CHAPTER 5

GRAPH BASED SEMI-SUPERVISED CLUSTERING WITH BAT ALGORITHM (GSSBAT) FOR OUTLIER DETECTION IN LARGE-SCALE DATA

5.1. INTRODUCTION

The problem concerned with outlier detection has been gaining focus recently owing to its importance in addressing different practical challenges. Outliers are often considered as noise, which requires to be eliminated from a dataset for the success of a particular model or algorithm. Outliers are patterns in data, which do not adhere to a normal behavior, or conform to an outlying behavior.

However in the most of the clustering methods, outlier detection is now described and formulated to be a problem of optimization. With a given $o$, the number of probable candidate sets for the objective function is $C_n^O = \frac{n!}{o!(n-o)!}$, which is extremely high. In addition, the optimal value of $o$ might have to be determined, i.e., how many number of outliers are presented in the data set becomes a difficult task. A probable theoretical mechanism to solve the problem is to look out for a range of outlier values. For this purpose in the recent work develops a Particle Swarm Optimization (PSO). The major disadvantages of PSO algorithm are that it easily falls into local optimum in high-dimensional space and its convergence rate is less during the iterative process. To solve this problem Bat Algorithm is introduced in this work.

However missing values is also another common problem to each of the data of the actual world. The existence of missing values in data leads to datasets, which Galaret et al., (2013) calls as “incomplete” datasets due to the unavailability of some information. Data pre-processing methods are used on raw data to render the data clear, free of noise and with consistency. Data Normalization does the standardization of the raw data by modifying them into particular range employing linear transformation that can create good quality clusters and enhance the accuracy of clustering algorithms.
This research work presents a novel outlier factor function, which is obtained from the weighted entropy and shows that calculation/updating of the outlier factor could be carried out with no necessity for estimating the joint probability distribution. Here the BAT optimization is inclusive of n number of data samples N which move around a D-dimensional search space for the optimization of a particular variational property. Moreover, an upper bound of the outliers for reducing the search space is also estimated. Depending on the best objective function values, the outliers are estimated for the given dataset. The proposed performance of GSSBAT is measured with regard to the DR, FAR, time comparison between the numbers of attributes, number of data objects

5.2. PREPROCESSING

Data Pre-processing is a very essential step as it can influence the result derived from a clustering algorithm. This module computes the tuples with missing values employing various options such as maximum, minimum, constant, average and standard deviation for treating missing values tuples prior to the application of normalization approach over the dataset. This procedure provides the treatment over the missing value data and then it uses it over the second portion (data normalization) of the data preparation.

5.2.1. Normalization Approach

Data Mining can produce efficient result when correct and resourceful data mining approach is applied to the dataset. As per the authors Visalakshi and Thangavel (2009), normalization is utilized in order to standardize every feature of the dataset into a particular predetermined criterion such that repetitive or noisy objects can be removed and valid and robust data are used that can effectively enhance the accuracy of the result.

Data normalization is an important step to find missing values in the samples. The significance of normalization is that it improves the accuracy of the results, which are got during the time of clustering De Souto et al., (2008). The results are better when data preparation is done with data preprocessing and normalization is
performed with diverse methodologies. Data normalizations methodologies consists of min-max normalization, Z-Score normalization. No globally defined rule exists for the normalization of datasets and therefore the selection of a specific normalization rule remains largely with the user’s discretion.

Among them three methods, Min-Max normalization technique is involved with the linear transformation over raw data. Min$_A$ and Max$_A$ stand for the minimum and maximum value for the attribute A. This method does the mapping of the value of attribute A into the range of [0, 1].

This data processing method is used for mining earlier, can considerably increase the overall quality of the patterns that are mined and/or the time needed for the real mining.

5.3. IMBALANCED DATASET METHODS

Other important problems of outlier detection with the rising complexity, imbalanced dataset, and diverse datasets, are concerned with catching similar kind of outliers totally as a group, and evaluating the outliers. To solve imbalanced dataset problem, traditional imbalanced methods is used in this research work.

Approaches are classically segregated into four groups: i) algorithmic level, ii) data level, iii) cost-sensitive techniques and iv) aggregates of classifiers.

5.3.1. Algorithmic level

Algorithmic level approaches force the classifier to approach a decision threshold that is hooked onto an accurate classification of the minority class by strategies such as adjusting the weights for every class. For example, in Chawla et al., (2002) a weighted Euclidean distance function was utilized for the classification of the samples employing k-nearest neighbors (k-NN). In a similar manner, a Support Vector Machine (SVM) with a kernel function Wu and Chang (2003) biased to the minority class to enhance the prediction of the minority class.
5.3.2. Cost-sensitive approaches

Cost-sensitive approaches assign diverse costs for training the examples belonging the majority and the minority classes Domingos (1999). Nonetheless, it is hard to fix the cost in a proper manner (can be performed by numerous ways) and might rely on the features of the data sets. The standardized public classification data sets do not have the costs Galaret et al., (2013) and there is a huge possibility of over-training, while searching to get the most suitable cost.

