Journal of Modelling and Simulation of Systems (Vol.3-2012/Iss.1)
Kuna et al. / Automatic Outliers Fields Detection in Databases / pp. 14-20

Automatic Outliers Fields Detection in Databases

H.D.Kuna*, R.García-Martinez, F.R.Villatoro*****

* Computer Science Department. School of Sciences. National University of Misiones. Argentina
Félix de Azara 1552 - C.P. N3300LQH - Posadas - Misiones - Argentina.
Email: hdkuna@unam.edu.ar
** Information Systems Research Group. Productive and Tecnological Development Department. National University of Lanus. Argentina.
29 de Septiembre 3901 (1826) Remedios de Escalada, Lanús. Argentina
rgarcia@unla.edu.ar
*** Language and Computer Science Department. University of Malaga. Spain.
Complejo Tecnológico, Campus de Teatinos 29071 Málaga.. Spain
Email: villa@lcc.uma.es

Accepted: 11/07/2012

Abstract— Information has become one of the most important assets companies need to protect. From this fact, the audit of systems has a central role in preventing risks related to information technology. In general, development and implementation of the computer-assisted audit technique is still incipient. Data mining applies in an embryonic and asystematic way to tasks related to systems audit. This work tries to show a procedure which uses a clustering process based on density, in particular the algorithm LOF, to identify each field which can be considered an outlier in a database.

Keywords: ACM Classification: H.2.7 Database Administration; H.2.8 Database Application; K.6.5 Security and Protection.

1. INTRODUCTION

Data mining is a powerful tool to discover automatically useful knowledge in databases (Fayyad and Piatetsky-Shapiro, 1996). Some of the typical features of data mining are (among others): generation of predictive models, cluster analysis and association analysis (Larose, 2006).

According to Hawkins (1980), an outlier is a data which is so much different to other data that it is suspected to have been created by different mechanisms. Historically, Statistics has had a paramount role in the detection of outliers, Data Mining at present plays a fundamental role in the process of discovery of those anomalous data.

Outlier detection is a topic widely studied from different disciplines such as statistics (Rousseeuw and Van Driessen 1999) (Hodge and Austin, 2004), Data Mining (Romero and Ventura , 2007), Machine Learning. Its objective is to detect and isolate the dirty data in order to make the data mining process more effective and efficient.

Another objective of outlier detection is the detection of various kinds of frauds (Van der Aalst and Medeiros, 2005) (Etheridge and Brooks , 1994). Automatic detection of these outliers has become a key element in the System auditor's job, making the task easier and objective.

To detect noises in databases implies that an audit requires the auditor's to conduct further tests in order to verify the possible reasons creating such noise and to be able to convert those tracks into evidence. There are many reasons which can generate this noise (Kuna, García Martínez and Villatoro, 2009), for example errors in the design of the interface, errors in the design of the database and unauthorized access to the database, among other.

1.1 Clustering for the detection of outliers

A clustering process can be defined as an unsupervised learning method where data are divided into similar groups, groups which share common characteristics. It is one of the main techniques to discover hidden knowledge, being much used in the discovery of patterns, and specifically the discovery of outliers.

The distance between objects is the main element used, since it is considered that the greater the distance among an object and the rest of the sample, the greater the possibility of considering such object as an outlier is. The main methods to measure that distance are the Euclidean distance, that of Manhatam and the Mahalanobis distance.

At present, the main clustering techniques can be classified as follows:

- Hierarchical clustering, there is a hierarchical decomposition of the data set, a graph known as dendogramm is created, representing the form in which the clusters are being created and the distance among them. This tree can fall under two categories, divisive (top-down) or agglomerative (bottom-up), the distance between clusters can be measured among the centroids, among the nearest neighbors, among the farthest neighbors and among other methods. Some of the hierarchical algorithms are: ROCK (Sudipto, Rajeev and Kyuseok , 2000) CHAMALEON (George, Eui-Hong and Vipin , 1999), CURE (Sudipto, Rajeev and Kyuseok , 1998), BIRCH (Tian, Raghu and Miron , 1996).

