Tribhuvan University

Institute of Science & Technology

Comparative Analysis of Proximity Based Outlier Detection Algorithms

Dissertation

Submitted To:

Central Department of computer Science & information Technology

Tribhuvan University

Kirtipur, Kathmandu

Nepal

In partial Fulfillment of the requirements for the Degree of Master of science in computer science and information technology

Submitted By:

Bhupendra Ram Luhar

November, 2017

Supervisor

Prof. Dr. Subarna Shakya

Pulchowk Campus, IOE

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

Student's Declaration

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work.

… … … … … … …….

Bhupendra Ram Luhar

Date: 12th November, 2017

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

Supervisor's Recommendation

I hereby recommend that the dissertation prepared under my supervision by **Mr. Bhupendra Ram Luhar** entitled **"comparative analysis of Proximity Based Outlier Detection Algorithms"** be accepted as in fulfilling partial requirement for completion of master Degree of science in computer science and information Technology.

… … … … … … … … … …

Prof. Dr. Subarna Shakya

Department of Electronics & Computer Engineering.

Institute of Engineering,

Pulchowk, Nepal

Date: **12th November, 2017**

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

LETTER OF APPROVAL

We certify that we have read this dissertation and in our opinion it is appreciable for the scope and quality as a dissertation in the partial fulfillment for the requirement of Masters Degree in Computer Science and Information Technology.

Evaluation Committee

…… …….

Asst. Prof. Nawaraj Paudel
Head of Department
Central Department of Computer science
& Information Technology
Tribhuvan University
Kirtipur

…….

Prof. Dr. Subarna Shakya
(Supervisor)
Department of Electronics & Computer
Engineering,
Institute of Engineering,
Pulchowk, Nepal

…….
(External Examiner)
Date: 12th November, 2017

…….
(Internal Examiner)

Acknowledgement

I would like to express my sincere thanks to my supervisor **Prof.Dr. Subarna Shakya**, Institute Of Engineering, Pulchowk Campus, Nepal for his support, motivation, suggestions and guidance. His advice was inevitable and with his help I was able to work on my own interested field and complete my thesis on time.

I am also thankful to **Mr. Nawaraj Poudel**, Head of Department, CDCSIT who has provided all the help and facilities, which I required, for the completion of my thesis.

Moreover, I would like to express my heartfelt gratitude to all my teachers at Central Department of Computer Science and Information Technology, Tribhuvan University who have imparted knowledge in various subjects.

Last but not the least; I would like to express my thanks to all my friends and my lovely parents for direct and indirect supports for the completion of this thesis.

Bhupendra Ram Luhar

12th November, 2017

Abstract

Outlier detection is the process of finding peculiar pattern from given set of data. Nowadays, outlier detection is more popular subject in different knowledge domain. Data size is rapidly increases every year there is need to detect outlier in large dataset as early as possible.

In this research, comparison of three different proximity based outlier detection algorithm i.e. distance based method (LDOF), cluster-based method (K-medoid based OD) and density based method (LOF) is presented. The main aim of this research is to evaluate their performance of those three different proximity based outlier algorithm for different dataset with different dimension. The dataset used for this research are chosen such way that they are different in size, mainly in terms of number of instances and attributes. When comparing the performance of all three proximity based outlier detection algorithms, density based method (LOF) is found to be better algorithm to detect outlier in most cases with accuracy level 94.47% as well as 66.93% precision, 83.14% recall and 73.18% F-measure value.

Keywords:

Outlier detection, LDOF, K-medoid based OD, LOF.

Table of Contents

List of Figures

Figure **Page**

List of Tables

Table **Page**

List of Abbreviations

Abbreviations	Full Form
OD	Outlier Detection
LDOF	Local Distance Based Outlier Factor
PAM	Partition around Medoid
LOF	Local Outlier Factor
DS	Data Set
DB	Distance Based
WEKA	Waikato Environment for Knowledge Analysis.
SDK	Software Development Kit
IDE	Integrated Development Environment

CHAPTER 1

1. INTRODUCTION

1.1 Introduction to outlier

Data mining refers to extracting hidden interesting patterns of data from massive data sets. Outlier detection is one of the important task of data mining which is actually find out the data object that are deviating from the common expected behaviors. Outlier detection and analysis is sometime known as outlier mining. An outlier is a data object that is significantly different from the remaining data object in massive data sets. Hawkins [1] provides formal definition of outlier as following:

"An outlier is an observation which deviates so much from other observation as to arose suspicious that it was generated by different mechanism."

Outlier detection is process of identifying the data object, which drastically different from the rest of data set. It is an important data mining task with broad applications such as credit card fraud detection, insurance claim detection, medical diagnosis, image processing, intrusion detection and event detection.

There are various causes of data as an outlier such as data quality poor/ contaminated data, malfunctioning of equipment, manual error and good but exceptional data. The outlier can be mainly categorized into mainly three categories which are point outlier, contextual outlier and collective outlier. If an individual data point or object can be considered as anomalous with respect to rest of the data, then the data point is called point outlier [2]. If data point is a rare occurrence with respect to some specific context and it is a normal occurrence with respect to some another context such type of data point is called contextual outliers. If an particular data point is not anomalous but it's with entire data set is anomalous, then it is called collective outliers.

The point outlier is simplest forms of outlier and is the key focus of majority of research in outlier detection [2]**.** Proximity based outlier detection method is one the best approach to detect the point outliers. The proximity outlier detection method can decomposed into three classes which are:

1.1.1 Distance based approach

Distance based outlier detection techniques are most widely and frequently used techniques and completely depends upon the concept of local neighborhood of data points. Data points with large k-nearest neighbor distance are defined as outlier [2]**.** In distance based method, first distance between data points must be computed with the help of distance measure metrics such as Manhattan distance and Euclidean distance. Secondly declare data point as outlier by nearest neighbor based technique [10]**.**

1.1.2 Cluster based approach

Clustering is the assignment of a set of observations into subsets called clusters so that observations in the same clusters are similar in some sense. It is a useful technique for the discovery of data distribution and patterns in the original data. The goal of clustering technique is to find out both the dense and the sparse region in a data set. It is a method of unsupervised learning and a common technique for statistical data analysis. It is an important technique used for outlier analysis. Outlier detection based on clustering approach provides new positive results. Clustering algorithms are used for outlier detection, where outliers (values that are “far away” from any cluster) may be more interesting than common cases.

1.1.3 Density based approach

Density based method use more complex mechanism as compared to the distance method. It not only finds out the local densities of the point being studied but also the densities of its nearest neighbors [2] Density based outlier detection institutes the density around an outlier data object is significantly different from the density around neighbors. It uses the relative density of a data object against its neighbors as factor the data object as outlier’s algorithm for given dataset is widespread problem.