5.3.3. Re-sampling

Another approach is the re-sampling of the data with the purpose of dealing with the challenges caused due to the irregular characteristics of data. This data level approach does not change the already available classifiers and is used in the form of a pre-processing strategy before training a classifier. The data set could be resampled by the oversampling of the minority class, and/or under sampling of the majority class. Although it might be advantageous being independent of the classifier, it is generally difficult to decide about the optimal re-sampling ratio without manual intervention. Moreover, it may be challenging to oversample the minority classes and still maintain the distribution to be the same, particularly in realistic applications in which there is a greater probability of overlaps between minority and majority classes. In the same way, when under-sampling the majority class, it is mostly tedious to maintain the new distribution of the majority class to be identical as the distribution from which it is sub-sampled.

5.3.4. Ensembles of classifiers

These classifiers have been popular in the last decade Galaret et al., (2012). There are two important techniques; bagging and boosting. Bagging comprises of a variety of classifiers that are employed over the data subsets. On the other hand, in boosting, the entire set is utilized in order to train the classifiers in every iteration whereas much attention is bestowed on the classification of the samples, which had been misclassified in the earlier iteration. This is carried out by adjusting the weights
toward their right classification. The most popular boosting technique includes Ada Boost Schapire and Freund (1997). Although ensembles are often utilized for classifying the irregular data sets, they are not capable of handling the imbalanced data sets all by themselves. Also, they need one or more approaches combined, including the ones mentioned above consisting of re-sampling data (SMOTE Boost Chawla et al., (2003), EUS Boost Galar et al., (2013) etc.). from these types SMOTE algorithm is used in this work.

**SMOTE algorithm**

An over-sampling approach is proposed where the minority class is over-sampled by generating “synthetic” examples instead of over-sampling with a replacement. Additional training data was created by carrying out specific operations on actual data. In this case, operations such as rotation and skew were the common means of perturbing the training data. Synthetic examples are generated in a manner that is application-specific, by working in “feature space” instead of “data space”. The minority class is then over-sampled by considering every minority class sample and presenting the man-made examples along the line segments that join any/all of the k minority class closes to neighbors. Based on the amount of over-sampling necessary, neighbours located in the k nearest neighbors are selected in random.

Synthetic samples are created in the following method, which is actually finding the difference between the feature vectors (sample) considered and its closest neighbor. This difference is multiplied by a random number between 0 and 1, and then it is added to the given feature vector. The result is a random point being chosen along the line segment between two particular characteristics. In this work propose the use of k-Nearest Neighbour (kNN) algorithm for estimating and substituting the imbalanced dataset samples. The significant advantages of this technique are:
kNN can do the prediction of both discrete attributes (the most frequent value among the k closest neighbours) and continuous attributes (the mean among the k closest neighbours);

It is not required to create a predictive model for every attribute with imbalanced dataset samples. In fact, the k-nearest neighbour does not generate the models when the training examples are exploited to act as model. Therefore, the kNN can be adjusted with ease to function as class with any attribute, by just changing which attributes will be taken into consideration in the distance metric. Moreover, this technique can be convenient in the treatment of examples with different imbalanced values.

5.4. BAT ALGORITHM

Heuristic optimization algorithms, generally influenced by the social behavior of animals, do not give the assurance to get the best or global solution. But these algorithms always try to get better solutions Yang (2009). In the field of heuristics, two components exists posing indispensability, exploration (also referred to as diversification) and exploitation (also referred to as intensification). Exploration indicates global search capability whereas exploitation refers to local search capability of algorithm. An algorithm’s success is always dependent on how well balanced these components are. Too less exploration but too much of exploitation may result in premature convergence, and contrastingly, too much of exploration but too little exploitation may lead to issues in algorithm convergence towards the optimal solutions Yang (2010); Gao and Liu (2012). BA, first introduced by Yang (2010), is one among the heuristic algorithms. A mechanism is observed in bats referred to as echolocation that guides them during hunting expeditions. The capability echolocation of bats is interesting, since these bats can locate their prey and differentiate between various kinds of insects even in pitch black darkness. BA is typically an algorithm influenced by the characteristic of echolocation of bats Yang (2010).

