Improved K-Means with Fuzzy-Genetic Algorithm for Outlier Detection in Multi-Dimensional Databases

Dr.C.Sumithirdevi1, M.Parthiban2, K. Manivannan3, P.Anbumani4, M.Senthil Kumar5
Department of Computer Science & Engineering,
V.S.B Engineering College, Karur, Tamilnadu, INDIA - 639111
dr.parthibanvsbce@gmail.com, sumithirdevic@yahoo.co.in

Abstract—Increasing processing and storage capacities of computer systems make it possible to record and store increasing amounts of data in an inexpensive way. Even though more data potentially contains more information, it is often difficult to interpret a large amount of collected data and to extract new and interesting knowledge. To protect all these data safely is more difficult. The term data mining is used for methods and algorithms that allow analyzing data in order to find rules and patterns describing the characteristic properties of the data. Due to significant development in information technology, larger and huge volumes of data are accumulated in databases. In order to make the most of this huge collection, well-organized and effective analysis techniques are essential that can obtain non-trivial, valid, and constructive information. Organizing data into valid groupings is one of the most basic ways of understanding and learning. Cluster analysis is the technique of grouping or clustering objects based on the measured or perceived fundamental features or similarity. The main objective of clustering is to discover structure in data and hence it is exploratory in nature. But the major risk for clustering approaches is to handle the outliers. Outliers occur due to the mechanical faults, any transformation in system behavior, fraudulent behavior, human fault, instrument mistake or any form of natural deviations. Outlier detection is a fundamental part of data mining and has huge attention from the research community recently. In this paper, the standard K-Means technique is enhanced using the Fuzzy-Genetic algorithm for effective detection and removal of outliers (EKMOD). Experiments on iris dataset reveal that EKMOD automatically detect and remove outliers, and thus help in increasing the clustering accuracy. Moreover, the Means Squared Error and execution time is very less for the proposed EKMOD. The Fuzzy controller helps to improve the performance of Genetic algorithm and it is more flexible in nature.

Keywords---K-Means, Outlier detection, Fuzzy-Genetic Algorithm, Cluster Analysis

I. INTRODUCTION

Outlier detection is one of the fundamental parts of data mining and has huge attention from the research community recently. In this paper, the standard K-Means technique is enhanced using the Fuzzy Genetic Algorithm for effective detection and removal of outliers (EKMOD). Experiments on iris dataset revealed that EKMOD automatically detect and remove the outliers that present in the clustering, and thus help in increasing the clustering accuracy. Moreover, the Means Squared Error and execution time is very less for the proposed Method.

Data mining is the technique deals with the detection of nontrivial, unseen and interesting information from several types of data. Due to the continuous growth of information technologies, there is a huge increase in the number of databases, in addition to their dimension and difficulty. A computerized technique is essential to analyze this huge amount of information [1]. The results of the analysis can be used for making a decision by a human or program.

Data clustering has been extensively utilized for the following three major purposes [2].

Underlying structure:
The Underlying Structure is used to expand insight into data, produce hypotheses, identify anomalies and recognize salient features.

Natural classification:
It is to recognize the degree of similarity among the forms or organisms that are present in the available database.

Compression:
It is a technique for organizing the data in an effective manner and summarizing the data through the cluster prototypes.

One of the fundamental difficulties in data mining is the outlier detection. Clustering is a significant tool for outlier analysis [3-5]. Outliers are a collection of objects that are significantly unrelated or irrelevant data remain in the data of database [6]. Outlier detection is a very important but a difficult one with a direct application in an extensive variety of application domains, together with fraud detection [7], recognizing computer network intrusions and bottlenecks [8], illegal activities in e-commerce and detecting mistrustful activities [9, 10].

Many data-mining approaches discover outliers as a side-product of clustering techniques. On the other hand, these approaches characterize outliers as points, which do not fit inside the clusters. As a result, the techniques unconditionally characterize outliers as the background noise in which the clusters are surrounded. Another class of techniques characterized outliers as points, which are neither a division of a cluster nor a division of the background noise; relatively they are specific points, which behave in a different way from the standard.

