Journal of Theoretical and Applied Information Technology
15th December 2021. Vol.99. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A MODEL FOR CLASS NOISE DETECTION USING TREE-BASED CLUSTERING ALGORITHM

[1] HELAL A. Suleiman, [2]MANAL A. ABDELFATTAH, [3]OSAMA E. EMAM
[1,2,3] Information Systems dept, Faculty of Computers and Artificial Intelligence, Helwan University, Cairo,Egypt

Corresponding Author: [1] helal.a.suleiman@fci.helwan.edu.eg
Author: [2]manal_8@hotmail.com,[3] emam_o_e@yahoo.com

ABSTRACT

Clustering is the most descriptive task in a mining data stream, because it is a method of grouping or merging data objects into disjoint clusters based on some criteria you choose. So, such data objects in the same cluster are similar hence data objects in other clusters are different. This paper presents a model that includes methods for ensuring not only detecting outliers but also handling noisy data in real-time and offline as well. Furthermore, by using the improved tree-based clustering algorithm there is no need to initialize number of clusters in advance. Experimental results, applied to hotels and flights, find that this proposed model achieves high-quality results without outliers in less time on real datasets.

Keywords: *Clustering, Data Stream, Tree-Based Clustering, Noisy Data, Outliers*

1. INTRODUCTION

Traditional data mining methods use real static data, which are considered qualified and more powerful when processing small data sets but are considered unqualified and slow in real-time processing. Most researchers tend to develop data flow algorithms to reduce execution time and memory consumption. Stratification, partition, model, grid-based, and density-based clustering are the four main types of clustering algorithms [1]. The first type of hierarchical algorithm consists of two methods: agglomerative method and split method, which try to evaluate specific data to establish a hierarchical structure of clusters. Tree-based clustering is a partition-based technique for identifying clusters [2]. Tree-based clustering is a partition-based technique for identifying clusters. They directly learn groups, such as K-means and K-medoids, which divide the set into multiple groups based on the use of seeds or centroids, and are the focus of this research [3]. Density-based and grid-based algorithms attempt to divide the data space by the number of cells that will be clustered to generate clusters [4]. Finally, there are model-based algorithms that can match data with mathematical models.

Clustering is the process of grouping similar items according to predefined criteria. Each group, called a cluster, consists of data similar to each other, called homogeneous so all groups are different in terms of its data, called heterogeneous. Most authors say that the intra-cluster between the projects is high and the similarity between the clusters is low. It mainly contributes to image processing, spatial data analysis, pattern recognition, especially market research, and supporting navigation [5].

Hierarchical clustering and partition clustering are two clustering algorithms. The nested sequence of partitions is hierarchical grouping which is suitable for top-down and bottom-up methods. Accordingly, the hierarchical grouping is divided into agglomerative grouping and split grouping. The bottom-up strategy is used for cohesive hierarchical techniques, while the top-down method is used for division techniques. Hierarchical grouping uses different metrics, such as link criteria, which specify the dissimilarity in the sets based on the pairwise distances observed in these sets. The linking criteria can be divided into three types: simple linking, average linking, and full linking [6].

Similarity method calculates distance between data objects to group massive amounts of data, and use similarity measures to group them into specific

starting groups. When new data is received, these groups must be updated quickly. The ability to group data in transmission or semi-sorting without a preset number of clusters, and the ability to maintain data clusters gradually and efficiently, are limitations imposed by data evolution. In a dynamic environment, some clustering algorithms are ineffective. There are two main reasons for this situation. First, these strategies assume that the number of clusters is constant, k, however, this parameter will change over time. Second, these methods completely recalculate the cluster attributes [7].

We provide an effective model based on a hierarchical clustering technique that includes three phases; first launching a bundles concept for large-size of requests focus on semi-structured data such as XML and JSON. Second applying pre-processing phase that is responsible for handling noises and overlapping in responses. Then the last Phase include running an agglomerative clustering algorithm, which generates a set of initial clusters that are represented by their centroid and improved agglomerative clustering algorithm to detect outliers. This model has been tested on a real data set and found a remarkable ability to detect abnormal data.

This paper is divided into the following: Section 2 Related work. Section 3 Clustering Analysis Techniques. In Section 4 we describe the proposed model. Section 5 shows the experimental settings and the results. The Conclusions and future work are presented in Section 6.

2. RELATED WORK

Hierarchical clustering algorithm is used in many fields and supervises the understanding of the basics of the data used. The mechanisms of similarity and differences between groups are considered through this technique of hierarchical grouping. All of these mechanisms were reviewed in this study.

