



A NOVEL PERFORMANCE ENHANCED AND OUTLIER RESISTANT HYBRIDIZED GINI_HDBSCAN DEEP CLUSTERING ALGORITHM FOR BIG DATA ANALYSIS

N.Valarmathy.¹ , Dr.Krishnaveni Sakkarapani²

*Research Scholar & Assistant Professor, PG & Research Department of Computer Science, Pioneer
College of Arts & Science, Coimbatore, Tamil Nadu, India.*

valarmathykamalam@gmail.com

*Assistant Professor, Department of Computer Science, PSGR Krishnammal College for Women,
Coimbatore, Tamil Nadu, India*

sss.veni@gmail.com

Abstract

HDBSCAN is a unique and most prominent density-based clustering algorithm in which it is possible to construct hierarchy trees and extract flat clusters from that tree using specific stability measures. Predominantly most of the hierarchical clustering algorithms used nowadays have a huge number of computations in obtaining pairwise dissimilarity measures. Such limitations can be overcome using a clustering algorithm that makes use of a single linkage concept and faces many problems like it is very much prone to outliers and can produce extremely skewed or slanted dendrograms. To overcome the limitations a hierarchical clustering linkage criterion commonly known as Genie is being used which can link two clusters with a chosen inequity measure (Gini Index or Bonferroni Index) so that the size of the cluster will not go more than the assigned threshold value. The additional use of the Gini index and threshold value can result in the potential benefit of this hybrid approach is the possibility of clustering data with variable densities. This hybrid GINI_HDBSCAN algorithm is suitable for handling various applications where low minimum cluster sizes are required and where there is a need to elude a huge number of small clusters which are seen in high-density regions. In this proposed hybrid algorithm to increase the speed, parallel execution is performed and can be executed using multiple threads. The memory overhead for this proposed algorithm is small and the distance matrix need not be pre-computed to obtain the desired clustering results. The proposed algorithm is experimentally tested on the educational dataset and the obtained results show that this proposed approach is efficient for clustering huge datasets in terms of all metrics.

5756

DOI Number:10.14704/nq.2022.20.8.NQ44603

NeuroQuantology 2022; 20(8): 5756-5771

1.INTRODUCTION

Clustering algorithms used by many researchers nowadays include several domains to extract and analyze similar patterns for the assigned dataset. The method

of dividing a set of data objects into subsets is called clustering or cluster analysis. All the partitioned subset is known as a cluster, in which each object inside a cluster is similar to each other and dissimilar to objects of other



clusters. Then all the set of clusters formed using this cluster analysis is known as the clustering process. This partitioning of objects into clusters is not performed by humans, but here this process is performed by the clustering algorithm. Hence, this process of clustering is useful when there is a necessity to discover unknown groups in a dataset. The clustering technique can also be used for detecting noise or outliers, where outliers are values that are “far away” from the cluster point of all the cluster groups. This process can be applied in several fields like the detection of credit card frauds and the monitoring of criminal activities in electronic commerce. Cluster analysis has become an active topic in data mining due to the huge amounts of data being collected in the databases. Cluster analysis is a branch of statistics that mainly focuses on distance-based clustering on high dimensional data which can contain numerous attributes or sizes and variable densities. To solve this problem the most prominent and commonly used algorithm is the HDBSCAN algorithm [18]. In this proposed method applying an additional threshold value and Gini index value can produce extraordinary results. When combining genie and HDBSCAN clustering algorithm for clustering variable density data many potential benefits were obtained. Our newly proposed algorithm can be useful in different situations where there is a need to cluster data with a low minimum cluster size and avoid small clusters in highly dense areas. This proposed algorithm is executed on HDBSCAN’s cluster tree candidates directly without making any modifications to the various levels of hierarchy. HDBSCAN is one of the modified version of the Density-based SCAN algorithm (DB SCAN)[3] in which clusters are made in such a way that each partition have a higher density than the partitions made near them. That is densely partitioned areas are very well distinguished from the areas of low density. The objects which do not tally with the assumed criterion are treated to be as noise and it is discarded. Most of the problems faced in single linkage criterion are solved by using Genie one of the latest clustering

fields or dimensions. In this process of clustering, each keyword is treated as a dimension, and there are at most thousands of keywords. Most traditional algorithms can handle only low dimensional data with two or three dimensions, but clustering high dimensional data is a challenging task. Clustering algorithms depend on several factors like scalability and the capability of this algorithm to tackle clusters of arbitrary shape, different types of attributes, incremental updates, noisy data, and constraints. Most clustering algorithms differ with the partitioning levels which will group the clusters by checking whether they are mutually exclusive or not using similarity measures. Clustering methods usually differ in the levels of partitions performed.

In this big data era, the most common problem faced is clustering datasets of variable

algorithms that can produce high-quality outputs and can perform faster computations. Hence in this paper, a new algorithm is proposed combining the techniques available in HDBSCAN and Genie clustering algorithm [10]. This newly proposed GINI_HDBSCAN algorithm can overcome all the problems faced in HDBSCAN and the genie algorithm. The experimental tests were conducted on the educational dataset. The results obtained have been compared with another clustering algorithm on different metrics. This paper is designed with various sections: the literature review of both the algorithms is listed in section II and in section 3 the overview of the HDBSCAN algorithm is presented. In section 4 the overview of the Genie clustering algorithm is given and in section 5 the newly proposed hybridized HDBSCAN-genie clustering algorithm is given. In section 6 experimental setup procedures using an educational dataset are explained. In section 7 results obtained for the proposed algorithm have been compared with other algorithms and in section 8 conclusion of the proposed method has been discussed and future research directions have been mentioned.

