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

Real-Time High-Speed High Dimension Data Streaming and Feature Extraction on Edge Computing Devices in Industrial Internet of Things (IIoT)

Senthil Velan G.[1], **Dr. B. Shanthini**[2], **Dr. V. Cyril Raj**[3]

Submitted: 28/05/2023 **Revised**: 16/07/2023 **Accepted**: 28/07/2023

Abstract: Industrial Internet of Things (IIoT) and Artificial Internet of Things (AIoT) attracted technologist due to its economical impact and advancement in prediction and decision making using collected data. Data collection and data processing in the IIoT and AIoT is experiencing its limitations in computational devices and storage devices. IIoT and AIoT pumps large amount of data to the network in the form of high dimension and high speed, the pumped data is collected processed and feature extracted. Key Challenge of the collected data is to extract insights (Features) about it, huge amount of data collection and data processing has become a most interesting and open problem statement for both Industrialist and Researchers. IIoT system is becoming a primary area for data mining research, most sophisticated data analysis tools are required to understand the stress of data coming for the IIoT sensors and the devices connected to the network. More the devices connected to the network, more the data we need to process at a given time. This makes the data mining more complex in the IIoT environment. As the data is collected for longer duration, there is huge chance of data drift; this is one of the core issues in the IIoT data streaming and mining system. This paper proposes an efficient data mining and feature extraction technique for IIoT, The proposed technique reduce the computation significantly and increase the feature extractions on Edge computing. The author proposes a frame work for IIoT data processing named as Information Flow of IIoT (IFIIoT). The Proposed Framework is assessed using both synthetic and real-world data, encompassing various streaming speeds and data drift scenarios. The evaluation takes into account the overall performance of existing state-of-the-art algorithms in the literature, while the generated data provides valuable insights into the clustering system. Consequently, the proposed framework demonstrates superior performance compared to the current state-of-the-art algorithms described in the literature.

Keywords*: Industrial Internet of Things, Information Flow of IIoT, Streaming Process, Enhanced BIRCH Clustering*

1. Introduction:

Industrial Internet of Things (IIoT),an advance version of IoT for Industries. IIoT is collecting data in-order to understand the machine for better predictive maintenance. IIOT Networks are connected with large number of sensors and actuators which will sense the physical world parameters, and send the data to processing nodes; this IIoT network is bit different form conventional network. In most of the conventional network we load a web pages, but in the IIoT network we have a specific protocols like JSON or XML to transfer the data (not in the form of page) from the devices to the servers [1]. This data are in huge size, because there are many connected devices and this connected devices send data at very short interval. This become an huge data streams, as times goes, the number of connected devices has been increasing exponentially, this leads to very large size of data streams will happen in real time in the near future [2]. The biggest challenge of the IIoT system is to collect the data and get insights from it (insights are always hidden in the data) understanding the data is becoming a most important research area in the field of data mining, as the data is keeps on increasing.

In fact, 90% of the worlds data has been collected in the last three years, just think of it, internet is in usage for the past four decade, but the amount of data has been collected for the past 38 years is equal to the data collected in the last two years. The key projection about the IIoT s system is to keep increasing the connected devices, which will increase the data size exponentially, so the data mining is one of the primary problems in the IIoT systems [3]. IIoT is one of the most sophisticated and advanced system which uses low power communication devices, low band width consuming internet protocols, smart sensors and actuators, and other communication and sensing devices. All this devices are connected to the network and sends data forgetting insights without a human Intervention. IIoT is developed on Machine to Machine (M2M) Communication Extension[4]. Machines sends data to the nearest configured cloud storage. Data stream model of the IIoT system need to handle high speed and large data, due to vast amount of data, it is necessary to develop an algorithm, which will extract insights from the received data with minimum computation time and optimized memory space for storing all the collected data from the IIoT system [5]. The primary problems of the data streaming and mining algorithms are restricted with computation time and storage memory and the secondary problem is to deal with data drift.

To reduce costs, it is essential to optimize resource utilization. In the context of data stream mining, our focus revolves around enhancing performance through the following aspects:

- Accuracy: Ensuring precise and reliable

Research Scholar[1], Professor[2], Department of Computer Science and Engineering, St.Peter's Institute of Higher Education and Research, Chennai.
Professor[3], Department of Computer Science and Engineering, Dr MGR Educational and Research Institute, Chennai

predictions or classifications.

- Memory Efficiency: Minimizing the amount of memory required for storing and processing data.
- Time Optimization: Streamlining the learning process from training data and expediting prediction generation.

Significant issues are presented by the configuration of Industrial Internet of Things (IIoT) systems. It calls for algorithms that are extremely effective in terms of time and memory usage, flexible enough to adapt to changes, and capable of uninterruptedly learning [6]. Additionally, these algorithms must be distributed and work without a hitch on top of Big Data infrastructures. Accurately meeting these needs in real-time is the main issue for IoT analytics solutions. Massive volumes of continuous data emanating from the physical world have accumulated as a result of the shift from the desktop computing era to ubiquitous computing and the Internet of Things.