Hierarchical clustering methods that consist of bottom-up and top-down methods are the two most common methods. The first method depends on all the objects in the data in one cluster and then the splitting takes place, which continues until each object is placed in a separate cluster in which this is done repeatedly, but the second type involves that each object is in a group and then each group is combined with the other. According to the percentage of similarity between these groups, they are eventually placed in one group, and this is the method I relied on in this research. CURE [8], ROCK [9], BIRCH [10] and Chameleon [11] are representative methods of hierarchical grouping. The ROCK method, which relies on links rather than spaces, is applied to a data set that is characterized by categorical features, but the BIRCH, depending on CF tree, is created before scanning groups that are far from each other and merging dense clusters into leaf nodes. Spherical or different large sizes are Chameleon, CURE.

Chameleon [11] is a method consists of three steps; first a high-quality algorithm h Metis is used in Segmentation graph that represents data objects in a graph structure; second, this algorithm divides the graph into subgroups and then in the third step we get final results by iterative merging of these initial groups that depend on the highest similarity by comparing the similarities between the individual groups.

CURE This method is based on that each group includes a fixed number of points created by selecting very distant points of the mass and then this mass is broken towards the center of mass and the noise is identified and recognized as slow groups and the distances between the closest pair of similarities are calculated from Different groups are then investigated by repeating the grouping process until the number of groups is less than a certain percentage of the original number noise is detected by groups with few points at this time. According to [8], the fraction should be 1/3 of the total number of points in a data set.

H. Singh [12] is a method that depends on birch and K-means. Birch is used to create a tree, then the number of groups is reduced using K-means, and a new method for the grouping process was done by birch, unlike the main algorithm that uses many thresholds instead of only one to eliminate noise data [13].

This algorithm DP [14) is considered new recently proposed, which is calculated for each point of local density and the specific distance, which was measured by calculating the shortest distance between the point and other points of high density, and the value is considered to be the maximum distance between this point and any other points of the point with the highest density, and we can define the centers of mass as being which contains points

with abnormally large values. And after discovering the centers of mass, the mass is assigned to each remaining point of the same thing as the nearest neighbor with the highest density. One of the advantages of this algorithm is that it is able to determine the results of clustering quickly, but among its disadvantages is that it is not able to discover clusters of arbitrary shape. In terms of DP issues, some novel algorithms have been developed [15], [16], [17].

3. CLUSTERING ANALYSIS TEACHNIQUES

Cluster analysis is one of the tasks of data mining and is therefore designed for making data easier to find, recommend, and organize. Clustering technology divides the data set into multiple groups, each group has its own set of attributes [18]. Unlike classification, clustering is an unsupervised learning technique in which similar objects in the data set are grouped to form different groups. Objects in different groups are totally different meanwhile objects in the same group are very similar to each other [19].

3.1 Hierarchical Cluster Analysis

The hierarchical clustering analysis depending on tree structure that produces nested clusters that computed using any distance measures then create distance matrix for all points so the same cluster must contain similar instances, the different clusters must be different instances and all similarity measure must be clear [13].

There is no one clustering algorithm that can solve all problems in batching or real-time nor a formula that can determine which clustering algorithm is best [20].

Before applying any clustering algorithm, we should determine the dataset behaviors and data types for all attributes so some algorithm designed to handle one issue or focus on a particular problem only. Accordingly, this algorithm will almost always fail on a data set with a completely different type of model as required to test on a sample of data before applying algorithm on a dataset.

The basic steps of hierarchical clustering are as follows:

1. if necessary, Compute the proximity matrix.
2. Repeat.
3. Merge the closest two clusters.
4. Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.
5. until only one cluster remains.

The core of the hierarchical clustering algorithm depends on the similarity between objects and uses the distance measure to connect objects and put it in clusters. The hierarchical clustering is called tree-based clustering and the number of clusters is calculated dynamically without putting it as input such as partitioning algorithms that depends on the number of clusters pre-request parameter so each iteration in hierarchical clustering changes the number of clusters. This number will increase in the split grouping function because all data instances start from the same cluster and split at each iteration [21].

The two main types of hierarchical clustering are agglomerative and divisive:

Divisive Method: : In this division method, which depends on the division from top to bottom and assigns all the points in one group and this is its main purpose, then they are divided into two groups at least based on the percentage of similarity between them, and then these groups or clusters are divided into a larger form until there is one group for each one of these points.

Agglomerative Method: This method is called the agglomerative approach and also known as the incremental approach, which starts from the bottom up and sets each point in a separate cluster, then two groups are merged based on calculating the distance or the percentage of similarity between them and this method continues until they are placed in only one cluster in calculating the distance between the groups based usually at Euclidean distance. But other metrics can be chosen to determine distances, such as Manhattan distance, where these methods differ on how to measure the distance between each group.

Single Linkage: This method depends on the minimum distance between the two groups, that is, the distance between the data points closest to the other group is taken as the distance between the two groups.

Complete Linkage*:* In this method, the farthest distance between the points is relied upon.

Average Linkage*:* In this method, it depends on the average distance between each point in one group to each point in the other group, and it is considered as the average of the hierarchical association of the correlation.

3.2 Data Streaming

Data flow is the generation of data continuously and in large quantities and at a high rate during real time. Usually, this data comes from

a source with a series of records that record events when they occur continuously. Examples of this are such as sensors that report the current temperature or weather, and the user clicking a link on a website.