2.LITERATURE REVIEW



About 100 research papers which has made used of DBSCAN, HDBSCAN and Genie clustering algorithms has been reviewed and analysed before proposing this hybridized

algorithm. The research papers which has been reviewed has been tabulated below for further reference.

S. No	Authors	Algorithm Used	Description
1	M. Ester, H.-P. Kriegel, J. Sander, and X. Xu [20]	DBSCAN	DBSCAN algorithm always assigns the density to the minimum number of objects which is represented using the variable minpts as radius. The radius value depends of distance threshold value epsilon.
2	Campello. et al.[2]	HDBSCAN	The FOSC frame work has been proposed to formalize the procedure for selecting clusters using local cuts which involves optimization problem. ["Framework for Optimal Selection of Clusters"]
3	Rodriguez and Laio [24]	density peaks clustering	Addition of a new third parameter to DBSCAN resulted in a new density-ratio based Clustering approach.
4	Zhu et al. [32]	density-ratio based clustering	A density peak clustering has been introduced in this paper. This algorithm uses decision graph method for selecting the cluster centers. This algorithm can also work with variable densities.
5	Ankerst et al. [19].	OPTICS	This OPTICS algorithm is mainly used in constructing ordered representation of data which permits us to search all the available clustered density regions.
6	Sander et al. [13].	automatic cluster selection	In this paper an automatic cluster selection method has been proposed which



		on method	selects the radius using a fixed parameter.
--	--	-----------	---

7	Dockhorn et al. [2]	Variable density based clustering-HDBSCAN,	HDBSCAN method is introduced to analyze variable density clusters using gradually decreasing minPts with fixed epsilon.
8	Dockhorn et al.[1]	DBSCAN-Edge quantile method	The edge quantile method has been proposed to view the complete hierarchy in the cluster form; the sub clusters which exceeds the edge length of 0.95 quantile are being cut off.[1].
9	Campello et al. [23]	AUTO-HDS	AUTO-HDS method is a hierarchical algorithm which is identical to HDBSCAN which is proposed by Campello et al. [23]
10	L. McInnes, J. Healy, and S. Astels[17]	HDBSCAN	An improved version of DBSCAN is the HDBSCAN and OPTICS has been proposed for investigating diverse areas of research dataset.[17]
11	T. Zhang, R. Ramakrishnan, M. Livny[27]	BIRCH method	BIRCH method is introduced to work with problems on real-valued vectors only.
12	F. Jiang, et al & J.B. MacQueen [7, 14, 25, 26]	k-modes, or k-medoids algorithms and fuzzy clustering schemes	This paper includes the execution and implementation of these three algorithms and fuzzy clustering schemes[19,28-30] by assigning the number of outputs in advance.
13	D. Müllner [5, 6, 25],	Fast cluster	a fast $O(n^2)$ -time and $O(n)$ -space algorithm practically discards all popular traditional hierarchical clustering



				algorithms except single linkage method
1 4	Gagolewski M., Bartoszek M., Cena A.[21]	Genie clustering algorithm		a new linkage criterion known as Genie clustering technique has been proposed to produce high quality clustering output within a limited execution time.
1 5	Yaswanth Kumar Alapati et al.[33]	Combining clustering and classification	A new method has been proposed to increase the classification accuracy by performing clustering.	can return a good clustering straight away with very little or no parameter tuning. Working procedure of HDBSCAN algorithm A series of steps are performed in the execution of the HDBSCAN algorithm Alter the space based on the density/sparsity. The MST is constructed using the weighted distance graph method.[MST-minimum spanning tree] Cluster hierarchy is built for the closely connected components. The cluster hierarchy is condensed using the minimum cluster size value. Then stable clusters are extracted using the condensed cluster hierarchy tree.

3. OVERVIEW OF HDBSCAN

Campello, Moulavi, and Sander proposed a novel clustering algorithm known as HDBSCAN[18]. HDBSCAN abbreviated as Hierarchical Density-Based Spatial Clustering of Applications with Noise is one of the latest and most prominent algorithms which can deal with varying density clusters. Mainly this algorithm is used for big data analysis as it is very fast and can return meaningful clusters. This algorithm works similarly to DBSCAN[3] but the epsilon values used can vary and gives the best result when there is a stable epsilon value. Usually, traditional DBSCAN algorithm a cut level epsilon value is used for the dendrograms, but in this approach, the dendrograms are condensed by viewing splits which results in a less number of points that is split as falling out of a cluster so this results in a smaller tree with a fewer cluster that loses points. From this tree, stable clusters can be chosen. This process of choosing the clusters from smaller cluster tree heights allows trees to be cut at varying heights. This algorithm is very robust to parameter selection and the main parameter used is minimum cluster size which can be easily selected from the tree. Therefore it can be specified that HDBSCAN

Pseudocode of HDBSCAN algorithm

can return a good clustering straight away with very little or no parameter tuning.

Working procedure of HDBSCAN algorithm

A series of steps are performed in the execution of the HDBSCAN algorithm

Alter the space based on the density/sparsity.

The MST is constructed using the weighted distance graph method.[MST-minimum spanning tree]

Cluster hierarchy is built for the closely connected components.

The cluster hierarchy is condensed using the minimum cluster size value.

