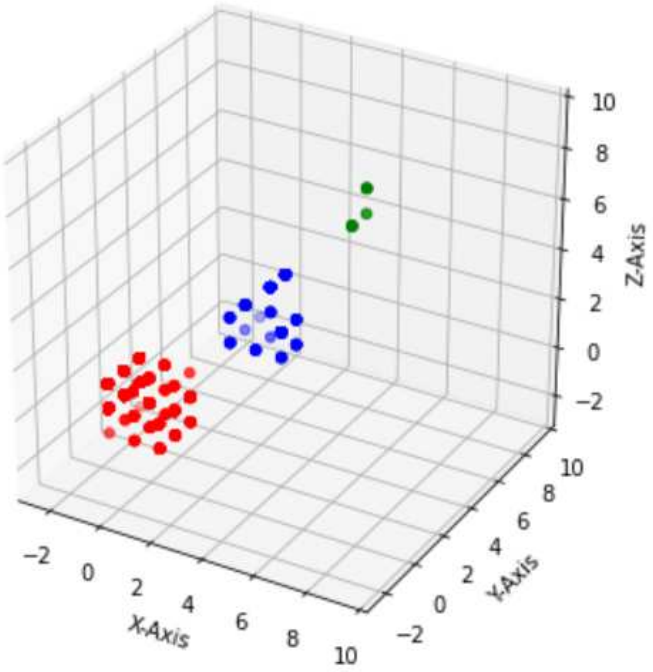
The RA, DEC, and Radial distance of galaxies computed from the Redshift are fed into a sparse matrix clustering algorithm to locate galaxy clusters. We preprocessed the data to remove missing and NULL values before applying the sparse matrix clustering algorithm to galaxies coordinates. After preprocessing the data, we used sparse matrix clustering to store and organize the processed data, as illustrated in the diagram below. First, we formed an HDF5 group to represent the group number, and inside that group, we created a sub-group if we found any clusters, otherwise, no sub-group is generated. The coordinate range in which a galaxy cluster has been identified is denoted by the sub-group that was generated. Following the discovery of the cluster, we constructed an HDF5 dataset to hold the cluster coordinates, as represented in Fig.5.

After storing the processed galaxy cluster coordinates we have tried to visualise the same using python matplotlib library.For our dataset, we can see in Fig.6 that three galaxy clusters/groups were discovered using our clustering technique.

Fig. 5 Storing and Organizing galaxy cluster coordinates using HDF5

```
Group1/Cluster/Coordinates
    [[2. 3. 4.]
     [3. 4. 4.]
     [2. 3. 3.]
     [3. 3. 4.]
     [2. 2. 2.]
     [2. 2. 3.]
     [2. 4. 2.]
     [4. 2. 2.]
     [4. 2. 3.]
Group2/Cluster/Coordinates
Group3
Group4
Group5
Group6
```

Fig. 6 Plotting Galaxy cluster identified using 3D sparse matrix technique



7 Advantages over Machine Learning Algorithms

7.1 Real-time data processing

Unlike machine learning techniques, which require storing data first to determine the optimal number of clusters before executing clustering, our suggested technique performs clustering in real-time. Whenever a new galactic coordinate is encountered, it is passed to our sparse matrix algorithm, which processes the coordinate and checks whether it is a member of any cluster without storing it.

7.2 Suitable for Big-Data

Our technique can manage a large amount of astronomical data since we only store galactic coordinates that have been detected as part of a cluster.We also use the HDF5 file system to store the processed coordinates, which is capable of managing enormous amounts of complex and heterogeneous data.

7.3 Avoids re-computation

While using machine learning clustering algorithms, whenever a new coordinate is encountered, the entire process must be recomputed/rerun, which is inefficient for astronomical data. To solve this difficulty, we're storing previously identified clusters in the HDF5 file system and then attempting to find the best appropriate cluster for newly added galactic coordinates.

8 Conclusion

In this paper, we describe an unique method for identifying galaxy clusters using a sparse matrix. Our proposed method optimizes the storage of huge data by processing the galactic coordinates RA, DEC, and radial distance determined from redshift in real-time. This strategy also aids in reducing a galaxy cluster's infinite directions to only 26. The HDF5 file system was also utilized to store and organize the obtained cartesian coordinates as well as the galaxy clusters derived from these coordinates. The use of the HDF5 file system helps to avoid processing of previously identified galaxy clusters when new galactic coordinates are fed to our sparse matrix clustering technique.Our proposed method is robust to outliers, and we do not need to calculate the most likely number of clusters before implementing our approach. We can easily determine the range containing the greatest number of galaxy clusters. Furthermore we are planning to work on finding the optimal delta values (Δx,Δy,Δz) to avoid duplicate cartesian coordinates and to store a 6 bit binary number with each galaxy object identified as part of cluster to represent direction.