1.2 Problem statement

Data mining application has got rich focus due to its significance of outlier detection algorithms. The comparison of outlier detection algorithm is complex and open problem. The point outlier detection from given data set is crucial task in data mining nowadays. There are various traditional algorithms for detecting point outlier from the dataset and most of them are vulnerable to detect outliers. So proximity based approach is one of the widely used approaches to detect point outliers. . There are different way define proximity of data object in dataset, which are distance based, density based and cluster based. The selection of best proximity based algorithm for given dataset is widespread problem. So algorithm performance measure parameter like accuracy, precision, recall and F-measure can be used for help in selection of best proximity based outlier detection algorithms for given data set.

1.3 Objective of thesis

The main objectives of this thesis are:

- To detect outlier points from given data set by using Distance based outlier detection algorithm, Cluster based outlier detection algorithm and Density based outlier detection algorithm.
- To perform comparative analysis of these proximity based outlier detection algorithms based on accuracy, precision, recall and F-measure parameters

1.4 Limitation of thesis

Limitations of this research were:

- This study had done by comparison between three approach of proximity based outlier detection algorithms. (LDOF for distance based approach, k-medoid based OD for cluster based approach and LOF for density based approach)
- This research had focused on comparison of Accuracy, Precision, Recall, and F-measure of implemented algorithms.
- All algorithms had implemented in commonly used java programming language.

1.5 Structure of Report

This report is organized is organized in six chapters including the following chapters.

- Chapter 1 of this dissertation work is introduction part, which is organized into subsequent four chapters.
    - First chapter is focused on introduction and overview of outlier and proximity based outlier approach.
    - Second chapter is about problem analysis of existing or previous works which demands further study to get better solutions.
    - Third chapter describe the main objective of this dissertation works.
    - Fourth chapter is about limitation of this dissertation works.
- Chapter 2 contains explanation of all previous studies related to this topic in detail under literature review.
- Chapter 3 includes details of all algorithms to be studied.
- Chapter 4 describes the implementation details.
- Chapter 5 contains all the details of data which is applied for analysis purpose and comparative performance measure of all three different proximity based outlier detection algorithms over three different datasets. The result of the study is shown in tabular form as well as in graph.
- Chapter 6 provides final conclusion and future works of the study.

CHAPTER 2

2. LITERATURE REVIEW

2.1 Background and Literature Review

Outlier detection is very essential of any modeling techniques. A failure to detect outliers or their ineffective handling can have serious effects on the strength of the inferences drained from the technique. There is large number of techniques available to perform this task, and often selection of the most suitable technique poses a big challenge to the practitioner. There is no standard technique for outlier detection. Therefore many approaches have been developed to detect outliers.

Statistical methods are one of the earliest algorithms that can be used by various outlier detection methodologies [3]. One of the single dimensional univariate methods is Grubb's method which is Extreme Studentized Deviate [4] which calculates a z value as the difference between the mean value of the attribute and the query value divided by standard deviation for the attributes. For example a simple statistical scheme for outlier detection is based on the use of box-plot rule. Solberg and Lathi [5] have applied this technique to eliminate outliers in medical laboratory references.

Box-plots graphically depicts groups of numerical data using five quantities: the smallest quantities, lower quartile, median, upper quartile, and largest quantities. In another example, Agrawal and Yu [6] recently proposed a variant of Grubb's test for multivariate data. The Grubb's test computes the distance of the test data points from the estimated sample mean and declares any point with distance above with certain threshold to be an outlier.

Distance-based outlier analysis method works with the assumption that the k-nearest neighbor distance of outlier data points are much larger than normal data points. Knorr and Ng [7] were the first to introduce distance based outlier techniques. An object p in a data set DS is a DB(q, dist)-outlier if at least fraction q of the object in DS lie a greater distance than dist from p. the simplest approach is nested loop, where two arrays are maintained- the first array contains the candidate for outlier data points and another array contains the data point which these candidates are compared in distance based processing.

Once more than k-data points have been identified to lie within distance D from a data point in the first array, that point is determined as non outlier.

Ramaswamy et,al, [8] proposed the extension of Knorr and Ng techniques. All these points are ranked based on the outlier score. Subsequently, Anguilli and Pizzuti [9,10,11] proposed a method to determine the outliers by considering the whole neighborhood of the objects. All the points are ranked based on the sum of distances from the k-nearest neighbors.

Nageswara Rao [12] proposed reverse nearest neighbor approach for distance based outlier detection. In this approach an outlier is defined as a point for which the number of reverse k-nearest neighbor is less than a predefined user threshold.

Breuing et.al [13] proposed a Local Outlier Factor for each each object in the data set, indicating its degree of outlierness. This is the first concept of density based outlier detection algorithms. Since LOF value of each object is obtained by comparing its density with those in its neighborhood.

Zhang et al [14] proposed a local distance-based outlier detection method to find outlier from the data set. The local distance based factor of an object determines the degree to which object deviates from its neighborhood.

Chawala and Gionis [15] presented techniques which is simultaneously cluster and discover outlier in data points. This is generalization of k-means approach. It is an iterative approach and it converges to local optima. This algorithm is not suitable for all similarity measures. However the number outlier cannot be detected automatically.

In [16] proposes a method based on clustering approaches for outlier detection. They first perform the PAM clustering algorithm in that, small clusters are detected in the remaining clusters based on calculating the absolute distances between the results show that their method works well. The paper [17] discusses outlier detection algorithms used in data mining system. Fundamental approaches currently used for solving this problem are considered, and their advantages and disadvantages are discussed. A new outlier detection algorithm is recommended. In presence of outliers, special concentration should be taken to assure the strength of the used estimators [18]**.**

CHAPTER 3

3. RESEARCH METHODOLOGY

3.1 Data collection

There are three different type of data set which are collected from UCI machine learning repository. The data set have been chosen such that they are differ in size , mainly in terms of number of instances and number of attributes.

3.1.1 Dataset 1

The first dataset is small iris data set. The dataset contains 3 attributes apart from the class attributes with 150 instances. But in this research only 60 instances have been taken in which 50 instances are normal and 10 instances taken as outlier.

3.1.2 Dataset 2

The second dataset is medium sized Seed dataset. The dataset contains 7 attributes apart from the class attributes with 210 instances. But in this research only 150 instances have been taken in which 140 instances taken as normal and 10 instances taken as outlier.

3.1.3 Dataset 3

The large data set is Breast Cancer Wisconsin (Diagnostic). The dataset contains 30 attributes apart from the id and class attributes with 569 instances. But in this research only 391 instances have been taken in which 357 instances taken as normal data and 34 instances taken as outlier data.