Echolocation is generally sonar that is used by bats to find the prey and to prevent obstructions. These bats cry very loudly and listen to the echo, which bounces
back from the objects in the surrounding Yang (2010). This way, a bat can calculate the distance between it and the object. Moreover, bats can differentiate between an obstruction and a prey even in conditions of pitch black darkness Nakamura et al., (2012).

With the aim of transforming these behaviours of bats onto algorithm, Yang formulated few rules Komarasamy and Wahi (2012).

1) All the bats make use of echolocation for sensing distance, and they also “understand” the difference between food/prey and obstacles by some magical instinct; Bats fly in random with velocity $v_i$ at position $x_i$ with a frequency $f_{\text{min}}$, differing wavelength and loudness $A_0$ to look out for prey. They can automatically adapt to the wavelength (or frequency) of their uttered pulses and then adjust the rate of pulse emission $r \in [0, 1]$, based on the proximity with their target.

2) Even though the loudness can differ in several manners, it is assumed that the loudness vary from a huge (positive) $A_0$ to a constant minimum value $A_{\text{min}}$.

The detailed description of the proposed Bat Algorithm is discussed in the proposed work.

5.5. SEMISUPERVISED CLUSTERING ALGORITHM

In several practical learning domains (e.g. text processing, bioinformatics), there is huge availability of unlabeled data but less amount of labeled data that can be quite costly for generation. As a result, semi-supervised learning, which is the learning from a combination involving both labeled and unlabeled data, has emerged to be a field of research interest recently.

Semi-supervised clustering, which uses class labels or pairwise constraints on few examples to help in unsupervised clustering, has remained in the limelight for various research projects. In case the initial labeled data denotes all of the relevant categories, then both the concepts semi-supervised clustering and semi-supervised
classification algorithms can be utilized for the purpose of categorization. But in several domains, information about the relevant categories lacks completeness.

Recently, Semi-supervised clustering algorithms have gained substantial attention within the machine learning and data mining communities. In classical clustering algorithms, just unlabeled data is brought into use for generating the clustering’s; in the case of semi-supervised clustering; the aim is to include previous information regarding the clusters into the algorithm so as to enhance the results of clustering. Several works in the recent times have explored on this challenge Bar-Hillel et al., (2003); Basu et al., (2004).

Among the existing techniques, graph-based semi-supervised clustering is an attractive approach owing to its low computation complexity and flexibility in practice. In general, a graph-based semi-supervised clustering consists of two key steps. First, a graph is built from all data samples including both labeled and unlabeled samples to model the relationships among the points. Then, label information of the labeled samples is propagated to the unlabeled samples over the graph. Though different graph-based Semi-supervised clustering techniques formulate the label propagation process via different objective functions, all of them share one common assumption, that is, points present on the same structure like a cluster, a subspace, or a manifold are expected to have the same label. As one generally does not possess an explicit model for the underlying structures, a graph constructed from the data samples often serves as an approximation to it. Therefore, constructing a good graph, which best captures the important data structure is crucial for every graph-based semi-supervised clustering algorithms.

5.6. PROPOSED GRAPH BASED SEMI-SUPERVISED CLUSTERING WITH BAT ALGORITHM (GSSBAT) METHODOLOGY

In the proposed system, the categorical, numerical and mixed data are evaluated by using the GSSBAT algorithm more effectively. In this section, the means by which entropy, Shannon, Jensen-Shannon Divergence (JSD) and total correlation
could be exploited for capturing the likelihood of outlier candidates is looked at. In this research, the unbalanced dataset is handled and the outliers are detected optimally. Moreover, an upper bound of the outliers for reducing the search space is also estimated. Depending on the best objective function values, the outliers are estimated for the given dataset. The proposed performance of GSSBAT is measured with regard to the DR, FAR, time comparison between the numbers of attributes, number of data objects, Normalized Mean Square Error (NMSE) for the comparison of error results, Area Under the Curve (AUC). It indicates that the proposed GSSBAT have lesser NMSE error, FAR, and more Detection Rate (DR) with lesser time consumed for completing the process.

5.6.1. Preprocessing Using Min-Max Normalization

The preprocessing is an extremely significant step as it can improve the result obtained from a clustering algorithm. This module computes the tuples with missing values employing various options such as maximum, minimum, constant, average and standard deviation for treating the missing values tuples before applying the normalization approach onto the dataset. This process treats the missing value data and afterwards it is applied to the process of data normalization of data preparation. Normalization is employed for the standardization of all the features of the dataset into a particular predetermined criterion such that repetitive or noisy objects can be removed and valid and acceptable data is used that can enhance the accuracy of the result obtained. Data normalization is a critical step that aids in preventing larger features from randomized value to the specific range. The significance of normalization is that it improves the accuracy of the results, which are achieved during the process of clustering Jain and Bhandare (2011).