Clustering is a kind of unsupervised or unsubstantiated classification technique, which is used to group the data into different classes or clusters, without class label predefined. The general criterion for a good clustering is that the data objects within a cluster are similar or closely related to each other but are very dissimilar to or different from the objects in other clusters. Clustering or cluster analysis has been used in many fields, including pattern recognition, signal processing, web mining, and animal behavior analysis. Clustering can also be used for outlier detection, where the outliers are usually the data objects not falling in any cluster [11]. There are different kinds of clustering methods, namely partitioning methods, hierarchical methods, distance-based methods, density-based methods, model-based methods, kernel-based methods, neural network-based methods, and so on [11, 12].
K-means clustering algorithm [13, 14] is a kind of distance-based partitioning method with the Sum of Squared Error (SSE) criterion. K-means algorithm iteratively groups the data into k clusters, with the mean value of the data objects in each cluster representing the cluster, until there is no change in them. The k-means algorithm is very simple and can be easily implemented. It has the run time complexity of \(O(nkt)\) with \(n\) being the number of the objects to be clustered, \(k\) the number of the clusters, and \(t\) the number of iterations. But the traditional k-means algorithm has some drawbacks: it is sensitive to the outliers, which substantially influence the cluster centroids; it is not suitable for discovering the clusters with non-spherical shapes; it is not suitable for discovering the clusters of different densities or of different sizes; it is sensitive to the initialization of the centroids and cannot guarantee convergence to the global optimum.

The difficulty in outlier detection in some cases is comparable to the classification problem. For instance, the major concern of clustering-dependent outlier detection approaches is to discover clusters and outliers, which are typically considered as noise that should be eradicated with the purpose of making more consistent clustering [15]. Few noisy points possibly will be distant from the data points, while the others might be nearer. The distant noisy points would influence the result more considerably since they are more dissimilar from the data points. It is necessary to recognize and eliminate the outliers, which are distant from all the other points in cluster. Therefore, in order to enhance the clustering accuracy, a perfect clustering approach is necessary that should detect and remove those outliers.

The remainder of this paper is organized as follows. The next section presents some basic concepts and the types of Outlier detection. Section 3 provides a brief revision of mapping problems. Section 4 describes about the generic structure of fuzzy logic controller. Section 5 explains the fuzzy genetic algorithm on FLCs. Section 6 explains the methodology that going to carry out in this paper. Section 7 describes the initialization of K-means algorithm in fuzzy genetic algorithm. Section 8 shows the experimental results; finally the conclusion is drawn in section 9.

II. RELATED WORK

Outlier detection is used widely in various fields. The theme about the outlier factor of an object is unrestricted to the case of cluster. There are two kinds of outlier detection methods: formal tests and informal tests. Formal and informal tests are usually called tests of discordancy and outlier labeling methods, respectively. Most formal tests need test statistics for hypothesis testing. They are usually based on assuming some well-behaving distribution, and test if the target extreme value is an outlier of distribution, i.e., whether or not it deviates from the assumed distribution. Some tests are for single outlier and others for multiple outliers. Selection of these tests mainly depends on numbers and type of target outliers, and type of data distribution. Many various tests according to the choice of distributions are discussed by Barnett and Lewis (1994) and Iglewicz and Hoaglin (1993). Iglewicz and Hoaglin (1993) . They reviewed and compared five selected formal tests which are applicable to the normal distribution, such as the Generalized ESD, Kurtosis statistics, Shapiro-Wilk, the Boxplot rule, and the Dixon test, through simulations. The theme about the outlier factor of an object is unlimited to the case of cluster. Based on this factor of the cluster, a clustering-based outlier detection method, which is named as CBOD, is projected by Sheng-yizhong Jiang and Qing-bo An [16]. This technique constitutes two levels, the first level is cluster dataset by one-pass clustering algorithm and second level is to find out the outlier cluster by outlier factor. The time difficulty of CBOD is almost linear with the amount of dataset and the number of attributes that end in good scalability and become accustomed to huge datasets.

Eliminating the objects that are noisy is one of the major goal of data cleaning as noise delays most type of data analysis. Mostly used data cleaning techniques focus on eliminating noise that is the product of low-level data errors that results from an imperfect data collection method, but data objects which are not related or only weakly related can also considerably hold back on data analysis. Therefore, if the goal is to improve the data analysis to the extent that is possible, these objects must also be considered as noise, at least with respect to the underlying analysis. As a result, there is a need for data cleaning techniques that eliminate both types of noise. Since data sets can include huge amount of noise, these methods also need to be able to remove a potentially large fraction of the data. Xiong et al. [17] discovered four methods projected for noise removal to improve data analysis in the occurrence of high noise levels.