3.2 Tool Used

All the algorithms are implemented in java language using NetBeans IDE 8.1 with partial use of WEKA' libraries.

3.2.1 Programming language

For the implementation of studied algorithm Java Programming Language is used. Java is general-purpose, concurrent, class-based, object-oriented computer programming language that is specifically designed to have as few implementation dependencies as possible. One

characteristics of java is portability, which means that computer programs written in java language must run similarly on any hardware/operating system platform. This is achieved by compiling the java language code to an intermediate representation called java byte code, instead of directly to platform-specific machine code. Java byte code instructions are analogous to machine code, but they are intended to ne interpreted by a virtual machine written specifically for the host hardware. End-user commonly use a java runtime environment installed on their own machine for standalone java applications, or in a web browser for java applets.

Java is a robust language. It provides many safeguards to ensure reliable code. It has strict compile time and run time checking for data types. It is designed as garbage collected language ease the programmers virtually all memory management problems. Java also incorporates the concept of exception handling which captures series errors and eliminates any risk of crashing the system.

3.2.2 NetBeans IDE

NetBeans is an integrated development environment for java which contains base workspace and extensible plug-in system for customizing the environment. NetBeans SDK is free and open source software mostly written in java. The initial software development can extend its ability by installing plug-ins written for NetBeans Platform, such as development toolkits for other programming languages and can write and contribute their own plug-in modules.

The NetBeans SDK includes the Eclipse Java development tools, offering an IDE with a built-in incremental java compiler and a full model of java source files. This allows advanced refactoring techniques and analysis. It provides the rich client platform for developing general purpose applications.

3.2.3 WEKA Workbench

The WEKA workbench is a collection of state –of-the-art machine learning algorithms and data preprocessing tools [20]. It includes the virtually all ML algorithms. It provides extensive supports for the whole process of experimental data mining, including preparing the input data, evaluating learning schemes statistically, and visualizing the input data and result of learning. As well as a variety of learning algorithms, it includes a wide range of preprocessing tools. This

diverse and comprehensive toolkit is accessed through a common interface so that its user can compare different methods and identify those that are most appropriate for the problem at hand.

WEKA was developed at the University of Waikato in New Zealand; the name stands for Waikato Environment for Knowledge Analysis. The system is written in Java and distributed under the terms of GNU General Public License. It runs on almost any platform and has been tested under Linux, Windows, and Macintosh operating system and even on a personal digital assistant. It provides a uniform interface to many different learning algorithms along with methods for pre- and post- processing and evaluating the result of learning scheme on any given dataset.

3.3. Proximity Outlier Detection Techniques

Proximity based outlier detection techniques define data point as an outlier. The proximity of data point may be defined in variety of ways, which are subtly different from one another. In this dissertation, total three proximity-based algorithms were studied for the analysis of point outlier detection in large datasets. The most common ways of identifying proximity for outlier detection and analysis are as follows:

3.3.1 Distance based Method

The distance of data points to its k-nearest neighbor is used in order to define proximity. Distance based outlier detection techniques are most widely and frequently used techniques and completely depends upon the concept of local neighborhood of data points. Data points with large k-nearest neighbor distance are defined as outlier.

3.3.1.1 Local Distance-Based outlier Detection Factor (LDOF)

LDOF uses the relative distance from a data points to its neighbor to measure how much data point deviate from their neighborhood. The higher outlier factor, it more likely to the data point is an outlier. It is used top n-manner to find the outlier data point based on local distance based outlier factor. The factor LDOF is calculated as follows [11]**:**

LDOF of x_p: The local distance-based outlier factor of x_p is defined as:

$$\text{LDOF}(x_p) = \frac{\bar{d}_{xp}}{\bar{D}_{xp}}$$

$\bar{d}_{xp}$ (KNN distance of x_p): Let N_p be the set of k-nearest neighbor of data point x_p (excluding x_p). the K-nearest neighbors distance of x_p equals the average distance from x_p to all data points in N_p. More formally, let dist(x_i,x_p)>0 be a distance between data points x_i and x_p . The k-nearest neighbors distance of x_p is defined as :

$$\bar{d}_{xp} = \frac{1}{k}\sum_{x_i \in Np} dist(x_i, x_p)$$

$\bar{D}_{xp}$(KNN inner distance of x_p) : Given N_p of data point x_p , the k-nearest neighbors inner distance of x_p is defined as the average distance between in N_p.

$$\bar{D}_{xp} = \frac{1}{k(k-1)}\sum_{x_i, \acute{x}_i \in N_p,\ i \neq i} dist(x_i, x_i)$$

Furthermore, LDOF is used as top n LDOF and follow the following steps:

Input: a given data set D, natural number n and k.

Steps:

1) For each data points x_p in D, retrieve x_p's k- nearest neighbors.
2) Calculate LDOF for each data points x_p . the data points with LDOF< LDOF_{lb} are directly discarded.
3) According to their LDOF values sort the data points.

Output: Highest LDOF values of first n data points.

3.3.2 Cluster based Method

The non–membership of a data point in any cluster, its distance from other clusters, and size of the closet cluster, are used as criteria in order to compute the outlier score. In simple way every data point, is either member of cluster or an outlier. Clustering is the assignment of a set of observations into subsets called clusters so that observations in the same clusters are similar in some sense. It is a useful technique for the discovery of data distribution and patterns in the

original data. The goal of clustering technique is to find out both the dense and the sparse region in a data set.

Outlier detection based on clustering approach provides new positive results. Clustering algorithms are used for outlier detection, where outliers (values that are “far away” from any cluster) may be more interesting than common cases.

3.3.2.1 PAM Algorithm [19]

Input: Set of n data.

Output: Set of k clusters containing all of n data in any of the clusters

1. Initialize: randomly select (without replacement) k of the n data points as the medoids
2. Associate each data point to the closest medoid. ("closest" here is defined using any valid distance metric, most commonly Euclidean distance orManhattan distance)
3. For each medoid m
    1. For each non-medoid data point o
        1. Swap m and o and compute the total cost of the configuration
4. Select the configuration with the lowest cost.
5. Repeat steps 2 to 4 until there is no change in the medoid.

Firstly each cluster with number of elements less than half the number of average elements of all clusters are considered as outliers and all the elements of those clusters are considered as outliers. Secondly, the mean distances between each non-medoid element and medoid element of the remaining clusters are calculated. Then, a threshold value (T) is set for the degree of consideration. Finally, the elements with their distance between medoid greater than corresponding average times the threshold value are considered as outliers.