Then stable clusters are extracted using the condensed cluster hierarchy tree.

Advantages of HDBSCAN

This algorithm can operate having varying density clusters

The epsilon parameter can be eliminated as and when it is not necessary to choose a cut off dendrograms

The epsilon parameter can be replaced by the min_cluster_size parameter which can be used to determine the falling out of a cluster point or splitting to form two new clusters.

The eom- excess of a mass parameter used in this method helps to return clusters with the best stability over epsilon.

Unlike DBSCAN distance threshold need not be chosen at the beginning where there are benefits when used as the epsilon threshold.



```

    """import the necessary library files need of for execution and plotting"""
    import numpy as np; import matplotlib.pyplot as plt; import seaborn as sns
    import sklearn.datasets as data

    %matplotlib inline
    sns.set_context('poster'); sns.set_style('white'); sns.set_color_codes()
    plot_kwds = {'alpha': 0.5, 's': 80, 'linewidth': 0}

    """import the sample dataset needed for execution and plotting"""
    edotrain_ = data.make_edotrain(n_samples=50, noise=0.05)
    edotest_ = data.make_edotest(n_samples=50, centers=[(-0.75, 2.25), (1.0, 2.0)],
    cluster_std=0.25)
    test_data = np.vstack([edotrain, edotest])
    plt.scatter(test_data.T[0], test_data.T[1], color='b', **plot_kwds)
    """load HDBSCAN library for execution and plotting"""
    import hdbscan
    clusterer = hdbscan.HDBSCAN(min_cluster_size=5, gen_min_span_tree=True)
    clusterer.fit(test_data)
    HDBSCAN(alg_order='best', alpha=1.0, approx_min_span_tree=True,
    gen_min_span_tree=True, leaf_size=40, memory=Memory(cachedir=None),
    min_size='wclust', min_cluster_size=5, min_samples=None, p=None)
    """Build the minimum spanning tree"""
    clusterer.minimum_spanning_tree_plot(edge_cmap='viridis', edge_alpha=0.6,
    node_size=80, edge_linewidth=2)
    """Build the cluster hierarchy"""
    clusterer.single_linkage_tree_plot(cmap='viridis', colorbar=True)
    """Condense the cluster tree"""
    clusterer.condensed_tree_plot()
    """Extract the clusters"""
    clusterer.condensed_tree_plot(select_clusters=True, selection_palette=sns.color_palette())
    """plotting of clusters"""
    palette = sns.color_palette()
    cluster_colors = [palette[cluster] for cluster in clusterer.labels_]
    if col >= 0 else (0.5, 0.5, 0.5) for col, sz in zip(clusterer.labels_, clusterer.probabilities_)
    plt.scatter(test_data.T[0], test_data.T[1], c=cluster_colors, **plot_kwds)
    output

```



From the above results, it can be stated that HDBSCAN is fast in execution and gives more accurate clustering results than the traditional K-Means algorithm. Hence HDBSCAN algorithm is chosen for the hybridization of this proposed method.

4. OVERVIEW OF GENIE CLUSTERING ALGORITHM

A more faster and dominant version of the Hierarchical clustering algorithm is Genie outlier resistant clustering algorithm designed and published in 2016 by Gagolewski, Bartoszek, and Cena [21]. The disadvantage of classical linkage criteria and single linkage clustering algorithm is overcome in this genie clustering algorithm. This algorithm can be executed on multiple threads parallel to speed up the process. This algorithm can link two clusters of the same economic inequality measure considered as the Gini index or Bonferroni Index of the variable cluster sizes which does not increase above the given threshold. This algorithm executes by assigning every cluster point as a member of its cluster and then keeps on merging the closest cluster pair's one after the other. The smallest point group is matched with its

nearest neighbor and this procedure when followed neglects the formation of highly imbalanced clusters [21].

Advantages of Genie clustering algorithm

1. The idea used behind this algorithm is very simple and easy to handle.
2. This algorithm is very much prone to outliers and can create extremely skewed dendrograms.
3. This algorithm can link two clusters of chosen Gini index and cluster size is also maintained.
4. It outperforms K-Means, BIRCH, ward or average linkage methods in terms of clustering quality.
5. The execution speed is drastically increased and this algorithm may run on multiple threads at a same time.
6. The memory overhead is small and so it need not pre-compute distance measure as performed for HDBSCAN.
7. It performs outlier detection using mutual reachability distance concept.
8. This algorithm is best suited for clustering larger datasets (BIG DATA).



The main simplicity feature of the single linkage algorithm is preserved in this Genie algorithm by making use of Gini-index value threshold values as $g \in (0, 1, 2, 3)$. Modifying these Gini threshold values can avoid a drastic increase in the chosen inequity measure and can force the merging of small clusters with some other clusters. The normalized Gini-

$$G(x) = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^n |x_i - x_j|}{(n-1) \sum_{i=1}^n x_i} \text{----- eqn(1)}$$

5. PROPOSED HYBRIDIZED GINI_HDBSCAN DEEP CLUSTERING ALGORITHM

OBJECTIVE OF PROPOSED METHOD

☐ The proposed method is designed to prove that the DNN classifier gives the best results when using clustered data

☐ To increase classification accuracy the data set to be used for classification when clustered gives best results than the usual unclustered dataset.

☐ For High-Dimensional datasets, it is necessary to apply Feature Selection Algorithm first for reducing the dimensions and then apply the clustering algorithm to get an exact number of clusters.