Three of the methods are based on usual outlier detection techniques: distance-based, clustering-based, and an approach based on the local outlier factor (LOF) of an object. The other technique, that is a new method that is projected, is a hyperclique-based data cleaner (HCleaner). These techniques are examined based on the terms of their contact on the subsequent data analysis, specially, clustering and association analysis.

The idea about outlier factor of an object is extended to the case of cluster. Outlier factor of cluster determines the difference degree of a cluster from the entire dataset and two outlier factor definitions are projected by Sheng-Yi Jiang and Ai-Min Yang [18]. A framework of clustering-based outlier detection, called as FCBOD, is suggested. This framework contains two stages, the initial stage cluster dataset and the next stage determine outlier cluster by outlier factor. The time difficulty of FCBOD is almost similar with respect to both size of dataset and the number of attributes.

III. MAPPING THE PROBLEM

a) Encoding Scheme

GAs work with the population of chromosomes, each of which can be decoded into a solution of the problem. Encoding scheme in genetic algorithm is the basis of its development, which directly affects the construction of genetic operators and performance of genetic algorithm. Real-coded scheme is used in our model. It is like a Matrix model.

b) Initial Population

An initial data is created from random selection of solutions. Note that the number of data that present is usually not equal. The data is selected at random.
c) Fitness Function

During the search procedure, each individual data is evaluated using the fitness function. It improves the effectiveness of the method. The fitness function can be defined as

\[ f(x) = \max \sum_{i=1}^{n} G_i \]  

\[ G_i = \{(Types \ of \ records \ in \ Database)X(P_{i1}X P_{i2}X \ldots \ldots, P_{iN})\} \]

\[ (P_{i1}X P_{i2}X \ldots \ldots, P_{iN}) = i^{th} \ \text{unit of exchanging Coefficients} \]

d) Selection

For selection process, two types of data are chosen at random at random from the population and they form a couple for crossover. Selection can be based on different probability distributions, such as uniform distribution or a random selection from a population where each individual data is assigned a weight dependent on its fitness of the data, so that the best individual data that has the greatest probability to be selected. We arrange in order the individual data of the population according to their fitness values. The selection probability for the individuals is a linear distribution.

e) Crossover

Crossover is usually applied to selected pairs of parents with a probability equal to a given crossover rate. Since the crossover technique is a key method in evolutionary algorithms for finding optimum combinations of solutions, it is important to implement this operation effectively for making the over all method work effectively. It mainly consists of two types of crossover

- Unbiased two-point crossover:
  - The standard procedure in evolutionary algorithms is to use uniform two-point crossover in order to create the recombinant children strings. The two-point crossover mechanism works by determining a point in the string at random called the crossover point, and exchanging the segments to the right of this point.

- Optimized Crossover:
  - The optimized crossover technique is a useful method for finding the best combinations of the features present in the two solutions. The idea is to create at least one child string from the two parent strings which is a better solution recombination than either parent. The nature of the children strings is biased in such a way that at least one of the two strings is likely to be an effective solution recombination of the parent strings.

f) Mutation

The mutation operation modifies an individual. In defining the mutation operator, we take into account the domain type of an attribute. Let us consider two important situations:

1. The individual data must be a feasible solution.
2. Those data which are not included for clustering process in the last generation, that data’s need to be exchanged into the new individual group of data.

g) End criterion

In the end criterion, the termination criterion (statistical or temporal) is not satisfied, then a return condition is to be executed as a third step; otherwise, the algorithm is terminated totally. The criterion, usually a sufficiently good fitness, or a maximum number of iterations, or the global best fitness is steady going with indeterminate iterations.

IV. GENERIC STRUCTURE OF A FUZZY LOGIC CONTROLLER

Fuzzy Logic (FL) is generally linked with the theory of fuzzy sets, a theory which relates to classes of objects with un-sharp boundaries in which membership function is a matter of degree. Fuzzy theory is essential and is applicable to many systems of consumer products like washing machines or refrigerators to big systems like trains or subways.

