ISSN 2278-3091

Volume 9 No.2, March - April 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse188922020.pdf

https://doi.org/10.30534/ijatcse/2020/188922020

Using Clustering techniques and Classification Mechanisms for Fault Diagnosis

Sonika Dahiya[1*], Harshit Nanda[2*], Jatin Artwani[3*], Jatin Varshney[4*]

[1]Delhi Technological University, India, sonika.dahiya11@gmail.com
[2]Delhi Technological University, India, hnnanda@gmail.com
[3]Delhi Technological University, India, jatinartwanii@gmail.com
[4]Delhi Technological University, India, jatinvarshneyy@gmail.com

ABSTRACT

In this paper we aim to provide a simulation of processing a real-life fault diagnosis system using a s-set benchmark dataset. In this proposal, firstly we determine the optimal number of clusters using three major techniques. Secondly, we perform clustering to build a comparative study of seven algorithms using evaluation metrics, time taken, shape and density of clusters and effectiveness in detecting the outliers. We select the best clustering algorithm to segment the dataset into the corresponding fault-based groups and label the dataset to be used for classification. Lastly, an automation tool called the TPOT Classifier is applied to find the best classification pipeline that yields a high accuracy. The results obtained are promising and outlay a detailed comparison of the clustering algorithms and classification mechanisms. This methodology can be employed on a real-time sensor data to carry out fault diagnosis effectively.

Key words: Classification Pipeline, Clustering Algorithm, Fault Diagnosis, Optimal Number of Clusters

1. INTRODUCTION

In industrial systems, it is integral to improve the efficiency of the processes involved so that the products have a higher quality alongside meeting the environmental and other regulations concerning industrial safety. In the industries, the faults that may occur in equipments can have a significant impact on the efficiency of a system and can be fatal in some cases. As a consequence, these faults need to be addressed and detected as early as possible. This detection helps in isolation of the faulty conditions and facilitates in analysing the cause behind occurrence of such faults.

In any Fault diagnosis system readings are recorded using sensor data which in turn are known as correlated process parameters. Subsequently, by applying techniques like principal component analysis (PCA), it is possible to reduce the dimension of the recorded dataset to make it fit for analysis by the proposed method. Clustering algorithms are employed when the data is unlabelled and therefore an unsupervised learning approach has to be used to carry out the analysis. They segment the dataset into groups based on the similarity indices calculated between the data points.

In most of the clustering algorithms, there is a direct input parameter for the number of clusters, or it can be controlled by hyper tuning parameters for some methods. As a consequence, to select the best method and validate the result it is an integral step to find the optimal number of clusters. Two of the most common techniques based on intra-cluster variation that is, elbow and silhouette method are used. Furthermore, an enhanced technique known as Gap statistic to calculate the optimal number of clusters is also employed. The essence of the Gap statistic approach is to compare within the cluster dispersion and it is known to outperform other methods in terms of accuracy and reliability [2].

There are a lot of aspects to be considered in choosing the best fit & suitable clustering algorithm for a particular dataset. In an attempt to funnel down to the preferred clustering technique for a dataset a comparison of seven clustering algorithms is carried out by taking similar experimental conditions throughout for all the techniques. This study consolidates the advantages and disadvantages of most prominent clustering algorithms used and helps in identification of the most suitable algorithm to be used.

Generally, any sensor related dataset is finite, unlabelled multi-variate. To further structure the dataset this paper introduces an approach to label the observations by numerical values in coherence with the cluster group reported by the most efficient clustering algorithm.

This labelled dataset is used to bring about an amalgamation of unsupervised and supervised learning in determination of faults. To build a classification model, an automation tool known as TPOT classifier which implicitly take into account different pre-processing mechanisms and each one of them is used on eleven commonly used classification algorithms to determine the classification pipeline that is suitable and performance efficient on the dataset chosen.

*All the authors contributed equally to this work.

This technique provides level-1 classification for each observation. Dataset is split between training and testing data to build a model to determine the accuracy of the classification model. This model can be further used to classify the data fetched from sensors in real time. This paper reports results from the synthetic dataset which has 2 dimensions and has uniform dispersion of observations across all the fault groups. Section II Describes the background algorithms for obtaining optimal number of clusters, clustering algorithms, evaluation metrics and classification algorithms used. Section III describes the proposed approach. Detailed description of the Dataset used for analysis and Results obtained are presented in section IV. A section on conclusion ends the paper.

2. BACKGROUND ALGORITHMS

Clustering is a method of exploratory information analysis focusing on fragmenting a limited, unlabeled, multivariate informational index into a set of homogeneous clusters, classifications or groups.

2.1 Determination of Optimal Number of Clusters

2.1.1 Elbow Method

This method takes into account a function of the number of clusters called WSS that is total within-cluster sum of square giving an idea of the compactness of clustering. These intra-cluster variations are required to be as minimal as possible [6]. The algorithm is as follows:

- Reckon a clustering algorithm for various values of like in k-means clustering by ranging the values of k from 1 to 10 and then compute the WSS for each value of k.
- Plot the readings of WSS in a form of a curve with each corresponding value of number of clusters i.e. K
- Observe the plotting and the point at which it distinctly bends over is a signal for the most appropriate number of clusters.

