5.1 **Datasets used:**

The proposed algorithm has been implemented and tested on some well-known real data sets, i.e. Iris, Wine, m-feature-factors, Thyroid, and Synthetic control which we get from UCI for comparison of results. And for large data set, we are using the KDD-CUP'99 [16] Network Intrusion Detection as real stream data set. Descriptions of data are as follow in table (1):

Table 1 Data Sets:

Datasets	Size	Features	Clusters
IRIS	150	5	3
Wine	178	12	3
mFeat-factors	2000	217	10
Thyroid	2800	27	3
letter	2000	17	26
Synthetic Control	600	62	6
KDDcup99	494040	42	23

5.2 **Parameters Used:**

The proposed algorithm has been implemented and tested on following parameters used:

Table 2 Parameter Values:

Parameters	Values
Weight Exponant (m)	1.8
termination criteria ($\in$)	0.001
max iteration	20
number of individual	50
size of chromosome	$\sqrt{sizeofchunks(n_l)}$
number of generation	20
Crossover (P_c)	0.5
Mutation (P_m)	0.1

5.3 **Tools Used:**

This project has been implemented in ANACONDA3 (64 bit), spyder compiler, with python3.7 coding language, All the experiments were performed on a 1.8GHz Intel Core i5 processor computer with 8GB memory, running on Windows 10.

5.4 **Experiments Results:**

5.5 Validation:

We compare our validation result with others result in order to get a proper number of cluster, we are using real data set, comparison with validations i.e. cluster entropy (CE), partition coefficient(PC), Cohesion(VP), inter cluster Separation(SAP), Silhouette(SIL) and Calinski-Harabasz index (CH) as shown in table(3).We are receiving the data in window, can be equal or different size. Results shows that how much data we are receiving at instant and what the proper number of cluster is in between.

Table 3 IRIS Data Sets:

Window Size	c	CE	PC	VP	SAP	SIL	CH	**PROPOSED**
50	2	**4.51E-02**	**9.26E-01**	9.99E-01	**1.18**	**6.80E-01**	454.50	1.63E-01
	3	9.76E-02	8.38E-01	**9.99E-01**	1.75E-02	5.49E-01	529.12	**1.15E-01**
	4	1.25E-01	7.81E-01	9.98E-01	9.98E-02	4.97E-01	**558.12**	2.25E-01
	5	1.61E-01	7.04E-01	9.98E-01	2.24E-01	5.49E-01	513.30	3.86E-01

Table 4 WINE Data Sets:

Window Size	c	CE	PC	VP	SAP	SIL	CH	**PROPOSED**
60	2	**5.70E-02**	**9.09E-01**	**9.99E-01**	2.44E-01	**5.17E-01**	**764.85**	3.01E-01
	3	9.60E-02	8.41E-01	9.97E-01	9.64E-02	5.56E-01	694.95	**1.92E-01**
	4	1.02E-01	8.26E-01	9.95E-01	**1.89E-01**	5.61E-01	535.24	2.91E-01
	5	1.16E-01	8.00E-01	9.99E-01	2.04E-01	6.55E-01	505.12	3.21E-01

Table 5 Thyroid Dataset:

Window Size	c	CE	PC	VP	SAP	SIL	CH	**PROPOSED**
700	2	1.78E-03	9.96E-01	**2.85E-03**	1.51	8.94E-01	262.00	2.54E-02
	3	**5.76E-04**	**9.98E-01**	4.28E-03	**1.83**	**9.38E-01**	**1571.38**	**9.60E-03**
	4	2.43E-02	9.54E-01	5.75E-03	2.89E-04	7.33E-01	122.53	2.28E-01
	5	7.94E-02	8.55E-01	6.98E-03	4.03E-01	5.32E-01	56.31	1.020
	6	2.47E-02	9.52E-01	8.52E-03	2.61E-03	7.49E-01	261.52	2.59E-01
	7	8.79E-02	7.98E-01	9.90E-03	1.92E-01	4.18E-01	19.84	1.023

5.6 **Comparison with existing methods:**

We compare our algorithm with existing traditional Fuzzy c-mean where we are usign whole data once to find weather we are getting results equal or not. Where we are using stream data as chunks in proposed algorithm

Table 6 IRIS Dataset:

number of clusters		2	**3**	4	5
FCM	-	4.63E-01	**4.62E-01**	6.53E-01	6.41E-01
	WindowSize				
PROPOSED	50	3.01E-01	**1.92E-01**	2.91E-01	3.21E-01
	75	6.33E-01	**2.38E-01**	6.48E-01	7.24E-01