3.3.2.2 Outlier detection algorithm:

Input: k Clusters of n data (Output from PAM Algorithm)

Output: Set of outliers

1. Set threshold value T
2. For each cluster c
    2.1 If total number of elements is less than half of (n/k), add all the elements into Outlier list

2.2 Else, calculate mean distance from mediod (md) for the cluster and for each non-medoid data point o

2.2.1 If distance from medoid, d is greater than (T*md), add the data point to Outlier list

3.3.3 Density based Method

The density based method is proximity based method using distance metrics. The number of other points within specified local region of data points is used in order to define to local density. These local density values may be converted into outlier scores.

3.3.3.1 Local Outlier Factor (LOF)

The Local Outlier Factor (LOF) [19] algorithm is powerful outlier detection techniques that has been widely used to outlier detection. The LOF algorithm utilizes the concept of local outlier that captures the degree to which an object is outlier based on the density of its local neighborhood. Each data point or object can be assigned LOF values which represent that data object being an outlier. High LOF values are used to identify data object as outlier, whereas low LOF values indicate a normal data object [19]**.** LOF use the relative density of an object against its neighbors as the indicator of the degree of the data object being outliers.

Algorithm of LOF:

Input: A given data set D, natural number n and k.

Step1: For each data point p, compute k-distance (p):

The K-distance(p) is distance to its k-th nearest neighbor. The k-distance (p) provides a measure of density around the data object p, when k-distance of p is small that means the area around p is dense and vice versa.

Step2: Finding K-distance neighborhood of data point p:

The k-distance neighborhood of p contains every data object whose distance for p is not greater than the k-distance.

$$N_{k\_distance(p)}(p) = \{q \in D \backslash \{p\} \mid d(p,q) \leq k\_distance(p)\}$$

Step3: Compute the reachability distance of p with respect to data object O:

For each data object q in the k-distance neighborhood of p, define the reachability distance of p with respect to q as max{k-distance(q), d(p,q)}. The reachability distance of data object p with respect to data object O is:

$reach\_dist_k(p,O) = \max\{\text{k-distance}(O), d(p,O)\}$

Step4 : Compute the local reachability density of data object p:

The local reachability density of an object p is the inverse of average reachabilty distance from the k-nearest neighbors of p.

$$lrd_k(p) = \frac{1}{\left[\frac{\sum_{o \in N_k(p)} reach - dist_k(p,o)}{|N_k(p)|}\right]}$$

Step 5: Compute the Local Outlier Factor of p:

LOF(p) is the average of the ratios of the local reachability density of p and that of p's k-nearest neighbors:

$$LOF_k(p) = \frac{\sum_{o \in N_k(p)} \frac{lrd_k(o)}{lrd_k(p)}}{|N_k(p)|}$$

Step 6: According to their LOF values sort the data points or objects

Output: Highest LOF values of first n data points.

3.4 Comparison Criteria

The comparative analysis or the result is made on the basis of following criteria.

3.3.1 Confusion Matrix:

A confusion matrix is a table for analyzing the result of outlier classifier. It deals with how outlier detection algorithm can recognize tuples of different outlier class (Either yes or no). in order to develop the confusion matrix , the following terms should be considered.

- **True Positive (TP):** Positive tuples that are correctly labeled by the outlier detection algorithm.
- **True Negative (TN):** Negative tuples that are correctly labeled by outlier detection algorithm.
- **False Positive (FP):** Negative tuples that are incorrectly labeled as positive.
- **False Negative (FN):** Positive tuples that are mislabeled as negative.

	Predicted outlier	Predicted Normal
Actual outlier	**TP**	**FN**
Actual Normal	**FP**	**TN**

Table 3-1: Confusion Matrix

3.3.2 Accuracy

Accuracy of outlier detection algorithm on given dataset is percentage of dataset tuples that are correctly classified as outlier or not. It also refers to the recognition rate of the outlier detection algorithm.

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+TN}$$

3.3.3. Precision

Precision refers to the measure of exactness that means what percentage of tuples labeled as positive (or outlier) are actually such.

$$\text{Precision} = \frac{TP}{TP+FP}$$

3.3.4 Recall

Recall refers to the true positive or outlier that means the proportion of positive tuples that are correctly identified.

$$\text{Recall} = \frac{TP}{TP+FN}$$

3.3.5 F- Measure

The F-measure or F-score also refers to F-measures that combines both measures precision and recall as the harmonic mean.

$$\text{F-measure} = \frac{2 \text{ x precision x recall}}{\text{precsion+recall}}$$

CHAPTER 4

4. RESULT, ANALYSIS AND COMPARISONS

4.1 Result Analysis and comparison

In this study, the accuracy of all three algorithms mentioned in chapter 3 is compared for three different dimensional dataset mentions in chapter 3.1 which is compared based on accuracy, precision, recall and F-measure. The results were achieved by using whole test dataset for different outlier algorithms.

4.1.1 Comparison results of proximity-based outlier detection algorithms for dataset1

Table 4-2 provides the summery output for comparison of all three algorithms studied over dataset (i.e. Iris dataset). After long observations for choosing the value for k (i.e., k nearest neighbor for LOF and LDOF and number of cluster for k-mediods OD algorithms). The optimal value of comparison parameter of all three algorithm occurred at k=12.

Algorithms	Accuracy	Precision	Recall	F- measure
LDOF	88.33%	61.54%	80%	68.57%
K-medoid based OD	90%	64.24%	90%	74.97%
LOF	93.33%	71.41%	100%	83.33%

Table 4-1: Result of outlier Detection for Data set 1.