PROPOSED METHODOLOGY

The proposed methodology can rapidly raise the accuracy value of any classifier by clustering the dataset before classifying it. [33] Commonly whenever high-Dimensional data the mining process will be slow and can also rapidly decrease the rate of accuracy. The ratio of several correct outcome instances to the total number of tests performed during classification is treated to be the accuracy value. This proposed framework includes different phases like Feature selection, Clustering, and Classification

1) Feature Selection

The procedure adopted in finding out a subset of the most relevant features from the entire set of features can produce more accurate results in classification. This procedure is known as feature selection. Features provide information about the dataset. Each sample is represented using many features in this high-dimensional data set.

The data sets have to be pruned for classification as they may contain irrelevant or

index value is obtained using the equation given below [eqn.1]. The results obtained show that high-quality clusters are obtained when the dendrograms value is at 0.4 height with 23 clusters and the Gini index = 0.76. The value up to the maximum of 0.85 is obtained for 10 clusters.

redundant features. The feature selection process is performed to select the set of most relevant features which can increase the learner's ability in pattern classification. Principal Component Analysis (PCA) dimensionality reduction algorithm is used as a feature selection algorithm in this proposed framework.

2) Clustering

Clustering algorithm is applied on the reduced dataset after performing dimensionality reduction. Then cluster id is added to each group of clusters in that dataset. The clustering algorithm used in this proposed framework is Hybridized GINI_HDBSCAN algorithm.

3) Classification

After clustering the data set using our proposed technique deep neural network classification algorithm is used to classify the student dataset. The flow of execution adopted in this proposed algorithm is clearly shown in figure. 1

WORKING PROCEDURE

The proposed algorithm works on the concept of MST which uses a pairwise distance graph for the given point set. The edges of the MST are arranged in increasing order of their weights and have been used for execution, this arrangement avoids the formation of clusters of highly imbalanced datasets. Whenever the Gini index value exceeds the threshold value a forced merge operation is performed to form a point group of the smallest size.

The clustering can now be computed concerning the mutual reachability distance usually the Euclidean metric which is used in the HDBSCAN algorithm.



- If the value of $M > 1$, then mutual reachability distance $m(i,j)$ with smoothing factor M is used instead of the chosen "raw" distance $d(i,j)$.
- It holds $m(i,j) = \max(d(i,k), c(i), c(j)) \setminus$, where $d(i,k)$ with k being the $(M-1)$ -th nearest neighbor of i . This makes "noise" and "boundary" points being "pulled away" from each other.

The proposed method can be evaluated by changing the epsilon values and assigning

the minimum cluster size as a single input parameter. The 'eom' (Excess of Mass) cluster selection function gives the clusters with the best stability over epsilon as its results. Let F be a fixed inequity measure or the Gini-index and $g \in (0, 1, 2, 3]$ be some threshold. The step-by-step execution of our proposed methodology is clearly explained in figure 2.

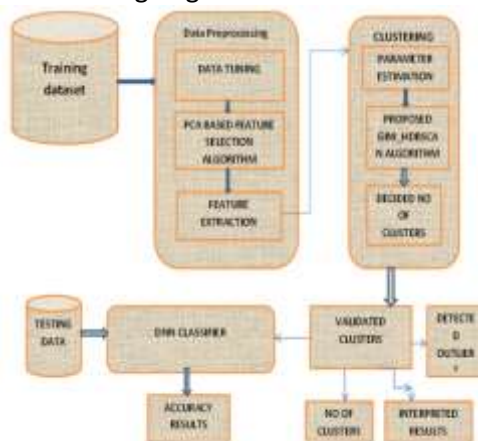


Figure 1. Block diagram of proposed GINI_HDBSCAN method.

5763

EXPERIMENTAL SETUP

DATASET

A new student dataset has been designed by combining various attributes from distinct

datasets. The newly formed dataset has 33 attributes and 2959 instances that were suitable for evaluating the proposed algorithm.

PARAMETER SETTING

GINI_HDBSCAN Parameters

```
Ghclust(d = NULL, objects = NULL, thresholdGini = 0.3, useVpTree = FALSE, ...)
```

d - an object of class *dist*, *NULL*, or a single string,
objects - *NULL*, numeric matrix, a list, or a character vector
thresholdGini - single numeric value in $[0,1]$, threshold for the Gini index, 1 gives the standard single linkage algorithm
useVpTree - single logical value, is used to decide whether vantage-point tree can be used to speed up nearest neighbour searching in low-dimensional spaces
 ... internal parameters used to tune up the algorithm

DNN Parameters

```
Number of hidden layers = (number of attributes + number of classes) / 2  

Number of neurons = 100/200/300  

Number of epoch = 500  

Learning rate = 0.3  

Reset = true
```