Nowadays, most fields rely entirely on the creation of data streaming. This data processing flow is very useful and it applies to the majority of industrial sectors and big data and usually some of companies are doing simple applications such as collecting system records that are done at the expense of min-max for the initial processing time. These applications provide more advanced processing in almost real time and manipulate data streaming to generate simple reports and responses such as issuing alarms when key metrics exceed certain thresholds. And eventually these applications will lead to more advanced data analysis such as machine learning algorithms to extract deeper insights from the data and apply a complex broadcast and event processing algorithm such as windows decomposing time to find the latest movies, which enriches ideas.

Data flow aggregation contains many challenges, including the following, for example, memory constraints, one-time aggregation, and the difficulty of forecasting. Therefore, techniques for aggregating data flows must meet the following requirements.

- **Real-Time Response**: Required in most data flow applications, such as medical applications, flights and hotel reservations, so in processing this data in real time and making decisions, all this is done in a very short time during the time taken to process the incoming data.
- **Single-Scan**: Traditional data aggregation techniques require reading data multiple times, while data stream aggregation is done by examining the object once because the I/O operations are more expensive than the base memory, so multiple passes are not possible.
- **Handling Noisy Data**: The clustering algorithm must be able to deal with and eliminate the noise that exists in the basic data because it has a significant impact on the quality of the information and another impact on decision-making.

So, in our model handle noises in pre-processing step and detect outliers to improving the hierarchical clustering algorithm.

4. PROPOSED MODEL

A detail of our proposed model is presented .it consist of three phases that apply preprocessing for all response data that responsible of handling noises and remove it then apply improved agglomerative clustering algorithm .in the last phase handle outliers depending on threshold on agglomerative algorithm. the phases are involved in the proposed model as shown in figure 1.

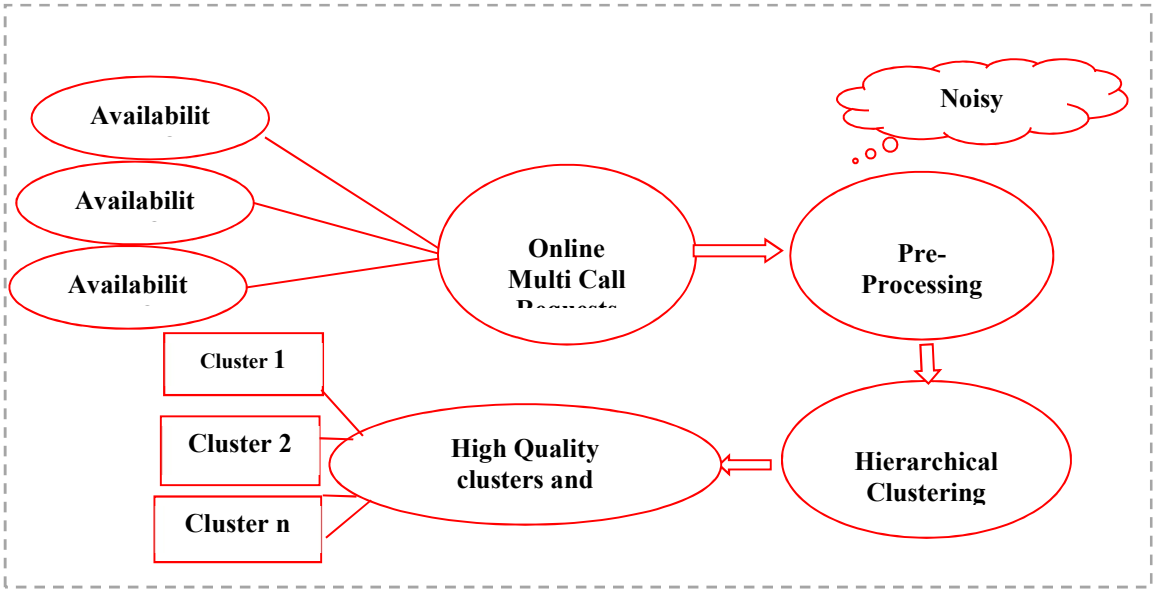


Figure. 1: Proposed Model Of Pre-Processing And Hierarchical Clustering Algorithm

The following are the steps of the proposed model implementation:

1- Determine The number of service systems used by providers to call normally depend on business rules and restrictions such as
   - The provider has a product for the requested service.
   - The provider has a white listed for this service.
   - The provider ranks algorithm according to performance and speed.
   - The status of the connections according to the last x previous connections.
2- Calling providers will not precisely mean that we will make one call for each provider. If the provider protocol permits and the volume of the services or products requested may exceed the capacity of our system or exhaust the provider system resources as the number of calls may reach millions per minutes multiplied by the number of requested services, it would be recommended to split this service to smaller bundles then we will have to make more call for each provider, the consumption of resources of each call will be lighter.
3- Thus, I determine the number of bundles according to the size of the request for each provider that is responsible for memory utilization and according to provider limitations.
4- Launch as many requests as bundles in a fork asynchronous way which we call multi call engine
5- Run a pre-configured daemon for a certain number of seconds in order to receive the responses of the asynchronous calls running in the background.
6- It is worth mentioning that we can also launch other calls to perform other preparatory procedures while we are waiting for the results.
7- Response data that is usually semi-structured data XML or JSON.
8- When receiving each provider, we pass its data through a pre-processed uniform model to structure it according to our data model.
9- This Model detects all overlapping data and remove or fill any missing data according to business logic.
10- Apply improved agglomerative clustering algorithm that applies the Z-Score function to receive normalized data, and then receive all clusters and array of outliers that depending on the threshold as input.

Functional Flow

4.1.1 Semi structured Data requests for Availability

The following step accepts various requests from different suppliers and each request received in different structure.

Some providers send requests in XML or JSON format or in a csv file. Figure 2 introduces samples of XML requests for two providers sending requests through multi-call function that is responsible for handling multi requests for any structured using order system to prevent any delay from the server.

After handling accepted requests which come in different format as described in figure 1, the online multi-call request starts to transform them into a unified format.

Here are two examples of received data in xml formate from two providers answering for the same request regarding flight availability. As we can see the xml is easy to understand but indifferent structure that need to be uniformed in order to able to perform different operations on it such as comparing prices, conditions, ..etc

```
<flightInformation>
        <productDateTime>
                <dateOfDeparture>011221</dateOfDeparture>
                <timeOfDeparture>0330</timeOfDeparture>
                <dateOfArrival>011221</dateOfArrival>
                <timeOfArrival>0700</timeOfArrival>
        </productDateTime>
        <location>
                <locationId>CAI</locationId>
                <terminal>2</terminal>
        </location>
        <location>
                <locationId>FCO</locationId>
                <terminal>3</terminal>
        </location>
        <companyId>
                <marketingCarrier>AZ</marketingCarrier>
                <operatingCarrier>AZ</operatingCarrier>
        </companyId>
        <flightOrtrainNumber>895</flightOrtrainNumber>
        <productDetail>
                <equipmentType>32S</equipmentType>
        </productDetail>
        <addProductDetail>
                <electronicTicketing>Y</electronicTicketing>

        <productDetailQualifier>LCA</productDetailQualifier>
        </addProductDetail>
</flightInformation>
```

```
<FlightSegment>
        <DepDate>01 December</DepDate>
        <DepTime>0805</DepTime>
              <ArrDate>01 December </ArrDate>
        <ArrTime>1240</ArrTime>
        <DepDay>Thu</DepDay>
        <ArrDay>Thu</ArrDay>
        <DepAirport>CAI</DepAirport>
        <DepAirportName>CairoInt'l</DepAirportName>
        <DepCityName>Cairo</DepCityName>
        <ArrAirport>CMN</ArrAirport>
        <ArrAirportName>Casablanca Mohammed V
International Airport</ArrAirportName>
        <ArrCityName>Casablanca</ArrCityName>
        <DepCountry>Egypt</DepCountry>
        <ArrCountry>Morocco</ArrCountry>
        <Airline>AT</Airline>
        <AirName>Royal Air Maroc</AirName>
        <FlightNo>273</FlightNo>
        <BookingClass>V</BookingClass>
        <AirCraftType>738</AirCraftType>
        <ETicket></ETicket>
        <NonStop>0</NonStop>
        <DepTer>2</DepTer>
        <ArrTer>2</ArrTer>
        <AdtFareBasis>VA0W0E1A</AdtFareBasis>
      <ChdFareBasis></ChdFareBasis>
        <InfFareBasis></InfFareBasis>
        <Dur>0535</Dur>
</FlightSegment>
```

We pass the above XML through a special interpreter for each provider and below we can see the uniformed results

```
[Segments][0][DepartureDate]      = 01-12-2021
[Segments][0][DepartureTime]      = 03:30
[Segments][0][ArrivalDate]        = 01-12-2021
[Segments][0][ArrivalTime]        = 07:00
[Segments][0][DepartureAirport]   = CAI
[Segments][0][ArrivalAirport]     = FCO
[Segments][0][FlightNumber]       = 895
[Segments][0][Class]              = Y
[Segments][0][Airline]            = AZ
[Segments][0][Departureterminal]= 2
[Segments][0][ArrivalTerminal]    = 3
```

```
[Segments][1][DepartureDate]      = 01-12-2021
[Segments][1][DepartureTime]      = 08:05
[Segments][1][ArrivalDate]        = 01-12-2021
[Segments][1][ArrivalTime]        = 12:40
[Segments][1][DepartureAirport] = CAI
[Segments][1][ArrivalAirport]     = CMN
[Segments][1][FlightNumber]       = 273
[Segments][1][Class]              = V
[Segments][1][Airline]            = AT
[Segments][1][Departureterminal]= 2
[Segments][1][ArrivalTerminal]    = 2
```

Figure. 2: Sample documents with different structures and tagging

4.1.2 Apply Request Bundles

Request Bundle is a data container that assembles multiple individual requests in a single request message. This means that some providers prefer splitting the availability request based on the provider`s restrictions.