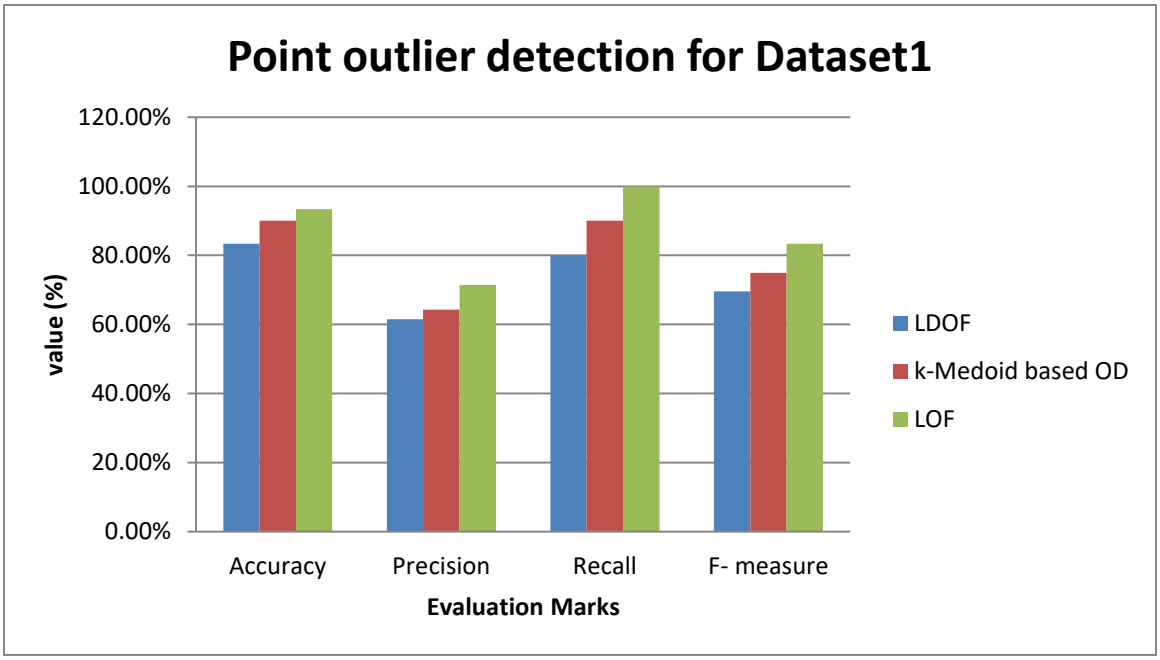

Figure 4-1: Graph of table 4-1.

Based on the Figure 4-1, it is clearly seen that the accuracy value of LOF is got high label of 93.33% and LDOF got less accuracy of label 88.33%. In case of precision and the Recall value of implemented proximity based outlier detection algorithms LOF had got high precision and recall level of 71.41% and 100% respectively. Whereas LDOF got less precision and recall level of 61.54% and 80 % respectively.

Figure 4-1 also show the F-measure of table 4-2 observed by implemented proximity based outlier detection algorithms where it ranges from 69.57% to 83.33%. Again LOF had got a victory over compared algorithms with value of 83.33% and LDOF had got minimum value of 69.57%

4.1.2 Comparison results of proximity based outlier detection algorithms for dataset 2:

Table 4-3 provides the summery output for comparison of all three algorithms studied over dataset (i.e. Iris dataset). After long observations for choosing the value for k (i.e., k nearest neighbor for LOF and LDOF and number of cluster for k-mediods OD algorithms). The optimal value of comparison parameter of all three algorithm occurred at k=20.

Algorithms	Accuracy	Precision	Recall	F- measure
LDOF	91.33%	38.46%	50%	43.48%
K-medoid based OD	94%	54.55%	60%	57.15%
LOF	94.67%	58.33%	70%	63.63%

.

Table 4-2 Result of outlier detection algorithm for dataset 2

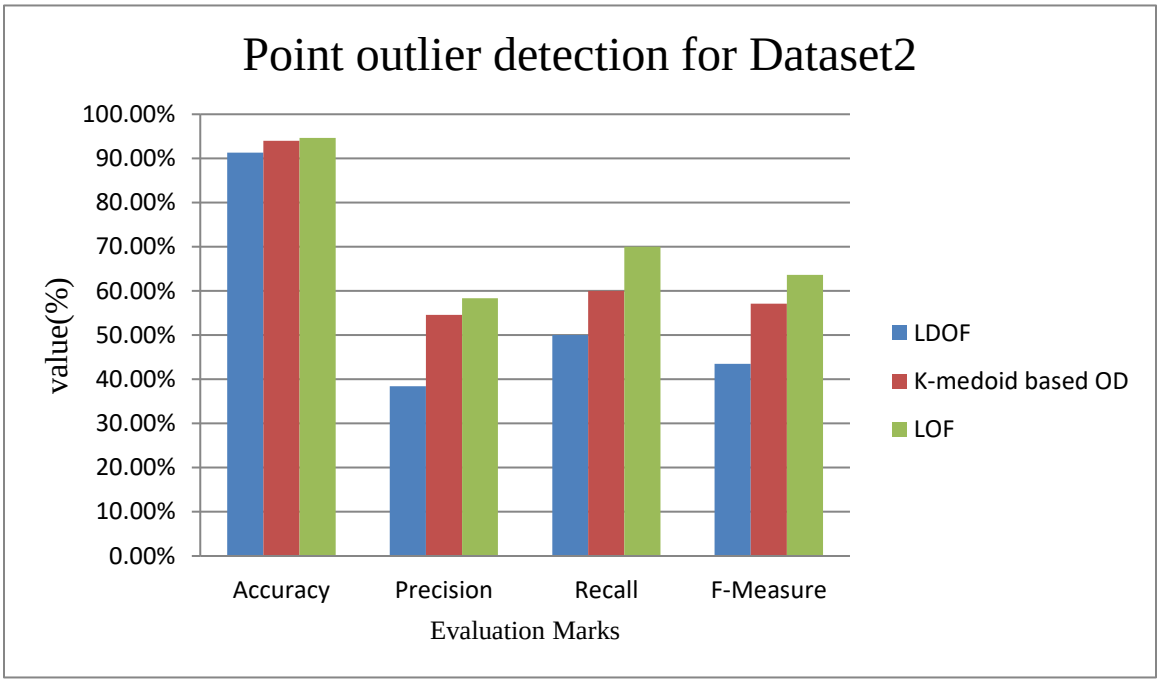

Figure 4-2: Graph of table 4-2.

Based on the Fig 4-2, it is clearly seen that the accuracy value of LOF is got high level of 94.67% and LDOF got less accuracy of label 91.33%. In case of precision and the Recall value of implemented proximity based outlier detection algorithms LOF had got high precision and recall level of 58.33% and 70% respectively. Whereas LDOF got less precision and recall level of 38.46% and 50 % respectively.

Fig 4-2 also show the F-measure of table 4-2 observed by implemented proximity based outlier detection algorithms where it ranges from 43.48% to 63.63%. Again LOF had got a victory over compared algorithms with value of63.63% and LDOF had got minimum value of 43.48%.

4.1.3 Comparison results of proximity based outlier detection algorithms for dataset 3

Table 4-3 provides the summery output for comparison of all three algorithms studied over dataset (i.e. Breast cancer Wisconsin (Dogonistic) dataset). After long observations for choosing the value for k (i.e., k nearest neighbor for LOF and LDOF and number of cluster for k-mediods OD algorithms). The optimal value of comparison parameter of all three algorithm occurred at k=35.

Algorithms	Accuracy	Precision	Recall	F- measure
LDOF	91.13%	50.00%	61.77%	55.27%
K-medoid based OD	88.52%	41.94%	76.47%	54.17%
LOF	95.41%	71.05%	79.41%	74.99%

Table 4-3: Result of outlier Detection for Data set 3.