2.1.2SilhouetteMethod

This method comes into role as an efficient method for specifying the likelihood of an object belonging to a particular cluster or in a way how one object is different from its neighbouring clusters. Distance metrics like the Euclidean or the Manhattan distances are usually applied to find out the silhouette value [11].Assuming that the dataset has been clustered beforehand into k clusters with a clustering algorithm like k means algorithm, an average silhouette value is calculated for every k and the value of k which has the maximum value of the silhouette coefficient s(i) determines the optimal number of clusters for the given. For every data point, following variables are accounted:

- C(i): The cluster allotted to the i^{th} data point.
- |C(i)|:total number of clusters allocated to the i^{th} data point,
- A(i): Likelihood of the data point to it's own cluster.
- B(i): Mean Difference from the neighbouring clusters.

2.1.3 Gap-Statistic Method

This technique finds its application to all the clustering methods. The goal of the method is to contrast the dispersion internal tothe clusters with the desire under a suitable null reference distribution of the dataset[4]. The measurement which will produce the largest statistical gap will give out the right number of clusters. The intra-cluster disparities for various vales of n make it unlike the randomised distribution of the data points. The working of the algorithm is:

- Cluster the given dataset to identify the number of clusters from n=1, ... , n_{max}, and compute the complete intra-cluster difference W_n.
- Create B reference datasets with the help of randomly induced uniform distribution. Then, cluster the generated datasets where the number of clusters are varied from n=1, . . . ,n_{max}, and compute the corresponding total within-cluster variation W_{nb}.
- Find out the probable Gap Statistics as the deviation of theexperimental value W_n with respect to the expected value W_{nb} under the null hypothesis.
- Root out the total number of clusters as the minimal value of n given that the gap statistic is in the range of one usual deviation of the gap n+1.

2.2Clustering Algorithms

2.2.1 K-Means

This technique holds loose terms with the k-nearest neighbour classifier which is a commonly used machine learning algorithm for classification. Application of the 1-nearest neighbour classifier to the mid points of the cluster observed by k-means classifies new data set into the clusters that are already existing. This is called the Rocchio algorithm, also known as the nearest centroid classifier[9]. K means techniques find its application in various domains such as feature learning, cluster analysis and vector quantisation[12].

2.2.2 Affinity Propagation

Applied in statistical mathematics and data warehouses and data mining, affinity propagation (AP) is schemed upon the idea of message passing within the data points. As compared to other clustering techniques such as the k-means or the k-mediods algorithms, this technique does not need the value of the number of clusters before its execution. In fact, affinity propagation coins the representative members of the clusters as the 'exemplars'. The method gives out good results for some computer vision applications and for some figurative biology assignments. The similarity between the data points is calculated using common indices. The algorithm continues by alternating between two message transfer procedures to recondition two matrices.

2.2.3 Spectral Clustering

It is a widely used technique to carry out exploratory analysis of the data provided. Initially, a similarity graph is created for the data points to cluster. Subsequently, computation of the first k-eigenvectors in its Laplacian matrix is carried out to illustrate a feature vector corresponding to each object [8].

Furthermore, K-means is applied to group objects into k-classes. Multiple k-means results are combined to group the objects belonging to irregular form groups.

2.2.4Agglomerative Clustering

Also termed as Agglomerative Nesting, this type of hierarchical clustering is idealised on the similarity of the objects in the clusters. The results of the technique are visualised by a tree-based formation of the objects called a dendrogram by a function hclust(). In short, each object is considered to be a singleton cluster, then pairs of clusters are consolidated one by one until all of them have been merged into a single cluster consisting of all the objects[7]. This algorithm is very much direct to follow:

- Compute the proximity of the clusters and generate a proximity matrix.
- Consider every data point as a singleton cluster.
- Reiterate this step as the successive pairs of clusters are merged and the proximity matrix is updated at every new change.
- Remaining result should be a single big cluster.

2.2.5 Mean-Shift Algorithm

It is a hierarchical clustering technique and it builds upon kernel density estimation. The parameter for the number of clusters doesn't have to be given as input beforehand and it doesn't put a constraint on the shape of the clusters but it is computationally expensive in comparison to the other algorithms.

Firstly, a bandwidth of the kernel is selected randomly which is to be placed on each data point. After calculating the mean distance of all the data points located inside the kernel, the centre of the window is displaced to the mean obtained. This is iteratively carried out until the result converges and we do not have to re-estimate the mean.

2.2.6DBSCANAlgorithm

Itis known as Density-based spatial clustering of applications with noise. It groups together a set of data points that are in proximity to each other based on an evaluative distance metric (usually Euclidean Distance) and a minimum number of points. The points present in regions having lesser density are marked as outliers[10].

DBSCAN makes use of two parameters: Epsilon and the minimum number of points required to make a dense region (minPts). The algorithm begins by taking an unvisited arbitrary starting point. It retrieves that it's neighborhood, and if that contains required threshold of points, a formation of cluster is initiated. Otherwise, the point retrieved is labeled as an outlier. However, it may happen that this point is later found in a big enough neighborhood of a different point and subsequently become a part of a cluster [13]. If a point is associated with a dense part of a cluster, it's neighborhood also becomes a part of that cluster. Therefore, all the points within the neighborhood are added, as it is actually their own neighborhood when they are also dense. The above methodology iteratively repeats until the density-connected cluster is found in a complete manner[3]. After all this, a newly found unvisited point is fetched and processed in order to determine if it's an outlier or if it belongs to a cluster.