In figure 3, we need to send 4 requests instead of two because some of the requests are more size

so need to split this request into bundles.
Bundles are more powerful and normalized data for all bundles thatare related to one provider so it detects overlapping data in all requests because don't found two bundles contain the same data request and improve memory time.

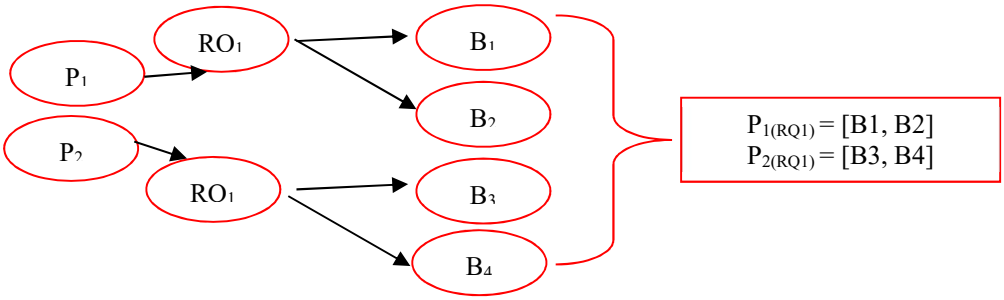


Figure. 3: Bundle's concept

P denoted Provider -RQ denoted the request - B denoted Bundle

Search for hotel availability in London for 2 rooms, 2 adults and 3 childrens for 3 nights knowing that each provider may serve more than 2000 hotels each hotel may have many different types of rooms and for each room different rates and within reach rates may receive different type of meals, the expected result XML for one hotel may be five hundred lines of code including description, conditions, rates,..etc for each option. Imaging that the provider machine needs to search the database and make other queries to third party providers to fetch the request information with weight of data and the execution time it takes can be very slow from the user`s experience and can decrease or block the whole system performance.

Thus, we prefer to split these 2000 hotels into 10 requests each one of them contains only 200 hotels depending on the provider system protocol. This of course lightens the load over both our server and the provider`s server and allows us to serve data to the user in a real time. Without splitting the 2000 hotels per provider in to smaller bundles, the above total time response could reach 50 seconds. But with bundles method we can lower this period of time to 12 seconds.

4.1.3 Pre-Processing Data

Before applying agglomerative clusting algorithm, pre-processing data is more important to handle many related issues such as:

- Missing and redundant data.
- Error responses reeived from some providers by the bundles concept.
- Different Responses structure such as XML or JSON.

Request Bundles Phase

Input: Calls /*the number of providers*/
Output: Requests without redundant items

```
Function GetBundles(Calls){./*Calls meaning the number of providers*/
  Initialize array variable (Bundles)
        Foreach (calls as call)
        Bundles [Call]=The number of permitted items/size of bundle; /*according to provider constraint*/
       Return Bundles;
Function Unique_Requests(Bundles)/*remove any redundant items on requests*/
    Foreach Bundles and determine the number of requests according to the number of bundles {
   Initialize array variable Newcalls=array()
   Append provider data in NewCalls
  Append bundle number in each provider /*show in fig 3*/
    }
```

We utilize the concept of data preprocessing for outlier reduction to build a uniform data function that takes the responses data and handles all previously mentioned issues. In other words, this method is used to optimize the input format for extracting higher quality features from the dataset. Hence this step is crucial as it has a strong effect on the clustering results.

Preprocessing Phase

Input: Responses
Output: Normalized data

```
Convert responses to array
Responses_array[]=xml_or_jsonToArray(Responses)
Responses_array =Array_merge between all responses for all providers
   Foreach  Responses_array as responses
/*this functios  remove any missing data and remove any redundant data on the same response */
       responses=BuildUniformdata(responses);
        NotMissingData[]=Checkmissingdata(responses);
Responses=NotMissingData;
```

Journal of Theoretical and Applied Information Technology
15th December 2021. Vol.99. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.1.4 Apply Hierarchical Clustering Algorithm

Each sample in the data set is considered to be in one cluster, but there are different methods for measuring the distance between two samples and different methods, and we will apply Euclidean distance indicated by the formula 1.

$$\text{dist}((x, y), (a, b)) = \sqrt{(x-a)^2 + (y-b)^2} \quad (1)$$

Here, (x,y),(a,b) denoted two points. The two closest categories are combined into one category and they are the closest. The distance between the two categories is measured in several ways and this is repeated until all categories are collected in one cluster.

