Date of publication DEC-31, 2023, date of current version OCT-02, 2023.
www.computingonline.net / computing@computingonline.net

*Print ISSN 1727-6209*
*Online ISSN 2312-5381*
*DOI 10.47839/ijc.22.4.3350*

# A Comparative Analysis of Data Stream Clustering Algorithms

**TAJUDEEN AKANBI AKINOSHO[1], ELIAS TABANE[2], ZENGHUI WANG[3]**

[1]Department of Computer Science, University of South Africa, South Africa (tajuakins1@gmail.com)
[2]Department of Computer Science, University of South Africa, South Africa (tabane@unisa.ac.za)
[3]Department of Electrical Engineering, University of South Africa, South Africa (wangz@unisa.ac.za)

Corresponding author: Tajudeen A Akinosho (e-mail: tajuakins1@gmail.com).

This work is supported in part by University of South Africa M&D Bursary and is partially supported by the South African National Research Foundation (Grant Nos. 132797 and 137951), South African National Research Foundation incentive grant (No. 114911), and South African Eskom Tertiary Education Support Programme.

**ABSTRACT** This study compares the performance of stream clustering algorithms (DenStream, CluStream, ClusTree) on Massive Online Analysis (MOA) using synthetic and real-world datasets. The algorithms are compared in the presence on noise level [0%, 10%, 30%] on the synthetic data. DenStream epsilon parameter was tune to 0.01 and 0.03 to improve its performance. We use the performance evaluation metrics CMM, F1-P, F1-R, Purity, Silhouette Coefficient, and Rand statistic. On synthetic data, our results show that ClusTree outperformed CluStream and DenStream on the almost all the metrics except in Purity and Silhouette were DenStream performs better at noise levels (10% and 30%). ClusTree outperform CluStream and DenStream on Forest Cover type dataset on metrics CMM, F1-P, F1-R, Silhouette Coefficient, and Rand statistic with 90%, 74%, 77% and 89% respectively. However, the tune DenStream epsilon parameter shows some improvements. On electricity data, DenStream outperform CluStream and ClusTree at epsilon parameter (0.03 and 0.05) on metrics F1-P, F1-R, and Purity. The investigation of DenStream epsilon parameter (0.03 and 0.05) on RandomBRF Generator with noise level [0%, 10%, 30%] shows that DenStream with epsilon 0.03 outperform other parameter adjustment.

**KEYWORDS** Data stream clustering; DenStream; CluStream; ClusTree; MOA; Python.

## I. INTRODUCTION

DATA has been described as the new oil, to underscore its economic importance. However, to extract maximum benefits from the constant stream of generated data, existing mining and analytical tools must adapt to the new ones developed from scratch. A vital tool in the investigation of large-scale data and data stream is Massive Online Analysis (MOA) [1, 2]. There has been great work in extending MOA into R and Python; these include RMOA [3], stream [4], streamMOA [5], scikit-multiflow [6], and River [7].

Data stream environment is different from traditional data mining settings [2, 8]. This study investigated and compared the performance of data clustering techniques DenStream [16], CluStream [17], and ClusTree [18] in MOA using six performance evaluation metrics (CMM, F1-P, F1-R, Purity, Silhouette Coefficient, and Rand statistic). We further investigate DenStream performance on epsilon parameter adjustment and RandomBRFGenerator noise levels. Some studies such as [28] have tried optimizing the distance threshold $\epsilon$ in the range [0,1] on RandomRBFGenerator while [16, 29] carried out work on the fading or decaying factor $\lambda$, but to the best of our knowledge we have not seen the epsilon combined with different noise levels.

The remainder of this paper is organized into four sessions. In Session II, we present the related works and major approaches of data stream clustering. In Session III, we present the methodology for the comparative analysis. In Section IV, we describe the experiment setup, the data set used, and output of the experiment. Finally, in Section V, we present future research and conclude this paper.

## II. RELATED WORK

### A. DATA CLUSTERING

Clustering is the process of grouping data objects into clusters based on their similarity or dissimilarity using certain characteristics or criteria. Clustering is an unsupervised learning problem [9]. Data Clustering is relevant in identifying structure when information about data is unavailable. Clustering evaluation measures is divided into external/extrinsic and internal/intrinsic measures [15, 26]. Data Clustering has a wide range of applications like in Breast cancer tumors diagnosis using Fuzzy c-means clustering [19],

in image segmentation using sine cosine algorithm [20], and multidimensional sequence clustering [21].

### B. DATA STREAM CLUSTERING APPROACHES

There are several data stream clustering approaches in literature. The major approaches are described as follows:

*Partitioning:* This approach produces spherical-shaped clusters. The most prevalent are the k-means and k-medoids. The k-means is not appropriate for spherical-shaped clusters but good when the number of clusters is known. Noise and outliers affect it.

*Density-based*: This approach produces arbitrary shaped clusters mostly in partitioning method. The density-based approach can detect arbitrary shapes clusters and outliers.

*Grid-based*: This method produces clusters based on grids to speedup clustering process.

*Model-based*: This method is based on statistical models and allows objects to belong to several clusters.

*Hierarchical based*: The hierarchical is grouped into agglomerative and divisive. Examples of agglomerative are Hierarchical Agglomerative Clustering (HAC) and BIRCH [10]. HAC, however, is not suitable for data stream due to multiple scans.

### C. DATA STREAM CLUSTERING

An analysis of benchmark stream clustering algorithms was presented in [8]. The authors appraised the performance of CluStream [17], DenStream [16] and ClusTree [18] using 4 real datasets, Electricity, Adult, Poker, and Forest Cover Type. The authors use different evaluation measures Sum of Squared Distance (SSQ), Purity, and Clustering Mapping Measure (CMM) available in MOA for the evaluation. The results show that DenStream performed better on clustering quality based on window size while both CluStream and ClusTree outclassed DenStream on CMM metric. However, the study only focused on real-world datasets Adult-Census, Poker-Hand, Covertype, and Electricity. The CMM, a unique assessment measure for evolving data stream was developed [11] and implemented in MOA.