IoT data is unique and differs from data used in common databases and machine learning. It originates from diverse sources and domains, including sensors and social media streams. Unlike traditional data streams with Gaussian distributions, IoT data consists of short-term snapshots with sporadic distributions. It is dynamic, with changing data distributions over time. Due to the large quantities and real-time nature of IoT data, specialized data analytics solutions are needed to handle its heterogeneity, dynamics, and velocity. While clustering or classification methods can be used to group the data, classification methods require labeled training data, which may not be feasible for IoT applications with vast amounts of data. In contrast, clustering methods do not require supervised learning but are more effective in offline scenarios with fixed data distributions. This paper introduces a clustering method capable of adapting to changes in the data stream, making it well-suited for the framework of IIoT data streams.

In data analysis, clustering is commonly used to group data based on similarity and homogeneity. The resulting clusters represent categories within the dataset and enable data assignment to different groups. This paper presents an adaptable clustering method that examines data distribution and dynamically updates cluster centroids to accommodate changes in the online data stream. This approach facilitates the creation of dynamic clusters and assigns data based on both their features and the distribution patterns observed at a specific time. The authors evaluate the proposed technique using real-time data obtained from live traffic streaming via GPS location. The system, referred to as an intelligent traffic analysis system, clusters traffic sensor measurements based on various features including average vehicle speed, vehicle types, and vehicle count. These clusters are then analyzed and assigned labels, such as "busy," based on factors like overall vehicle density at a specific time and road capacity. By abstracting further, the technique can identify events such as traffic jams, which can be utilized by automated decision-making systems like GPS navigators to enable automatic rerouting [10].

The study discusses relevant research on the analysis of idea and data drifts in stream data. The mathematical underpinnings of the Elbow technique, which is used as a metric to assess cluster quality, are discussed. The Enhanced BIRCH clustering approach, which automatically chooses the ideal number of clusters based on the distribution of the data, is introduced. The suggested method is tested against cutting-edge techniques using both synthetic and real-time datasets to determine how well it performs.

2. Related Work

In the context of solving the clustering problem, one of the methods that stands out is the k-means algorithm, commonly referred to as Lloyd's algorithm [11]. This algorithm has been selected for its simplicity and its effectiveness in improving streaming data clustering. Additionally, the concept of utilizing data distribution can also be applied to determine the parameter k for k-median [12] or to determine the number of classes in unsupervised multi-class support vector machines and other clustering algorithms [13].

A clustering technique called the k-means algorithm seeks to divide a given dataset into k different clusters. Each data point is initially assigned to the closest cluster centroid after the algorithm chooses k random cluster centroids from the dataset. On the basis of the mean value of the data points within each cluster, the centroids are then changed iteratively. Up until the centroids stabilise, this procedure continues. However, numerous rounds with various beginning values are necessary because the ultimate outcomes largely depend on the initial centroids. This can lead to significant computational overhead, especially for streaming data. Additionally, the use of random restarts for convergence can result in lower quality clusters and increased computational time. To address these challenges, the k-means++ algorithm [14] introduces an intelligent approach for selecting initial centroids based on randomized seeding, considering their potential proportion to the overall dataset.

When dealing with large data sets that exceed memory capacity, the STREAM algorithm [15] offers a solution by treating the data as a stream and performing one-pass clustering. However, this approach may lead to misclustering as the data stream evolves. To overcome this limitation, Aggarwal et al. [16] introduced CluStream, which addresses the problem by dividing it into an online Microcluster component and an offline Macroclustering component. The number of clusters must be predetermined, fixed, or selected by the user at each phase in the CluStream process, necessitating human oversight at every stage.

StreamKM++ [17] is a widely used stream clustering approach that builds upon the k-means++ [14] algorithm. However, similar to other methods, StreamKM++ requires prior knowledge of the number of clusters and is not well-suited for evolving data streams. The challenge of determining the appropriate number of clusters in a dataset has been extensively investigated, and Chiang and Mirkin [18] proposed a novel method called k-means, which demonstrates good performance in selecting the number of clusters and cluster recovery.

This method identifies clusters by detecting new anomalies in the data and eliminates small clusters using a threshold based on Hartigan's rule [19].

To avoid misleading results caused by single strong outliers, it is important to eliminate singleton clusters from the results, as noted by Rousseuw. In a streaming setting, the time frame for measuring cluster quality using the Elbow Values should be defined carefully [20]. The last time the centroids were recalculated can serve as a natural time frame, as this is when data drift was detected and the new clustering has to adapt to the data stream. By analyzing existing algorithms, this paper proposes an efficient Enhanced BIRCH clustering technique for the IIoT environment.

3. Proposed Enhanced BIRCH Clustering Technique for High-Dimensional Data in Iiot

BIRCH employs the Clustering Feature tree (CF tree) to partition incoming data points incrementally and dynamically. In the standard BIRCH algorithm, once a data point has identified the CF-node based on the closest distance calculation, it will be assigned to the CF-leaf if the radius of the leaf does not surpass the threshold (T). However, if the radius exceeds the threshold, a new leaf will be created. Additionally, if the leaf limit is exceeded, a split parent operation will be performed.

In Enhanced BIRCH, if a new data point exceeds the threshold, it will be adjusted to the threshold value. This adjustment is achieved by increasing the leaf radius scale, aiming to minimize the need for split parent operations, which are common in the original BIRCH algorithm.