- Methods based on partitions, successive partitions of the data set are being created, objects are organized in k clusters so that the deviation of each object is minimized in relation to the center of the cluster. This method is ideal when working with a great quantity of data, its disadvantage being that the value of k should be previously defined and the selection of the center of each cluster is arbitrary. Some of the algorithms which use this method are: K-MEANS (Zhexue , 1998), CLARA (Kauffman and Rousseuw, 1990). CLARANS (Han, 1994).

- Methods based in density, where the data cluster in relation to density measures and objects located in regions with low density are considered anomalous. Among the methods based in density are: LOF (Breuning, Kriegek, Raimond, NG and Sander, 2000) , DBSCAN (Pang-Ning, Michael and Vipin, 2005), OPTICS (Ankerst, Breuning, Kriegel and Sander , 1999).

- There are other methods like those based on Grid (Cho and Lee, 2011), diffuse methods (Cateni, Colla and Vannucci, 2007), based in neural networks (Chuang and Jeng, 2007) (Zhang, Qiu and Li, 2001) , methods based in evolutionary algorithms (Whitacre, Pham and Sarker, 2006), methods based on entropy (Ni, Chen, Lu, Wu and Sun , 2008), etc., which are having an interesting development.

1.2 LOF Algorithm

Most clustering algorithms used to detect outliers were not created for that purpose. The LOF algorithm developed by Breuning (2000) was specifically created to detect outliers, differing from most of the other clustering algorithms, where the detection of outliers is a secondary benefit. LOFvalue of an object p represents the degree to which p is an outlier. Below is an explanation of the LOF developed algorithm :

Definition 1. K-distance 'p':

Considering: an integer 'k'> 0; two points 'p' and 'or' belonging to the dataset D; the K-distance 'p', denoted k-distance (p) is: d (p, o) <==> (if and only if). at least 'k' elements of D, denoted by 'o *' (except 'p') have d (p, o*) <= d (p, o)

at least 'k' - 1 elements of D, denoted by 'o*' (except 'p') have d (p, o*) <d (p, o). Then the k-distance of an element 'p' is its distance from one element 'or' the dataset provided is greater (or equal) to the distance between 'p' and 'k' and 'k' - 1 items dataset.

Definition 2. K-distance neighborhood of 'p'

The neighborhood in the k-distance 'p', denoted Nk (p) is:

Nk (p) = {q € D / {p} / d (p, q) <= k-distance (p)

Once we have calculated the k-dist, (p) has to be the neighborhood around them, they are all 'q' of D whose distance from 'p' is lower than the k-dist (p). A property that has is that the cardinality of Nk (p) is greater than 'k'.

Definition 3. Reachability-distance 'p' over 'or' distance (range)

reach-dist (p, o) = max {k-dist (o) d (p, o)}

This is the reach-dist (p, o) (distance range) of 'p' over 'o' is the maximum of the distance between two points or the k-dist (o). If the distance among the points is small, statistical deviations will occur in the results of the calculations. This problem can be controlled with the 'k', higher values of 'k' minor fluctuations will suffer results. This part is added in the definition of parameter 'MinPts' taking in account that it comes from the density based clustering algorithms and is going to be used to define the density in the neighborhood of an object 'p' through:

reach-distMinPts(p, o) / for all or € NMinPts(p) This means that the parameter will be used to replace the 'k' defined above.

Definition 4. Local reachability density (lrd) de 'p' (local reachability density)

$$lrd_{MinPts}(p) = 1 - \frac{\sum reach - dist_{MinPts}(p, o)}{|N_{MinPts}(p)|} \quad (1)$$

This value is the local density of an object 'p', lrd(p) is calculated by inverting the average of reach-dist(p,o) for all points 'o' of neighborhood of 'p' N(p), i.e., the sum of all r-d on the cardinality of the neighborhood (number of elements), inverted (1 - x).

Definition 5. Local outlier factor (LOF) of 'p' (local outlier factor)

$$LOF_{MinPts}(p) = \frac{\sum \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|} \quad (2)$$

Finally the value of LOF to the point 'p' is the average of the sum of the relationship between all of the lrd of the objects the in neighborhood of 'p' and lrd of 'p'. Considering that, one can say that: lower lrd(p) and higher lrd(o) will be the higher value of LOF.