In [12], the authors presented the challenges and solutions of data stream clustering like noise, limited time, memory, evolving data, and high dimensionality.

In [13], the authors proposed "Clustering Evolving Data streams into Arbitrary Shaped" (CEDAS). The technique uses the Euclidean distance measure and can join or separate macro-clusters in a fully online method. The proposed technique was evaluated for processing speed, dimensional effects, purity, adaptation to evolving data, detection of intrusion, and Big Data, using both the KDDCup99 and real-world London Air Quality dataset and compared with DenStream, CluStream and MR-Stream and CEDAS performed comparatively well.

To scale up large data arriving at fast rate [14] propose a clustering algorithm that uses a two-stage strategy: a fast scale distance-based algorithm and a slower scale density-based algorithm. The authors evaluate the algorithm against CluStream and DenStream using concept drift experiment, robust path-based test, and multi-density test. The results proved that their algorithm performed better than the DenStream and CluStream algorithms.

In [15], the authors a new grid and density-based algorithm known as DGStream suitable stock markets and appropriate for handling outliers and noise. This algorithm uses feature vectors and DBSCAN algorithm at the online and offline phases, respectively. DGStream algorithm was evaluated against density-based algorithms: DenStream, ClusTree, and DStream are used for both synthetic and real-world datasets with different scales. On the synthetic dataset, DGStream outperforms some of these algorithms based on the following performance metrics: time, recall, purity, precision, and F1-score. DGStream also shows better performance than other algorithms on real-world datasets (Adult, KDDCup'99, Covertype and the National Stock Exchange of India (NSE Stocks, 2017).

In [22], the authors applied persistent homology on streaming data. The authors use *data summarization* and *computation of persistent intervals* approaches which serves as the online and offline component of data stream clustering algorithms such as CluStream [17], ClusTree [18], DenStream [16], and streaming *k*-means [23]. The online component of ClusTree was used for the model. The model identifies and detect *horizontal* or *reticulate genomic exchanges* during the evolution of Influenza and HIV viruses.

Several of the available data stream clustering algorithms are restrictive models [24]. To address the limitations, [24] proposed a novel data stream clustering approach improved streaming affinity propagation (ISTRAP). ISTRAP can detect and monitor clusters in evolution. However, it cannot handle high-dimensional data stream due to the Euclidean distance.

The advances in IoT device networks increase demands for effective security systems. In [25], the authors proposed an online and unsupervised scheme to detect attacks in smart home IoT networks. The scheme combines the algorithms CluStream [17] and Page-Hinkley test [27]. According to the experimental results, the overall detection rate is about 97% and the precision above 87%.

## III. METHODOLOGY

This study evaluates data stream clustering algorithms on the MOA platform using benchmark clustering algorithms. The study applied three datasets from both synthetic and real-world sources. The datasets are the synthetic data generator RandomRBFGenerator since it is the only implemented stream generator for clustering in MOA and two real-world datasets (Forest Covertype and Electricity). Since MOA cannot display the performance of three or more algorithms, we saved the output of the various algorithms as a csv file, merge, and populated in Microsoft Excel. We used Python open-source libraries (Pandas, hvplot, and Plotly) to visualization the graphs and charts offline on Jupyter Notebook. Other notable programming languages of choice like R and Gnuplot can also be used as an alternative.

## IV. EXPERIMENTAL SETUP

This study used MOA release-2021.7.0 conducted on the HP ProBook 450 G7, Processor: Intel(R) Core (TM) i5-10210U CPU @ 1.60GHz 2.11 GHz; RAM: 16.00 GB. System type: 64-bit, x64, Operating System: Windows 10 Pro.

The default parameters used for the stream clustering algorithms are given as follow: CluStream (with KMeans): horizon = 1000, maxNumKernels = 100, kernelRadiFactor = 2, k=5. ClusTree: horizon = 1000, maxHeight = 8. DenStream (with DBSCAN): horizon = 1000, epsilon = 0.02, beta = 0.2, mu = 1, initPoints = 1000, offline = 2, lambda = 0.25, and processingSpeed = 100.

### A. RANDOMRBF GENERATOR

We compare CluStream and ClusTree algorithms using synthetic data with 10% generated using RandomRBFGenerator with 205000 instances for the evaluation. The stream setting started initial with merge/split events every 5000 examples and gradually to every 50000 examples totaling 205000 and presented the output of CluStream in Fig. 1, output of ClusTree in Fig. 2 and output of DenStream in Fig. 3.

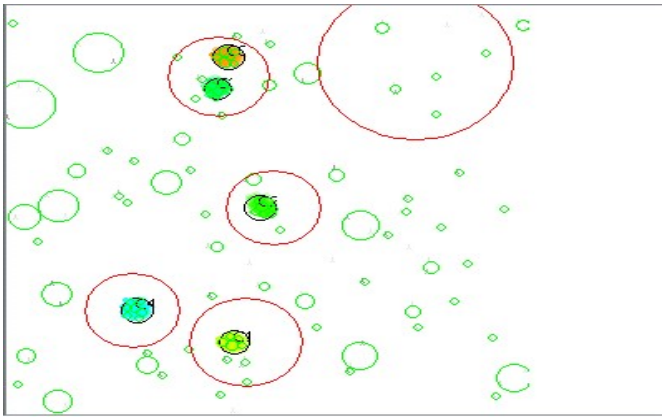

Figure 1. CluStream for RandomRBFGenerator with 10% noise. The clustering is indicated with red ring, the micro-clustering is in green rings, ground-truth is the black ring over the points.

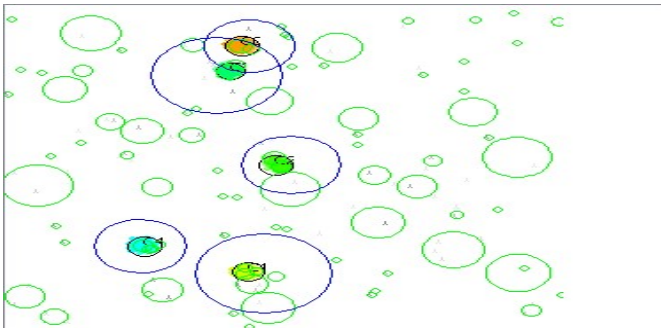

Figure 2. ClusTree for RandomRBFGenerator with 10% noise. The clustering is indicated with blue rings, the micro-clustering are in green rings, ground-truth is the black ring over the points.