The need for Enhanced BIRCH clustering that can adjust their parameters and clustering approach based on changes in data streams is crucial for high dimensional data in IIoT environments. Traditional stream clustering methods require prior knowledge of the number of clusters or different parameterization, which is not feasible in dynamic environments where the data distribution can change over time. To fully utilize the abundance of data produced, it is important to consider new possibilities and insights that have not been previously explored. However, prior knowledge and assumptions can also enhance or alter the results obtained. The proposed data streaming and clustering framework heavily relies on the incoming data from industrial sensors via gateway, and all sensors are optimized to minimize data transmission loss. The data flow model is illustrated in Figure 1.

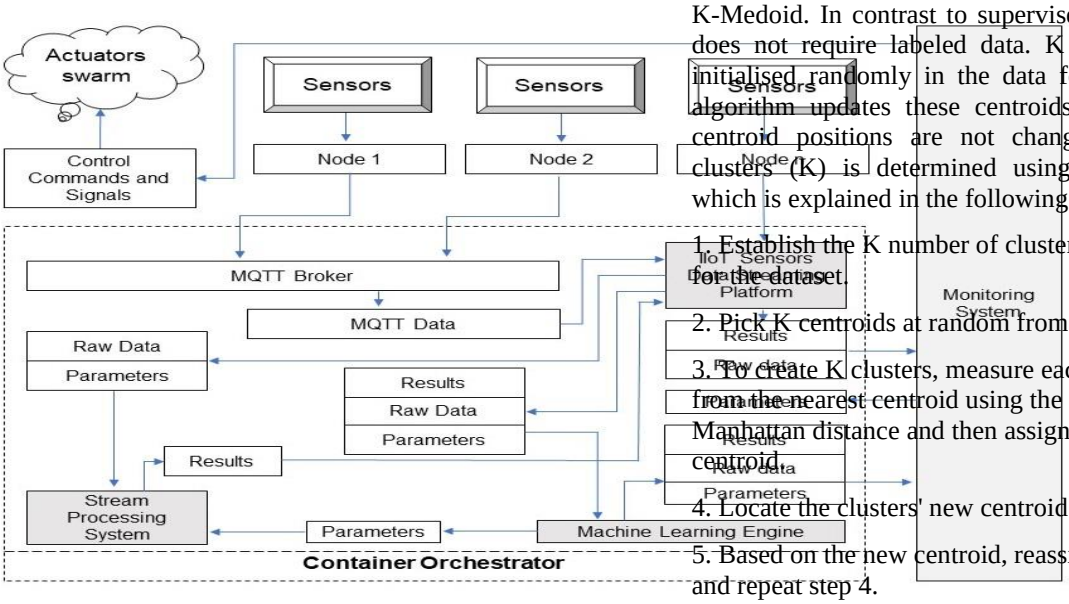

Fig 1 –Framework for the High Speed data streaming for the IIoTApplication.

3.1 Predicting the Clusters Requirement:

In dealing with unfamiliar data, a fundamental issue is identifying the number of clusters present in various data segments. To address this, we suggest that the distribution of the data can provide useful insights into the categories. Since data usually contains multiple features, we examine the distribution of each feature to estimate the necessary number of clusters.

3.2 Elbow Method

K-Means is the most widely used iterative unsupervised learning algorithm, which is simpler than other algorithms like PCA (Principal Component Analysis) or K-Medoid. In contrast to supervised learning, K-Means does not require labeled data. K cluster centroids are initialised randomly in the data for K-Means, and the algorithm updates these centroids iteratively until the centroid positions are not changed. The number of clusters (K) is determined using the Elbow Method which is explained in the following steps

1. Establish the K number of clusters that are necessary for the dataset.
2. Pick K centroids at random from the dataset.
3. To create K clusters, measure each point's distance from the nearest centroid using the Euclidean or Manhattan distance and then assign the point to that centroid.
4. Locate the clusters' new centroid.
5. Based on the new centroid, reassign each data point, and repeat step 4.

Repeat this procedure until the centroid positions stop changing after a predetermined number of iterations.

Identifying the ideal number of clusters is a crucial aspect of the K-Means algorithm. The Where in the Elbow method, which involves plotting the Within-Cluster Sum of Squares (WCSS) against the number of clusters (K) and choosing the K value at the "elbow" point of the resulting graph, where the WCSS begins to decrease at a slower rate with the addition of more clusters, is a popular method for determining the ideal value of K.

BIRCH employs the Clustering Feature tree (CF tree) to partition incoming data points incrementally and dynamically. The section provides an overview of the

four phases in the BIRCH algorithm and explains the construction of the CF tree. Theoretical basics of Clustering Features are discussed, highlighting how distance metrics can operate solely on these features. The section concludes by revisiting how BIRCH utilizes Clustering Features to compress data, enabling the algorithm to handle large datasets within limited RAM capacity.