2. PROBLEM FORMULATION

The algorithms used to detect outliers have several problems:

- There are no formal procedures to define the tasks to be developed for identifying outliers step by step.
- Other of the problems clustering algorithms present is that they identify the tuple considered to contain outliers, but they do not identify which field of that record in particular, contains the anomalous data. In big databases with complex structures, this can be a complication in the task of the systems auditor, since he must carry out a subsequent analysis to identify this field.
- Most of the clustering methods consider an outlier as a binary property. For many scenarios it is more convenient to assign a degree which measures the possibility of each datum being an outlier.

3. SOLUTION PROCEDURE PROPOSED

The parameters to be defined for the LOF algorithm execution are:

- MinPtsLowerBound (hereafter LB) - Must be a positive integer value only.
- MinPtsUpperBound (hereafter UB) - Must be a positive integer value only.
- Kind of distance - default: Euclid. The other types of measurement are: squarted, cosine, inverted angle cosine and radiant.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3}$$

The first 2 parameters are used to define the neighborhood that the algorithm will form around each tuple for particular analysis. The LB is the limit of the minimum number of tuples which must be used to calculate the value of LOF, while the UB address the upper limit of rows to be used for the same task, defining the 'neighborhood' of tuples where each one is going to be compared to determine their value outlier.

The objective of the procedure proposed is to combine the LOF algorithm with the detection of metadata database cleanup, so as to indentify which field in particular, in a great database, contains anomalous data or data with noise. The steps of this process are shown in Figure 1.

The procedure does the following:

Step 1.- LOF is executed in a database

Step 2.- Two separate databases according to a given value of LOF in order to separate a clean database and another database with outliers.

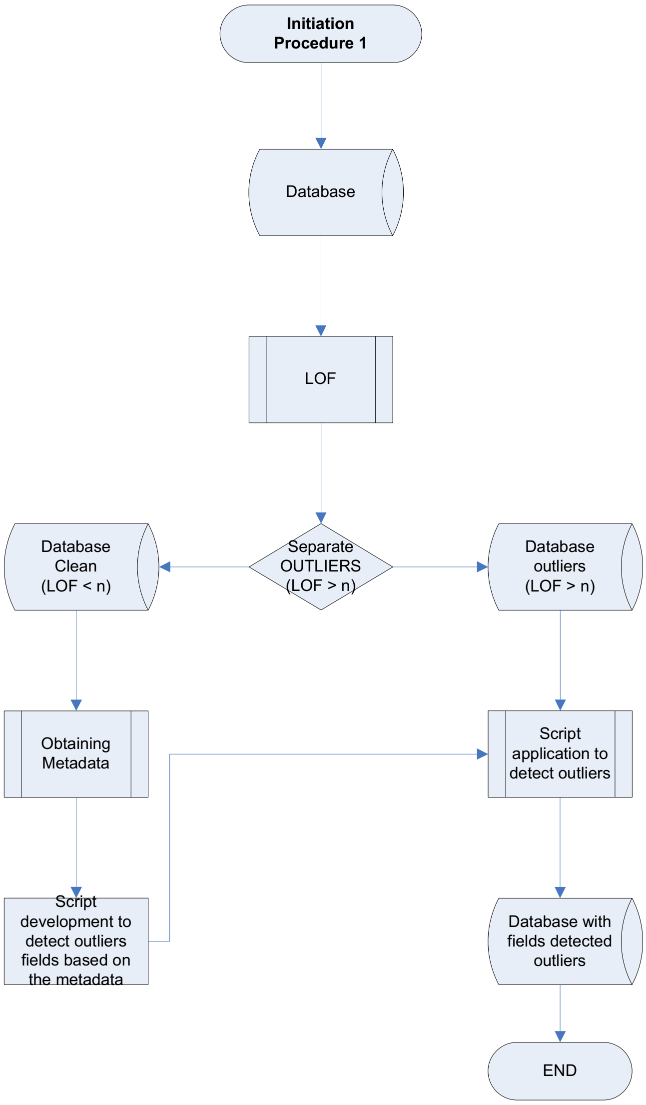


Fig. 1. Process Steps

Step 3.- Metadata are determined on the clean database, maximum, minimum and average of each column

Step 4.- A script that runs all the columns and compares the maximum and minimum values "normal" with each field is developed, if the field value is greater or less than the "normal" values the belonging field is marked as a possible outlier.