Figure 2. Algorithm of our proposed GINI_HDBSCAN approach

```

ALGORITHM OF OUR PROPOSED GINI_HDBSCAN ALGORITHM
STEP 1.1 Import necessary library files and define functions for drawing plots
STEP 1.2 Import the necessary dataset needed for execution and plotting
STEP 3. EXECUTE HDBSCAN library function using necessary arguments and build its parameters
STEP 4. BUILD the minimum spanning tree
STEP 5. BUILD the cluster hierarchy using distance based membership
STEP 5.1 define the structure using cluster it and condensed tree
STEP 5.2 collect the cluster elements of the tree excluding the singleton points
STEP 5.3 get the leaf cluster nodes under the cluster to be considered.
STEP 5.4 collect up the last remaining points of each leaf cluster using labels and max_label values calculated.
STEP 5.5 using the membership and condensed tree calculate the minimum distance to the neighbor and return the distance value
STEP 5.6 using the distance value and the softmax value find out the cluster points using the criteria set up
STEP 5.7 identify the saturation points by using the membership function defined as membership_vector = dist_membership_vector(x, min_label, data), color = np.argmax(membership_vector), saturation = membership_vector[color]
STEP 5.8 return the obtained saturation points using distance membership function.
STEP 6. OUTLIER detection using membership function
STEP 6.1 define min_label val, points in clusters, merge_height and per cluster score using the predefined functions with suitable parameters
STEP 6.2 identify whether all the identified points cluster points are within the cluster height already defined.
STEP 6.3 create the outlier membership function using the parameters like min_label values, points in cluster, merge height and per cluster score and soft max value.
STEP 6.3.1 based on the soft max value the resulting cluster points are obtained.
STEP 6.3.2 assign the cluster id, use tree and find out all possible clusters and compare it with the label value
STEP 6.3.3 now using the outlier membership vector function the outliers can be identified and plotted separately with different color.
STEP 7. Combine this two membership functions and return the results as the product of direct and indirect.
STEP 8 identify the saturation points using this approach also and plot the points.
STEP 9. APPLY a conditional probability to find out the nearest cluster points and append it to the parent clusters
STEP 10. REPEAT step 8 to 9 until all points in the clusters have been identified and grouped correctly.
STEP 11 USING this cluster groups assign labels to each group
STEP 12. APPLY DNN classification by using the predefined labelled dataset
STEP 13. DISPLAY the classification results and clustered results separately.
    
```

6

. EVALUATION MEASURES

The performance of the proposed clustering algorithms is evaluated by estimating number of clusters, estimating number of noise points detected, Adjusted Rand Index, Silhouette Coefficient, and execution time. Each experiment was iterated about 30 times to get an unbiased result. The calculation of the overall performance of this algorithm is obtained by averaging the results obtained in the 30 individual runs.

The performance of classification is evaluated in two aspects as

- Classification performed without clustering (Normal classification)
- Clustering prior to classification (Deep clustering)

The performance of the DNN classifier is evaluated using accuracy, precision and recall.

7. RESULTS AND DISCUSSION

Clustering result obtained for proposed GINI_HDBSCAN Algorithm

The performance of the proposed GINI_HDBSCAN clustering algorithm can be measured in terms of nine different metrics as shown in table 3. The gap statistics chart given below shows the intra-cluster variation between observed data and reference data within a uniform random distribution. Figure 2 shows the gap statistics values obtained for the proposed clustering method. The optimal number of clusters obtained for different values of k is shown in figure 3.