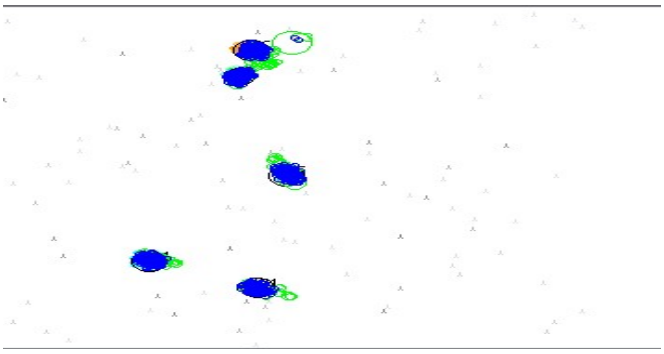

Figure 3. DenStream for RandomRBFGenerator with 10% noise. The clustering is indicated with blue dots, the micro-clustering is in green rings, ground-truth is the black ring, and the points are the other colors.

We compare the CluStream, ClusTree, and DenStream algorithms on a synthetic data generated using the RandomBRFGenerator with 10%, 30%, and 0% noise. The performance evaluation metrics (CMM, F1-P, F1-R, Purity, Silhouette Coefficient, and Rand statistics) used for the evaluation tabulated (see Table I). We used 200,000 instances of the synthetic data set for the evaluation.

The performance evaluation measures for the three algorithms using CMM measure is shown in Fig. 4 with ClusTree dropping below 0.50 at timestamp (53,000). Fig. 5 shows the barchart for the noise level 10%, 30%, and 0% on the synthetic data set. The result indicate that ClusTree outperform DenStream and CluStream virtually in all the metrics at noise level 0%,10%, 30% respectively except on Purity and Silhouette Coefficient (see Table 1, Table 2, and Table 3).

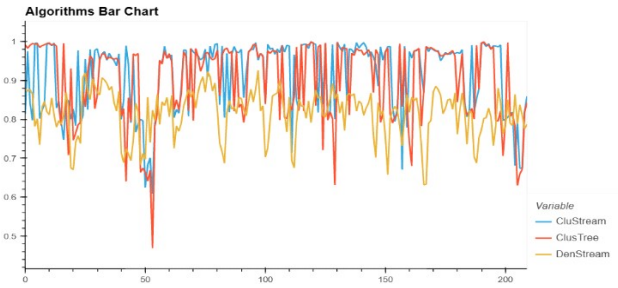


Figure 4. CMM result of CluStream ClusTree, and DenStream on RandomRBFGenerator with 10% noise.

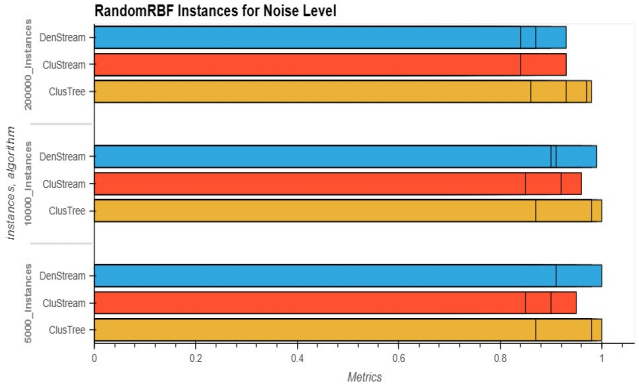


Figure 5. Noise level barchart of CluStream ClusTree, and DenStream on RandomRBFGenerator.

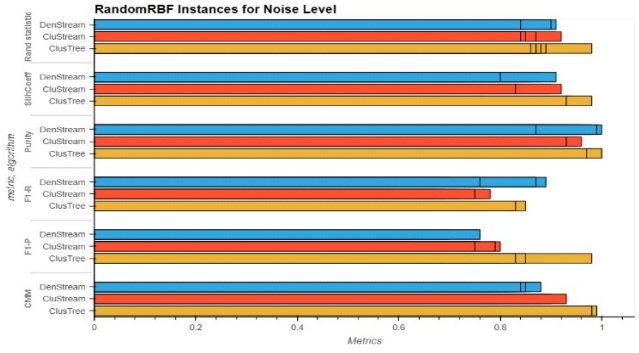


Figure 6. Noise level barchart of CluStream ClusTree, and DenStream on RandomRBFGenerator with performance metrics.

**Table 1. Average Evaluation measure of the algorithms on the RandomRBF Generator with noise level 0%**

| Metrics | CluStream | ClusTree | DenStream |
|---|---|---|---|
| CMM | 0.903 | **0.980** | 0.857 |
| F1-P | 0.767 | **0.843** | 0.740 |
| F1-R | 0.767 | **0.843** | 0.840 |
| Purity | 0.947 | **0.990** | 0.953 |
| Silhouette | 0.883 | **0.963** | 0.863 |
| Rand statistic | 0.847 | 0.867 | **0.883** |

**Table 2. Average Evaluation measure of the algorithms on the RandomRBF Generator with noise level 10%**

| Metrics | CluStream | ClusTree | DenStream |
|---|---|---|---|
| CMM | 0.823 | **0.960** | 0.837 |
| F1-P | 0.750 | **0.900** | 0.623 |
| F1-R | 0.710 | **0.787** | 0.660 |
| Purity | 0.867 | 0.940 | **0.953** |
| Silhouette | 0.793 | **0.883** | 0.820 |
| Rand statistic | 0.900 | **0.950** | 0.823 |

**Table 3. Average Evaluation measure of the algorithms on the RandomRBF Generator with noise level 30%**

| Metrics | CluStream | ClusTree | DenStream |
|---|---|---|---|
| CMM | 0.693 | **0.807** | 0.800 |
| F1-P | 0.767 | **0.780** | 0.323 |
| F1-R | 0.660 | **0.723** | 0.557 |
| Purity | 0.783 | 0.790 | **0.947** |
| Silhouette | 0.680 | 0.700 | **0.713** |
| Rand statistic | 0.837 | **0.863** | 0.700 |