2.2.7 HDBSCAN Algorithm

The method transforms DBSCAN into a hierarchically clustered principle [1] while extending the use of it as a technique to excerpt a flat clustering method formed on the constancy of various clusters. HDBSCAN can be illustrated with 5 steps:

- Transform the space based on density or sparseness of data points
- After that create a minimum weight spanning tree having edges quantified by distance in the graph
- Take the connected components and build a hierarchical group
- Concentrate the above cluster hierarchy considering the minimum size of a cluster
- Extract the clusters that are stable from the tree obtained

2.3 Evaluation Metrics

2.3.1 Adjusted Rand Index

Both the basic assignments of truth class that is "labels_true" and the assignments given by the clustering algorithm of the identical specimen "labels_pred" are known in this case. The Adjusted Rand Index is known as the measure of similitude of the two mentioned assignments [5]. The unadjusted Rand Index is given by the following:

$$\mathrm{RI} = \frac{a+b}{C_2^{n_{samples}}} \qquad (1)$$

Here,C and K refer to two sets and 'a' is the number of total pairs of elements which are a part of the same set in Cand in the same set in K, b is the total pairs of elements which are invaried sets in C and in varied sets in K, the denominator is the number of unordered pairs in the dataset. Upon reducing the value of the expected Ri value that is E[RI]of random tags, the adjusted rand index is given by:

$$\mathrm{ARI} = \frac{\mathrm{RI} - E[\mathrm{RI}]}{\max(\mathrm{RI}) - E[\mathrm{RI}]} \qquad (2)$$

2.3.2 Mutual Information based Scores

Both the basic assignments of truth class that is "labels_true" and the assignments given by the clustering algorithm of the identical specimen "labels_pred" are known in this case. The Mutual Information is defined as the measure of concordance of the two mentioned assignments. Let there be two assignments U and V respectively of the similar N objects, the entropy of a partition set is given as:

$$H(U) = -\sum_{i=1}^{|U|} P(i)\log(P(i)) \quad (3)$$

where P(i) is {|Ui|/N} is the probability of an object at random picking from U that falls into class U_i. Similarly it is done for V. The adjusted mutual information (AMI) is given by:

$$\mathrm{AMI} = \frac{\mathrm{MI} - E[\mathrm{MI}]}{\mathrm{mean}(H(U), H(V)) - E[\mathrm{MI}]} \quad (4)$$

2.3.3Homogeneity, Completeness and V-measure

Both the basic assignments of truth class “labels_true” and the assignments given by the clustering algorithm of the identical specimen “labels_pred” are known in this case. Homogeneity is defined as the desirable goal ofa cluster containing only members belonging to a single class and completeness is defined as the desirable goal of all members of a particular class assigned to the same cluster. V-measure is defined as the harmonic mean of the homogeneity and completeness. The mathematical equations are given by:

Homogeneity,

$$h = 1 - \frac{H(C|K)}{H(C)} \quad (5)$$

Completeness,

$$c = 1 - \frac{H(K|C)}{H(K)} \quad (6)$$

where H(C—K) is defined as the conditional entropy associated with the classes given the cluster assignments and is as follows:

$$H(C|K) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|} \frac{n_{c,k}}{n} \cdot \log\left(\frac{n_{c,k}}{n_k}\right) \quad (7)$$

Here, H(C) is defined the entropy of the classes and is given by:

$$H(C) = -\sum_{c=1}^{|C|} \frac{n_c}{n} \cdot \log\left(\frac{n_c}{n}\right) \quad (8)$$

The V-measure is given by:

$$v = 2 \cdot \frac{h \cdot c}{h + c} \quad (9)$$

2.3.4Fowlkes-Mallows Scores

Both the basic assignments of truth class that is “labels_true” and the assignments given by the clustering algorithm of the identical specimen “labels_pred” are known in this case. The Fowlkes-Mallows index (FMI) is computed as the geometric mean of the pairwise recall and precision. The FMI Score is :

$$\mathrm{FMI} = \frac{\mathrm{TP}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})}} \quad (10)$$

where ‘TP’ isTrue Positives, ‘FP’is False Positivesand FN is False Negatives. The range of the score is 0 to 1.

2.3.5Silhouette Coefficient

In this case, the ground truth labels are unknown. The Silhouette Coefficient gives an idea of the definition of the clusters, and the score is applied to every sample consisting of two major scores:

a: The average distance between a selected sample and rest of the points belonging to the same cluster.

b: The average distance between a selected sample and rest of the points in the cluster which is next nearest.