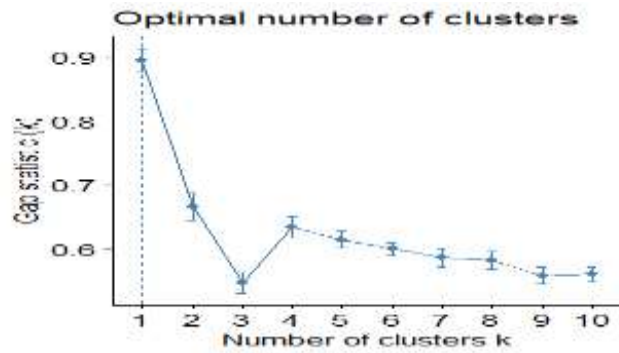


Figure 2: Gap Statistics Chart

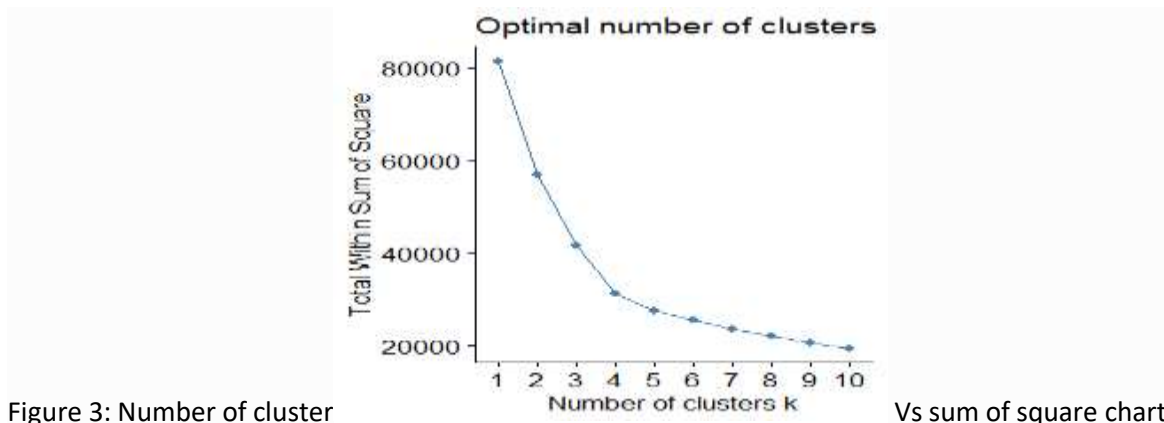


Figure 3: Number of cluster

Vs sum of square chart

5765

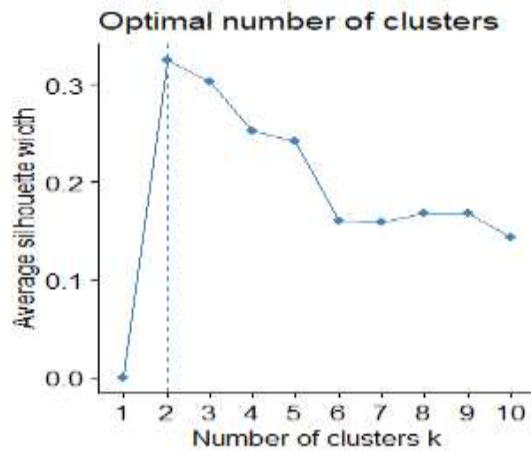


Figure 4: number of cluster Vs Silhouette width chart



Gini Hierarchical DBSCAN Cluster

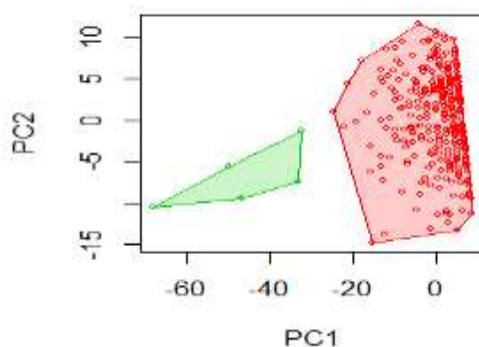


Figure 5: number of cluster chart

Clustering Algorithms used	Execution time(sec)	Estimated no of clusters formed	Estimated number of noise points detected	Silhouette measure
K means	0.832	5	10	0.785
DBSCAN	0.431	3	18	0.6681
HDBSCAN	0.4423	3	20	0.7809
GENIE	0.2312	2	22	0.8912
GINI_H DBSCAN	0.12	2	25	0.952

5766

Table 2: performance comparison of proposed method with other clustering algorithms

The average silhouette width obtained for different values of k is shown in figure 4 and the number of Clusters formed is shown in figure 5. Figure two clearly shows that only 2 clusters are formed and all other noise points have been removed. Figure 6 clearly shows the execution time taken by different clustering algorithms. The time taken by the proposed algorithm is very less than compared with other clustering algorithms. Figure 7 shows the

number of clusters formed by different clustering algorithms and this chart clearly shows that number of clusters formed is less when compared with the HDBSCAN algorithm. Figure 10 shows an accuracy-based comparison performed using different clustering algorithms. The charts clearly show that the proposed GINI_HDBSCAN algorithm gives the best clustering results than other clustering algorithms.





Figure 6: Execution time comparison chart

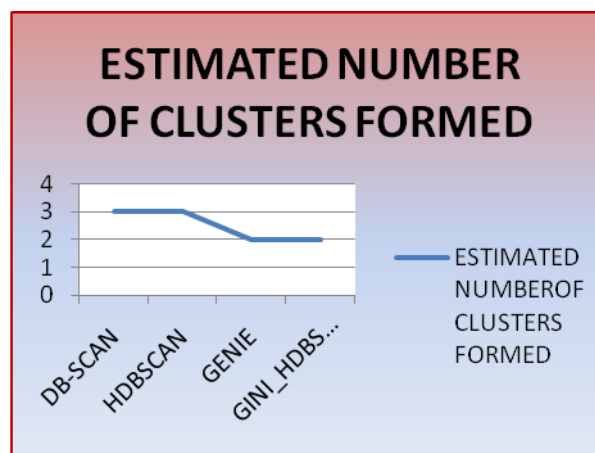


Figure 7: No. of clusters formed comparison chart

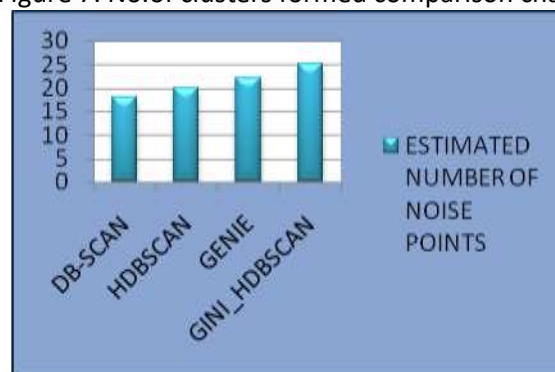
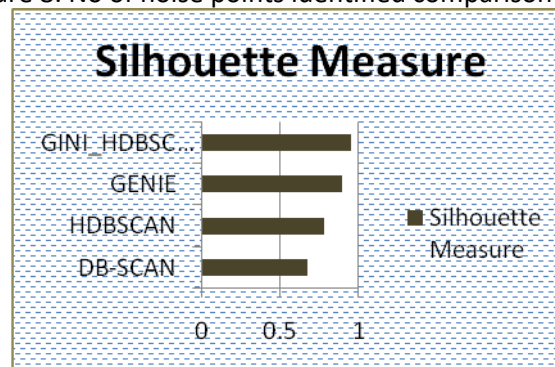


Figure 8: No of noise points identified comparison chart



HDBSCAN OUTPUT	GINI_HDBSCAN OUTPUT
Estimated number of clusters: 3	Estimated number of clusters: 3
Estimated number of noise points: 18	Estimated number of noise points: 25
Homogeneity: 0.901	Homogeneity: 0.971
Completeness: 0.821	Completeness: 0.897
V-measure: 0.891	V-measure: 0.945
Adjusted Rand Index: 0.878	Adjusted Rand Index: 0.9234
Adjusted Mutual Information: 0.901	Adjusted Mutual Information: 0.916
Silhouette Coefficient: 0.681	Silhouette Coefficient: 0.7809
Total running time of this algorithm: (0 minutes 0.451 seconds)	Total running time of the script: (0 minutes 0.12 seconds)
Estimated memory usage: 4 MB	Estimated memory usage: 1.5 MB