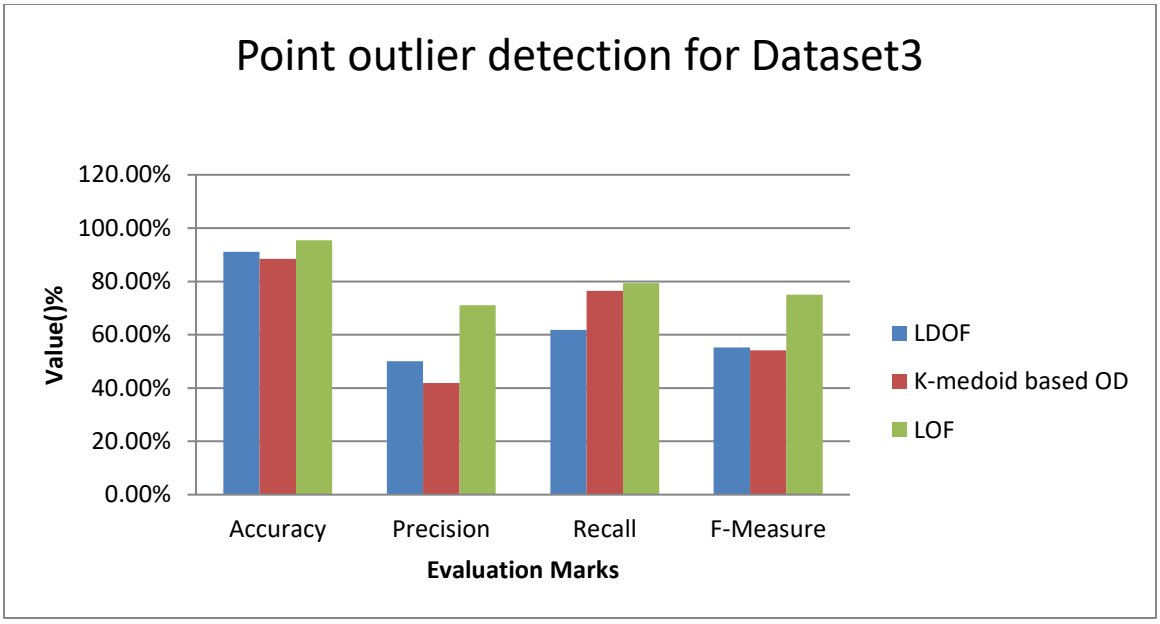

Figure 4-3: Graph of table 4-3

Based on the Figure 4-3, it is clearly seen that the accuracy value of LOF is got high level of 95.41% and K-medoid based OD got less accuracy of label 88.53%. In case of precision and the Recall value of implemented proximity based outlier detection algorithms LOF had got high

precision and recall level of 71.41% and 79.41% respectively. Whereas LDOF got less recall level of 61.77% and K-medoid based OD had got less precision level of 41.94%

Figure 4-3 also show the F-measure of table 4-4 observed by implemented proximity based outlier detection algorithms where it ranges from 54.17% to 74.99%. Again LOF had got a victory over compared algorithms with value of 74.99% and K-medoid based OD had got minimum value of 54.17%.

4.1.4 Comparison of average results of proximity based outlier detection algorithms:

Table 4-3 provides the summery of average output for comparison of all three algorithms studied over different three dataset.

Algorithms	Accuracy	Precision	Recall	F- measure
LDOF	90.26%	50.00%	63.92%	55.77%
K-medoid based OD	90.84%	53.58%	75.46%	62.10%
LOF	94.47%	66.93%	83.14%	73.98%

Table 4-4: Averages result of outlier Detection for all dataset.

Table 4-5 showed that comparisons between averages of all evaluation metrics of all implemented proximity based outlier detection algorithms. From that comparison, LOF had got rich as well as motivating and encouraging performance in every aspect, whereas LDOF had got minimum or less performance in every aspect as compared to LOF and K-medoid based outlier detection algorithm.

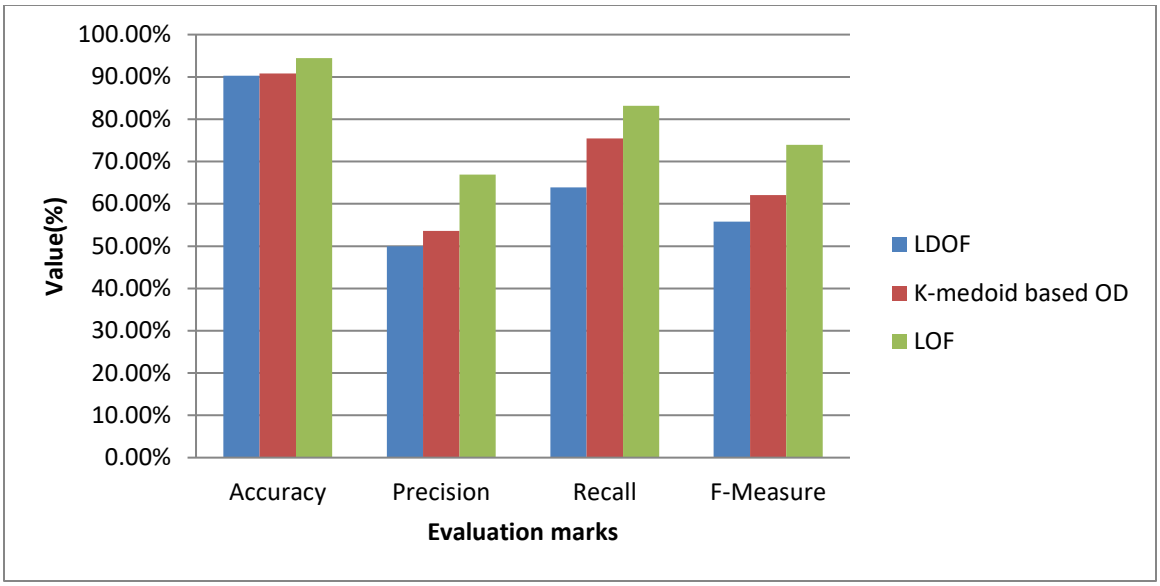

Figure 4-4: Graph of table 4-4

CHAPTER 5

5. CONCLUSION

5.1. Conclusion

In this research, the comparative analysis of proximity-based outlier detection algorithms (i.e. distance based method (LDOF), cluster-based method (K-mediod based OD), density based method (LOF)) using various measure parameter like accuracy, precision, recall and F-measures over the three different dataset with different dimension and size are evaluated. From the result analysis, density based method (LOF) has higher accuracy as well as higher precision, recall and F-measure with level of 94.47 %, 66.93%, 83.14 and 73.98% respectively on average as compared to the cluster based method (K-medoid based OD) and distance based method(LDOF). However distance based method has less accuracy as well as precision, recall and F-measure on average for all dataset.