Recently, Fuzzy theory has been a strong tool for combining new theories (called soft computing) such as genetic algorithms or neural networks to get a wide and deep knowledge from real data. Fuzzy logic is conceptually easy to understand, tolerant of inaccurate data and flexible. Moreover this method can model non-linear functions of arbitrary complexity and it is based on natural language. Natural language has been shaped by thousands of years of human history to be convenient and efficient. Since fuzzy logic is built atop of the structures of qualitative description used in everyday language, fuzzy logic is easy to use. Fuzzy inference system (FIS) is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or pattern discerned.

Figure 1 the structure of Fuzzy Logic Controller mainly consists of
1. Fuzzification
2. Inference Engine
3. Defuzzification

A. Fuzzification

The Fuzzification comprises the process of transforming crisp values into grades of membership for linguistic terms.
of fuzzy sets. The membership function is used to associate a grade to each linguistic term.

B. Inference Engine

Using If-Then type fuzzy rules convert the fuzzy input to the fuzzy output.

Firstly, if we want to determine the Fuzzy inputs, then we should apply them to the antecedents of the fuzzy rules. If a given fuzzy rule has multiple antecedents, the fuzzy operator (AND or OR) is used to obtain a single number that represents the result of the antecedent evaluation. This number (the truth value) is then applied to the consequent membership function.

To evaluate the disjunction of the rule antecedents, operation OR fuzzy operation is employed.

Similarly, in order to evaluate the conjunction of the rule antecedents, we apply the AND fuzzy operation intersection.

The result of the antecedent evaluation can be applied to the membership function of the consequent. The most common method of correlating the rule consequent with the truth value of the rule antecedent is to cut the consequent membership function at the level of the antecedent truth. This method is called clipping. Since the top of the membership function is sliced, the clipped fuzzy set loses some information. However, clipping is still often preferred because it involves less complex and faster mathematics, and generates an aggregated output surface that is easier to defuzzify.

While clipping is a frequently used method, scaling offers a better approach for preserving the original shape of the fuzzy set. The original membership function of the rule consequent is adjusted by multiplying all its membership degrees by the truth value of the rule antecedent. This method, generally loses less information, can be useful in fuzzy expert systems.

The process of fuzzy inference involves: membership functions (MF), a curve that defines how each point in the input space is mapped to a membership value or degree of membership between 0 and 1; fuzzy logic operators (and, or, not); if-then rules. Since decisions are based on the testing of all of the rules in an FIS, the rules must be combined in some manner in order to make a decision. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. Aggregation only occurs once for each output variable, just prior to the final step, defuzzification. Due to the linguistic formulation of its rule basis, the FIS provides an optimal tool to combine more criteria among those that were above illustrated according to a reasoning that is very similar to the human one. So doing, in practical application, the knowledge of the technical expert personnel can easily be exploited by the system designer.

C. Defuzzification

The Defuzzification converts the fuzzy output of the inference engine into crisp using membership functions analogous to the ones used by the fuzzifier. Fuzziness helps us to evaluate the rules, but the final output of a fuzzy system has to be a crisp number. The input for the defuzzification process is the aggregate output fuzzy set and the output is a single number.

\[
\text{COG} = \frac{\int_{a}^{b} \mu_{A}(x) x \, dx}{\int_{a}^{b} \mu_{A}(x) \, dx}
\]  

\text{(2)}

Centroid defuzzification method finds a point representing the centre of gravity of the fuzzy set A on the interval, on the interval, ab. A reasonable estimate can be obtained by calculating it over a sample of points.

V. STRUCTURE OF A FUZZY GA BASED ON FLCS

The performance of the genetic algorithm is correlated directly with its careful selection of parameters. It is possible to destroy a high fitness individual when a large crossover probability is set. The performance of the algorithm would fluctuate significantly. For a low crossover probability, sometimes, it is hard to obtain better individuals and does not guarantee faster convergence. High mutation introduces too much diversity and takes longer time to get
the optimal solution. Low mutation tends to miss some near-optimal points. The use of fuzzy logic controllers to adapt genetic algorithm parameters is useful to improve the genetic algorithm performance [26]. An FLC is composed of a knowledge base, that includes the information given by the expert in the form of linguistic control rules, a Fuzzification interface which has the effect of transforming crisp data into fuzzy sets, an inference system that uses them together with the knowledge base to make inference by means of a reasoning method, and a defuzzification interface that translates the fuzzy control action thus obtained to a real control action using a defuzzification method. The structure of an FLC [27] is shown in Figure 1.