Author Contribution Alok Kumar Rai and Snigdha Sen carried out the experiment and drafted the manuscript. Pavan Chakraborty reviewed the manuscript and helped with the idea of developing a real-time clustering algorithm.

Funding No fund was received to assist with the preparation of this manuscript.

Data Availability All the datasets used for this work can be received upon request to the corresponding author.

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical approval Not applicable.

References

1. casjob homepage. https://skyserver.sdss.org/casjobs/, accessed: 2022-02-01
2. cosmos homepage. https://cosmos.astro.caltech.edu/, accessed: 2022-02-01
3. hdf5 homepage. https://www.hdfgroup.org/solutions/hdf5/, accessed: 2022-02-01
4. kids homepage. http://kids.strw.leidenuniv.nl/, accessed: 2022-02-01
5. sdss homepage. https://sdss.org, accessed: 2022-02-01
6. Ball, N.M., Brunner, R.J.: Data mining and machine learning in astronomy. International Journal of Modern Physics D **19**(07), 1049–1106 (2010)
7. Barchi, P., Costa, F., Sautter, R., Moura, T., Stalder, D., Rosa, R., de Carvalho, R.: Improving galaxy morphology with machine learning. Journal of Computational Interdisciplinary Sciences **7** (01 2016)
8. Eke, V.R., Baugh, C.M., Cole, S., Frenk, C.S., Norberg, P., Peacock, J.A., Baldry, I.K., Bland-Hawthorn, J., Bridges, T., Cannon, R., et al.: Galaxy groups in the 2dfgrs: the group-finding algorithm and the 2pigg catalogue. Monthly Notices of the Royal Astronomical Society **348**(3), 866–878 (2004)
9. Ginsburg, A., Sipőcz, B., Brasseur, C., Cowperthwaite, P., Craig, M., Deil, C., Guillochon, J., Guzman, G., Liedtke, S., Lim, P., Lockhart, K., Mommert, M., Morris, B., Norman, H., Parikh, M., Persson, M., Robitaille, T., Segovia, J., Singer, L., Woillez, J.: Astroquery: An astronomical web-querying package in python. The Astronomical Journal **157**, 98 (02 2019). https://doi.org/10.3847/1538-3881/aafc33
10. Gladders, M.D., Yee, H.: A new method for galaxy cluster detection. i. the algorithm. The Astronomical Journal **120**(4), 2148 (2000)
11. Huchra, J., Geller, M.: Groups of galaxies. i-nearby groups. The Astrophysical Journal **257**, 423–437 (1982)
12. Kepner, J., Fan, X., Bahcall, N., Gunn, J., Lupton, R., Xu, G.: An automated cluster finder: the adaptive matched filter. The Astrophysical Journal **517**(1), 78 (1999)
13. Li, I., Yee, H.K.: Finding galaxy groups in photometric-redshift space: the probability friends-of-friends algorithm. The Astronomical Journal **135**(3), 809 (2008)
14. Perren, G.I., Vazquez, R.A., Piatti, A.E.: Asteca: Automated stellar cluster analysis. Astronomy & Astrophysics **576**, A6 (2015)
15. Sarazin, C.L.: X-ray emission from clusters of galaxies. Reviews of Modern Physics **58**(1), 1 (1986)

16. Saviane, I., Ivanov, V.D., Borissova, J.: Groups of Galaxies in the Nearby Universe: Proceedings of the ESO Workshop Held at Santiago de Chile, December 5-9, 2005. Springer Science & Business Media (2007)
17. Selim, I., Elkafrawy, P., Dabour, W., Eassa, M.: Virgo cluster membership based on k -means algorithm. International Journal of Astronomy and Astrophysics **10**, 1–10 (01 2020)
18. Sen, S., Agarwal, S., Chakraborty, P., Singh, K.P.: Astronomical big data processing using machine learning: A comprehensive review. Experimental Astronomy pp. 1–43 (2022)
19. Sen, S., Saha, S., Chakraborty, P., Singh, K.P.: Implementation of neural network regression model for faster redshift analysis on cloud-based spark platform. In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. pp. 591–602. Springer (2021)
20. Wen, Z., Han, J., Liu, F.: Galaxy clusters identified from the sdss dr6 and their properties. The Astrophysical Journal Supplement Series (06 2009)