3.3 Handling the Data Drift:

The clusters produced by the k-means method with random restarts become constant when the input data is fixed and can be used to classify related datasets. In the case of streaming data, however, same data viewed at various time periods may have various interpretations and belong to several clusters, leading to various clustering findings. For instance, if the volume of traffic is high at 7 p.m., it is taken into consideration; however, if it is heavy at 7 a.m., it may be exceptional and signify the presence of a special event. In order to solve this, we alter the cluster centroids in accordance with the data stream's current distribution. Data drift is then addressed by keeping an eye on changes in the statistical properties. Our method is based on the characteristics of stochastic convergence, which show that mean square convergence implies convergence in probability and distribution. The formula for convergence in the mean square is given as

$$\lim_{n\to\infty} E|Xn - X|^2 = 0$$

The proposed algorithm comprises three sets, the first of which involves identifying the centroids of the received data from IIoT. This algorithm is presented below in Algorithm 1, which outlines the process of identifying the centroids using Elbow techniques. At each step, adjustments are performed, and the new centroid value is updated in the system. The Probability Density Function (PDF) is computed and represented in an array of discrete elements. The second step involves counting the turning point to deal with data drift. To determine the turning point value, we first calculate the derivative of the system, where dx/dy=0, with dy being the closest point in the y array and dx being the closest point in the x array.

Algorithm 1: Proposed Method of Initial Cluster Centroids

DS dataset and K clusters are the inputs.

Consequently, K clusters have the best starting centroids.

Procedure: Proposed Method(Input)

1. Each of DS's n attributes, a1, a2, a3,..., an, must be a number. Simply change any attributes with non-numeric values to numeric values.

2. Apply Principal Component Analysis to the DS dataset using 2 components.

3. Utilise percentile to divide the entire dataset into K equal portions based on the first component.

4. From the primary data, extract the split dataset using the index.

5. Calculate the mean for each attribute in the split dtasets

6. Take each dataset's mean as the initial cluster centroids, CE= ce1, ce2,.., cek, where ce1, ce2,.., cek are the initial centroids for the first, second,.., k clusters sequentially.

7. The Centroids should be assigned to the K Mean clustering algorithm

The initial cluster is the key identification problem, once the cluster is identified, then the feature is extracted using K- Mean in the dataset, where the centroid found at the initial conditions, this centroid value is passed into the turning point validation system to correct its value based on the streaming data. On the fly, centroid and the turning point value is calculated and updated for the next calculation. Algorithm 2 explains about the identifying the K mean of the live streaming data in the Industrial IoT environment.

Initial Centroid is found by Algorithm 1and subsequently used in the Algorithm 2. Executing of initial K means value take the computation of the O(ndk), because there is no loop iteration in the algorithm. For the clustering the given IIoT data, it's necessary to identify the elbow method values. Elbow values are a special value which says the streaming data validity and the consistency. Calculating the score is a computational intensive job. Instead of calculating the distance between two points, we can calculate the distance between the point and the K mean and pass it to the Elbow value as a distance pair matrix. As the data counts increases, constructing the distance matrix become difficult, so in order to reduce the computation, sampling the distance value is done and then clustering is done. Now the challenge is to validate the distance value obtained by sampling which is not accurate, so it's a trade-off between the matrix and the sample value. The best way to solve this problem is to sample the right data for the distance matrix calculation.

Assigning the closest value is one of the key operations in the streaming and clustering. It's a low computation intensive task, but if there is a requirement to recalculate the cluster values, then the computation cost of the system grows as fast as possible with number of input data.

3.4 Enhanced BIRCH algorithm:

1. To generate the CF form for all data points, apply the formula CF = (N, LS, SS), where SS is the total of the attribute values' squared values (X2)., where LS stands for the total of the attribute values (X) and N stands for the number of data points.

2. The CF-Tree begins operating by integrating numerous CFs after the data is translated into CF form. You will be prompted to enter the value of B (Branching) at this point.

3. It is essential to initialise the CF tree's initial threshold before extracting any data points from the database. This threshold acts as the initial value for every new CF entry and doesn't alter throughout the grouping process.

4. The parameter L (number of leaf nodes), which is

dynamically inserted as needed for clustering, is not initially necessary in the improved version of BIRCH.

5. Two other parameters, m and b, are added to make calculations easier. While m counts the number of leaf branches in CF leaf nodes, the parameter b maintains track of the number of branches in CF non-leaf nodes.

6. When a CF subcluster is introduced, BIRCH uses the Euclidean distance to compare its placement with the created CFs at the root node. The CF subcluster is then assigned to the nearest CF root node via BIRCH.

7. From the selected CF node, the subcluster then descends to the chosen non-leaf child node. The location of each non-leaf CF is compared by BIRCH with the location of the subcluster. The non-leaf CF node that is closest to the incoming subcluster is where BIRCH continues the descent.

8. From the chosen non-leaf CF node, the subcluster then falls to the leaf child node. BIRCH compares the positions of each leaf node and the subcluster. The closest leaf node along the descent path receives the subcluster's temporary assignment from BIRCH.

9. After determining which leaf node is nearest, the subcluster determines if the radius of the chosen leaf, including the newly formed subcluster, does not go above the threshold T. The subcluster is assigned to that leaf (modified leaf-CF) if it satisfies the requirement.

10. The system increases the cluster scale if the radius of the selected leaf, including the CF subcluster, is larger than the threshold T. The radius is once more verified, and if it does not exceed the new threshold, the T threshold is changed and the subcluster is assigned to the leaf(modified leaf-CF).

11. A new leaf (modified leaf-CF), made up only of the incoming subcluster, is generated if the updated threshold value still causes the radius to exceed T. To incorporate the new data points, the CF parent is updated all the way up to the root.