Applications of FLCs for parameter control of GAs are to be found in [28]. The main idea is to use an FLC whose inputs are any combination of GA performance measures or current control parameters and whose outputs are GA control parameters. Current performance measures of the GA are sent to the FLCs, which computes control parameters values that will be used by the GA. In our FGA approach, the crossover probability and the mutation probability are defined on specific individuals of the population using several FLCs that take into account fitness values of individuals and their distances. The next subsections present the design of the FLC that adapts the crossover probability Pc and the mutation probability Pm.

Our strategy for updating the crossover and mutation probabilities is to consider the changes of the maximum fitness and average fitness in the GA population of two continuous generations. The occurrence probabilities would be increased if it consistently produces a better offspring during the recombination process; however, Pc would be decreased and Pm increased when \( f_{\text{ave}}(t) \) approaches to \( f_{\text{max}}(t) \) or \( f_{\text{ave}}(t-1) \) approaches to \( f_{\text{ave}}(t) \). This scheme is based on the fact that it encourages the well-performing operators to produce more offspring, while reducing the chance for poorly performing operators to destroy the potential individuals during the recombination process. The FLC proposed has two inputs: A two-dimension FLC system is used in our GA, in which there are two parameters \( e_1 \) and \( e_2 \):

\[
e_1(t) = \frac{f_{\text{max}}(t) - f_{\text{ave}}(t)}{f_{\text{max}}(t)} \quad (3)
\]

\[
e_2(t) = \frac{f_{\text{ave}}(t) - f_{\text{ave}}(t-1)}{f_{\text{max}}(t)} \quad (4)
\]

where

- \( t \) is timestep,
- \( f_{\text{max}}(t) \) is the best fitness at Iteration \( t \),
- \( f_{\text{ave}}(t) \) is the average fitness at Iteration \( t \),
- \( f_{\text{ave}}(t) \) is the best fitness at Iteration \( t \),
- \( f_{\text{ave}}(t-1) \) is the average fitness at Iteration \( (t-1) \).

The membership functions are shown in Figure 4, where NL is Negative large, NS is Negative small, ZE is Zero, PS is Positive small, PL is Positive large. The inputs of the mutation FLC are the same as those of the crossover FLC. But the membership function for \( \Delta P_m(t) \) was scaled by

![Membership Function](image.png)
Table 1: Fuzzy rules for crossover operation \((\Delta P_c(t))\)

<table>
<thead>
<tr>
<th>(c_1)</th>
<th>NL</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>PL</td>
<td>NS</td>
<td>ZE</td>
<td>NS</td>
<td>PS</td>
<td>PL</td>
</tr>
<tr>
<td>PS</td>
<td>ZE</td>
<td>ZE</td>
<td>NL</td>
<td>ZE</td>
<td>ZE</td>
</tr>
<tr>
<td>ZE</td>
<td>NS</td>
<td>NL</td>
<td>NL</td>
<td>NL</td>
<td>NL</td>
</tr>
</tbody>
</table>

Table 2: Fuzzy Rules for Mutation Operation \((\Delta P_m(t))\)

<table>
<thead>
<tr>
<th>(c_1)</th>
<th>NL</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>PL</td>
<td>PS</td>
<td>ZE</td>
<td>PS</td>
<td>NS</td>
<td>NL</td>
</tr>
<tr>
<td>PS</td>
<td>ZE</td>
<td>ZE</td>
<td>PL</td>
<td>ZE</td>
<td>ZE</td>
</tr>
<tr>
<td>ZE</td>
<td>PS</td>
<td>PL</td>
<td>PL</td>
<td>PL</td>
<td>PL</td>
</tr>
</tbody>
</table>

10%. Fuzzy rules describe the relation between the inputs and outputs. Tables 1 and 2 show the Rule-Base used by the FLCS presented. For the parameter control in our GAs, the outputs \(\Delta P_c(t)\) and \(\Delta P_m(t)\) of fuzzy logic controllers are translated the fuzzy control action thus obtained to a real control action. Center of gravity [29] is used as our defuzzification method. Then we use the crisp value to modify the parameters \(P_c\) and \(P_m\) as follows:

\[
P_c(t) = P_c(t-1) + \Delta P_c(t) \tag{5}
\]

\[
P_m(t) = P_m(t-1) + \Delta P_m(t) \tag{6}
\]

VI. METHODOLOGY

Clustering is a basic method used to detect potential outliers. From the viewpoint of a clustering algorithm, potential outliers are the data which are not located in any cluster. Furthermore, if a cluster significantly differs from other clusters, the objects in this cluster might be outliers. A clustering algorithm should satisfy three important requirements [19]

- Discovery of clusters with an arbitrary shape
- Good efficiency on large databases
- Some heuristics to determine the input parameters.

