Outlier Based Fraud Detection System

Prerana M More, Mrudul K Rushipathak, Shweta G Bairagi, Trupti S Nalnikar
Department of Computer Engineering
KKWIEER, Nashik
Email id: prerana.more2012@gmail.com, mrudul40@gmail.com , shwetabairagi12@gmail.com,
trupti.nalnikar@gmail.com

Abstract

Data mining has the vital task of Outlier detection which aims to detect an outlier from given datasets. The analysis or detection of outlier data is referred to as Outlier Mining. In Data mining, outlier detection is the identification of unusual or distant data records that might be require further investigation or analysis. This paper provides the data driven methods for various fraud detection systems based on literature review, fraudulent activities or cases and comparative research. Outlier detection is the technique which discovers such type of data from the given data set. Several techniques of outlier detection have been introduced which requires input parameter from the user. The goal of this proposed work is to partition the input data set into the number of clusters using K-NN algorithm. Then the clusters are given as an input to the outlier detection methods namely cluster based outlier algorithm and Local Outlier Factor Algorithm. The Performance evaluation of this algorithm confirms that our approach of finding local outliers can be practically implemented.

Keywords: *Data Clustering, K-NN, Outlier Detection, ODA, LOF*

INTRODUCTION

A number of surveys, review articles and books cover outlier detection techniques in the field of data mining. In this paper we make an attempt to bring together many different outlier detection techniques in a structured and generic description.

Outlier detection has been used for long time to find and remove identical observations from data. Outliers are generated due to system faults, errors in data entry, fraudulent behaviour, data scaled inappropriately or simply through natural deviations in populations. Their detection can identify system faults and fraud before they escalate with potentially identical consequences. It can identify errors and remove their significant effect on the data set and as such to purify the data for processing. The earlier outlier detection methods were complexed but now, different techniques are used, related to Statistics. In this paper, we introduce different algorithms for outlier detection. We identify the motivations of it and differentiate among their advantages and disadvantages in a comparative review.

The LOF algorithm is an outlier detection method which is unsupervised method and it calculates the density deviation of a given data point with respect to its neighbors in the database. It notes the data as outlier samples that have a relatively lower density than their neighbors in the database.

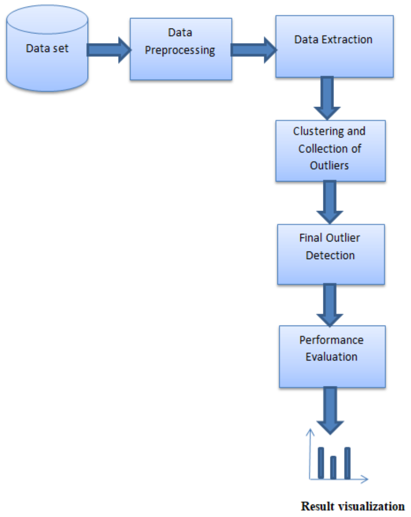
In this paper basic methodologies currently used for solving this problem are considered, and their advantages and disadvantages are discussed. It is based on methods of fuzzy set theory and the use of kernel functions and possesses a number of advantages compared to the existing methods. The performance of the algorithm used is studied by the example of outlier detection arising in credit card systems, the so-called fraud detection systems.

Literature Survey

a. Kadam, Pund and Aggarwal went through various approaches to detect outliers including the cluster-based approach.
b. Aparna and Nair proposed the CHB-K-Means algorithm by using a weighted attribute matrix to detect outliers.
c. Jiang et al. introduced a two-phase clustering algorithm for outlier detection. In the first phase, the k-means algorithm is changed to partition the data in such a way that a data point is assigned to be a new cluster center only if the data point is far away from all clusters. In the second phase, a minimum spanning tree is constructed based on the cluster centers obtained from the first phase.
d. Zhang et al. (2009) proposed a measure called Local Distance-based Outlier Factor (LDOF) to measurethe outliers present in scattered data.
e. He et al. declared the concept of cluster based local outlier and also designed a measure called clusterbased local outlier factor (CBLOF) .

Proposed Methodology

The proposed system works in three phase's viz. preprocessing, clustering and outlier detection. Before performing clustering input dataset is preprocessed and then the clustering is performed using k-nn algorithm. After clustering, each cluster out of k clusters is given as an input to the outlier detection methods. There are different methods that are used for detecting an outlier. Output provided is set of outliers from input dataset. System architectural diagram is explained below.