12. Once the previous steps have been finished, BIRCH moves on to phase 3, also known as global clustering.

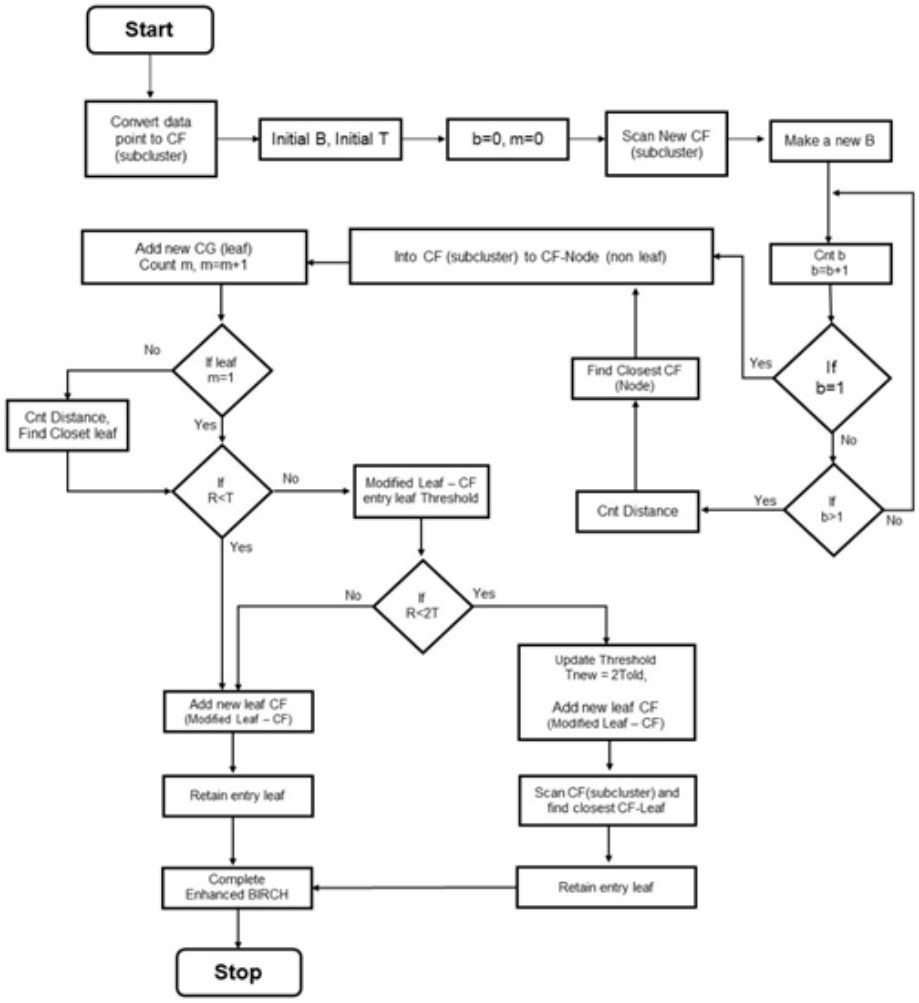

Fig 2 –Data Flow in Enhanced BIRCH Algorithm.

The quality of the cluster can be significantly improved by altering the BIRCH algorithm at the threshold setting. In this study, we suggest changing the CF-entry threshold value to improve cluster quality dynamically. In contrast to the conventional BIRCH method, which applies the CF formula CF = (N, LS, SS), our study makes changes to the CF-Leaf component. To be more precise, we use the modified leaf-CF formula: CF-Leaf

(modified leaf-CF) = (N, LS, SS, T), where T is the threshold value. The T parameter, which has been added to CF-Leaf (modified leaf-CF), enables the storage and tracking of the most recent threshold modifications. It's vital to remember that CF-Leaf only uses the T parameter, but CF-Node continues to use the equation CF = (N, LS, SS). Changes in threshold values may be able to enhance cluster quality.

4. Evaluation of Information Flow (IF) of Iiot (IF-Iiot):

Novel Framework is introduced for generating data streams and the data drift. The drift is achieved by two ways, one by shifting the random intervals and the other technique is to shift the centroid values obtained by the Algorithm1, by keep changing the centroid value, and we can achieve it via best data drift on the live streaming. Our proposed framework can be highly suitable for the Heavy IIoT environment and the time critical operations. All the setting has been kept forward to develop a reasonable synthetic data for our framework. This paper uses a two clustering algorithm to develop a data, the first one is Clustream [6], it has horizon value of 1000, maximum number of kernels is 100 and the radifunction is 2. By using this value, Random RBFGenerator is used to get the data drift. Based on the random generate value, the weight and the distribution is equally validated and assigner.

This paper introduces an novel way of generating the values suitable of the data drift. In the HyperCubing sample, the dimensions and the number of clusters are fixed, based on the deviation in the values and the distribution of the value in the paper way, we can consider that the data extracted as a feature is valid and it will be valid until the next cluster is formed. The biggest challenge in the Long term data and the IIoT data is all about its heterogeneity. The proposed framework support all kind of distribution, they are Cauchy distribution, Triangular distribution, Gaussian distribution, Exponential distribution, Poisson distribution and so on. Hence, the Proposed framework is highly suitable for the both the homogeneous and the heterogeneous data sets.