On balance scale, Density based method (LOF) algorithm has predicted better outlier result than other proximity based outlier detection algorithms studied for all dataset.

More algorithms from the proximity based outlier detection can be incorporated for further study to be studied dataset or other dataset which have numeric as well as categorical value. Moreover some algorithms can be customized for the specific domain so that outlier detection could have more accurate and reliable.

References

[1] Hawkins D. Identification of Outliers, *Chapman and Hall*, 1980.

[2] Malik K., H.Sadawart Kalra G.S. "Comparative Analysis of Outlier Detection Techniques" International Journal of Computer Applications (0975 – 8887) Volume 97– No.8, July 2014

[3] Barnett, V. and Lewis, T.: 1994, Outliers in Statistical Data. John Wiley & Sons.3rd edition.

[4] Huber, P. 1974. Robust Statistics.Wiley, New York.

[5] Solberg E. Lathi H et.al , Detection of outlier in reference distribution: performance of Horns algorithm. Clinical chemistry 51. 2326-32. 10.1373/clinchem.2005.058339.

[6] Aggarwal, Charu & Yu, Philip. (2008). Outlier Detection with Uncertain Data. 483-493. 10.1137/1.9781611972788.44.

[7] Knorr E. M and Ng. R. T. ,Algorithms for mining distancebased outliers in large datasets. In Proc. 24th Int. Conf. Very Large Data Bases, VLDB, pages 392–403, 1998.

[8] Ramaswamy S., Rastogi, R. and Shim K.. Efficient algorithms for mining outliers from large data sets. pages 427–438, 2000.

[9] Angiulli F., Basta S., and Pizzuti. C., Distance-based detection and prediction of outliers. IEEE Transactions on Knowledge and Data Engineering, 18:145–160, 2006.

[10] Angiulli F. and Pizzuti C., Fast outlier detection in high dimensional spaces. In PKDD '02: Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery, pages 15–26, 2002.

[11] Angiulli F. and Pizzuti C., Outlier mining in large high dimensional data sets. IEEE Transactions on Knowledge and Data Engineering, 17:203–215, 2005.

[12] Nageswar Rao A., unsupervised distance-based outlier detection in reverse nearest neighbor, South Asian Journal of Engineering and Technology vol2 (2016)

[13] Breuning M., Kriegel H-P., R. Ng, and Sander J., LOF: Identifying Density-Based Local Outliers. In Proc. of 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD'00), Dallas, Texas, pp 93-104, 2000.

[14] Zhang K, Hutter M., and Jin H. A new local distance-based outlier detection approach for scattered real-world data. In PAKDD '09: Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, pages 813–822, 2009

[15] Al-Zoubi, M. (2009) An Effective Clustering-Based Approach for Outlier Detection, European Journal of Scientific Research.

[16] Petrovskiy I. Outlier Detection Algorithms in Data Mining Systems., Department of Computational Mathematics and Cybernetics, Moscow State University, Vorob‟evy gory, Moscow, 119992 Russia.e-mail: michael@cs.msu.suReceived February 19, 2003.

[17] Ben-Gal I., OUTLIER DETECTION, Department of Industrial Engineering,Tel-Aviv University,Ramat-Aviv, Tel-Aviv 69978, Israel.

[18] Theodoridis S. and Koutroumbas K., Pattern Recognition 3rd ed. (2006). p. 635.

[19] Alshawabkeh M., Jang B., and Kaeli D., "Accelerating the Local Outlier Factor Algorithm on a GPU for Intrusion Detection Systems", *GPGPU-3* March 14, 2010, Pittsburg, PA, USA.

[20] V. VeeraLaxmi, Dr. D. Ramyachitra, Ripple Down Rule Learner (RIDOR) classifier for IRIS Dataset, International Journal of Computer Science and Engineering (IJCSE), 2015.

[21] http:// www.cs. Waikato.ac.nz/ml/WEKA/

APPENDIX

(Sample Dataset)

1 Sample data for dataset1 (IRIS dataset)

Att1	Att2	Att3	Att4
5.1	3.5	1.4	0.2
4.9	3	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5	3.6	1.4	0.2
5.4	3.9	1.7	0.4
4.6	3.4	1.4	0.3
5	3.4	1.5	0.2
4.4	2.9	1.4	0.2
4.9	3.1	1.5	0.1
5.4	3.7	1.5	0.2
4.8	3.4	1.6	0.2
4.8	3	1.4	0.1
4.3	3	1.1	0.1
5.8	4	1.2	0.2
5.7	4.4	1.5	0.4
5.4	3.9	1.3	0.4
5.1	3.5	1.4	0.3
5.7	3.8	1.7	0.3
5.1	3.8	1.5	0.3
5.4	3.4	1.7	0.2
5.1	3.7	1.5	0.4
4.6	3.6	1	0.2
5.1	3.3	1.7	0.5
4.8	3.4	1.9	0.2
5	3	1.6	0.2
5	3.4	1.6	0.4
5.2	3.5	1.5	0.2
5.2	3.4	1.4	0.2
4.7	3.2	1.6	0.2
4.8	3.1	1.6	0.2
5.4	3.4	1.5	0.4
5.2	4.1	1.5	0.1
5.5	4.2	1.4	0.2
4.9	3.1	1.5	0.1
5	3.2	1.2	0.2

5.5	3.5	1.3	0.2
4.9	3.1	1.5	0.1
4.4	3	1.3	0.2
5.1	3.4	1.5	0.2
5	3.5	1.3	0.3
4.5	2.3	1.3	0.3
4.4	3.2	1.3	0.2
5	3.5	1.6	0.6
5.1	3.8	1.9	0.4
4.8	3	1.4	0.3
5.1	3.8	1.6	0.2
4.6	3.2	1.4	0.2
5.3	3.7	1.5	0.2
5	3.3	1.4	0.2
7.1	3	5.9	2.1
6.3	2.9	5.6	1.8
6.5	3	5.8	2.2
7.6	3	6.6	2.1
4.9	2.5	4.5	1.7
7.3	2.9	6.3	1.8
6.7	2.5	5.8	1.8
7.2	3.6	6.1	2.5
6.5	3.2	5.1	2
6.4	2.7	5.3	1.9