Step 1 :Pre-processing

Input : Open the file or URL to be processed.

Dataset consists of numeric data from various sources. Dataset should be in xml or csv format. The data can be generated by different transactions, inventory management systems and DBMS.

User need to mention attributed which are unwanted or don't play any role in outlier detection as well as mark class attribute from imported file.

Save the file to proceed further.

Processing : System will now remove undesirable attributes and prepare the file for further processing.

Step 2 : Outlier Detection Algorithm

Input : User has to click on Start Algorithm

Processing : Data Records which may contain outliers are detected and differentiated from remaining records

Step 3 : LOF Algorithm

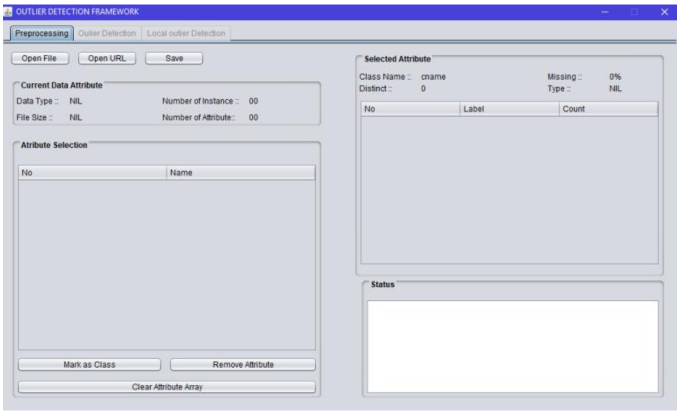
Input : Records which may contain outliers are considered for execution of Local Outlier Factor algorithm.

User has to mention Number of Parameters for KNN algorithm

Processing: KNN algorithm will be run followed by LOF to detect outliers

Detected outliers will be displayed followed by time in ms

Results



Current Data Attribute

Data Type :: NIL
Number of Instance :: 00
File Size :: NIL
Number of Attribute:: 00

Atribute Selection

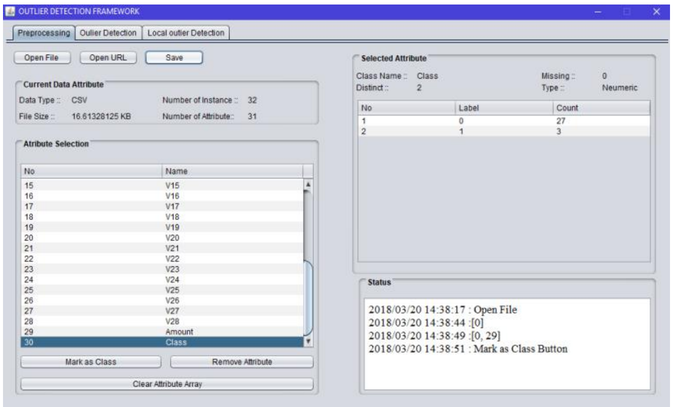
No	Name

Selected Attribute

Class Name :: cname
Missing :: 0%
Distinct :: 0
Type :: NIL

No	Label	Count

After selecting a file Remove unwanted attributes &mark some attributes as class and save the file



Current Data Attribute

Data Type :: CSV
Number of Instance :: 32
File Size :: 16.61328125 KB
Number of Attribute:: 31

Atribute Selection

No	Name
15	V15
16	V16
17	V17
18	V18
19	V19
20	V20
21	V21
22	V22
23	V23
24	V24
25	V25
26	V26
27	V27
28	V28
29	Amount
30	Class