Hierarchical Clustering Algorithm

Input: Normalized data, Threshold
Output: clusters

```
Convert responses to array
Responses_array[]=xml_or_jsonToArray(Responses)
    Foreach  Responses_array as responses
/*this function remove any missing data and remove any redundant data on the same response */
        NotMissingData[]=Checkmissingdata(responses);
Responses=NotMissingData;
normalized_data = zscore( Responses ); /*get the normalized data original data */
      construct a distance matrix using the distance values.
      Calculate Euclidian distance of all data in the dataset.
      If distance> Threshold value this data is considered as outlier.
      Outlier[]=datavalue;
      If distance ≤ Threshold value this data is not outlier.
     Pair of clusters with the shortest distance.
     Remove the pair from the matrix and merge them.
     Evaluate all distances from this new cluster to all other clusters, and update the matrix.
     Repeat until the distance matrix is reduced to a single element.
    Destroy outlier array
```

To enhance the accuracy of anomaly detection in clusters and reduce complexity, all while ensuring the quality of the data's original information, the original data should be normalized and reduced. Z-score is used to improve data comparability and standardization of data. After normalization, the average value of the processed data is 0 and the standard deviation is 1. The function is shown in formula 2.

$$Z = \frac{(x-\mu)}{\sigma} \quad (2)$$

Here, μ represents the average value of the samples and σ represents the sample standard deviation and Z-scores follow a standard normal distribution, N (0,1). The Z-score reflects the number of sample standard deviations observation x differs from μ an outlier is identified as $|Z| > 3$, based on the idea that 99.7% of the observations are within 3 standard deviations from the mean in a normal distribution and anything greater would be rejected as an outlier belonging to a different distribution.

5. EXPERIMENTS AND RESULTS

To validate efficiency of our proposed model we applied it to three datasets of customer flights, hotels and sightseeing bookings were obtained from an existing travel platform which is integrated to different booking service provider companies such as (TripAdvisor, Hotelbeds, Amadeus, …etc.). The characteristics of datasets are shown in Table 1. Each company returns

booking data in XML or JSON so to apply our model After gathering data then apply to pre-processing step to remove missing values and redundant data on a dataset that contains more than 3000 flight records and 5000 hotel bookings were collected from 4 window times. Retrieved fight and hotel booking records were joined because hotel reservations or flight tickets were made by the same person.

Table 1: Characteristics of Datasets

Dataset Name	Dataset Characteristics	Number of Instances	Number of Attributes
Hotels	Multivariate	5000	19
Flights	Multivariate	3000	12
Sightseeing	Multivariate	900	8

1)Accuracy

To determine the accuracy, there is a proposed technical performance in the following classification, which depends on the confusion matrix, in the following formulas, True Negative (TN)refers to correctly rejected samples, True Positive (TP) refers to correctly identified samples, False Positive (FP) refers to incorrectly identified samples and False Negative (FN) means incorrectly rejected samples:

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+F\ )} \quad (3)$$

Table 2: Comparative analysis of the best result of the proposed model and the results obtained before Pre-processing in four windows

Datasets	No. of windows	Accuracy	
		Before Pre-processing Step	After Pre-processing Step
Flights	W1	72.45	91.4
	W2	78.6	91.5
	W3	72.45	94.42
	W4	71.45	92.22
Hotels	W1	76.85	93.54
	W2	75.44	94.63
	W3	88.42	92.55
	W4	81.25	94.22
Sightseeing	W1	88.45	96.47
	W2	89.45	95.75
	W3	86.50	96.85
	W4	8.20	94.47