Figure 9: silhouette measure comparison chart

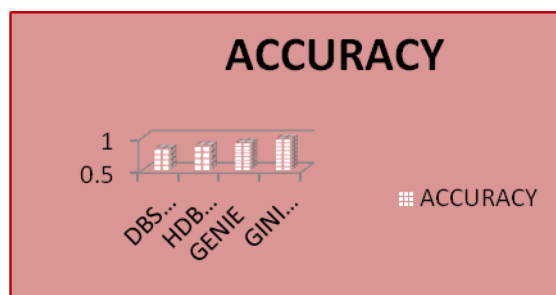


Figure 10: accuracy based comparison

The number of noise points identified by different clustering algorithms is clearly shown in figure 8. This chart shows that our proposed algorithm is a powerful outlier resistant algorithm as it has detected and removed many noise points than the DBSCAN algorithm. The proposed algorithm has also been evaluated in terms of other metrics like homogeneity, completeness, v-measure, adjusted rand index, adjusted mutual information, and silhouette coefficient. The proposed algorithm is also compared with the HDBSCAN algorithm in terms of all the above-mentioned metrics. The results obtained shown in table 3 clearly show that the proposed algorithm works best in terms of all metrics.

Table 3. Results metrics comparison

The obtained results clearly show that the proposed GINI_HDBSCAN algorithm outperforms all other algorithms in terms of all metrics. Hence it can be proved that the proposed algorithm is best suited for clustering huge datasets with very less execution time.

Impact of Classification without clustering

The performance of the clustering algorithm can also be evaluated using classifiers, as clustering when performed before classification gives best results than normal classification. Hence in this framework, the evaluation has been performed without clustering and with clustering. Table 4 clearly shows the results obtained for different classifiers like naïve Bayes, SVM, and DNN.

Classifiers used	Accuracy	Precision	Recall
Naïve Bayes	82.33	0.7862	0.7905
SVM	87.34	0.8124	0.8906
DNN	91.02	0.8954	0.9023

Table 4: Classification results obtained for different classifiers

Impact of applying clustering prior to Classification

The performance of DNN classifier and other classifiers like SVM and Naïve bayes has been obtained after performing clustering

with DBSCAN, HDBSCAN, Genie and GINI_HDBSCAN algorithm. The accuracy results obtained is tabulated in table 5. The results obtained clearly shows that performance of DNN classifier has been



increased when clustering is performed prior to classification.

The accuracy of the DNN classification algorithm after performing clustering using our proposed GINI_HDBSCAN algorithm is 98.6% which is higher than any other classification algorithms. Hence our proposed algorithm is best suited for clustering huge

datasets and can also give higher accuracy when executed prior to classification. The chart shown in figure 10 clearly shows the accuracy values obtained for different classifiers after performing clustering using DBSCAN, HDBSCAN, Genie and proposed GINI_HDBSCAN algorithm.

Classifiers Used	Accuracy			
	DBSCAN	HDBSCAN	GENIE	GINI_HDBSCAN
Naïve Bayes	68.7	76.43	78.79	90.205
SVM	71.4	86.21	87.12	94.65
DNN	83.12	88.12	78.79	98.652

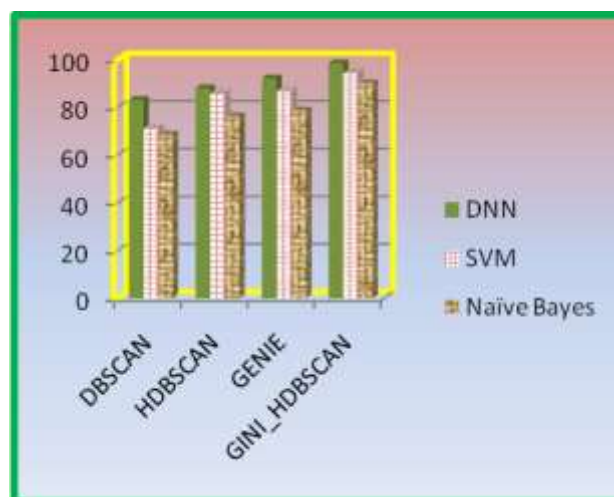


Figure 11: Accuracy based comparison of different classifiers

Hence the results obtained show that GINI_HDBSCAN is the best algorithm for clustering. GINI_HDBSCAN. The evaluation results have proven that this proposed algorithm can identify the outliers and has given a very less number of clusters. In terms of all metrics, our proposed algorithm has shown significant results. In this proposed framework deep clustering has been performed the results obtained have been judged in terms of two impacts and the evaluation results have proven that classification results are good when performed after clustering the datasets.