Table 7 Letter Dataset:

number of clusters		24	**25**	26	27
FCM	-	1.53E-01	**1.38E-01**	1.42E-01	1.43E-01
	WindowSize				
PROPOSED	500	3.440	**3.275**	3.426	3.678
	200	3.046	**2.659**	3.084	3.126

Table 8 Synthetic Control Dataset:

number of cluster		2	3	4	5
FCM	-	5.04E-01	**5.03E-01**	1.1210	1.0131
	WindowSize				
PROPOSED	50	6.58E-01	**2.62E-01**	3.97E-01	3.18E-01
	100	6.33E-01	**3.26E-01**	1.1485	1.1102
	120	6.75E-01	**3.64E-01**	7.15E-01	6.86E-01

Table 9 mfeat-factors Dataset:

number of cluster		7	8	**9**	10	11
FCM	-	2.211	2.336	**2.127**	2.524	2.604
	WindowSize					
PROPOSED	200	1.583	1.823	**1.162**	1.832	2.219
	400	1.815	1.736	**1.526**	2.155	1.936

5.7 Results of various data sets

DataSets	#Cluster
IRIS	3
Wine	3
Yeast	2
Morphological	3
mfeat-factors	9
Heart	5
Thyroid	3
Letter	25
KDDcup'99	22

5.8 **Memory complexity:**

It is the most important characteristic of continuous data algorithm for restriction in memory such as finite memory. If we are comparing our model with previous implemented model, we are utilizing memory space more than previous methods.

The memory complexity of Fuzzy c-mean is O (n (d + c)). Whereas, for our algorithm stream data space complexity is O (np (d + c)). Where n is the number of pattern, c is the number of cluster, d is the dimension of pattern and $p = \frac{n_i}{n}$, $1 \leq i \leq s$. The number of data in chunks X_i is n_i, for conclusion because $p < 1$, Our algorithm perform on small memory consumption.

6 Conclusion:

In this paper we are determining the number of cluster when we are dealing with stream data. For clustering stream data we update the Fuzzy c-means algorithm as Real time fuzzy c-means clustering algorithm to work with live data or stream data and give the optimal number of clusters. This algorithm is able to work with both kind of data i.e. stream data and large data which are out of bound of memory space of our systems. To avoid drawback of previous algorithm such as to reduce the large search area 2 to n-1 to find optimal clusters we use genetic algorithm, which help to reduce large search area and give global optimal result, rather than choosing local optimal by previous algorithms. We also proposed best suitable validation index for find number of clusters for real time fuzzy c-means. To achieve the goal, we compare the result of used validation index with various other cluster validation indices. Proposed algorithm use less memory than tradition algorithm in case of large data, and give same result as given by traditional algorithm. To deal with massive live data, which are coming more frequently than our algorithm's computation time, we need more storage until we didn't get the result of previous chunk. To avoid this problem we can use parallel processing on distributed system.

References

1. MK Masood, Wooi Ping Hew, and Nasrudin Abd Rahim. Review of anfis-based control of induction motors. *Journal of Intelligent & Fuzzy Systems*, 23(4):143–158, 2012.
2. Leszek Rutkowski, Maciej Jaworski, and Piotr Duda. *Stream Data Mining: Algorithms and Their Probabilistic Properties*. Springer, 2019.
3. Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.
4. Enrique H Ruspini. A new approach to clustering. *Information and control*, 15(1):22–32, 1969.
5. Andre D Lascari, Michael H Graham, and John C MacQueen. Intervertebral disk infection in children. *The Journal of pediatrics*, 70(5):751–757, 1967.
6. Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.
7. Dewan Md Farid, Li Zhang, Alamgir Hossain, Chowdhury Mofizur Rahman, Rebecca Strachan, Graham Sexton, and Keshav Dahal. An adaptive ensemble classifier for mining concept drifting data streams. *Expert Systems with Applications*, 40(15):5895–5906, 2013.
8. Shifei Ding, Fulin Wu, Jun Qian, Hongjie Jia, and Fengxiang Jin. Research on data stream clustering algorithms. *Artificial Intelligence Review*, 43(4):593–600, 2015.
9. Moh'd Belal Al-Zoubi, Amjad Hudaib, and Bashar Al-Shboul. A fast fuzzy clustering algorithm. In *Proceedings of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases*, volume 3, pages 28–32, 2007.
10. Timothy C Havens, James C Bezdek, Christopher Leckie, Lawrence O Hall, and Marimuthu Palaniswami. Fuzzy c-means algorithms for very large data. *IEEE Transactions on Fuzzy Systems*, 20(6):1130–1146, 2012.
11. Jinyin Chen, Xiang Lin, Qi Xuan, and Yun Xiang. Fgch: a fast and grid based clustering algorithm for hybrid data stream. *Applied Intelligence*, 49(4):1228–1244, 2019.