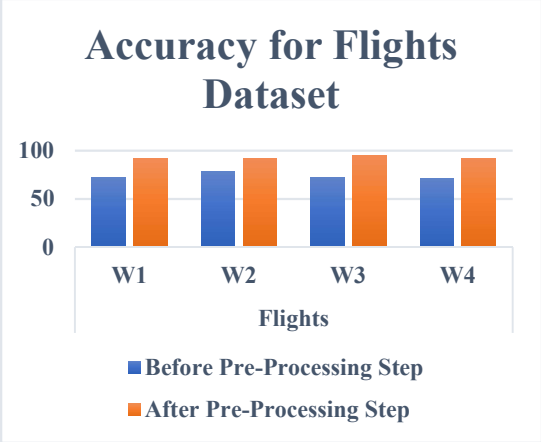


Figure. 4: Accuracy for flights dataset

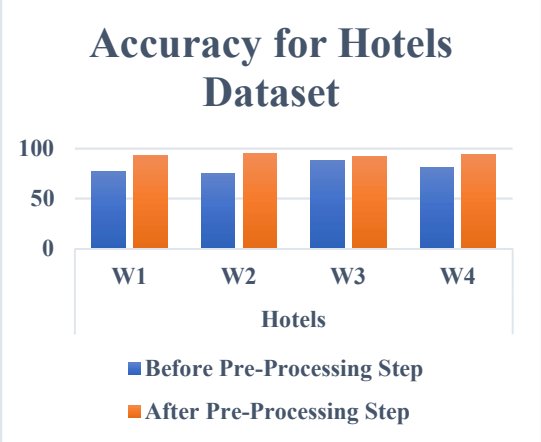


Figure. 5: Accuracy for hotels dataset

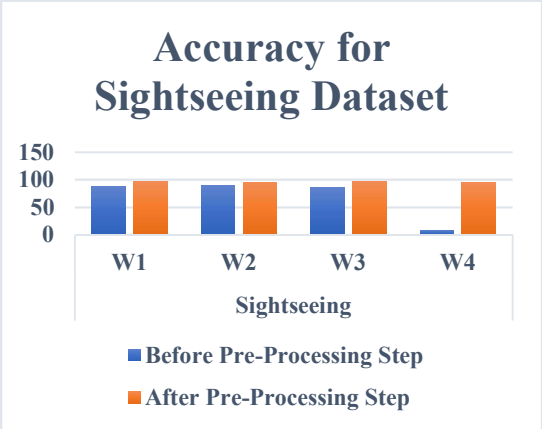


Figure. 6: Accuracy for sightseeing dataset

2)Precision and Recall

Precision means the percentage of positive feedback correctly expected to the total positive feedback High precision relates to the low false-positive rate. Our proposed model We have got 0.94 pretty good precision.

$$\text{Precision} = \frac{(TP)}{(TP+FP)} \quad (4)$$

where TP is the number of true positives and FP is the number of false positives

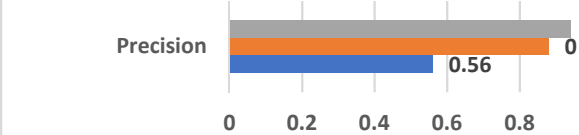
Instead, it is the percentage of all related groups returned by this algorithm. From Figre. 7, it is clear that the proposed model has the highest precision whereas the k-means algorithm has the lowest.

The recall measure always works on a pair of data points and is part of the actual pairs that were identified from the data points and is also directly proportional to the quality of the set.

From Figure. 8, it is clear that the hybrid algorithm has the highest recall whereas the k-means algorithm has the lowest.

$$\text{Recall} = \frac{(TP)}{(TP+FN)} \quad (5)$$

where TP is the number of true positives and FN the number of false negatives.



	Precision
Proposed Model	0.94
Agglomerative	0.88
K-Means	0.56

Figure. 7: Precision Results for Clustering Algorithms

	Precision
Proposed Model	0.94
Agglomerative	0.88
K-Means	0.56

Figure. 8: Recall Results for Clustering Algorithms

3)F-score and execution Time

The quality of clusters is evaluated according to the value of accuracy and the value of recall resulting from the information that was presented in the previous point. Therefore, each cluster is treated as if it were the result of a query. Both accuracy p and r call to the test must be taken into account, and this is to calculate the score p is the number of valid results divided by the number of all results returned and r is the number of valid results divided by the number of valid results that should be returned as shown in Equation 6.

$$\text{F-measure} = \frac{(2*p*r)}{(p+r)} \quad (6)$$

From Figure. 9, it is clear that the hybrid algorithm has the highest F-measure, thus a greater accuracy of clusters formed, whereas the k-means algorithm has the lowest.

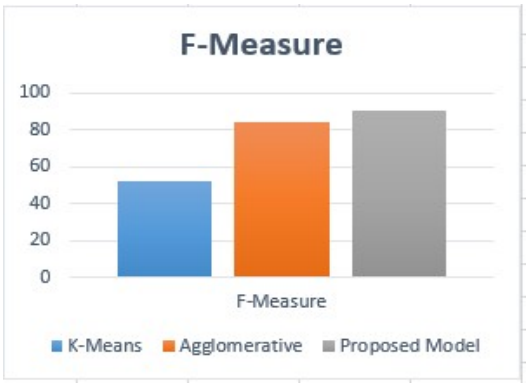


Figure.9: F-measure Results for Clustering Algorithms

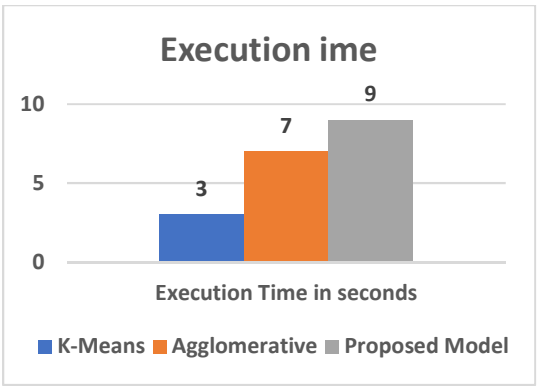


Figure.10: Executions time Results for Clustering Algorithms

From figure.10 The execution time or the complexity for the proposed model take more time from k-means and agglomerative.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have implemented the proposed model that have been detailed above. It has three phases which can save time and optimize the performance. Firstly, it detects and handles missing and overlapping values in preprocessing phase. Secondly, it highly improves the agglomerative clustering algorithm to detect outliers in the data stream is one of the challenging research problems. From the experimental results, it is come to know that the outlier detection and clustering accuracies are more efficient in the proposed model while compared to k-means and agglomerative clustering algorithms that applied on new datasets. But found limitation in our proposed model take more time in execution In the future, we are interested in investigating a good way to automatically determine the density threshold of noise points and reduce our proposed model in execution time.