8.CONCLUSION AND FUTURE ENHANCEMENT

Anovel hybrid clustering algorithm has been designed to rapidly increase the performance and rate of accuracy. Theevaluation results obtained in terms of all metric shows that the proposedalgorithm is best suited for clustering high dimensional big dataset. Theproposed clustering algorithm has identified the outliers and has increased theaccuracy of results by offering a limited number of clusters. This proposed

algorithmwhen used before classification will give the best classification results. Theexperimental results obtained to state that classification when doneusing clustered data gives better classification results. To conclude, it canbe stated that the Accuracy of a classifier can be improved by applyingGINI_HDBSCAN clustering before using a deep neural network. In the future, this proposed framework will be tested on differentdatasets of different sizes to prove that this proposed methodology is bestsuited for clustering huge datasets.

REFERENCE

1. A. Dockhorn, C. Braune, and R. Kruse, "An alternating optimization approach based on hierarchical adaptations of DBSCAN," in 2015 IEEE Symposium Series on Computational Intelligence, Dec 2015, pp. 749–755.
2. A. Dockhorn, C. Braune, and R. Kruse, "Variable density based clustering," 2016 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8, 2016.
3. Campello, Ricardo J. G. B. and Moulavi, Davoud and Sander, Joerg, "Density-based



clustering based on hierarchical density estimates,"in *Advances in Knowledge Discovery and Data Mining*. Berlin,Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–172.

4. Claudia Malzer¹ and Marcus Baum²:" A Hybrid Approach To Hierarchical Density-based Cluster Selection" arXiv:1911.02282v4 [cs.DB] 21 Jan 2021

5. D. Müllner, fastcluster: Fast hierarchical, agglomerative clustering routines for R and Python, *Journal of Statistical Software* 53 (2013) 1–18.

6. D. Müllner, Modern hierarchical, agglomerative clustering algorithms, ArXiv:1109.2378 [stat.ML] (2011).

7. F. Jiang, G. Liu, J. Du, Y. Sui, Initialization of k-modes clustering using outlier detection techniques, *Information Sciences* 332 (2016) 167–183.

8. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

9. G. Gupta, A. Liu, and J. Ghosh, "Automated hierarchical density shaving: A robust automated clustering and visualization framework for large biological data sets," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 7, no. 2, pp. 223–237, April 2010.

10. Gagolewski M., Bartoszek M., Cena A., Genie: A new, fast, and outlier-resistant hierarchical clustering algorithm, *Information Sciences* 363, 2016, pp. 8–23, doi:10.1016/j.ins.2016.05.003. 22

11. H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nusenes: A multimodal dataset for autonomous driving," arXiv preprint arXiv:1903.11027, 2019.

12. H.P. Kriegel, P. Krieger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 231–240, 2011.

13. J. Sander, X. Qin, Z. Lu, N. Niu, and A. Kovarsky, "Automatic extraction of clusters from hierarchical clustering representations," in *Proceedings of the 7th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, ser. PAKDD '03. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 75–87.

14. J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob.*, volume 1, University of California Press, Berkeley, 1967, pp. 281–297.

15. J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Springer, 1981.

16. K. Chaudhuri, S. Dasgupta, S. Kpotufe, and U. von Luxburg, "Consistent procedures for cluster tree estimation and pruning," *IEEE Transactions on Information Theory*, vol. 60, no. 12, pp. 7900–7912, 2014.

17. L. McInnes and J. Healy, "Accelerated hierarchical density based clustering," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, Nov 2017, pp. 33–42.

18. L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," *The Journal of Open Source Software*, vol. 2, no. 11, mar 2017.

19. M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," in *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '99. New York, NY, USA: Association for Computing Machinery, 1999, p. 49–60.

20. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, pp. 226–231.

21. Marek Gagolewski^{a,b}, Maciej Bartoszek^{b,c}, Anna Cena^{a,c} Genie: A new, fast, and outlier-resistant hierarchical clustering algorithm: Preprint submitted to *Information Sciences* October 20, 2020



22. N. Avermann and J. Schlöter, "Determinants of customer satisfaction with a true door-to-door DRT service in rural Germany," in *Research in Transportation Business & Management*, vol. 32, 2019, paper 100420.

23. R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander, "A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies," *Data Mining and Knowledge Discovery*, vol. 27, no. 3, pp. 344–371, Nov 2013.

24. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

25. R. Xu, D.C. Wunsch II, *Clustering*, Wiley-IEEE Press, 2009.

26. S. Zahra, M.A. Ghazanfar, A. Khalid, M.A. Azam, U. Naeem, A. Prugel-Bennett, Novel centroid selection approaches for kmeans-clustering based recommender systems, *Information Sciences* 320 (2015) 156–189.

27. T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: An efficient data clustering method for very large databases, in: *Proc. ACM SIGMOD'96 Intl. Conf. Management of Data*, ACM, 1996, pp. 103–114.

28. W. Pedrycz, A. Bargiela, Granular clustering: A granular signature of data, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 32 (2002) 212–224.

29. W. Pedrycz, Conditional fuzzy c-means, *Pattern Recognition Letters* 17 (1996) 625–631.

30. W. Pedrycz, J. Waletzky, Fuzzy clustering with partial supervision, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 27 (1997) 787–795.

31. W. Stuetzle, "Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample," *Journal of Classification*, vol. 20, no. 1, pp. 025–047, May 2003.

32. Y. Zhu, K. M. Ting, and M. J. Carman, "Density-ratio based clustering for discovering clusters with varying densities," *Pattern Recognition*, vol. 60, pp. 983 – 997, 2016.

33. Yaswanth Kumar Alapati et al. / Combining Clustering with Classification:A Technique to Improve Classification Accuracy

International Journal of Computer Science Engineering (IJCSE) Vol. 5 No.06 Nov 2016 56

5771