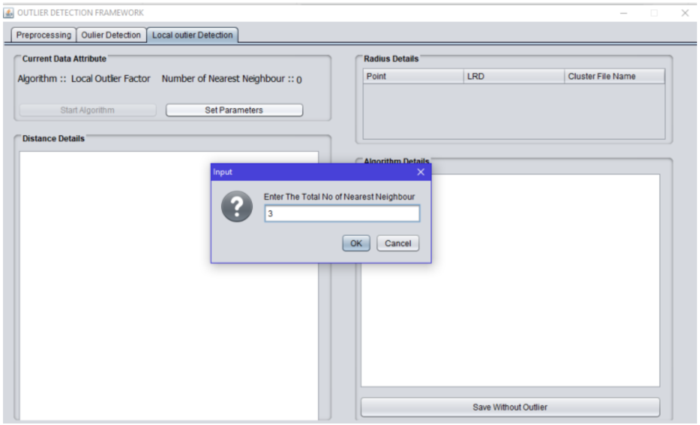
Selected Attribute

Class Name :: Class
Missing :: 0
Distinct :: 2
Type :: Neumeric

No	Label	Count
1	0	27
2	1	3

```
2018/03/20 14:38:17 : Open File
2018/03/20 14:38:44 :[0]
2018/03/20 14:38:49 :[0, 29]
2018/03/20 14:38:51 : Mark as Class Button
```

In order to preprocess dataset we have to select & import file through Open file or Open URL



Current Data Attribute

Algorithm :: Local Outlier Factor
Number of Nearest Neighbour :: 0

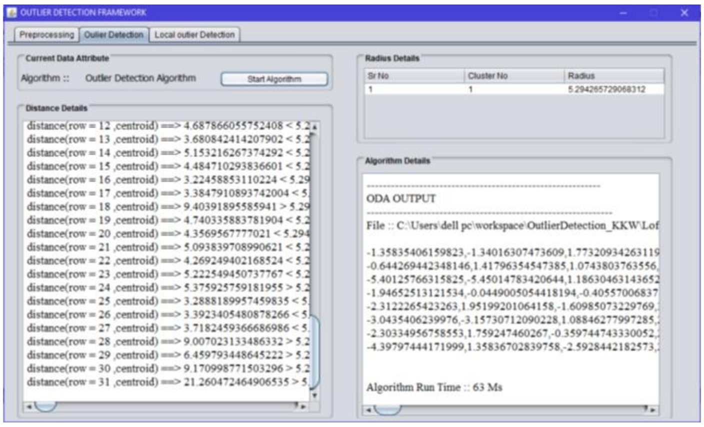
Radius Details

Point	LRD	Cluster File Name

Enter The Total No of Nearest Neighbour

3

After clicking on Outlier detection tab/button ODA algorithm will run and entire dataset will be checked and suspicious records will be extracted for further processing



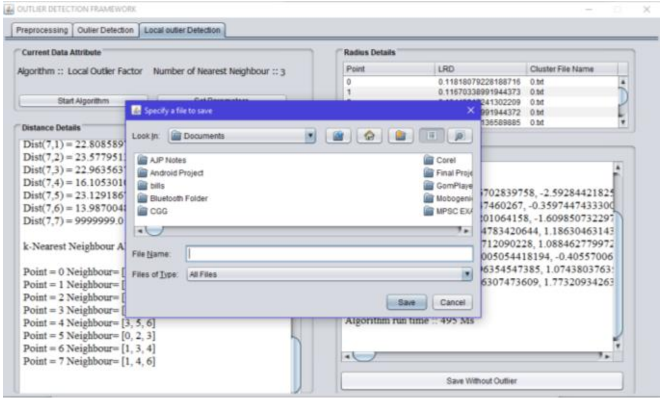
Algorithm :: Outlier Detection Algorithm

```
distance(row = 12 ,centroid) ==> 4.687866055752408 < 5.2
distance(row = 13 ,centroid) ==> 3.680842414207902 < 5.2
distance(row = 14 ,centroid) ==> 5.153216267374292 < 5.2
distance(row = 15 ,centroid) ==> 4.484710293836601 < 5.2
distance(row = 16 ,centroid) ==> 3.22458853110224 < 5.29
distance(row = 17 ,centroid) ==> 3.3847910893742004 < 5.
distance(row = 18 ,centroid) ==> 9.40391895585941 > 5.29
distance(row = 19 ,centroid) ==> 4.740335883781904 < 5.2
distance(row = 20 ,centroid) ==> 4.3569567777021 < 5.294
distance(row = 21 ,centroid) ==> 5.093839708990621 < 5.2
distance(row = 22 ,centroid) ==> 4.269249402168524 < 5.2
distance(row = 23 ,centroid) ==> 5.222549450737767 < 5.2
distance(row = 24 ,centroid) ==> 5.375925759181955 > 5.2
distance(row = 25 ,centroid) ==> 3.2888189957459835 < 5.
distance(row = 26 ,centroid) ==> 3.3923405480878266 < 5.
distance(row = 27 ,centroid) ==> 3.7182459366686986 < 5.
distance(row = 28 ,centroid) ==> 9.007023133486332 > 5.2
distance(row = 29 ,centroid) ==> 6.459793448645222 > 5.2
distance(row = 30 ,centroid) ==> 9.170998771503296 > 5.2
distance(row = 31 ,centroid) ==> 21.260472464906535 > 5.
```