Silhouette Coefficient is given by :

$$s = \frac{b - a}{max(a, b)} \quad (11)$$

2.3.6Calinski-Harabasz Index

In this case, the ground truth labels are unknown. TheCalinski-Harabasz index known as the Variance Ratio Criterion can be applied to assess the model on the basis of the cluster definition. Higher the value, better the model becomes. The score computed is basically computed as the ratio of sum of intra-cluster inter-cluster dispersion for all the clusters respectively. Given a dataset E of size nE having k clusters, the ratio s is given as :

$$s = \frac{\mathrm{tr}(B_k)}{\mathrm{tr}(W_k)} \times \frac{n_E - k}{k - 1} \quad (12)$$

where $\mathrm{tr}(B_k)$ is the trace of intra-group dispersion matrix and $\mathrm{tr}(W_k)$ is the trace of inter-cluster dispersion matrix. The mathematical representations are given as:

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T \quad (13)$$

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T \quad (14)$$

2.3.7Davies-Bouldin Index

The Davies-Bouldin Index is used to asses a modelbased on the separation or distance between the clusters. This calculated index highlights the mean ‘similarity’ between different clusters, and values approaching to zero indicates a better partitioning.The index R_{ij} has the following elements:

- s_i:calculatedas the mean distance between every cluster point i and the centroid computed for that cluster
- d_{ij}:as the inter-cluster centroids distance between i and j

$$R_{ij} = \frac{s_i + s_j}{d_{ij}} \quad (15)$$

The Davies-Bouldin Index is given by:

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} R_{ij} \quad (16)$$

2.4 TPOT Classifier

TPOT is an automation tool in Python that is used to find the most efficient classification regression pipeline for smooth machine learning workflows. It is built on top of sci-kit learn and sk-learn APIs. Therefore, TPOT is considered an automation tool for Machine Learning algorithms and not Deep Learning algorithms. More technically, TPOT is a Genetic Search heuristic that realizes the finest model parameters and entities. In other words, it can also be called a natural selector/ evolutionary computation. As a wholesome, TPOT considers a pipeline, assess its performance, and makes alterations in some randomly selected parts of the pipeline in lieu of finding those algorithms which will give better results. A pipeline is basically a combination of various preprocessing methods like the PCA, Scalers, etc. followed by various regressors or classifiers. However, more in-depth tuning of the model is still performed by hyper-parameter tuning which is carried out by methods like Grid Search. The process followed by TPOT to choose the classification pipeline optimal for the dataset is shown in figure-1.

3. PROPOSED APPROACH

To build on our simulation research of a fault diagnosis system we funnelled down on a synthetic two-dimensional dataset of 5000 observations which had an equi-partition of data points across the range which was cleaned and structured by P. Franti and S. Sieranoja. Figure-2 provides the overview of the proposed methodology.

3.1Pre-Processing

The dataset has two principal components. To ease the visualisation in terms of density and shape of clusters, standard scaler was applied to plot and see the results effectively. This technique helps to standardize the feature by removal of the calculated mean and scaling it further to unit variance. Scaling and centering takes place on each feature of the data. It is an essential step to make a standard normally distributed data.

3.2Stage-1

In this stage, a brief comparison was carried out between the 3 major techniques to calculate the optimal number of clusters that are Elbow method, Silhouette Method & Statistical-Gap method to calculate the optimal number of clusters in the s-set benchmark dataset. This step is essential as most of the clustering algorithms take this parameter to segregate the dataset into the respective clusters as input. The direct and simple method to compute this is using elbow method which uses within the cluster sum of squares (WSS) to minimise the intra cluster variation. After locating the bend in the graph obtained between WSS vs K (number of clusters) the optimal number of clusters are obtained. This is a conventional approach and the result obtained is found ambiguous as the result vary for every iteration the score is computed. It is more of a decision metric and not used as a computation metric. Average Silhouette method is a more refined and deterministic metric which computes the optimal number of clusters. An average silhouette of observations is computed for each K. The location of maximum is considered as the measure of good compact clustering. The gap statistic method provides a statistical foundation to measure the total within intra-cluster variations. It is an effective technique as it gives precise results even if it is tested for multiple runs.

3.3Stage-2

This stage is the backbone of our study as a comprehensive comparative study of 7 clustering algorithms is done on the same dataset to provide a comparative reasoning about the time taken to perform clustering, the density shape of clusters, outlier detection and various evaluation metrics. First, the clean and pre-processed data is fed into each of the clustering algorithms and the results are noted. Then the next step is to prepare a tabulation of the parameters and algorithms taken into consideration. Four of the clustering algorithms K-Means, Affinity Propagation, Spectral Clustering Agglomerative Clustering group the data points without detecting the outliers. A tabulation is prepared based on 5 evaluation metrics such as Adjusted Rand Index, Adjusted Mutual Information Score, Homogeneity, Completeness and V-Measure. The best clustering algorithm is chosen as the preferred choice only if clustering has to be performed without considering the outliers. The important aspect to take into account is the ability of some of the clustering algorithms to detect the outliers effectively. We compare the 3 algorithms namely Mean-Shift, DBSCAN and HDBSCAN visually based on the shape and density of data points and 3 evaluation metrics namely Silhouette Score (SS), CalinskiHarabasz Score (CHS) and Davies Bouldin Index (DBI).