Step 5.- The script is applied on the "dirty" database where the LOF value of the tuple presents a possible outlier, resulting in the fields probably being outliers.

4. EXPERIMENTATION

We define the Objective of the experiment (section 4.1), describe the data set (section 4.2), present the experimentation with databases created in accordance with the normal distribution. (section 4.3), and introduce the results using a real database of breast cancer (section 4.4).

4.1. Objective of the experiment

The following goals were addressed:

- Determine the best value of LOF to separate clean databases from dirty ones.

- Determine the values MinPtsUpperBound MinPtsLowerBound that give the best result to detect outliers. These parameters are used to define the neighborhood that the algorithm will form around each tuple to be analyzed. The MinPtsLowerBound is the limit of the minimum number of tuples which must be used to calculate the value of LOF, while MinPtsUpperBound marks the upper limit of tuples to be used for the same task, since this is said to define this measures definingthe 'neighborhood' of tuples against which they will be compared to determine their outlier value.

- Test the procedure in a real database.

4.2. Data Set

In this section we address variables to execute the procedure through a database with normal distribution (section 4.2.1) and present a case study with a real database (section 4.2.2).

4.2.1 Identification of variables to execute the procedure through a database with normal distribution

To perform the experiment, initially there were two databases created randomly, considering the normal distribution, one with 200 entries and the other with 400. To do this was used Matlab v7.6.0 sentence: m = random ('Normal', 1000,30,200 / 400.4) where:

- Normal: is the distribution.

- 100: is the median, range for generating numbers from 1 to x <200

- 30: is the dispersion

- 200/400: the number of records to be generated

- 4: the number of columns.

Figure 2 and 3 shows the histogram of both databases

Since the databases were created considering the normal distribution, outliers were calculated with the same values that are greater or less than two standard deviations from the mean.

From these values of outliers (statistically built), the procedure proposed was applied in order to compare results, determining the best value of LOF to separate the outliers and the best values of the variables and MinPtsUpperBound MinPtsLowerBound.

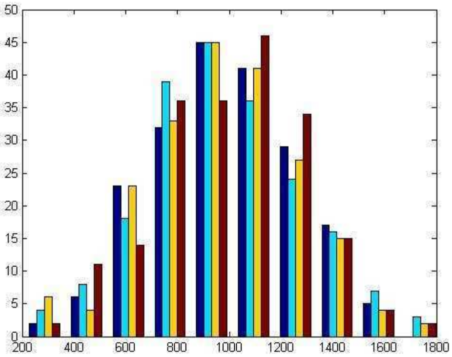


Fig. 2. Histogram based on 200 records

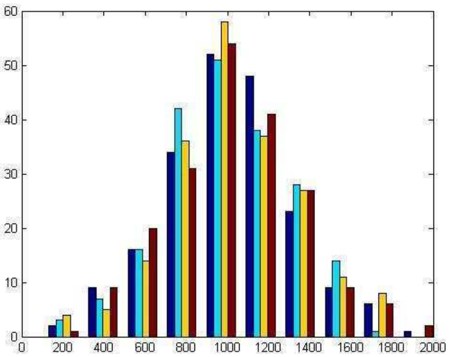


Fig. 3. Histogram based on 400 records

To achieve this goal 30 experiments were run using proposed algorithm with different values of LOF, MinPtsUpperBound and MinPtsLowerBound and the identified outliers were compared with those detected statistically. That is, those values that exceeded two standard deviations (positive or negative) the mean value of the database. To perform the test on a real database, the variables values used, were those which were more efficient when the algorithm was proved in comparison with the statistical identification previously conducted.