Sr No	Cluster No	Radius
1	1	5.294265729068312

```
------------------------------------------------------------
ODA OUTPUT
----------------------------------------------------------------
File :: C:\Users\dell pc\workspace\OutlierDetection_KKW\Lof

-1.35835406159823,-1.34016307473609,1.7732093426311
-0.644269442348146,1.41796354547385,1.0743803763556,
-5.40125766315825,-5.45014783420644,1.18630463143652
-1.94652513121534,-0.0449005054418194,-0.40557006837
-2.3122265423263,1.95199201064158,-1.60985073229769,
-3.0435406239976,-3.15730712090228,1.08846277997285,
-2.30334956758553,1.759247460267,-0.359744743330052,
-4.39797444171999,1.35836702839758,-2.5928442182573,


Algorithm Run Time :: 63 Ms
```

Set parameters i.e. Nearest neighbor for outlier detection algorithm and click on start algorithm



Algorithm :: Local Outlier Factor   Number of Nearest Neighbour :: 3

Look In: Documents

File Name:

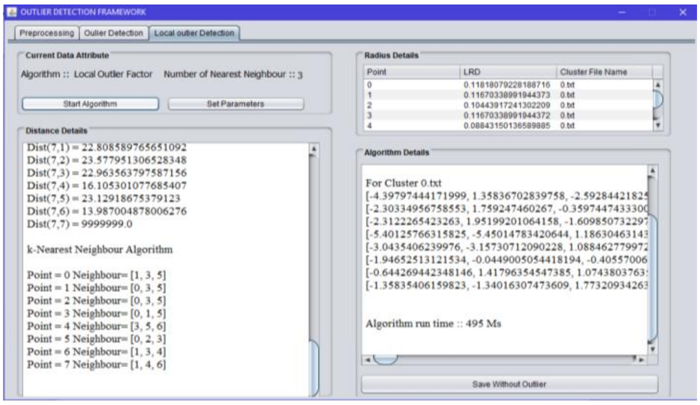
Files of Type: All Files

```
Dist(7,1) = 22.808589
Dist(7,2) = 23.577951
Dist(7,3) = 22.963563
Dist(7,4) = 16.105301
Dist(7,5) = 23.129186
Dist(7,6) = 13.987004
Dist(7,7) = 9999999.0

k-Nearest Neighbour A

Point = 0 Neighbour= [
Point = 1 Neighbour= [
Point = 2 Neighbour= [
Point = 3 Neighbour= [
Point = 4 Neighbour= [3, 5, 6]
Point = 5 Neighbour= [0, 2, 3]
Point = 6 Neighbour= [1, 3, 4]
Point = 7 Neighbour= [1, 4, 6]
```

Point	LRD	Cluster File Name
0	0.11818079228188716	0.txt
1	0.11670338991944373	0.txt
	241302209	0.txt
	991944372	0.txt
	136589885	0.txt

```
702839758, -2.59284421825
7460267, -0.3597447433300
01064158, -1.609850732297
4783420644, 1.18630463143
712090228, 1.088462779972
005054418194, -0.40557006
6354547385, 1.0743803763
6307473609, 1.77320934263

Algorithm run time :: 495 Ms
```

Set parameters i.e. Nearest neighbor for outlier detection algorithm and click on start algorithm



Algorithm :: Local Outlier Factor   Number of Nearest Neighbour :: 3

```
Dist(7,1) = 22.808589765651092
Dist(7,2) = 23.577951306528348
Dist(7,3) = 22.963563797587156
Dist(7,4) = 16.105301077685407
Dist(7,5) = 23.12918675379123
Dist(7,6) = 13.987004878006276
Dist(7,7) = 9999999.0

k-Nearest Neighbour Algorithm

Point = 0 Neighbour= [1, 3, 5]
Point = 1 Neighbour= [0, 3, 5]
Point = 2 Neighbour= [0, 3, 5]
Point = 3 Neighbour= [0, 1, 5]
Point = 4 Neighbour= [3, 5, 6]
Point = 5 Neighbour= [0, 2, 3]
Point = 6 Neighbour= [1, 3, 4]
Point = 7 Neighbour= [1, 4, 6]
```