For the unbalanced dataset, the min-max normalization method is employed for identifying the missing values effectively. Min-Max normalization is a simple strategy in which the method can effectively fit the data within a pre-determined boundary. It carries out a linear transformation on the actual data. Min-max normalization does the mapping of a value \( d \) of \( P \) to \( d' \) within the range
[new_min(p), new_max(p)]. The min-max normalization is computed by the following expression:

\[ d' = \frac{[d - \min(p)] \times [\text{new}_{\max(p)} - \text{new}_{\min(p)}]}{[\max(p) - \min(p)]]} \] (5.1)

Where \( \min(p) = \text{minimum value of attribute} \)

\( \max(p) = \text{maximum value of attribute} \)

By using the above formula (5.1), the missing values are identified efficiently. It is used to list out the outliers from the given dataset.

This study regarded outlier detection to be hugely overlapped unbalanced data clustering issue; in which original samples largely outnumber the outlier samples. Generally, the clustering algorithms display a not-so-great performance when working with unbalanced datasets and the results are bent towards majority class. For this kind of problems, the time, error and cost corresponding to the outlier sample gets predicted. An unbalanced data set is defined to be a data set, where one class of data highly outnumbers the other class of data. In order to predict the class of a data record in the data set, classification technique can be brought into use. It learns the model from historical data that is already labeled. It employs the learned models for the prediction of the class of unobserved or unknown data.

5.6.2. Synthetic Minority Over-sampling Technique (SMOTE) for sampling

This research Synthetic Minority Over-sampling Technique (SMOTE) on the complete class samples, outlier regions might not be defined well due to sparsely positioned outlier samples. Generation of synthetic samples blindly along with the sparse samples are resulting in a greater possibility of class mix. Therefore the minority samples may not be detected well. In order to prevent this class mix in the training data distribution, this research employs extreme outlier elimination from the minority class by utilizing \( k \) Nearest Neighbor (kNN) concept as a data cleaning technique. The \( k \)NNs cardinality value denotes if the point is positioned in a sparse
region or a dense region. It is referred to as the points, which are located very sparsely are called extreme outliers. By removing the extreme outliers, it is neglecting the points, which are far from the minority decision boundary for performing SMOTE Padmaja et al., (2007).

The generation of the synthetic samples is performed in the following manner: The difference between the sample considered and its closest neighbour is taken. This difference is multiplied by a random number between 0 and 1, and then added to the feature vector taken. This results in selecting a random point along the line segment between two particular features. This technique potentially pushes the decision region of the minority dataset samples with class to be more generalized. The amount of over-sampling is a parameter of the system, and thereafter a series of ROC curves can be created for diverse populations and ROC analysis is carried out.

Assume the data be referred to as the \(X\) comprising the number of the data objects as \(n(x_1, \ldots, x_n)\) every \(x_i\) for \(1 < i < n\) being a vector of the categorical attributes \([y_1, y_2, \ldots, y_m]^T\), where \(m\) refers to the number of categorical and discrete data attributes, \(y_j\) indicates the value of the attribute which belongs to either a categorical and discrete value. For the purpose of measuring the attribute value importance, the Shannon, JSD and the holoentropy of the attribute which is discussed in detail in the previous chapter 4 in section 4.4.2.

Therefore, the formulation of the outlier detection is now expressed to be a problem of optimization. Considering a provided \(O\), the number of probable candidate sets for the objective function is \(C_n^O = \frac{n!}{O!(n-O)!}\), that is extremely huge. In addition, one may need to decide the optimal value of \(O\), i.e., the number of outliers a data set actually possesses. A probable theoretical approach to this issue is searching for a range of values of \(O\) and then deciding over an optimal value of \(O\) through the optimization of a particular variational property of \(J_x(Y, O)\). Assume this as a research focus proposed in this research work. At present, focus will be on the development of practical solutions for the optimization issue.
5.6.3. Bat Algorithm

Bat Algorithm is influenced by the characteristic of echolocation in bats. Echolocation is generally sonar that bats exploit for the detection of prey and to prevent barriers. The sound of these bats is very loud and they listen to the echo, which reverberates back from the objects in the surrounding Yang (2010). This way, a bat can calculate on how distant they are located from an object. Moreover, the bats can distinguish between a barrier and a prey even in pitch black darkness Nakamura et al., (2012). Any bat flies randomly with velocity $V_i = [v_{i1} ... v_{id}]$ and pulse frequency $freq_i \in [freq_{min}, freq_{max}]$ at $P_i = [p_{i1} ... p_{id}]$, varying the rate of pulse emission $rate_i \in [rate_{min}, rate_{max}]$ and loudness $L_i \in [L_{min}, L_{max}]$.