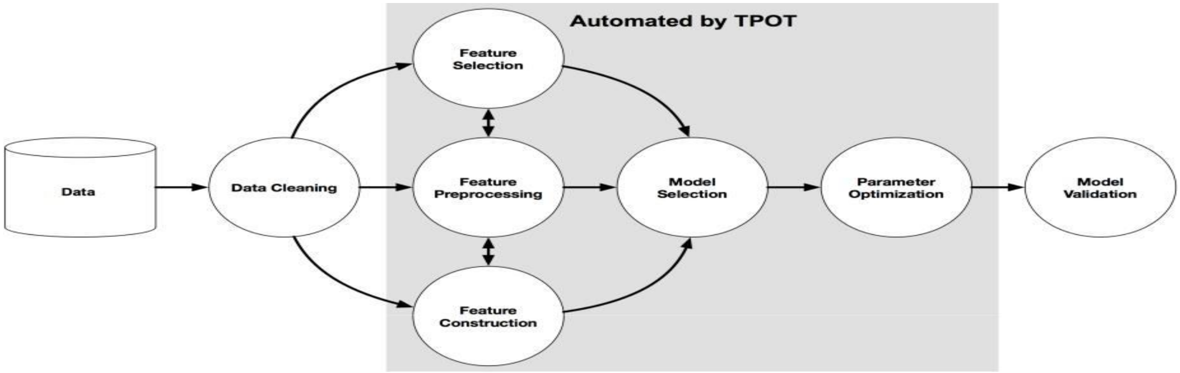


Figure 1: Step-by-step process of TPOT Classifier

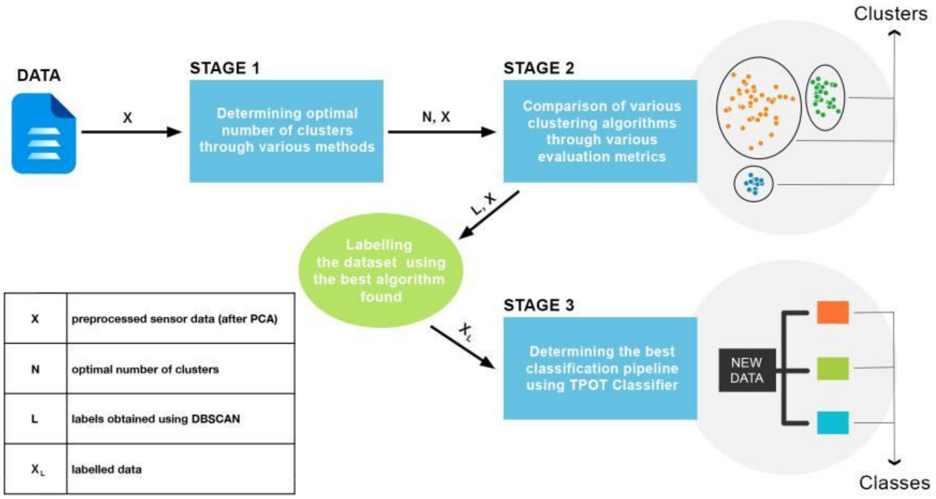

DATA
X
STAGE 1
Determining optimal number of clusters through various methods
N, X
STAGE 2
Comparison of various clustering algorithms through various evaluation metrics
Clusters
L, X
Labelling the dataset using the best algorithm found
X_L
STAGE 3
Determining the best classification pipeline using TPOT Classifier
NEW DATA
Classes
X preprocessed sensor data (after PCA)
N optimal number of clusters
L labels obtained using DBSCAN
X_L labelled data

Figure 2: Proposed Flow of the Process

3.4 Intermediate Step

To blend supervised learning, selected DBSCAN scan algorithm to be used for labelling of observations and further Building a classification model.

The dataset is labelled in coherence with the respective clusters associated by DBSCAN. Numerical values ranging from 0 to 14 are chosen for the 15 fault groups and -1 for outliers. This labelling is done so that the result can be verified by training the model and testing it on a small chunk of testing data by splitting the dataset into 4:1 ratio.

3.5 Stage-3

To build the classification methodology and select the best algorithm suitable for the dataset, we employed TPOT classifier which is based on a genetic algorithm. This algorithm combines various pre-processing methods followed by eleven classifiers to build multiple pipelines. Furthermore, it computes the accuracy of each pipeline on the testing chunk of data. The pipeline that yields the best accuracy is then chosen. Hyper parameter tuning is performed to obtain the best results. This approach can be employed when the features of a dataset has correlated parameters and therefore can be dimensionally reduced by using dimensionality reduction algorithm. This provides the basis to determine the faults in a real time sensor data recorded in any industrial system.

4. RESULTS AND DISCUSSIONS

4.1 Optimal Number of Clusters

In order to determine the optimal number of clusters, three methods namely Elbow Method, Silhouette Method and Statistical Gap Method have been applied.