### *B. FOREST COVER TYPE*

The data set has 581,012 instances with 54 attributes, 10 are continuous while the rest are binary, and each data is classified as one of seven forest cover type. The dataset is from the US Forest Service (USFS) and available at the UCI machine learning site. The Forest Cover type has been extensively used for much data stream research. The normalized version of the data set is used. The stream setting is like the RandomRBFGenerator above with 200000 instances for evaluation. The screenshot of the algorithms is in Fig. 7, Fig. 8, and Fig. 9 respectively. The average evaluation measure of the algorithms on the Forest Cover type with default settings, DenStream epsilon 0.03 parameter, and DenStream epsilon 0.05 parameter are tabulated (see Table 4, Table 5, and Table 6).

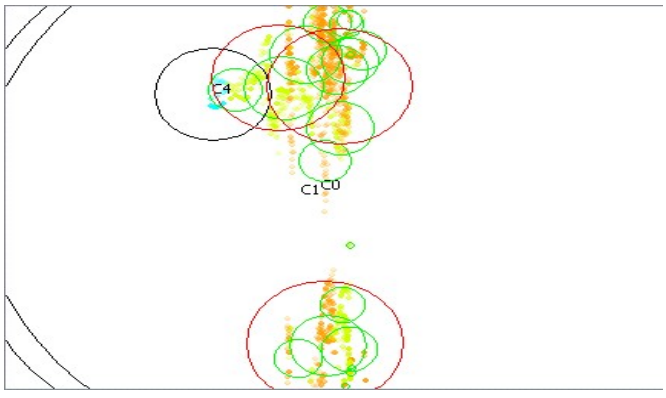


Figure 7. CluStream for Forest Cover type data set. The clustering is indicated with red rings, the micro-clustering is in green rings, ground-truth is the black ring over the points.

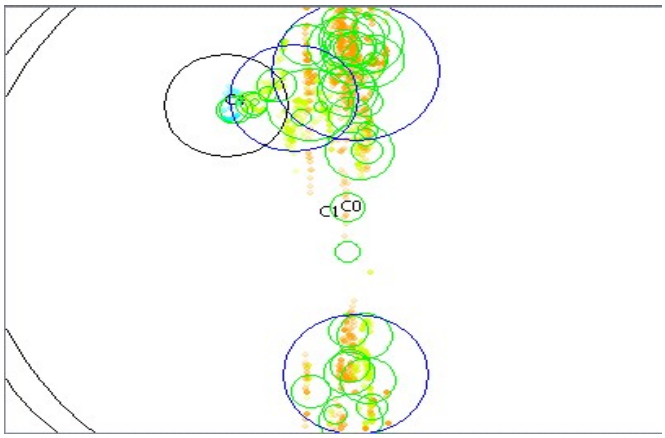


Figure 8. ClusTree for Forest Cover type data set. The clustering is indicated with blue rings, the micro-clustering is in green rings, ground-truth is the black ring over the points.

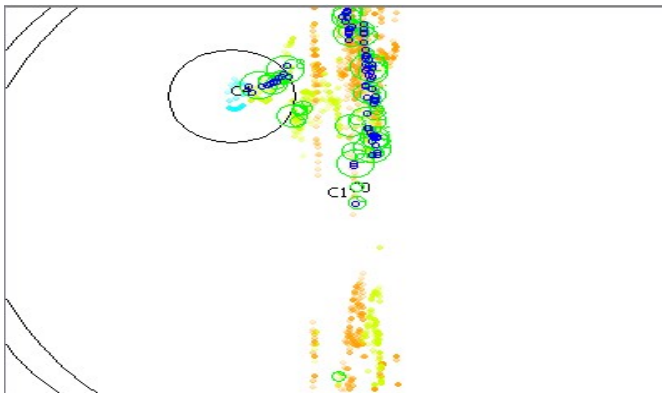


Figure 9. DenStream for Forest Cover type data set. The clustering is indicated with red rings, the micro-clustering is in green rings, ground-truth is the black ring over the points.

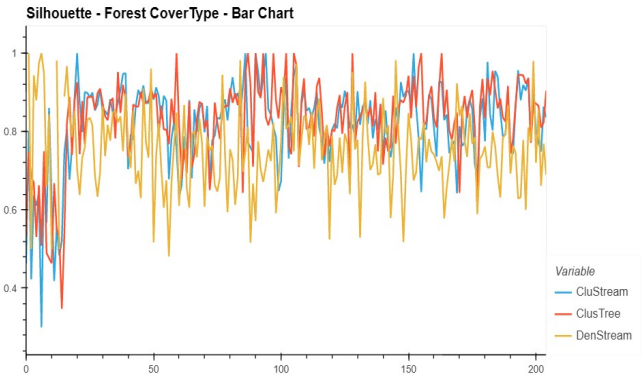


Figure 10. Silhouette Coefficient graph of CluStream ClusTree, and DenStream on Forest Covertype dataset.

**Table 4. Average Evaluation measure of the algorithms on Forest Cover Type with default settings**

| Metrics | CluStream | ClusTree | DenStream |
|---|---|---|---|
| CMM | 0.860 | **0.900** | 0.820 |
| F1-P | 0.690 | **0.740** | 0.620 |
| F1-R | 0.670 | **0.700** | 0.600 |
| Purity | 0.840 | 0.860 | **0.900** |
| Silhouette | 0.730 | **0.770** | 0.760 |
| Rand statistic | 0.870 | **0.890** | 0.790 |

**Table 5. Average Evaluation measure of the algorithms on the Forest Cover Type with Epsilon 0.03**

| Metrics | CluStream | ClusTree | DenStream |
|---|---|---|---|
| CMM | 0.860 | **0.900** | 0.860 |
| F1-P | 0.690 | **0.740** | 0.690 |
| F1-R | 0.670 | **0.700** | 0.670 |
| Purity | 0.840 | **0.860** | 0.840 |
| Silhouette | 0.730 | **0.770** | 0.730 |
| Rand statistic | 0.870 | **0.890** | 0.870 |

**Table 6. Average Evaluation measure of the algorithms on the Forest Cover Type with Epsilon 0.05**

| Metrics | CluStream | ClusTree | DenStream |
|---|---|---|---|
| CMM | 0.860 | **0.900** | 0.800 |
| F1-P | 0.690 | **0.740** | 0.690 |
| F1-R | 0.670 | **0.700** | 0.670 |
| Purity | 0.840 | 0.860 | **0.840** |
| Silhouette | 0.730 | 0.770 | **0.730** |
| Rand statistic | 0.870 | **0.890** | 0.870 |

The effect of adjusting the DenStream parameter to improve its performance is described in Fig. 11.