If the food is closer, the $rate_i$ will be bigger and the $L_i$ will be lower.

The rules of the updating of a bat are shown as follows

$$freq_i = freq_{min} + (freq_{max} - freq_{min}) \rho_i \quad (5.2)$$
$$V_i = V_i + (P_i - P_*)freq_i \quad (5.3)$$
$$P_i = P_i + V_i \quad (5.4)$$

Where $\rho_i \in [0,1]$ refers to a random value according to a uniform distribution. $P_*$ indicates the current global best solution among all of the $nbats$, which is chosen through the fitness function $F(P_i)$. The pseudo code of Bat Algorithm 5.1 could be summarized as follows

Algorithm 5.1. Bat Algorithm

**Input:** The fitness function $F$

**Output:** The best solution

**Method**

1. Initialize the bat population $P_i (i = 1, 2, ... n)$ and $V_i$
2. Set pulse frequency $rate_i$ at $P_i$
3. Initialize pulse rates \( r_{ae} \) and the loudness \( L_i \)
4. For \( 1 \rightarrow \text{iter} \)
5. Generate new solutions by changing frequency, and updating velocities and positions by using (5.2)-(5.4)
6. If \( rand < r_{ae} \) then
7. Choose a solution among the best solutions
8. Generate a position around the selected best one
9. End if
10. Generate a new solution by flying in random
11. If \( rand < L_i \& F(P_i) > F(P_*) \)
12. Get the new solutions
13. Increase \( r_{ae} \) and decrease \( L_i \)
14. End if
15. Rank the bats and get the current best \( P_* \)
16. End for
17. Return \( P_* \)

Bat consist of \( n \) number of data samples \( N \) which move around a \( d \)-dimensional search space for the optimization of a particular variational property of \( J_X(Y, O) \). The process of BAT starts with a population that contains a number of the data objects as \( n(x_1, \ldots, x_n) \) with every \( x_i \) with \( r_1 \ldots r_i \) refer to the attribute numbers selected from 1 to \( m \) for each data sample and the optimization appropriately next searches for the best range of values for \( O \) through continuously updating the generations. The location of the \( i^{th} \) data samples of cluster bats can be referred to by \( l = (l_1, \ldots, l_d) \). The velocity with respect to the \( i^{th} \) cluster of data points can be denoted as \( v_i = (v_{i1}, v_{i2}, \ldots, v_{iD}) \). The velocities corresponding to the data points in the cluster are restricted within \([V_{min}, V_{max}]^D \) respectively. The frequency rules are updated and global best solution is selected among several bats. By using the best fitness function value, the best solution is ranked.
At each generation, the position and also the velocity of each $i^{th}$ data point in the cluster gets revised by current best flies in bats. It occurs in the space which data point’s discrete problem, with the intent of resolving this issue, bat which is used for discrete binary variables. In binary space, a bat which is a data point in the cluster probably will get moved to the nearly corners of a hypercube through the flipping of multiple numbers of bits; consequently, the bat velocity on the whole may be defined by the number of bits modified according to the number of processes. In each step, the pulse rate and loudness is updated using best bat values. Hence the outlier in the given dataset is recognized more optimally.

In this research $\rho$ refers to the random value for the optimization of a particular dataset samples and random numbers within $(0,1)$. Velocities $V_i$ and $F(P_i)$ refer to the current best and old velocities for outliers. $P_i$ indicates the global current best position, and Increase rate $i$ and reduce $L_i$, updated outlier detection position. In Equation (5.2) (5.3), outlier detection position velocities and frequency of each dataset sample are tried to be at a maximum velocity and maximum frequency. Thus it is more suitable for the outlier dataset compare than preceding algorithms.

5.6.4. Graph Based Semi-Supervised Clustering With Bat Algorithm (GSSBAT)

In this research, to improve the clustering result, the graph-based semi-supervised clustering algorithm is employed that, in turn, uses the Gaussian random field to perform semi-supervised learning, where the mean of the field is characterized with regard to harmonic functions. The technique comprises of two important steps: graph construction and classification. It describes about the two steps of the algorithm in detail:

Let $\chi = \{x_1, \ldots, x_l, x_{l+1}, \ldots, x_n\}$ indicate a set of $n$ microarray data objects. The first $l$ points $x_i \in X(i \leq l)$ are labeled and the rest of the points $x_u \in X(l + 1 \leq u \leq n)$ are unlabeled. $y$ refers to the class label set.