REFERENCES:

[1] Alshaikhdeeb, B., & Ahmad, K. (2015). Integrating correlation clustering and agglomerative hierarchical clustering for holistic schema matching. Journal of Computer Science, 11(3), 484.

[2] Zubaroğlu, A., & Atalay, V. (2021). Data stream clustering: a review. Artificial Intelligence Review, 54(2), 1201-1236.

[3] Schubert, E., & Rousseeuw, P. J. (2019, October). Faster k-medoids clustering: improving the PAM, CLARA, and CLARANS algorithms. In International conference on similarity search and applications (pp. 171-187). Springer, Cham.

[4] Joshi, A., & Kaur, R. (2013). A review: Comparative study of various clustering techniques in data mining. International Journal of Advanced Research in Computer Science and Software Engineering, 3(3), 55-57.

[5] Ahmad, H. P., & Dang, S. (2015). Performance Evaluation of Clustering Algorithm Using different dataset. International Journal of Advance Research in Computer Science and Management Studies, 8.

[6] Khorshid, S. F., & Abdulazeez, A. M. (2021). breast cancer diagnosis based on k-nearest neighbors: A review. PalArch's Journal of Archaeology of Egypt/Egyptology, 18(4), 1927-1951.

[7] Al-Shammari, A., Zhou, R., Naseriparsaa, M., & Liu, C. (2019). An effective density-based clustering and dynamic maintenance framework for evolving medical data streams. International journal of medical informatics, 126, 176-186.

[8] Bhattacharjee, P., & Mitra, P. (2021). A survey of density based clustering algorithms. Frontiers of Computer Science, 15(1), 1-27.

[9] Gan, G., Ma, C., & Wu, J. (2020). Data clustering: theory, algorithms, and applications. Society for Industrial and Applied Mathematics.

[10] Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases. ACM sigmod record, 25(2), 103-114.

[11] Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., ... & Lin, C. T. (2017). A review of clustering techniques and developments. Neurocomputing, 267, 664-681.

[12] Jarman, A. M. (2020). Hierarchical Cluster Analysis: Comparison of Single linkage, Complete linkage, Average linkage and Centroid Linkage Method.

[13] Ros, F., & Guillaume, S. (2019). A hierarchical clustering algorithm and an improvement of the single linkage criterion to deal with noise. Expert Systems with Applications, 128, 96-108.

[14] Alla, A., Falcone, M., & Saluzzi, L. (2019). An efficient DP algorithm on a tree-structure for finite horizon optimal control problems. SIAM Journal on Scientific Computing, 41(4), A2384-A2406.

[15] Cheng, D., Zhu, Q., Huang, J., Yang, L., & Wu, Q. (2017). Natural neighbor-based clustering algorithm with local representatives. Knowledge-Based Systems, 123, 238-253.

[16] Wang, G., & Song, Q. (2016). Automatic clustering via outward statistical testing on density metrics. IEEE Transactions on Knowledge and Data Engineering, 28(8), 1971-1985.

[17] Xie, J., Gao, H., Xie, W., Liu, X., & Grant, P. W. (2016). Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors. Information Sciences, 354, 19-40.

[18] Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A.Y., Foufou, S. and Bouras, A., 2014. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. IEEE transactions on emerging topics in computing, 2(3), pp.267-279.

[19] Pandove, D., & Goel, S. (2015, February). A comprehensive study on clustering approaches for big data mining. In 2015 2nd international conference on electronics and communication systems (icecs) (pp. 1333-1338). IEEE.

[20] Estivill-Castro, V. (2002). Why so many clustering algorithms: a position paper. ACM SIGKDD explorations newsletter, 4(1), 65-75.

[21] Farinelli, A., Bicego, M., Bistaffa, F., & Ramchurn, S. D. (2017). A hierarchical clustering approach to large-scale near-optimal coalition formation with quality guarantees. Engineering Applications of Artificial Intelligence, 59, 170-185

[22] Nematzadeh, Z., Ibrahim, R., & Selamat, A. (2015, September). A method for class noise

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

detection based on k-means and SVM algorithms. In International Conference on Intelligent Software Methodologies, Tools, and Techniques (pp. 308-318). Springer, Cham.

[23] Nematzadeh, Z., Ibrahim, R., Selamat, A., & Nazerian, V. (2020). The synergistic combination of fuzzy C-means and ensemble filtering for class noise detection. Engineering Computations.