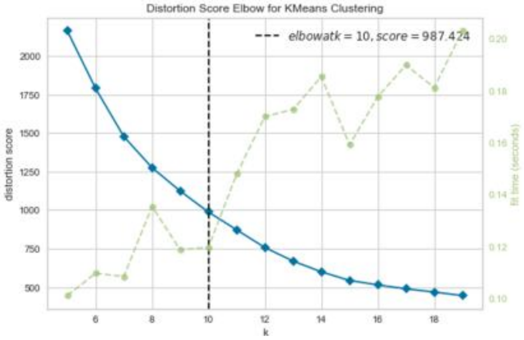
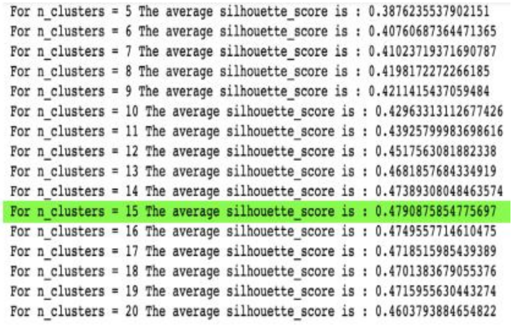


Figure 3: Elbow Method

The results of the proposed approach have been obtained on the S-Set benchmark Clustering Dataset .Figure 3 gives the visual result of the elbow method at k (number of clusters) = 10. Moreover, on the application of Silhouette Method, the average silhouette score is highest for k (number of clusters) = 15 as depicted in figure 4. Lastly, the statistical gap method gives optimal number of clusters=15 as shown in the figure 5. Since, the original Dataset has 15 clusters over 5000 data points, the results obtained by Silhouette Method and Statistical Gap Method are more accurate than those obtained by the Elbow Method.



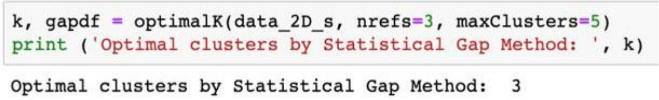
```
For n_clusters = 5 The average silhouette_score is : 0.3876235537902151
For n_clusters = 6 The average silhouette_score is : 0.40760687364471365
For n_clusters = 7 The average silhouette_score is : 0.41023719371690787
For n_clusters = 8 The average silhouette_score is : 0.4198172272266185
For n_clusters = 9 The average silhouette_score is : 0.4211415437059484
For n_clusters = 10 The average silhouette_score is : 0.42963313112677426
For n_clusters = 11 The average silhouette_score is : 0.43925799983698616
For n_clusters = 12 The average silhouette_score is : 0.4517563081882338
For n_clusters = 13 The average silhouette_score is : 0.4681857684334919
For n_clusters = 14 The average silhouette_score is : 0.47389308048463574
For n_clusters = 15 The average silhouette_score is : 0.4790875854775697
For n_clusters = 16 The average silhouette_score is : 0.4749557714610475
For n_clusters = 17 The average silhouette_score is : 0.4718515985439389
For n_clusters = 18 The average silhouette_score is : 0.4701383679055376
For n_clusters = 19 The average silhouette_score is : 0.4715955630443274
For n_clusters = 20 The average silhouette_score is : 0.4603793884654822
```

Figure 4: Silhouette Method

4.2 Clustering Algorithms

4.2.1 No Outlier Detection

A comparison between 5 algorithms that don't detect outliers has been made in order to determine the best one out in terms of various evaluation metrics, ease of use and time taken or complexity. Figure 6 depicts the clusters determined by the K-Means algorithm whereas Figure 7 shows the clusters determined by Affinity Propagation algorithm. Both algorithms are able to determine 15 clusters in the dataset. K-Means is easy to use due to availability of input on the number of clusters to be found. How-ever, Affinity Propagation requires parameter tuning in order to get the required number of clusters. Figure 8 depicts Agglomerative Clustering whereas Figure 9 shows Spectral Clustering. Both algorithms are well able to determine 15 clusters in the dataset. Both algorithms have the availability of direct input on the number of clusters to be found and are easy to use. Figure 10 depicts the variation of Mean Shift algorithm that doesn't detect outliers. Mean Shift has a special parameter known as 'cluster all' which when set 'True' assigns every data point to one of the clusters identified, whichdoesn't detect outliers.

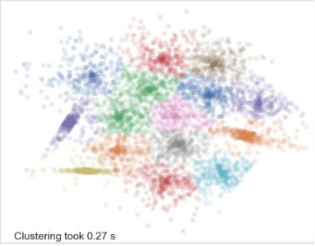


```
k, gapdf = optimalK(data_2D_s, nrefs=3, maxClusters=5)
print ('Optimal clusters by Statistical Gap Method: ', k)

Optimal clusters by Statistical Gap Method:  3
```

Figure 5: Gap Statistical Method

The evaluation metrics used for these 5 algorithms are:- Adjusted Rand Index(ARI), Adjusted Mutual Information Score (AMIS), Homogeneity (H), Completeness (C), V-measure (V), Fowlkes-Mallows Score (FMS)



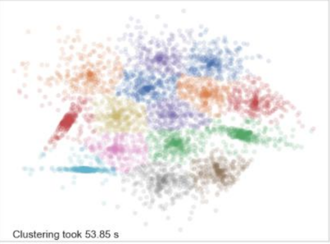
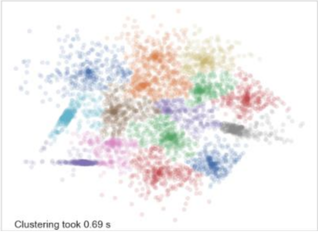


Figure 6: K-Means **Figure 7:**Affinity Propagation

Table 1 shows the comparison of 5 clustering algorithms on the basis of the 6-evaluation metrics. K-Means and Affinity Propagation are the top two performing algorithms for this dataset. It is clear in Table 1 that the K-Means algorithm has higher scores for 4 metrics namely Adjusted Mutual Information Score, Completeness, V-measure and Fowlkes-Mallows Score. Affinity Propagation has higher scores for Adjusted Rand Index and Homogeneity. It is concluded that K-Means outperforms all other algorithms. In Table 1, 'agglo' stands for agglomerative.