The main purpose of clustering algorithm modifications is to improve the performance of the underlying algorithms by fixing their weaknesses. Because randomness is one of the techniques used in initializing many of clustering techniques, and giving each point an equal opportunity to be an initial one, it is considered the main point of weakness that has to be solved. However, because of the sensitivity of K-Means to its initial points, which is considered very high, we have to make them as near to global minima as possible in order to improve the clustering performance. [20, 21]

Clustering approaches can be largely segmented into two divisions: hierarchical and partitional. Hierarchical clustering approaches recursively discover nested clusters either in agglomerative mode (initializing with each data point in its individual cluster and integrating the most similar pair of clusters consecutively to generate a cluster hierarchy) or in divisive (topdown) mode (initializing with all the data points in one cluster and recursively dividing each cluster into smaller clusters). Dissimilar to hierarchical clustering approach, partitional clustering approaches discover all the clusters at the same time as a partition of the data and do not enforce a hierarchical structure [2]. The most recognized hierarchical approaches are single-link and complete-link; the extensively used and the simplest partitional approach is K-Means. Partitional approaches widely used in pattern recognition owing to its nature of available data in the data base. K-Means has a wealthy and diverse history as it was separately discovered in several scientific fields.

A. K-Means Algorithm

Consider \(X = \{x_i\}, i = 1, ..., n\) is a set of \(n\) d-dimensional points to be clustered into a set of \(K\) clusters, \(C = \{c_k, k = 1, ..., K\}\). K-Means algorithm discovers a partition of the data which are not located in any cluster. The squared error between the empirical mean of a cluster and the points in the cluster is reduced. Consider \(\mathbb{E}_k\) be the mean of cluster \(c_k\). The squared error between \(\mathbb{E}_k\) and the points in cluster \(c_k\) is given as

\[
J(c_k) = \sum_{x_i \in c_k} ||x_i - \mathbb{E}_k||^2 \tag{7}
\]

The main objective of K-Means is to reduce the sum of the squared error over all \(K\) clusters,

\[
J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} ||x_i - \mathbb{E}_k||^2 \tag{8}
\]

Reducing this objective function is recognized to be an NP-hard problem (even for \(K = 2\)) [15]. As a result K-Means, which is a greedy algorithm, can only converge to a local minimum, although current study has shown with a large probability K-Means could converge to the global optimum when clusters are well separated [16]. K-Means begins with an initial partition with \(K\) clusters and allocate patterns to clusters in an attempt to lessen the squared error. While the squared error constantly decrease with an increase in the number of clusters \(K\) (with \(J(C) = 0\) when \(K = n\)), it can be reduced only for a constant number of clusters. The major steps of K-Means algorithm are as follows.

- Choose an initial partition with \(K\) clusters; reiterate steps 2 and 3 until cluster membership becomes constant.
- Produce a new partition by assigning each pattern to its closest cluster center.
- Generate new cluster centers.

Features of the data streams include their huge volume and potentially unstructured size, sequential access and dynamically evolving nature. This enforces further necessities to conventional clustering approaches to quickly process and sum up the enormous amount of constantly arriving data. It also necessitates the capability of adapting to changes in the data distribution, the capability of detecting emerging clusters and differentiates them from outliers in the data and the capability of incorporating old clusters or remove expired ones. All of these necessities
make data stream clustering a considerable challenge. Hence in order to detect and remove the outliers, K-Means is enhanced by integrating it with Fuzzy-Genetic Algorithm.

VII. INITIALIZATION OF K-MEANS USING Fuzzy-GENETIC ALGORITHM (IKMFGA)

In this section, proposed Initialized K-Means using Fuzzy-Genetic Algorithms (IKMFGA), which is efficient and has the potential to identify and remove the outliers. The design genetic algorithms help to solve the problem outlined above.