In the graph-based semi-supervised learning algorithms described in Zhu et al., (2003), the method first defines an undirected graph $W$ on the entire data set. In
the graph \( W \), the nodes indicate data instances in the graph and the edges stand for the strength of two data instances in the graph. Then the technique constructs a k nearest neighbors graph with the Gaussian function of Euclidean distance to weigh the edges 

\[
W_{ij} = \begin{cases} 
\exp \left(-\frac{(x_i-x_j)^T(x_i+x_j)}{\sigma^2}\right) & \text{if } i \sim j \\
0 & \text{otherwise}
\end{cases}
\] (5.5)

where \( i \sim j \) represents that node \( i \) and \( j \) has an edge in between them. Thus, the graph can be represented as below

\[
W = \begin{pmatrix}
W_u & W_{lu} \\
W_{ul} & W_{uu}
\end{pmatrix}
\] (5.6)

where \( W_{li} \) stands for the weight of the edges between two labeled microarray data instances and \( W_{uu} \) represents the weight of the edges between two unlabeled microarray data instances present in the graph. \( W_{uu} \) indicates the weights of the edges from the labeled microarray points to the unlabeled microarray points in the graph and \( W_{ul} \) stands for the weights of the edges from the unlabeled microarray points to the microarray labeled points in the graph. All the weights in \( W \) are weighed by \( w_{ij} \).

With the help of the above constructed graph, the graph-based techniques can be considered as the estimating a function \( f \) on the graph \( W \). \( f \) refers to a real-value class assign matrix that does the assigning of the class labels. In graph-based semi supervised learning, \( f \) must satisfy two rules: (1) the value of \( f \) has to be in proximity with the class labels of the labeled data samples and the regularizer in the graph can expressed in the equation below

\[
\sum_{i \in L} (f_i - y_i)^2
\] (5.7)

and (2) \( f \) must meet the second condition (5.8) and the \( f \) has to be sufficiently smooth on the entire graph. Thus the sufficiently smooth graph along with the regularizer can be expressed as below


\[
\frac{1}{2} \sum_{ij} w_{ij} (f_i - y_i)^2
\] (5.8)

This way, the clustering problem concerned with the graph-based semi-supervised learning have been observed to be the combination of the two regularizer given above

\[
\sum_{i \in L} (f_i - y_i)^2 + \frac{1}{2} \sum_{ij} w_{ij} (f_i - y_i)^2 = (f - y)^T (f - y) + \frac{1}{2} f^T \Delta f
\] (5.9)

In which \(\Delta\) is called the graph which is used to indicate the outliers more effectively \(f = \left(\begin{array}{c} f_l \\ f_u \end{array}\right)\) and the \(f\) are formulated as \(f = Pf\) where the \(P = D^{-1}W\), \(f_u\) stands for the label of the unlabeled microarray data examples and \(f_l\) specifies the values of the labeled examples.

\[
f_u = (I - P_{uu}^{-1}) P_{ul} f_l
\] (5.10)

where \(I\) refers to identity matrix and the clustering results are obtained from the \(f_u\). The outliers are optimally detected by using GSSBAT approach. For the mixed dataset, the outliers are detected which is sued to increase the clustering result than the previous approaches.

5.7. PERFORMANCE EVALUATION

This section carries out the efficiency and effectiveness tests for analyzing the performance of the novel GSSBAT technique. In order to test effectiveness, the result is compared with the existing methods for artificial data sets. The Area Under the Curve (AUC) results of various techniques and the characteristics of all the test data sets, like the numbers of objects (#n), attributes (#m) and outliers (#o), and the upper bound on outliers (#UO), are tabulated in the upper portion of Table 5.1.
Table 5.1. AUC Results of Tested Algorithms

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ITB-SP</th>
<th>ITB-SS</th>
<th>AMCEM</th>
<th>EMPWC</th>
<th>GSSBAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast-c</td>
<td>0.991</td>
<td>0.993</td>
<td>0.996</td>
<td>0.997</td>
<td>0.998</td>
</tr>
<tr>
<td>Credit-a</td>
<td>0.985</td>
<td>0.992</td>
<td>0.995</td>
<td>0.996</td>
<td>0.997</td>
</tr>
<tr>
<td>Diabetes</td>
<td>0.75</td>
<td>0.912</td>
<td>0.945</td>
<td>0.945</td>
<td>0.957</td>
</tr>
<tr>
<td>Ecoli</td>
<td>0.96</td>
<td>0.99</td>
<td>0.996</td>
<td>0.998</td>
<td>0.999</td>
</tr>
</tbody>
</table>

5.7.1. Efficiency of Real Data Sets

Gives the measurement of the time consumption results with an increase in number of objects, attributes and outliers.