2 Sample data for dataset2 (Seed Data)

att1	att2	att3	att4	att5	att6	att7
15.26	14.84	0.871	5.763	3.312	2.221	5.22
14.88	14.57	0.8811	5.554	3.333	1.018	4.956
14.29	14.09	0.905	5.291	3.337	2.699	4.825
13.84	13.94	0.8955	5.324	3.379	2.259	4.805
16.14	14.99	0.9034	5.658	3.562	1.355	5.175
14.38	14.21	0.8951	5.386	3.312	2.462	4.956
14.69	14.49	0.8799	5.563	3.259	3.586	5.219
14.11	14.1	0.8911	5.42	3.302	2.7	5
16.63	15.46	0.8747	6.053	3.465	2.04	5.877
16.44	15.25	0.888	5.884	3.505	1.969	5.533
15.26	14.85	0.8696	5.714	3.242	4.543	5.314
14.03	14.16	0.8796	5.438	3.201	1.717	5.001
13.89	14.02	0.888	5.439	3.199	3.986	4.738

13.78	14.06	0.8759	5.479	3.156	3.136	4.872
13.74	14.05	0.8744	5.482	3.114	2.932	4.825
14.59	14.28	0.8993	5.351	3.333	4.185	4.781
13.99	13.83	0.9183	5.119	3.383	5.234	4.781
15.69	14.75	0.9058	5.527	3.514	1.599	5.046
14.7	14.21	0.9153	5.205	3.466	1.767	4.649
12.72	13.57	0.8686	5.226	3.049	4.102	4.914
14.16	14.4	0.8584	5.658	3.129	3.072	5.176
14.11	14.26	0.8722	5.52	3.168	2.688	5.219
15.88	14.9	0.8988	5.618	3.507	0.7651	5.091
12.08	13.23	0.8664	5.099	2.936	1.415	4.961
15.01	14.76	0.8657	5.789	3.245	1.791	5.001
16.19	15.16	0.8849	5.833	3.421	0.903	5.307
13.02	13.76	0.8641	5.395	3.026	3.373	4.825
12.74	13.67	0.8564	5.395	2.956	2.504	4.869
14.11	14.18	0.882	5.541	3.221	2.754	5.038
13.45	14.02	0.8604	5.516	3.065	3.531	5.097
13.16	13.82	0.8662	5.454	2.975	0.8551	5.056
15.49	14.94	0.8724	5.757	3.371	3.412	5.228
14.09	14.41	0.8529	5.717	3.186	3.92	5.299
13.94	14.17	0.8728	5.585	3.15	2.124	5.012
15.05	14.68	0.8779	5.712	3.328	2.129	5.36
16.12	15	0.9	5.709	3.485	2.27	5.443
16.2	15.27	0.8734	5.826	3.464	2.823	5.527
17.08	15.38	0.9079	5.832	3.683	2.956	5.484
14.8	14.52	0.8823	5.656	3.288	3.112	5.309
14.28	14.17	0.8944	5.397	3.298	6.685	5.001
13.54	13.85	0.8871	5.348	3.156	2.587	5.178
13.5	13.85	0.8852	5.351	3.158	2.249	5.176
13.16	13.55	0.9009	5.138	3.201	2.461	4.783
15.5	14.86	0.882	5.877	3.396	4.711	5.528
15.11	14.54	0.8986	5.579	3.462	3.128	5.18
13.8	14.04	0.8794	5.376	3.155	1.56	4.961
15.36	14.76	0.8861	5.701	3.393	1.367	5.132
14.99	14.56	0.8883	5.57	3.377	2.958	5.175
14.79	14.52	0.8819	5.545	3.291	2.704	5.111
14.86	14.67	0.8676	5.678	3.258	2.129	5.351
14.43	14.4	0.8751	5.585	3.272	3.975	5.144
15.78	14.91	0.8923	5.674	3.434	5.593	5.136
14.49	14.61	0.8538	5.715	3.113	4.116	5.396
14.33	14.28	0.8831	5.504	3.199	3.328	5.224
14.52	14.6	0.8557	5.741	3.113	1.481	5.487
15.03	14.77	0.8658	5.702	3.212	1.933	5.439

14.46	14.35	0.8818	5.388	3.377	2.802	5.044
14.92	14.43	0.9006	5.384	3.412	1.142	5.088
15.38	14.77	0.8857	5.662	3.419	1.999	5.222
12.11	13.47	0.8392	5.159	3.032	1.502	4.519
11.42	12.86	0.8683	5.008	2.85	2.7	4.607
11.23	12.63	0.884	4.902	2.879	2.269	4.703
12.36	13.19	0.8923	5.076	3.042	3.22	4.605
13.22	13.84	0.868	5.395	3.07	4.157	5.088
12.78	13.57	0.8716	5.262	3.026	1.176	4.782
12.88	13.5	0.8879	5.139	3.119	2.352	4.607
14.34	14.37	0.8726	5.63	3.19	1.313	5.15
14.01	14.29	0.8625	5.609	3.158	2.217	5.132
14.37	14.39	0.8726	5.569	3.153	1.464	5.3
13.07	13.92	0.848	5.472	2.994	5.304	5.395
13.32	13.94	0.8613	5.541	3.073	7.035	5.44
13.34	13.95	0.862	5.389	3.074	5.995	5.307
12.22	13.32	0.8652	5.224	2.967	5.469	5.221
11.82	13.4	0.8274	5.314	2.777	4.471	5.178
11.21	13.13	0.8167	5.279	2.687	6.169	5.275
11.43	13.13	0.8335	5.176	2.719	2.221	5.132
12.49	13.46	0.8658	5.267	2.967	4.421	5.002
12.7	13.71	0.8491	5.386	2.911	3.26	5.316
10.79	12.93	0.8107	5.317	2.648	5.462	5.194
11.83	13.23	0.8496	5.263	2.84	5.195	5.307
12.01	13.52	0.8249	5.405	2.776	6.992	5.27
12.26	13.6	0.8333	5.408	2.833	4.756	5.36
11.18	13.04	0.8266	5.22	2.693	3.332	5.001
11.36	13.05	0.8382	5.175	2.755	4.048	5.263
11.19	13.05	0.8253	5.25	2.675	5.813	5.219
11.34	12.87	0.8596	5.053	2.849	3.347	5.003
12.13	13.73	0.8081	5.394	2.745	4.825	5.22
11.75	13.52	0.8082	5.444	2.678	4.378	5.31
11.49	13.22	0.8263	5.304	2.695	5.388	5.31

3 Sample Data for dataset 3 (Breast cancer Wisconsin (Diagnostic) data):

att1	att2	att3	att4	att5	att6	att7	att8	……	att30
13.54	14.36	87.46	566.3	0.09779	0.08129	0.06664	0.04781	……	0.05766
13.08	15.71	85.63	520	0.1075	0.127	0.04568	0.0311	……	0.06811