4.2.2 Case Study with a real database

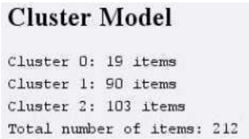
Once these values were determined, the procedure was applied on a real data set, one of the difficulties encountered was the difficulty to find a database with the security about which records are outliers. To experiment with real data a WDBC database from the UCI ML repository was used, the database contained information obtained from nuclear studies of breast cancer diagnoses, the data set had 569 records with 30 attributes, plus an ID attribute and a diagnosis t having two values "benign" and "evil", 357 were benign, and for this experiment were considered the "evil" ones as outliers (Rousseeuw and Van Driessen, 1999). To determine the efficiency of the algorithm, the 10 “evil” records considered extreme outliers were used. To obtain the 10 records to be considered, it be proceeded to consider the clustered ones with RM software (Rapid Miner), in its version 5 using K-Means algorithm which is configured:

- K=3;
- Max runs = 10
- Max optimizations steps = 100

Reduction Process SVD (Singular Value Decomposition);

- With a size = 2

After running the process, the cluster model obtained was the one seen in Figure 4.



Cluster Model

Cluster 0: 19 items
Cluster 1: 90 items
Cluster 2: 103 items
Total number of items: 212

Fig. 4. Cluster Model

Analyzing this model graphically (Figure 5), we see that the cluster is farther from is Cluster 0:

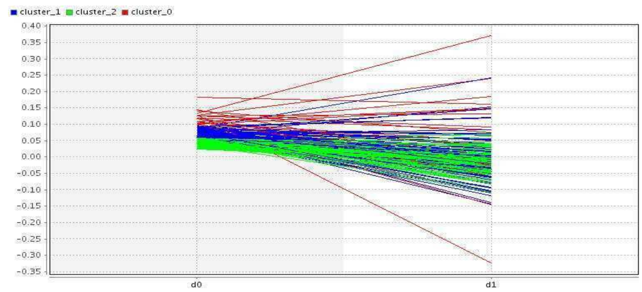


Fig. 5. Resulting Clustering Figures (Parallel - RM).

Cluster 0 contains the following values of evil dataset: to perform the test detection of outliers, the values were ordered from highest to lowest and the first 10 values were taken. As a result of this process, 357 benign records and 10 evil records were taken (considered outliers). The objective of this experiment was to determine the effectiveness of the procedure, according to the parameters determined in experiments with databases created in accordance with the normal distribution.

4.3. Experimentation with databases created in accordance with the normal distribution.

Outliers were determined from the point of view of statistics, outliers. With this purpose, it was determined that all values greater or less than two standard deviations from the mean were considered outliers. The procedure proposed was implemented considering the following parameters:

- Table 1 shows the values of LOF used to separate the Database with values considered "clean" (step b. For the procedure, see section 3.3):

Table 1. LOF Values used

LOF Values	1.3	1.5	1.7	1.9	2

- Table 2 shows the values of MinPtsUpperBound and MinPtsLowerBound used in experiments

Table 2. LOF Values used

MinPtsMin	1	10	5	5	20	50
MinPtsMax	2	20	15	10	40	100

For each LOF value eight tests were performed, giving a total of 30 trials, which compared the result of applying the procedure proposed with the outliers detected obtained from a statistical perspective. The best result was obtained considering a value of LOF = 1.5. Table 3 shows the results based on 200 and 400 records.

Table 3. Experimentation results with 200 records database

LOF Limit value	1,5					
MinPtsMin value	1	5	10	5	20	50
MinPtsMax value	2	15	20	10	40	100
Real DB Outliers	35	35	35	35	35	35
Procedure Detected Outliers	12	35	26	26	0	0
False Positives	25	165	16	16	0	0
Effectiveness	34,2857143	100	74,2857143	74,2857143	0	0

Table 4. Experimentation results with 400 records database

LOF Limit value		1,5				
MinPtsMin value	1	5	10	5	20	50
MinPtsMax value	2	15	20	10	40	100
Real DB Outliers	65	65	65	65	65	65
Procedure Detected Outliers	17	65	40	65	0	0
False Positives	53	335	25	335	0	0
Effectiveness	26,1538462	100	61,5384615	100	0	0

As a result, the greater effectiveness of the procedure was achieved with the following parameters:

- LOF = 1.5
- MinPtsMin = 10
- MinPtsMax = 20

4.4. Experimenting with the database of breast cancer

A procedure was applied to the real database, considering the best parameters found in tests performed on the database created according to the normal distribution, whose results are shown in Table 5.