Figure 5.1. Time results analysis of real data sets with data objects vs. methods

Figure 5.1 shows, the execution times of GSSBAT, EMPWC, AMCEM, ITB-SP, ITB-SS, and FIB with number of objects. From the figure 5.1 it concludes that the proposed GSSBAT has lesser execution time results of 5.63 seconds for 200 data objects which is 1.37 second, 2.37 seconds, 5.37 seconds and 6.37 seconds lesser when compared to EMPWC, AMCEM, ITB-SS, and ITB-SP methods respectively.
The results of all the methods in terms of data objects are discussed and tabulated in table 5.2.

Table 5.2. Time analysis for data objects vs. methods

<table>
<thead>
<tr>
<th>No. of dataset points</th>
<th>ITB-SP</th>
<th>ITB-SS</th>
<th>AMCEM</th>
<th>EMPWC</th>
<th>GSSBAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>25</td>
<td>22</td>
<td>12</td>
<td>11.5</td>
<td>10.36</td>
</tr>
<tr>
<td>100</td>
<td>23</td>
<td>21</td>
<td>11</td>
<td>10</td>
<td>9.31</td>
</tr>
<tr>
<td>150</td>
<td>18</td>
<td>15</td>
<td>10</td>
<td>8.75</td>
<td>7.21</td>
</tr>
<tr>
<td>200</td>
<td>12</td>
<td>11</td>
<td>8</td>
<td>7</td>
<td>5.63</td>
</tr>
</tbody>
</table>

Figure 5.2. Time results analysis of real data sets with data attributes vs. methods
Figure 5.2 illustrates that the execution times of the GSSBAT and other existing methods. This increases quickly with the number of attributes increases in a quadratic manner. The result concludes that the proposed GSSBAT, increase quickly with the number of attributes that is average of 5.6 seconds. From the figure 5.2 it concludes that the proposed GSSBAT has lesser execution time results of 3 seconds for 30 data attributes which is 2 second, 3 seconds, 6 seconds, and 9 seconds lesser when compared to EMPWC, AMCEM, ITB-SS, and ITB-SP methods respectively. The results of all the methods in terms of data attributes are discussed and tabulated in table 5.3.

Table 5.3. Time analysis for data attributes vs. methods

<table>
<thead>
<tr>
<th>No. of attributes</th>
<th>Time (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ITB-SP</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>15</td>
<td>18</td>
</tr>
<tr>
<td>20</td>
<td>16</td>
</tr>
<tr>
<td>25</td>
<td>15</td>
</tr>
<tr>
<td>30</td>
<td>12</td>
</tr>
</tbody>
</table>
Figure 5.3. Time results analysis for percentage of the outliers vs. methods

Figure 5.3 depicts the run time comparison in terms of the percentage of “outliers” present in the data set. Figure 5.3 concludes that the newly introduced GSSBAT has lesser execution time results of 0.003 seconds in terms of percentage of outliers (0.5) which are 0.007 seconds, 0.009 and 0.011 seconds lesser when compared to EMPWC, AMCEM, and ITB-SP methods respectively. The results of all the methods in terms of percentage of the outliers are discussed and tabulated in table 5.4.

Table 5.4. Time results analysis for percentage of the outliers vs. methods

<table>
<thead>
<tr>
<th>Percentage of outliers</th>
<th>ITB-SS</th>
<th>AMCEM</th>
<th>EMPWC</th>
<th>GSSBAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.028</td>
<td>0.023</td>
<td>0.017</td>
<td>0.011</td>
</tr>
<tr>
<td>0.2</td>
<td>0.026</td>
<td>0.021</td>
<td>0.015</td>
<td>0.011</td>
</tr>
<tr>
<td>0.3</td>
<td>0.021</td>
<td>0.018</td>
<td>0.013</td>
<td>0.007</td>
</tr>
<tr>
<td>0.4</td>
<td>0.018</td>
<td>0.016</td>
<td>0.014</td>
<td>0.005</td>
</tr>
<tr>
<td>0.5</td>
<td>0.014</td>
<td>0.012</td>
<td>0.01</td>
<td>0.003</td>
</tr>
</tbody>
</table>
5.7.2. Outlier detection measurements

Figure 5.4 illustrates the results of the performance comparison of the NMSE for the existing techniques like ITB-SP, ITB-SS, AMCEM, EMPWC and the proposed GSSBAT algorithm. From the figure 5.4 it concludes that the proposed EMPWC algorithm has lesser NMSE results of 6% which is 4%, 5%, 7% and 9% lesser error value when compared to EMPWC, AMCEM, ITB-SS, and ITB-SP methods respectively.