The fuzzy logic improves the performance of the algorithm, flexible in nature and it improves the clustering accuracy in an efficient manner. However, one of its features is a tendency for all of the population to converge to a single solution which is suboptimal. If all the members of the population are very similar, the crossover operator has little function and mutation turns out to be the primary operator. This effect is known as premature convergence [24].

Adaptive Fuzzy genetic algorithms, which dynamically adapt selected control parameters or genetic operators during the evolution, have been built to avoid the premature convergence problem and improve GA behaviour. One of the adaptive approaches is the parameter setting techniques based on the use of fuzzy logic controllers (FLCs), the fuzzy genetic algorithm (FGA) [25].

In this section, proposed Initialized K-Means using Fuzzy Genetic Algorithms (IKMFGA), which is efficient and has the potential to identify and remove the outliers. In Figure 5, t represents the generation number, and P stands for population and the k is number of identified outlier in the clusters.

1) The fuzzy genetic algorithm is begin. First the k is initialized next the collection of records is stored in a file on the disk and each record t is read in sequence. Then the population is initialized by coding it into a specific type of representation (i.e. binary, decimal, float, etc) then assigned to a cluster.

During the initialization phase of the Fuzzy genetic algorithm, each record is represented as non-outlier and hash tables for attributes are also built and updated.

2) Then the second step is to check while condition and the iteration process is carried out. During the process Fitness is calculated in the evaluation step. While the termination condition is not met, which might be number of generations or a specific fitness threshold, the processes of selection, recombination, mutations and fitness calculations are done. The selection process chooses individuals from population for the process of crossover. Crossover is done by exchanging a part between the chosen individual data, which is dependent on the type of crossover.

3) Mutation is done after that by replacing a few points among randomly chosen individual data. Then call for the fuzzy logic controller, it transferred to fuzzy control action then the crisp value is used to modify the parameters.

4) Then fitness value has to be recalculated to be the basis for the next cycle. Hence the condition is satisfied the process came to an end.

In the genetic procedure, the dataset is scanned for k times to discover exact k outliers, that is, one outlier is found and removed in each pass. In every scan over dataset, read each record t that is represented as non-outlier, its label is changed to outlier and the changed entropy value is calculated. A record that accomplishes maximal entropy impact is chosen as outlier in current scan and accumulated to the set of outliers. From the following theorem the entropy value which depends upon the attribute value of the record and the record is eliminated completely, usually the FGA method performs significantly better than the other methods.

Algorithm:

Begin FGA
K=No. of identified Outliers
\( t = 0 \) for each record
Initialize \( P(t) \)
Evaluate \( P(t) \)
while (the end criterion is not met)
do
\( t = t + 1 \)
Select \( P(t) \) from Crossover \( P_i(t) \)
Mutate \( P_m(t) \)
Call fuzzy logic controller \( (e_1, e_2) \)
Update according to equations (5) and (6)
Evaluate \( P(t)=k \)
End while
End FGA

Fig 5: Algorithm of Initialization of K-means using Fuzzy Genetic Algorithm

VIII. EXPERIMENTAL RESULTS

To evaluate the Initialized K-Means using Fuzzy-Genetic Algorithms, experiments were carried out using University of California, Irvine (UCI) Machine Learning Repository [30]. For the purpose of evaluating the proposed technique iris dataset [31] is used and the results are compared with standard K-Means and Fuzzy-Genetic Algorithm initializes K-means FGA IK [32].

Clustering results are generated using Standard K-Means, Fuzzy-Genetic Algorithm and the proposed IKMFGA for the iris dataset.

The performance of the proposed IKMFGA scheme is evaluated against the Standard K-Means, FGA IK based on the following parameters.

- Clustering Accuracy,
- Mean Squared Error and
- Execution Time

A. Clustering Accuracy

To compute the purity of the clusters, each cluster is assigned to the class which is more frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned documents and dividing by N.