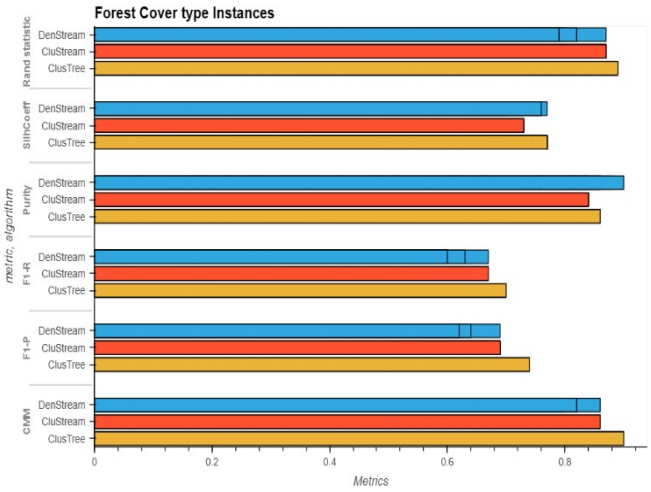


Figure 11. Barchart of CluStream ClusTree, and DenStream on Forest Cover type data set.

### C. ELECTRICITY

The dataset is a contribution from the Australian New South Wales Electricity Market. The dataset contains 45,312 instances. This research used the normalized version of the data set. The stream settings start with 5,000 examples initially and then 40,000 examples. At the end 45,000 records were used for the analysis. The screenshot of CluStream, ClusTree, and DenStream is given in Fig. 12, Fig. 13, and Fig. 14 respectively. In Fig. 15, the three algorithms are combined. The average evaluation measure of the algorithms on Electricity data set with default settings, DenStream epsilon 0.03 parameter, and DenStream epsilon 0.05 parameter are tabulated (see Table 7, Table 8, and Table 9).

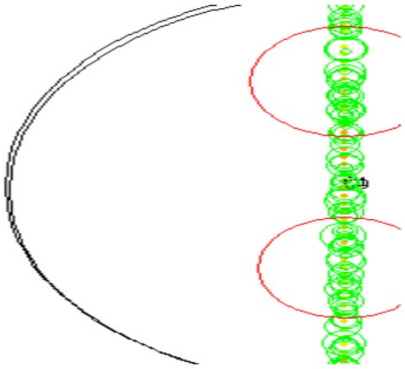

Figure 12. CluStream for Electricity data set

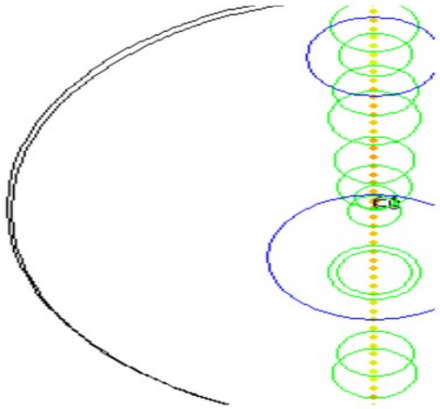

Figure 13. ClusTree for Electricity data set

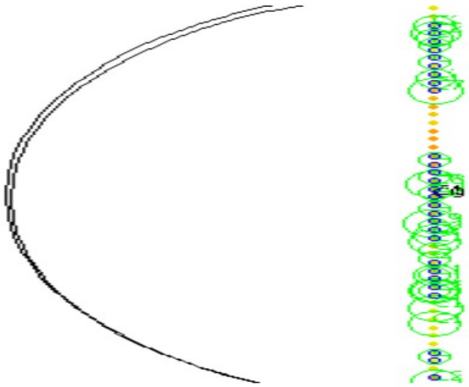

Figure 14. DenStream for Electricity data set

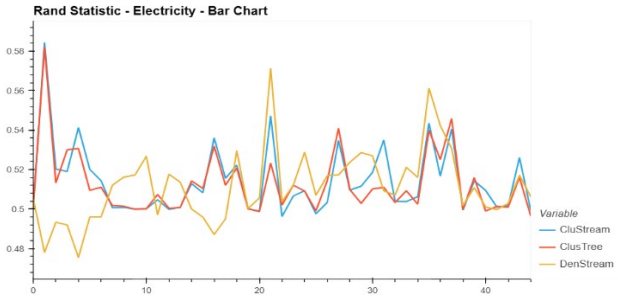


Figure 15. Rand statistic graph of CluStream ClusTree, and DenStream on Electricity dataset.