Point	LRD	Cluster File Name
0	0.11818079228188716	0.txt
1	0.11670338991944373	0.txt
2	0.10443917241302209	0.txt
3	0.11670338991944372	0.txt
4	0.08843150136589885	0.txt

```
For Cluster 0.txt
[-4.39797444171999, 1.35836702839758, -2.59284421825
[-2.30334956758553, 1.759247460267, -0.3597447433300
[-2.3122265423263, 1.95199201064158, -1.609850732297
[-5.40125766315825, -5.45014783420644, 1.18630463143
[-3.0435406239976, -3.15730712090228, 1.088462779972
[-1.94652513121534, -0.0449005054418194, -0.40557006
[-0.644269442348146, 1.41796354547385, 1.0743803763
[-1.35835406159823, -1.34016307473609, 1.77320934263


Algorithm run time :: 495 Ms
```

It displays outliers. We can now save file without outliers

This system recognizes the outliers from given dataset successfully. In order to produce precise and accurate details it first pre-processes the data and then perform Outlier Detection Algorithm i.e. ODA and then it displays outliers and processing time in mini seconds.

CONCLUSION

Outlier detection is a significant problem. It has direct applications in a wide variety of domains. A salient observation with outlier detection is that it was not a precisely formulated problem. Several approaches have been proposed by many researchers to target a particular application domain. To detect cluster based outliers first input data set is clustered into number of clusters using K-nn algorithm and then outlier is detected from each cluster by applying a cluster based outlier detection algorithm and the

Outlier Detection Algorithm. The ODA algorithm gives better accuracy as compared to cluster based outlier detection algorithm for same set of datasets.

We introduce the notion of the local outlier factor LOF, which captures exactly this relative degree of isolation. We show that our definition of LOF enjoys many desirable properties. There are two directions for ongoing work. First is how to describe or explain why the identified local outliers are exceptional. This is specifically meant for high-dimensional data or datasets, as a local outliers may be outlying only on some, but not on all dimensions. The second one is to further improve the performance of LOF computation. On the one hand, such an algorithm may provide more detailed information about the local outliers, e.g., by analyzing the clusters relative to which they are outlying. Also, computation may be shared between LOF processing and clustering. The shared computations may contain k-nn queries and reachability distances.

Acknowledgement

We are grateful to our guide Prof. N M Shahane for his valuable guidance for this research work. We would also like to show gratitude towards our K K Wagh Institute of Engineering Education and Research, Nashik and Head of Computer Engineering Department Dr. S S Sane for providing us the platform required for the work . We would also like to thank our colleagues for their useful insights towards his work.

REFERENCES

1. Aggarwal, C.C., 2015. Data Mining: The Textbook. Springer, New York, NY.Aggarwal, C.C., Reddy, C.K. (Eds.), 2013. Data Clustering: Algorithms andApplications. CRC Press, Boca Raton, FL, USA.
2. Ahmed, M., Naser, A., 2013. A novel approach for outlier detection and clusteringimprovement, in: Proceedings of the 8th IEEE Conference on IndustrialElectronics and Applications (ICIEA), pp. 577–582.
3. Aparna, K., Nair, M.K., 2016. Computational Intelligence in Data Mining.Springer. volume 2. chapter Effect of Outlier Detection on Clustering Accuracyand Computation Time of CHB K-Means Algorithm. pp. 25–35.
4. Chawla, S., Gionis, A., 2013. *k*-means: A unified approach to clustering and outlier detection. SIAM. chapter 20. pp. 189–197.
   Dave, R., Krishnapuram, R., 1997. Robust clustering methods: a unified view.IEEE Transactions on Fuzzy Systems 5, 270–293.
5. Duan, L., Xu, L., Liu, Y., Lee, J., 2009. Cluster-based outlier detection. Annalsof Operations Research 168, 151–168.
6. Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Y.Zomaya, A., Khalil, I., Foufou,S., Bouras, A., 2014. A survey of clustering algorithms for big data: Taxonomyamp; empirical analysis. IEEE Transactions on Emerging Topics inComputing PP, 1–1