![Figure 5.4. NMSE results comparison vs. methods](image)

Detection Rate

Correct detection rate, the number of outliers that are detected correctly by every approach.
Figure 5.5. Detection Rate (DR) results comparison vs. methods

Figure 5.5 illustrates the performance comparison results of the outlier Detection Rate (DR) for the methods such as ITB-SP, ITB-SS, AMCEM, EMPWC and proposed GSSBAT algorithm. From the figure 5.5 it concludes that the proposed GSSBAT algorithm has higher average DR of 87.98571 % which is 5.5%, 6.8285%, 11.4%, and 7.81% higher when compared to EMPWC, AMCEM, ITB-SS, and ITB-SP methods respectively. The detection accuracy of this proposed EMPWC is higher when compared to existing methods, since the proposed work data pre-processing and imbalanced dataset problem is solved by using data mining methods. Thus increases clustering accuracy as well as reducing error rate. The results of each of the methods in terms of detection rate are discussed and tabulated in table 5.5.
Table 5.5. Detection Rate (DR) results comparison vs. methods

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Detection Rate(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ITB-SP</td>
</tr>
<tr>
<td>0.1</td>
<td>73</td>
</tr>
<tr>
<td>0.2</td>
<td>73.5</td>
</tr>
<tr>
<td>0.3</td>
<td>74.1</td>
</tr>
<tr>
<td>0.4</td>
<td>75.3</td>
</tr>
<tr>
<td>0.5</td>
<td>76.4</td>
</tr>
<tr>
<td>0.6</td>
<td>77.4</td>
</tr>
<tr>
<td>Average DR(%)</td>
<td>74.671429</td>
</tr>
</tbody>
</table>

Figure 5.6. False Alarm Rate (FAR) results comparison vs. methods
Figure 5.6 depicts the performance comparison results of the False Alarm Rate (FAR) for the existing methods such as ITB-SP, ITB-SS, AMCEM, EMPWC and proposed GSSBAT algorithm. From the figure 5.6 it concludes that the proposed GSSBAT algorithm has lesser average FAR value of 15% which is 2.33%, 3.75%, 8.066%, and 10.21% lesser value when compared to EMPWC, AMCEM, ITB-SS, and ITB-SP methods respectively. The results of all the methods in terms of FAR are discussed and tabulated in table 5.6.

Table 5.6. False Alarm Rate (FAR) comparison vs. methods

<table>
<thead>
<tr>
<th>Threshold</th>
<th>FIB</th>
<th>ITB-SP</th>
<th>ITB-SS</th>
<th>AMCEM</th>
<th>GSSBAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>27</td>
<td>25</td>
<td>21</td>
<td>19</td>
<td>17</td>
</tr>
<tr>
<td>0.2</td>
<td>26.5</td>
<td>24.4</td>
<td>21</td>
<td>19</td>
<td>16</td>
</tr>
<tr>
<td>0.3</td>
<td>25.9</td>
<td>23</td>
<td>19</td>
<td>18</td>
<td>16</td>
</tr>
<tr>
<td>0.4</td>
<td>25.7</td>
<td>23</td>
<td>18</td>
<td>17</td>
<td>14</td>
</tr>
<tr>
<td>0.5</td>
<td>23.6</td>
<td>22</td>
<td>17</td>
<td>16</td>
<td>14</td>
</tr>
<tr>
<td>0.6</td>
<td>22.6</td>
<td>21</td>
<td>16.5</td>
<td>15</td>
<td>13</td>
</tr>
<tr>
<td>Average FAR(%)</td>
<td>25.216667</td>
<td>23.066667</td>
<td>18.75</td>
<td>17.33333</td>
<td>15</td>
</tr>
</tbody>
</table>

5.8. SUMMARY

In this chapter, the efficiency of the proposed GSSBAT outlier detection technique is evaluated using the attribute frequency results from a weighted entropy optimization. In this research, the important phases are preprocessing, outlier detection and clustering. The preprocessing is done with the help of min-max normalization approach. Then the unbalanced dataset problem is handled by using
SMOTE with kNN approach which is focused to increase the dataset efficiency. Apply the BAT optimization algorithm for optimizing the outlier attributes. At finally GSS based clustering algorithm is proposed for clustering and outlier detection. The result proves that the newly introduced GSSBAT approach has superior performance in terms execution, NMSE time and false alarm rate than the previous approaches.