Table 5. Experimentation results with Cancer database

LOF Limit value	**1,5**
MinPtsMin value	**10**
MinPtsMax value	**20**
Real DB Outliers	**10**
Procedure Detected Outliers	10
False Positives	0
Effectiveness	100

The experiment allowed to detect within 10 tuples considered as outliers, which attributes were exactly outliers. Table 6 shows some of the attributes outliers.

Table 6. Detected outliers result

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10
17,99	10,38	122,8	1001	0,1184	0,2776	0,3001	0,1471	0,2419	0,07871
20,57	17,77	132,9	1326	0,08474	0,07864	0,0869	0,07017	0,1812	0,05667
19,69	21,25	130	1203	0,1096	0,1599	0,1974	0,1279	0,2069	0,05999
11,42	20,38	77,58	386,1	0,1425	0,2839	0,2414	0,1052	0,2597	0,09744
20,29	14,34	135,1	1297	0,1003	0,1328	0,198	0,1043	0,1809	0,05883
12,45	15,7	82,57	477,1	0,1278	0,17	0,1578	0,08089	0,2087	0,07613
18,25	19,98	119,6	1040	0,09463	0,109	0,1127	0,074	0,1794	0,05742
13,71	20,83	90,2	577,9	0,1189	0,1645	0,09366	0,05985	0,2196	0,07451
13	21,82	87,5	519,8	0,1273	0,1932	0,1859	0,09353	0,235	0,07389

5. CONCLUSIONS

Experiments based on real data showed an effectiveness of 100%, being able to detect not only the tuples considered outliers but attributes of these tuples which are outliers. This is an important contribution to the task of the auditor since they automatically detect which attributes are specifically abnormal.

The automation of the task of the auditor in the detection of outliers through the implementation of the algorithm proposed makes their efforts more effective and efficient. This allows a level of objectivity in the work that ensures higher quality results and enables less experienced auditors to use the procedure and can thus improve the quality of their work.

REFERENCES

Ankerst, M., Breuning, M., Kriegel, H.P., and Sander, J. (1999). Optics: Ordering points to identify clustering structure. *In Proceedings of the ACM SIGMOD Conference*, pages 49–60. Philadelphia, PA.

Breuning, M., Kriegel, H., Ng, R.T., Sander, J. (2000). LOF: identifying density-based local outliers. *In Proceedings of the ACM SIGMOD Conference*, 93-104. Dallas, TX.

Cateni, S., Colla, V., and Vannucci, M. (2007). A fuzzy logic-based method for outliers detection. *Paper presented at the Proceedings of the IASTED International Conference on Artificial Intelligence and Applications*, AIA 2007, 561-566.

Cho, K.C. and Lee, J. S. (2011). Grid-based and outlier detection-based data clustering and classification. *Communications in Computer and Information Science*, Volume 150, 129-138.

Chuang, C., and Jeng, J. (2007). CPBUM neural networks for modeling with outliers and noise. *Applied Soft Computing Journal*, 7(3), 957-967.

Etheridge, H.L., Brooks, R.C. (1994). Neural networks: A new technology. *The CPA Journal*, 31(2), 531-550.

Fayyad, U. and Piatetsky-Shapiro, G. (1996). KnowledgeDiscovery and Data Mining: Towards a Unifying Framework. *Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining*, pp 82-88. Portland.

Han, J. (1994). Very large data bases. *In Proceedings of the 20th International Conference on Very Large Data Bases*, pages 144–155. Berkeley, CA.

Hawkins, D. (1980). Identification of Outliers. *Chapman and Hall*, London.

Hodge, V.J. and Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2), 85-126.

Karypis, G., Han, E. and Kumar, V. (1999). Charmaleon: Hierarchical clustering using dynamic modeling. *Computer*, 32(8), 68–75.

Kauffman, L., Rousseuw, P. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, New York, NY.