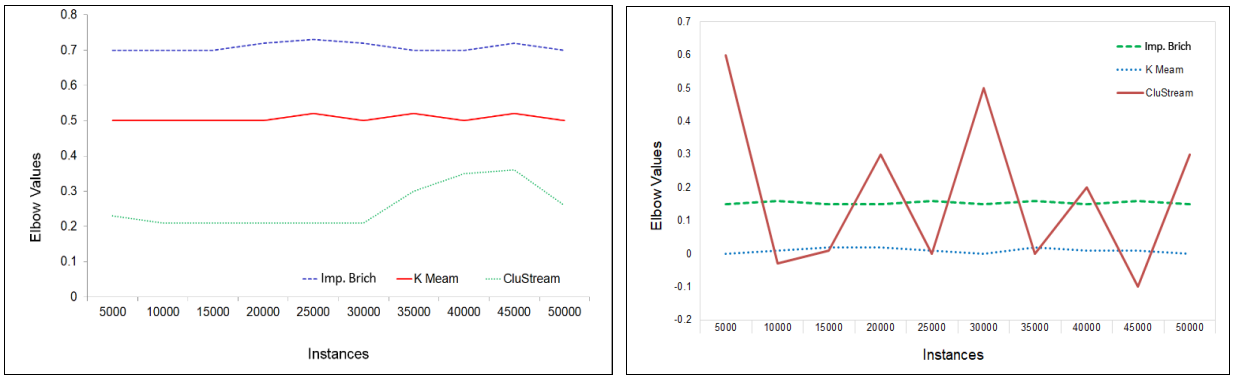

Fig 3 : a) Elbow values comparison on synthetic data sets using RBFGenerator.

b) Elbow values of synthetic data sets suing 3 features.

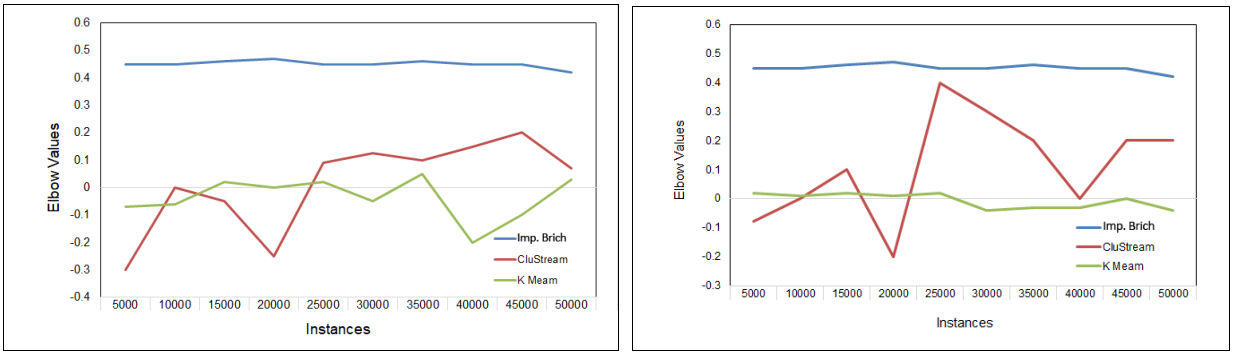

Fig 3 : c) Elbow values of synthetic data sets using 4 features. d) Elbow values of synthetic data sets using 5 features.

The proposed technique is dealing with the huge data drift and its data heterogeneity; it's all achieved by using centroid distribution. This means that the provided IIoT data is dealing with huge traffic. The below table is calculated using Elbow Values with the initial clustering data, then the number of clustering data, for calculating the performance of the proposed system, we have taken a sample of 1000 has been chosen and validated with all the possible cases in the system.

Table 1: Elbow Values with Different Cluster Value

Number of Clusters	Elbow Values
3	0.450232
4	0.364321
5	0.334389
6	0.408786
7	0.398814

The proposed framework is very efficient in clustering the right value into the right Cluster, for the better performance is calculated and compared with the proposed Framework. The cluster value drops to 0.5 and below, then mis-clustering will start affecting the performance of the framework. So, maintaining the value above half is very important while the live streaming and the clustering on the fly. The proposed framework is best for on the fly clustering and streaming. Figure 3 explains about the proposed technique with different clustering activities and its mis-clustering punctuations.