**Table 7. Average Evaluation measure of the algorithms on Electricity with default settings**

| Metrics | CluStream | ClusTree | DenStream |
|---|---|---|---|
| CMM | **0.760** | 0.750 | 0.490 |
| F1-P | 0.060 | **0.110** | 0.020 |
| F1-R | 0.090 | **0.110** | **0.110** |
| Purity | 0.780 | 0.700 | **0.900** |
| Silhouette | 0.670 | **0.730** | 0.490 |
| Rand statistic | 0.510 | 0.510 | 0.510 |

**Table 8. Average Evaluation measure of the algorithms on Electricity with Epsilon 0.03**

| Metrics | CluStream | ClusTree | DenStream |
|---|---|---|---|
| CMM | **0.760** | 0.750 | 0.600 |
| F1-P | 0.060 | 0.110 | **0.120** |
| F1-R | 0.090 | 0.110 | **0.230** |
| Purity | 0.780 | 0.700 | **0.830** |
| Silhouette | 0.670 | **0.730** | 0.490 |
| Rand statistic | 0.510 | 0.510 | 0.510 |

**Table 9. Average Evaluation measure of the algorithms on Electricity with Epsilon 0.05**

| Metrics | CluStream | ClusTree | DenStream |
|---|---|---|---|
| CMM | 0.760 | 0.750 | **0.780** |
| F1-P | 0.060 | 0.110 | **0.310** |
| F1-R | 0.090 | 0.110 | **0.480** |
| Purity | **0.780** | 0.700 | 0.750 |
| Silhouette | 0.670 | **0.730** | 0.570 |
| Rand statistic | 0.510 | 0.510 | 0.510 |

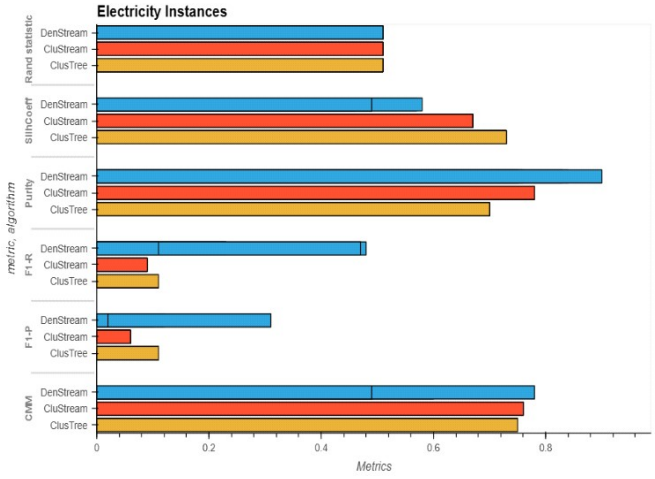


Figure 16. Barchart of CluStream ClusTree, and DenStream on Eletricity data set.

### *D. DENSTREAM WITH RANDOMRBF NOISE*

We further investigated the performance of DenStream with adjusted epsilon parameter on different noise level [0%, 10%, 30%] of RandonBRFGenerator. The output is tabulated in Table 10, Table 11, and Table 12 respectively. DenStream with epsilon 0.03 performance better on CMM and F1-P metrics on RandomBRFGenerator with 0% noise level than DenStream with default settings and DenStream with epsilon parameter 0.05 (see Table 10 and Fig. 17).

**Table 10. DenStream and RandomBRF Generator with 0% noise level.**

| Metrics | default | Epsilon_03 | Epsilon_05 |
|---|---|---|---|
| CMM | 0.840 | **0.850** | 0.760 |
| F1-P | 0.760 | **0.830** | 0.790 |
| F1-R | **0.760** | **0.760** | 0.690 |
| Purity | **0.870** | 0.820 | 0.720 |
| Silhouette | **0.800** | 0.790 | 0.750 |
| Rand statistic | **0.840** | 0.820 | 0.740 |

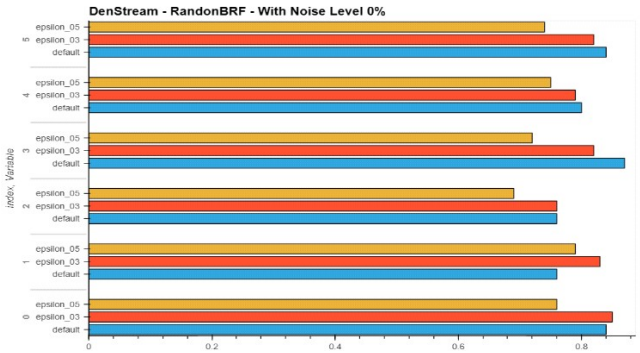


Figure 17. Barchart of DenStream on RandomBRF Generator with 0% noise level.

DenStream performance was also investigated on RandomBRFGenerator with 10% noise level. The result also shows that DenStream with epsilon parameter 0.03 outperform DenStream with default settings and DenStream with epsilon 0.05 on performance metrics CMM, F1-P, F1-R, Silhouette, and Rand statistic (see Table 11 and Fig. 18).

**Table 11. DenStream and RandomBRF Generator with 10% noise level.**

| Metrics | default | Epsilon_03 | Epsilon_05 |
|---|---|---|---|
| CMM | 0.820 | **0.850** | 0.810 |
| F1-P | 0.620 | **0.650** | 0.500 |
| F1-R | 0.600 | **0.620** | 0.610 |
| Purity | **0.900** | 0.870 | 0.860 |
| Silhouette | 0.760 | **0.770** | 0.700 |
| Rand statistic | 0.790 | **0.820** | 0.800 |

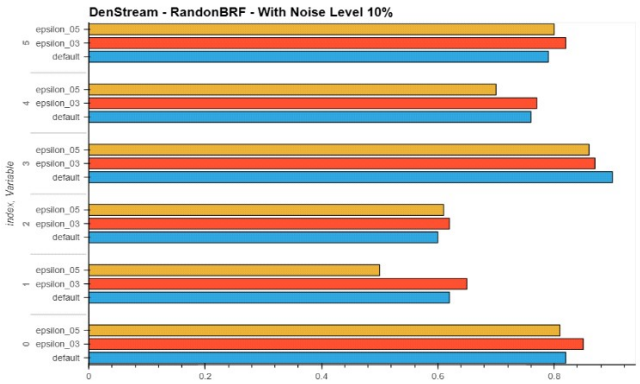


Figure 18. Barchart of DenStream on RandomBRF Generator with 10% noise level.

Lastly, DenStream performance was compared on RandomBRF Generator with 30% noise level. DenStream with epsilon parameter 0.03 also outperforms DenStream with default settings and DenStream with epsilon 0.05 on performance metrics CMM, F1-P, F1-R, Silhouette, and Rand statistic (see Table 12 and Fig. 19).