4.2.2 Outlier Detection

A comparison between 3 algorithms that detect outliers has been made in order to determine the best one out in terms of various evaluation metrics, ease of use, shape and density of the clusters, time taken and complexity. Figure 11 shows the variation of Mean Shift algorithm that detects outliers. Here, the parameter 'cluster all' has been set to 'False', which enables this algorithm to assign data points to either appropriate clusters or outliers. Figure 12 and figure 13 show the clusters identified by HDBSCAN & DBSCAN algorithms respectively. The main objective for these clustering algorithms is to correctly identify outliers as well as identify dense clusters that correspond to various faults in the system.



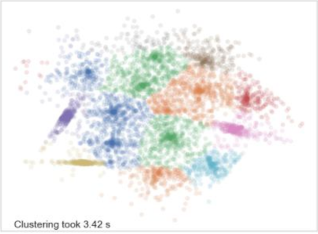
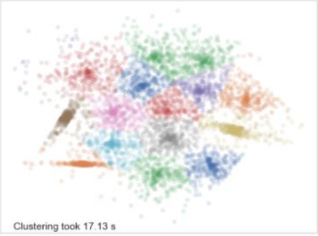


Figure 8: Agglomerative **Figure 9:** Spectral



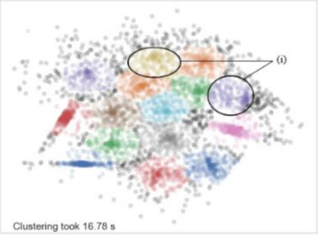


Figure10: Mean-Shift **Figure 11:**MeanShift(Outliers)

A visual comparison has been made on the basis of density and shape of clusters and cited in figure 11, figure 12 and figure 13 for each algorithm separately. The clusters identified by Mean Shift Algorithm are not dense as shown in figure 10. HDBSCAN identifies dense clusters however due to the ''variable density'' property of HDBSCAN, some points that lie considerably far away from the centroid of the clusters are still considered a part of that cluster. This in turn falsely assigns outliers to one of the clusters as shown in figure 12. The clusters identified by DBSCAN are dense to act analogous to the system faults. Due to these reasons HDBSCAN and DBSCAN easily outperform Mean Shift algorithm. Hence, evaluation based on metrics has been performed only for HDBSCAN and DBSCAN further. The evaluation metrics used for these 2 algorithms are: Silhouette Score (SS), Calinski-Harabasz Score (CHS) and Davies Bouldin Index (DBI)

Table 1: Evaluation Metrics for Clustering Algorithm

Algo/Metric	ARI	AMIS	H	C	V	FMS
K-Means	0.6329	0.71919	0.72060	0.7218	0.72124	0.6593
Affinity	0.6348	0.71916	0.72068	0.7217	0.72122	0.6575
Spectral	0.5176	0.68618	0.66050	0.7192	0.68861	0.5627
Agglo	0.5945	0.70591	0.70663	0.7094	0.70806	0.6218
Mean-Shift	0.5714	0.68584	0.68095	0.6955	0.68818	0.6014

Table 2 shows the comparison of HDBSCAN & DBSCAN on the basis of the 3 selected evaluation metrics. These metrics evaluate the clusters formed on the basis of their density. Clearly, DBSCAN outperforms HDBSCAN, which indicates that DBSCAN performs accurate outlier detection and is able to detect highly dense clusters.

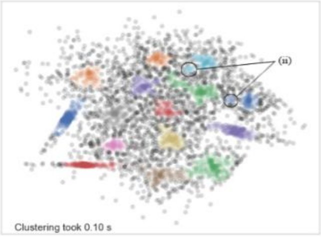


Figure 12:HDBSCAN

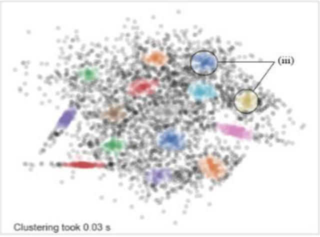


Figure 13:DBSCAN

Table 2: Evaluation Metrics

Algo/Metric	SS	CHS	DBI
DBSCAN	0.775258	17250.069197	0.301975
HDBSCAN	0.692781	10362.624316	0.407998

Table 3 shows the comparison between 7 clustering algorithms. Here, 'k' parameter for number of clusters.