Kuna, H., García Martínez, R., Villatoro, F.R. (2009). Pattern Discovery in University Students Desertion Based on Data Mining. *Book of Abstracts of IV International Meeting on Dynamics of Social and Economic Systems*, Pág. 48.

Larose, D. (2006). *Data Mining Methods and Models.* John Wiley & Sons, Inc., Hoboken, NewJersey.

Ni, W., Chen, G., Lu, J., Wu, Y., and Sun, Z. (2008). Local entropy based weighted subspace outlier mining algorithm. *Jisuanji Yanjiu Yu Fazhan/Computer Research and Development*, 45(7), 1189-1194.

Pang-Ning, T., Michael, S., Vipin, K. (2005). *Introduction to Data Mining.* Addison Wesley.

Romero, C., and Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33(1), 135-146.

Rousseeuw, P. J., and Van Driessen, K. (1999). Fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3), 212-223.

Sudipto, G., Rajeev, R., Kyuseok, S. (1998). Cure: an efficient clustering algorithm for large databases. *In SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 73–84. ACM Press, New York, NY, USA.

Sudipto, G., Rajeev, R., Kyuseok. S. (2000). A robust clustering algorithm for categorical attributes. *Information Systems*, 25(5):345–366.

Tian, Z., Raghu, R., Miron, L. (1996). Birch: an efficient data clustering method for very large databases. *In SIGMOD '96: Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, pages 103–114. ACM Press, New York, NY, USA.

Van Der Aalst, W.M.P, Medeiros, A.K.A. (2005): Process Mining and Security: Detecting Anomalous Process Executions and Checking Process Conformance.

Electronic Notes in Theoretical Computer Science, pp 121.

Whitacre, J. M., Pham, T.Q., and Sarker, R.A. (2006). Use of statistical outlier detection method in adaptive evolutionary algorithms track: Genetic algorithms. *Paper presented at the GECCO 2006 - Genetic and Evolutionary Computation Conference*, 2 1345-1352.

Zhang, G., Qiu, J., and Li, J. (2001). Outlier identification and justification based on neural network. *Zhongguo Dianji Gongcheng Xuebao/Proceedings of the Chinese Society of Electrical Engineering*, 21(8), 104-107+113.

Zhexue, H. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3), 283–304.

AUTHORS PROFILE

Horacio Kuna received BSc degree (2004) on Information Systems from University of Morón, MSc degree (2006) in Software Engineering from Polytechnic University of Madrid, and is PhD Candidate on Computer Science at Málaga University. He was Visiting Researcher (2009) and (2011) of Málaga University; and Visiting Researcher (2001-2003) of the Intelligent Systems Laboratory of the School of Engineering of the Buenos Aires University. MSc Kuna is Head and Senior Researcher of Data Mining Research Group, Head of the Computer Science Department and Full Professor of Software Engineering within School of Sciences. National University of Misiones. Argentina.

Ramón Garcia Martinez received BSc degree (1988) in Computer Science from National University of La Plata and MSc degree (1992) in Informatics Engineering and PhD degree (1997) in Computer Science from Polytechnic University of Madrid. He was Founder Director of the Intelligent Systems Lab (1994-2010), School of Engineering, University of Buenos Aires, Director (1992-1995) of the Research and Development Program on Artificial Intelligence, Lujan National University, Deputy Director (1996-2002) and Director (2002-2008) of the Master Degree Program on Software Engineering within agreement among Polytechnic University of Madrid and Buenos Aires Institute of Technology. He was Professor of Knowledge Engineering and Intelligent Technologies Based Data Mining within the PhD Degree Program on Computer Science of the Computer Science School of La Plata National University. Dr. Garcia-Martinez is Head and Senior Researcher at the Information Systems Research Group and Full Professor of Software Engineering at Lanus National University.

Francisco R. Villatoro received a MSc degree (1992) in Computer Science from Universidad de Málaga, Spain, a MSc degree (1993) in Physics from UNED, Madrid, Spain, and a PhD degree (1998) in Mathematics from Universidad de Málaga. Dr. Villatoro is Professor in Computational Science and Artificial Intelligence at the University of Málaga. His research was focused in the area of computational methods in nonlinear partial differential equations.