**Table 12. DenStream and RandomBRF Generator with 30% noise level.**

| Metrics | default | Epsilon_03 | Epsilon_05 |
|---|---|---|---|
| CMM | 0.780 | **0.830** | 0.700 |
| F1-P | **0.350** | **0.350** | 0.180 |
| F1-R | 0.500 | **0.600** | 0.520 |
| Purity | **0.930** | **0.930** | **0.930** |
| Silhouette | 0.670 | **0.740** | 0.640 |
| Rand statistic | 0.660 | **0.780** | 0.740 |

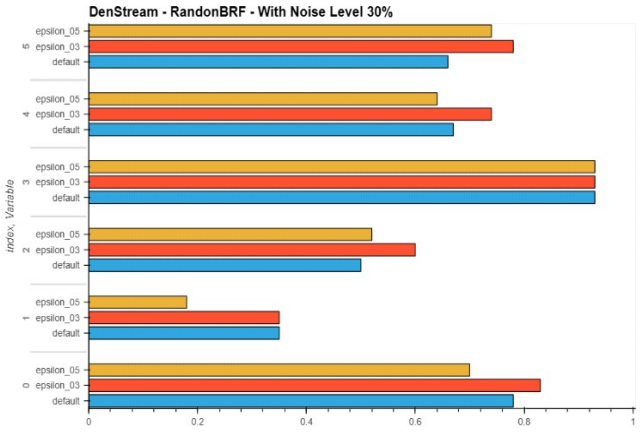


Figure 19. Barchart of DenStream on RandomBRF Generator with 30% noise level.

## V. CONCLUSIONS AND FUTURE WORK

In this paper, we have investigated and compared three data stream clustering algorithms on MOA framework using synthetic dataset RandomRBFGenerator and two real data sets Forest Cover and Electricity. The comparison was performed on RandomBRFGenerator using noise level [0%, 10%, 30%] and on the real data sets using DenStream epsilon parameter adjustment. We further investigated DenStream epsilon adjusted parameter on RandomBRFGenerator noise level [0%, 10%, 30%]. We have used performance evaluation metrics CMM, F1-P, F1-R, Purity, Silhouette Cofficient, and Rand statistics for the comparison. The data visualization was performed using Python libraries (pandas and hvplot).

From experiments, we observe ClusTree outperform DenStream and CluStream using RandomBRFGenerator with 0% noise level on performance metrics CMM, F1-P, F1-R, Purity, and Silhouette. Coefficient with 98%, 84.3%, 84.3%, 99%, and 96.8% respectively. ClusTree also performed better on noise level 10% and 30%. On Forest Cover Type, we observe data set, we observe ClusTree outperform DenStream and CluStream on most of the metrics. with adjusted epsilon parameter to 0.03, DenStream improves a bit and dropped with epsilon 0.05. On Electricity data set, DenStream outperform both CluStream and ClusTree on Purity and with epsilon 0.03 outperformed the two on F1-P, F1-R, and Purity. DenStream likewise outperform CluStream and ClusTree on CMM, F1-P, and F1-R.

In the further works, we will implement adjustment on DenStream other parameters on RandomBRFGenerator noise levels and real data sets.

## VI. ACKNOWLEDGEMENTS

This is to acknowledge the support University of South Africa M&D Bursary, the South African National Research Foundation, South African National Research Foundation incentive, and South African Eskom Tertiary Education Support Programme.

## References

[1] A. Bifet, J. Read, G. Holmes, B. Pfahringer, *Chapter 1: Streaming Data Mining with Massive Online Analytics (MOA)*. Data Mining in Time Series and Streaming Databases, 2018, pp. 1-25. https://doi.org/10.1142/9789813228047_0001

[2] A. Bifet, G. Holmes, B. Pfahringer, J. Read, P. Kranen, H. Kremer, T. Jansen, and T. Seidl, "MOA: A real-time analytics open-source framework." *in Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, September 2011, pp. 617-620. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-23808-6_41

[3] J. Wijffels, RMOA: Connect R to MOA to Perform Streaming Classifications. R package version 1.0, 2014. [Online]. Available at: https://CRAN.R-project.org/package=RMOA.

[4] M. Hahsler, M. Bolanos, and J. Forrest, "Introduction to stream: An extensible framework for data stream clustering research with R," *Journal of Statistical Software*, vol. 76, no. 14, pp. 1-50, 2017. https://doi.org/10.18637/jss.v076.i14.

[5] M. Hahsler, M. Bolanos, and J. Forrest, *streamMOA: Interface for MOA Stream Clustering Algorithms. R package version*, 2015, 51 p.

[6] J. Montiel, J. Read, A. Bifet, and T. Abdessalem, "Scikit-multiflow: A multi-output streaming framework," *Journal of Machine Learning Research*, vol. 19, issue 72, pp. 1-5, 2018.

[7] J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, A., H. Murilo Gomes, J. Read, T. Abdessalem, A. Bifet, "River: machine learning for streaming data in Python," *Journal of Machine Learning Research*, no. 22, pp. 1-8, 2021.

[8] S. Mansalis, E. Ntoutsi, N. Pelekis, and Y. Theodoridis, "An evaluation of data stream clustering algorithms," *Statistical Analysis and Data Mining*: *The ASA Data Science Journal*, vol. 11, no. 4, pp. 167-187, 2018. https://doi.org/10.1002/sam.11380.

[9] L. S. Agrawal, and D. S. Adane, "Models and issues in data stream mining," *International Journal on Computational Science & Applications (IJCSA)*, vol. 9, no. 1, pp. 6-10, 2016.

[10] T. Zhang, R. Ramakrishnan, M. Livny, "Birch: an efficient data clustering method for very large databases," *Proceeding of the SIGMOD*, 1996, pp. 103-114. https://doi.org/10.1145/235968.233324.