Table 3:Comparison of Clustering Algorithms

Algorithm	Time(s)	K	Outliers	Complexity
K-means	0.24	Yes	No	$O(n^2)$
Affinity	44.95	No	No	$O(n^2*t)$
Agglomerativ	0.8	Yes	No	O(n3)
Mean -Shift	17.22	No	Both	$O(n^2*t)$
Spectral	3.36	Yes	No	$O(n^2)+O(n^3)$
DBSCAN	0.03	No	Yes	O(n*logn)
HDBSCAN	0.17	No	Yes	O(n*logn)

The TPOT Classifier iterates for 8 generations giving out the internal CV score for each generation. With every generation having 100 pipelines, TPOT successfully compares 900 pipelines on the basis of accuracy obtained on testing data. For this particular dataset, TPOT selects ExtraTrees-Classifier as the base classifier and RBF-Sampler as the pre-processing mechanism and subsequently the best classification pipeline is obtained. This classification pipeline yields an accuracy of 98.7 percent on the testing data.

5. CONCLUSION

In this paper, we presented a novel approach to be used for fault diagnosis. The proposed approach has 3 stages which can be employed on a real-life sensor data to extract insights about the number of faults, fault-grouping, labelling the faults and further using supervised learning to build a classification model. We validated our approach on s-set benchmark dataset. After a comprehensive comparison, gap-statistic outperformed the other techniques to determine the number of clusters. Secondly, DBSCAN algorithm was the most efficient in comparison to other algorithms sin terms of determining dense clusters and its ability to detect outliers as well. Furthermore, TPOT algorithm facilitated to funnel down on Extra -Trees algorithm as the best classifier to build the supervised learning model. Next, we plan to implement the above approach on a real time sensor data which has 14 process parameters that are co-related and we intend to explore sophisticated dimensionality reduction algorithms as well.

REFERENCES

1. Ali, Tariq, Sohail Asghar, and Naseer Ahmed Sajid.**Critical analysis of DBSCAN variations**, IEEE *International Conference on Information and Emerging Technologies*, pp. 1-6, June 2010. https://doi.org/10.1109/ICIET.2010.5625720
2. U. Vignesh, G. Sivanageswara Rao, B. Manjula Josephine and Puvvada Nagesh. **Food Waste Protein Sequence Analysis using Clustering and Classification Techniques**, *IJATCSE International Journal of Advanced Trends in Computer Science and Engineering*, Volume 8, No.5, September - October 2019. https://doi.org/10.30534/ijatcse/2019/67852019
3. Cheng, Tang. **An Improved DBSCAN Clustering Algorithm for Multi-density Datasets**, *Proceedings of the 2nd International Conference on Intelligent Information Processing*, pp. 1-5, July 2017. https://doi.org/10.1145/3144789.3144808
4. Huang, Shyh-Jier, and Jeu-Min Lin. **Artificial neural network enhanced by gap statistic algorithm applied for bad data detection of a power system**, *IEEE/PES Transmission and Distribution Conference and Exhibition*, vol. 2, pp. 764-768, October 2002.
5. Kapil, Shruti, and Meenu Chawla. **Performance evaluation of k-means clustering algorithm with various distance metrics**, *IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, pp. 1-4, July 2016.
6. Marutho, Dhendra, Sunarna Hendra Handaka, and Ekaprana Wijaya. **The determination of cluster number at k-mean using elbow method and purity evaluation on headline news**, *IEEE International Seminar on Application for Technology of Information and Communication*, pp. 533-538, Sept. 2018 https://doi.org/10.1109/ISEMANTIC.2018.8549751.
7. Kaur, Puneet Jai. **Cluster quality based performance evaluation of hierarchical clustering method**, *IEEE 1st International Conference on Next Generation Computing Technologies (NGCT)*, pp. 649-653, Sept. 2015.
8. Sapkota, Niroj, AbeerAlsadoon, P. W. C. Prasad, Amr Elchouemi, and Ashutosh Kumar Singh. **Data Summarization Using Clustering and Classification: Spectral Clustering Combined with k-Means Using NFPH**, *IEEE International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 146-151, Feb. 2019.
9. Dr.M.Karthikeyan, ArangaArivarasan and D.Kumaresan. **Performance Assessment of Various Text Document Features through K-Means Document Clustering**

Approach, *IJATCSE International Journal of Advanced Trends in Computer Science and Engineering*, Volume 8, No.5, September - October 2019.
https://doi.org/10.30534/ijatcse/2019/21852019

10. Smiti, Abir, and ZiedEloudi. **Soft dbscan: Improving dbscan clustering method using fuzzy set theory**, *IEEE 6th International Conference on Human System Interactions (HSI)*, pp. 380-385, June 2013.
https://doi.org/10.1109/HSI.2013.6577851
11. Sudheera, P., V. Ramakrishna Sajja, S. Deva Kumar, and N. Gnaneswara Rao. **Detection of dental plaque using enhanced K-means and silhouette methods**, *IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, pp. 559-563, May 2016.
https://doi.org/10.1109/ICACCCT.2016.7831702
12. Wenchao, Li, Zhou Yong, and Xia Shixiong. **A novel clustering algorithm based on hierarchical and K-means clustering**, *IEEE Chinese Control Conference*, pp. 605-609, July 2007.
13. Beri, Saefia, and Kamaljit Kaur. **Hybrid framework for DBSCAN algorithm using fuzzy logic**, *IEEE International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*, pp. 383-387, February 2015.
https://doi.org/10.1109/ABLAZE.2015.7155024