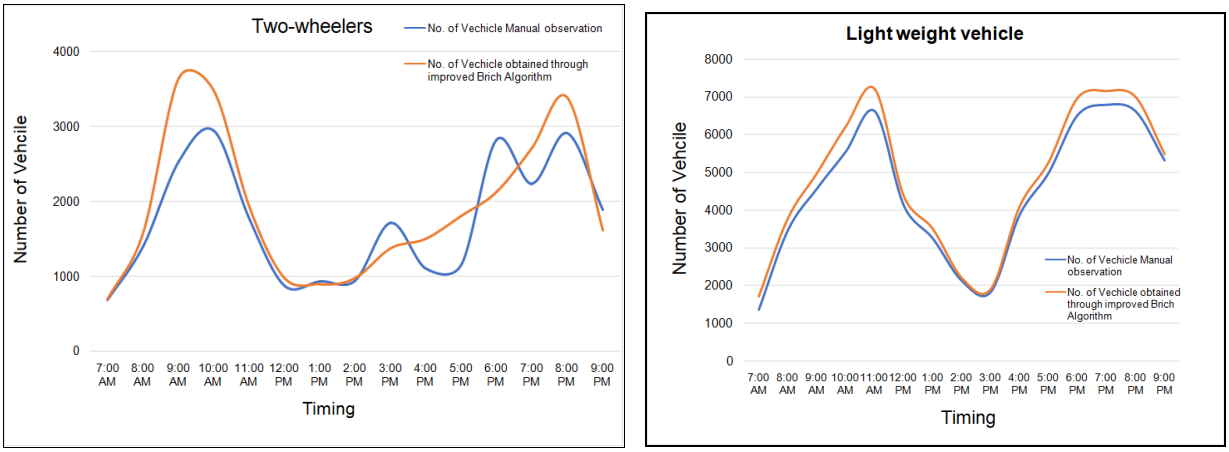

Fig 4 : a) Real time traffic application of Enhanced BRICH algorithm for Two wheelers

b) Real time traffic application of Enhanced BRICH algorithm for Light Vehicles

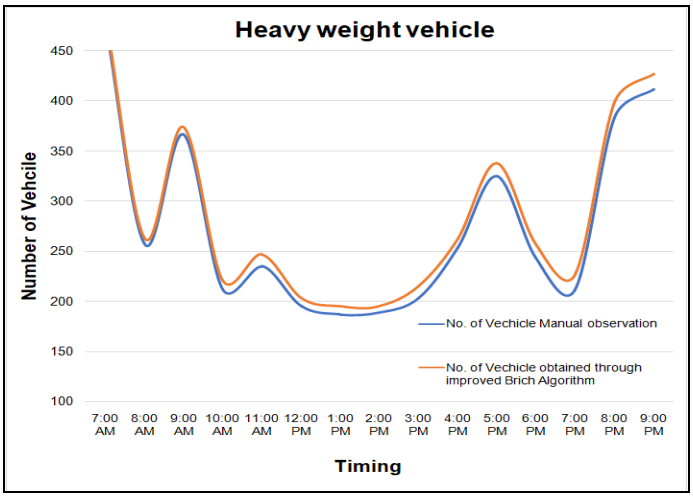

Fig 4 : c) Real time traffic application of Enhanced BRICH algorithm for Heavy Vehicles

To test our methodology in a real-world scenario we used real time traffic data from Chennai city. Though several thousands of sensors are available, we have identified 800 sensors which include (camera, speed sensors and proximity sensors, etc.). Once in every fifteen minutes, data from these sensors are collected and given as input to our clustering algorithm. During the peak traffic time between 7am to 11am and 4pm to 9pm. In a day 28,800 data are collected. We used 10days windows for this case study. We used the following attributes: the number of two-wheelers, the number of four-wheeler and the number of heavy vehicles near the sensor at the time of observation. The average speed of the above vehicles is also considered.

For better understanding of the traffic pattern and visualization time stumping and location details are also added in the sensor input. The understanding is type of vehicle, average speed of the vehicle type, number of such vehicle type are varying in the peak time period. Figure 4a, 4b, 4c are drawn based on cluster centroid provided by our algorithm for various type of vehicle over the above period. The graph shows that traffic density is high in peak time in the morning between 8am to 11am and 4pm to 9pm in the evening. In order to counter check our algorithm we obtained data from the Toll Plaza during the observation period to have exact count of various type of vehicle which have crossed the sensor.

Figure 4a. shows that sensor output for two wheelers is not consistent with the real number of vehicles. This may be due to sensor inability to distinguish between cycles and powered two wheelers. Where in figure 4b and 4c shows a consistent observation between our algorithm and real number of vehicle data. On average, the proposed system exhibits a 12.1% improvement factor in cluster quality metrics compared to state-of-the-art techniques. When utilizing a random RBF Generator, the performance exceeds Clustream by approximately 30%. Despite K-Mean being a superior streaming and clustering technique, our Enhanced BRICH algorithm surpasses its performance by 15%.

5. Conclusion:

This paper introduces an Enhanced BRICH algorithm for clustering in live streaming situation. This adaptive clustering technique work on the fly for Industrial Internet of Things (IIoT) applications, by considering the data drift as a primary challenge. This paper proposes a new technique to categorise the data drift based on its nature of occurrence and the PDF values, followed by the Centroid and the Turning point calculation. The clustering on the streaming data is a computationally intensive problem. For which we come up with the solution of sampling the data based on the centroid and the turning point value, this gives a better clustering compared with technique. The proposed technique works on real time data and as well in synthesized datasets. As a result, the clustering accuracy and the computational efficiency of the system has improved significantly. The proposed system is independent of data types like homogeneous and heterogeneous data. As well its can deal with multi-dimensional data where conventional algorithms fails while doing clustering on the fly. The results of the proposed system have an improvement factor of 12.1% on an average on the cluster quality metrics compared with the state of the art techniques. Comparing with Clustream the performance is about 30% more when random RBF Generator is used. Though K-Mean is better streaming and clustering technique, our proposed Enhanced BRICH algorithm overtook the performance by 15%. As a future work in the proposed framework, the author will deal with hyper-dimensional data which will be a huge challenge in the IIoT environment.