Since the outliers are detected and removed using the proposed IKMFGA, clustering accuracy is drastically increased. Table 4.1 shows the comparison of the accuracy of clustering accuracy for the proposed method with the standard K-Means and FGA IK method.
Table 3
Comparison of Clustering Accuracy in Iris Dataset

<table>
<thead>
<tr>
<th>Clustering Technique</th>
<th>Clustering Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard K-Means</td>
<td>89.80</td>
</tr>
<tr>
<td>GAIK</td>
<td>94.25</td>
</tr>
<tr>
<td>IKMTGA</td>
<td>98.23</td>
</tr>
<tr>
<td>IKMFGA</td>
<td>99.02</td>
</tr>
</tbody>
</table>

Figure 6: Comparison of Clustering Accuracy in Iris Dataset

From the figure 6, it can be observed that the accuracy of clustering result using standard K-Means is 89.80%, GAIK is 94.25% and IKMTGA method is 98.23% and the proposed method IKMFGA is 99.02% for iris dataset.

B. Mean Squared Error (MSE)

As mentioned above the formula for MSE is

\[ J(C) = \sum_{k=1}^{K} \sum_{x \in C_k} \| x - \mu_k \|^2 \]  

(5)

MSE of the iris dataset for the two cluster centers of the three methods are provided in table 4.

Table 4
Comparison of Mean Squared Error in Iris Dataset

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Standard K-Means</th>
<th>GAIK</th>
<th>IKMTGA</th>
<th>IKMFGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 1</td>
<td>0.6923</td>
<td>0.6010</td>
<td>0.4314</td>
<td>0.3312</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>0.5256</td>
<td>0.4699</td>
<td>0.3025</td>
<td>0.2469</td>
</tr>
</tbody>
</table>

Figure 7: Comparison of Mean Squared Error in Iris Dataset

From figure 7, it is observed that the proposed IKMFGA gives very low MSE values for both the clusters than the existing methods like Standard K-Means, GAIK and IKMTGA.

C. Execution Time

The execution time is calculated based on the machine time (i.e., the time taken by the machine to run the proposed algorithm). The fuzzy genetic algorithm is the fastest implementation of the algorithm which improves the performance of the algorithm in efficient way.

Table 5
Comparison of Execution Time in Iris Dataset

<table>
<thead>
<tr>
<th>Clustering Technique</th>
<th>Execution Time (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard K-Means</td>
<td>2.31</td>
</tr>
<tr>
<td>GAIK</td>
<td>1.40</td>
</tr>
<tr>
<td>IKMTGA</td>
<td>0.90</td>
</tr>
<tr>
<td>IKMFGA</td>
<td>0.70</td>
</tr>
</tbody>
</table>

Table 5 shows the execution time taken by the Standard K-Means, GAIK, IKMTGA and the proposed IKMFGA in iris dataset. It can be observed that the time required for execution using the proposed IKMFGA scheme for iris dataset is 0.70 seconds, whereas more time is needed by other three clustering techniques for execution.

From figure 8, it is observed that the proposed IKMFGA takes very low execution time when compared with the existing methods like Standard K-Means, GAIK and IKMTGA which takes 2.31, 1.40 and 0.90 seconds respectively in iris dataset.
IX. CONCLUSION

The Fuzzy-Genetic algorithm is Fuzzy logic based controllers are applied to fine-tune with dynamism the crossover and mutation probability in the genetic algorithms, in an attempt to improve the algorithm performance. Here fuzzy logic based controllers are used to adapt the parameters of genetic algorithms, thereby improving their performance. The standard genetic algorithm converges quickly with a larger probability to get trapped in local optima, while the fuzzy genetic algorithm spends more time to explore more feasible solutions with a larger probability to find global optimal solutions. Empirical results reveal that the proposed FPGA method for the Outlier Detection is more efficient when compared to all other approaches. The major concern in this clustering approach is the detection and removal of outliers. Outlier detection is an essential subject in data mining, particularly it has been extensively utilized to identify and eliminate anomalous or irrelevant objects from data cluster. This paper proposes an Initialized K-Means using Fuzzy-Genetic Algorithms (IKMFGA), which uses Fuzzy genetic algorithm to identify and remove the outliers in the clusters. The effectiveness of the proposed approach is tested using the iris dataset based on the clustering accuracy, MSE and execution time. From the results, it is revealed that the proposed IKMFGA provides the very accurate cluster results with low MSE. Moreover the execution time of this approach is very low when compared to other three clustering approaches.

REFERENCES


[34] Iglewicz, B., Hoaglin, D. How to detect and handle outliers. ASQC Quality Press, 1993.
Acuna, E., Rodriguez, C. A Meta analysis study of outlier detection methods in classification. Technical paper, Department of Mathematics, University of Puerto Rico at Mayaguez, 2004