[11] H. Kremer, P. Kranen, T. Jansen, T. Seidl, A. Bifet, G. Holmes, and B. Pfahringer, "An effective evaluation measure for clustering on evolving data streams," *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, August 2011, pp. 868-876. https://doi.org/10.1145/2020408.2020555

[12] A. Amini, T. Y. Wah, M. R. Saybani, and S. R. A. S. Yazdi, "A study of density-grid based clustering algorithms on data streams," *Proceedings of the IEEE 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, July 2011, vol. 3, pp. 1652-1656. https://doi.org/10.1109/FSKD.2011.6019867

[13] R. W. Hyde, P. Angelov, A. R. Mackenzie, F. Nie, "Fully online clustering of evolving data streams into arbitrarily shaped clusters," *Inf Sci (N Y)*, 382, pp. 96-114, 2017. https://doi.org/10.1016/j.ins.2016.12.004

[14] N. B. Roa, L. Travé-massuyès, V. Grisales, "A novel algorithm for dynamic clustering: properties and performance," *Proceedings of the 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2019, pp. 565-570.

[15] R. Ahmed, G. Dalkılıç, Y. Erten, "DGStream: High quality and efficiency stream clustering algorithm," *Expert Syst Appl*, vol. 141, pp. 112947-112959, 2019. https://doi.org/10.1016/j.eswa.2019.112947

[16] F. Cao, M. Ester, W. Qian, A. Zhou, "Density-based clustering over an evolving data stream with noise," *Proceedings of the SIAM Conference on Data Mining*, 2006, pp. 328–339. https://doi.org/10.1137/1.9781611972764.29.

[17] C. C. Aggarwal, J. Han, J. Wang, P. S. Yu, "A framework for clustering evolving data streams," *Proceedings of the 29th International Conference*

*on Very Large Data Bases*, Berlin, Germany, September 9-12, 2003, pp. 81–92. https://doi.org/10.1016/B978-012722442-8/50016-1.

[18] P. Kranen, I. Assent, C. Baldauf, T. Seidl, "The ClusTree: Indexing micro-clusters for anytime stream mining," *Knowl Inf Syst*, vol. 29, pp. 249–272, 2011. https://doi.org/10.1007/s10115-010-0342-8.

[19] D. Krasnov, D. Davis, K. Malott, Y. Chen, X. Shi,. and A. Wong, "Fuzzy c-means clustering: A review of applications in breast cancer detection," *Entropy*, vol. 25, issue 7, p.1021. 2023. https://doi.org/10.3390/e25071021.

[20] A.A. Ewees, M. Abd Elaziz, M.A. Al-Qaness, H.A. Khalil, and S. Kim, "Improved artificial bee colony using sine-cosine algorithm for multi-level thresholding image segmentation," *IEEE Access*, vol. 8, pp. 26304-26315, 2020. https://doi.org/10.1109/ACCESS.2020.2971249.

[21] S. Mashtalir, O. Mikhnova, & M. Stolbovyi, "Multidimensional sequence clustering with adaptive iterative dynamic time warping," *International Journal of Computing*, vol. 18, issue 1, pp. 53-59, 2019. https://doi.org/10.47839/ijc.18.1.1273.

[22] A. Moitra, N. O. Malott, P. A. Wilsey, "Persistent homology on streaming data," *Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW)*, 2020, pp. 636–643. https://doi.org/10.1109/ICDMW51313.2020.00090.

[23] M. Shindler, A. Wong, & A. Meyerson, "Fast and accurate k-means for large datasets," *Advances in Neural Information Processing Systems*, 2375–2383, 2011.

[24] J. Sui, Z. Liu, A. Jung, L. Liu, & X. Li, "Dynamic clustering scheme for evolving data streams based on improved STRAP," *IEEE Access*, vol. 6, pp. 46157–46166, 2018. https://doi.org/10.1109/ACCESS.2018.2864553

[25] F. H. Y. Nakagawa, S. Barbon Junior, & B. B. Zarpelao, "Attack detection in smart home IoT networks using CluStream and Page-Hinkley test," *Proceedings of the 2021 IEEE Latin-American Conference on Communications, LATINCOM'2021*, Santo Domingo, Dominican Republic, 2021, pp. 1-6. https://doi.org/10.1109/LATINCOM53176.2021.9647769

[26] J. Fang, C. Chan, K. Owzar, L. Wang, D. Qin, Q. J. Li, & J. Xie, "Clustering Deviation Index (CDI): a robust and accurate internal measure for evaluating scRNA-seq data clustering," *Genome Biology*, vol. 23, issue 1, article number 269, 2022. https://doi.org/10.1186/s13059-022-02825-5

[27] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41 (1/2), pp. 100-115, 1954. https://doi.org/10.1093/biomet/41.1-2.100.

[28] M. Carnein, H. Trautmann, A. Bifet, B. Pfahringer, "Towards automated configuration of stream clustering algorithms," *Communications in Computer and Information Science 1167 CCIS*, 2020, pp. 137–143. https://doi.org/10.1007/978-3-030-43823-4_12

[29] M. Carnein, H. Trautmann, A. Bifet, B. Pfahringer, "confstream: Automated algorithm selection and configuration of stream clustering algorithms," *Proceedings of the 14th International Conference Learning and Intelligent Optimization, LION 14*, Athens, Greece, May 24–28, 2020, Revised Selected Papers 14, pp. 80–95. https://doi.org/10.1007/978-3-030-53552-0_10.

***TAJUDEEN AKANBI AKINOSHO.*** ***He is a current MSc. student in Computing. He holds Honors degree in Computing and another in Computer Science. He also holds a Higher National Degree in Statistics. His research interest includes Data Engineering, Data Science, Machine Learning, Big Data, and High-Performance Computing with focus on Quantum Computing.***

***ELIAS TABANE*** ***is an IT professional with a Decade of experience on both software and hardware. His area of specialty includes big data analytics, IoT, digital transformation, information /cyber security, and last machine learning (data science). He holds national diploma in IT, bachelor's degree in IT support services and a master's degree in business information systems. He is currently the operational manager for prestige ops and is also doing consultancy for digital transformation and emerging technologies.***

***ZENGHUI WANG*** ***graduated with Doctoral degree (2007) in Control Theory and Control Engineering, Nankai University, China. He is currently a Professor at the University of South Africa. His research interests include industry 4.0, control theory and control engineering, engineering optimization, image/video processing, artificial intelligence, and chaos.***