As future work we want to integrate data from vehicle such as speed at various time and location, gear position, average milage fuel consumption data to add multidimensional model data and study the data drift and clustering details.

Reference:

[1] Sisinni, E., Saifullah, A., Han,S., Jennehag,U.and Gidlund,M., 2018. Industrial internet of things: Challenges, opportunities, and directions. *IEEE transactions on industrial informatics*, *14*(11), pp.4724-4734.

[2] Malik, P.K., Sharma, R., Singh, R., Gehlot,A., Satapathy, S.C.,Alnumay, W.S., Pelusi, D., Ghosh, U. and Nayak, J., 2021. Industrial Internet of Things and its applications in industry 4.0: State of the art. *Computer Communications*, *166*, pp.125-139.

[3] Wan,J., Tang,S., Shu,Z., Li,D., Wang, S.,Imran,M. and Vasilakos,A.V.,2016.Software-defined industrial internet of things in the context of industry 4.0. *IEEE Sensors Journal*, *16*(20), pp.7373-7380.

[4] Bahga,A. and Madisetti,V.K.,2016.Block chain platform for industrial internet of things. *Journal of Software Engineering and Applications*, *9*(10), pp.533-546.

[5] Lu, J., Liu,A., Song,Y. and Zhang, G., 2020. Data-driven decision support under concept drift in streamed big data. *Complex & Intelligent Systems*, *6*(1), pp.157-163.

[6] Seraj, R. and Ahmed, M., 2020. Concept Drift for Big Data. In *Combating Security Challenges in the Age of Big Data* (pp. 29-43). Springer, Cham.

[7] Sun,H.,He,Q.,Liao,K.,Sellis,T.,Guo,L.,Zhang,X.,Shen,J.andChen,F.,2019,December.Fastanomaly detection in multiple multi-dimensional data streams. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 1218-1223). IEEE.

[8] R oy,A., 2015. A classification algorithm for high-dimensional data. *Procedia Computer Science*, *53*, pp.345-355.

[9] Subbiah,S.S. and Chinnappan,J.,2021. Opportunities and Challenges of Feature Selection Methods for High Dimensional Data: A Review. *Ingénierie des Systèmesd' Information*, *26*(1).

[10] S.Lloyd, "Least squares quantization in PCM," IEEE Trans .Inf .Theory, vol.IF-28, no.2, pp.129–137,Mar. 1982.

[11] Henning, S. and Hasselbring, W., 2019, December. Scalable and reliable multi-dimensional aggregation of sensor data streams. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 3512-3517). IEEE.

[12] P. S. Bradley, O. L. Mangasarian, and W. N. Street, "Clustering via concave minimization," in Proc. Adv. Neural Inf. Process. Syst., Denver, CO, USA, 1997, pp. 368–374.

[13] L. Xu and D. Schuurmans, "Unsupervised and

semi-supervised multi-class support vector machines," in Proc. 20th Nat. Conf.Artif. Intell., Pittsburgh, PA, USA, 2005, pp. 904–910.

[14] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms Soc. Ind. Appl. Math., New Orleans, LA, USA, 2007, pp. 1027–1035.

[15] S.Guha, N.Mishra, R.Motwani, and L.O'Callaghan, "Clustering data streams," in Proc.41stAnnu.Symp. Found. Comput. Sci., Redondo Beach, CA, USA, 2000, pp. 359–366.

[16] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in Proc. 29th Int. Conf. Very Large Data Bases, Berlin, Germany, 2003, pp. 81–92.

[17] M. R. Ackermann et al., "StreamKM++: A clustering algorithm for data streams," J. Exp. Algorithmics, vol. 17, no. 1, pp. 173–187, 2012.

[18] M.M.-T.Chiang and B. Mirkin, "Intelligent choice of the number of clusters in k-means clustering: An experimental study with different cluster spreads," J. Classif., vol. 27, no. 1, pp. 3–40, 2010.

[19] J.A. Hartigan, Clustering Algorithms. New York, NY, USA: Wiley, 1975.

[20] Sun, H., He, Q., Liao, K., Sellis, T., Guo, L., Zhang, X., Shen, J. and Chen,F.,2019, December. Fast anomaly detection in multiple multi-dimensional data streams. In 2019 IEEE International Conference on Big Data (Big Data) (pp. 1218-1223). IEEE.

[21] Borade, J. L. ., & Muddana, A. . (2023). Performance Analysis of Different Optimization Algorithms for Multi-Class Object Detection. International Journal on Recent and Innovation Trends in Computing and Communication, 11(4), 175–191. https://doi.org/10.17762/ijritcc.v11i4.6400

[22] Kamau, J., Goldberg, R., Oliveira, A., Seo-joon, C., & Nakamura, E. Improving Recommendation Systems with Collaborative Filtering Algorithms. Kuwait Journal of Machine Learning, 1(3). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/134

[23] Chang Lee, Deep Learning for Speech Recognition in Intelligent Assistants , Machine Learning Applications Conference Proceedings, Vol 1 2021.

[24] Kumar, A., Dhabliya, D., Agarwal, P., Aneja, N., Dadheech, P., Jamal, S. S., & Antwi, O. A. (2022). Cyber-internet security framework to conquer energy-related attacks on the internet of things with machine learning techniques. Computational Intelligence and Neuroscience, 2022 doi:10.1155/2022/8803586