Outliers Detection in Multi-label Datasets

Peer-reviewed author version

BELLO GARCIA, Marilyn; NAPOLES RUIZ, Gonzalo; Morera, Rafael; VANHOOF, Koen & Bello, Rafael (2020) Outliers Detection in Multi-label Datasets. In: Martínez-Villaseñor, Lourdes; Herrera-Alcántara, Oscar; Ponce, Hiram; Castro-Espinoza, Félix A. (Ed.). Advances in Soft Computing (MICAI 2020), Springer International Publishing AG, p. 65 -75.

DOI: 10.1007/978-3-030-60884-2_5
Handle: http://hdl.handle.net/1942/32465

Outliers Detection in Multi-label Datasets

Marilyn Bello[1,2], Gonzalo Nápoles[2,3], Rafael Morera[1],
Koen Vanhoof[2], and Rafael Bello[1]

[1] Computer Science Department, Universidad Central de Las Villas, Cuba
[2] Faculty of Business Economics, Hasselt University, Belgium
Department of Cognitive Science & Artificial Intelligence,
Tilburg University, The Netherlands
mbgarcia@uclv.cu

Abstract. In many knowledge discovery applications, finding *outliers*, i.e. objects that behave in an unexpected way or have abnormal properties, is more interesting than finding *inliers* in a dataset. Outlier detection is important for many applications, including those related to intrusion detection, credit card fraud, and criminal activity in e-commerce. Several methods of outlier detection have been proposed, and even many of them from the perspective of Rough Set Theory, but at the moment none of them is specifically intended for multi-label datasets. In this paper, we propose a method that measures the degree of anomaly of an object in a multi-label dataset. This score or measure quantifies the degree of irregularity of an object with respect to the dataset. In addition, a method for generating anomalies in this type of datasets is proposed. From these synthetic datasets, the efficacy of the proposed method is proved. The results show the superiority of our proposal over other methods in the literature adapted to multi-label problems.

Keywords: Outlier Detection · Outlier Generation · Multi-label Datasets · Rough Set Theory · Knowledge Discovery

1 Introduction

The detection of outliers (anomalies or irregularities) is a key task in knowledge discovery. Roughly speaking, the process consists in detecting small groups of data objects that are deemed "exceptional" when compared with the rest of data, in terms of certain sets of properties. While there is no a single, generally accepted, formal definition of an outlier, Hawkins [11] defined an outlier as an observation that deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism.

Initially, the main reason for outlier detection was to remove outliers from the training data, since some pattern recognition algorithms are quite sensitive to outliers in the data [1]. However, for many applications [4, 10], such as fraud detection in e-commerce [23, 24], it is more interesting to detect rare events than to common ones, from a knowledge discovery standpoint.

Generally speaking, the existing approaches for outlier detection can be classified into the following five categories [19]: distribution-based approach [25], depth-based approach [17], distance-based approach [18], density-based approach [6], and clustering approach [13]. In addition, some authors [14, 16, 15] have employed the Rough Set Theory (RST) [21] for detecting outliers. For instance, Shaari et al. [26] proposed a new method to detect outliers using the concept of Non-Reduct as defined in RST. Chen et al. [8] proposed an outlier detection algorithm based on the neighborhood rough set model. In [15] the authors proposed a boundary-based outlier detection method, while in [16] they presented a rough membership function-based outlier detection method, by virtue of the notion of rough membership function in rough sets.

Although many of these techniques have proven useful and effective in detecting outlier pattern, none of them are specifically intended to deal with multi-label datasets at the moment. In a multi-label dataset [12], every object x is described by a number of input features $\{f_1, f_2, \ldots, f_m\}$, and is associated with a set of labels $\{l_1, l_2, \ldots, l_k\}$ instead of a single class label. Hence in this type of problem, an observation can belong to several classes at the same time.

In this paper, we propose a method to detect outliers in multi-label dataset. With this goal in mind, we rely on the definition of *outlier* given by Barnet and Lewis [3]. They defined an outlier as an observation (or subset of observations) which appear to be inconsistent with the remainder of the dataset. This idea could be modeled by using the extended RST approach, in which the consistency of an object is defined from the relation between its predictive and decision part. In other words, if the object's similarity class (i.e., the objects that are similar to it taking into account its predictive characteristics) and its equivalence class (i.e., the objects that are identical to it taking into account its labels) are similar, it could be said that it is consistent with respect to the rest of the objects in the dataset. Then, the degree to which an object is an outlier could depend on the extent to which the object satisfies this relation. Therefore, our method provides an anomaly degree for each object in the dataset instead of using the binary labeling (i.e., whether the object is an anomaly or not). The degree assigned to each object will be between $[0, 1]$, where 0 denotes a normal object (inlier), whereas 1 indicates a strong anomaly (outlier).

The evaluation of our detection method is difficult due to the lack of multi-label datasets with objects that have already been identified as outliers. Thus, as a second contribution of our paper, we also propose a method that generates outliers for datasets reported in the multi-label literature. The idea is to build an object from those objects that are similar to it, and whose irregularity is caused by the variation of its behavior, in terms of their labels. This method not just allows assessing our method but also provides the machine learning community with a procedure to generate more changeling datasets.

The rest of the paper is organized as follows. In Section 2, we briefly introduce the fundamentals of the Rough Sets Theory. Section 3 presents the outlier generation method in multi-label datasets, and Section 4 describes the outlier detection method. Experimental results on benchmark problems are discussed

in Section 5 while Section 6 concludes the paper.

2 Preliminaries on Rough Sets

RST is a methodology proposed in the early 1980's for handling uncertainty that is manifested in the form of inconsistent data [21]. The underlying notion behind the rough set analysis is the indiscernibility of objects. By modeling the indiscernibility as an equivalence relation, one can partition a finite universe of objects into a family of pair-wise disjoint subsets.

Let $DS = (\mathcal{U}, \Psi \cup \{d\})$ denote a decision system where $\mathcal{U}$ is a non-empty finite set of objects called the universe of discourse, Ψ denotes a non-empty finite set of features describing any object in $\mathcal{U}$, and $d \notin \Psi$ represents the decision class. In this mathematical formalism, an equivalence class $[x]_\Phi$ of $x \in \mathcal{U}$ comprises the set of objects in $\mathcal{U}$ that are deemed inseparable from x according to the information contained in the feature subset $\Phi \subseteq \Psi$. Two objects are considered inseparable if they have identical values for all features.

This definition is adequate for nominal features but is too rigid when dealing with numerical ones, given that marginal differences between two numerical values could toss two nearly identical objects into different inseparability classes. This problems can be alleviated in some extent by extending the concept of inseparability relation, and replacing the equivalence relation with a weaker binary relation [27]. Equation (1) shows an indiscernibility relation, where $0 \leq S(x, y) \leq 1$ is a similarity function. The similarity function could be formulated in a variety of ways. In this study, we assume that $S(x, y) = 1 - \delta(x, y)$, where $\delta(x, y)$ stands for the Heterogeneous Euclidean-Overlap Metric [30] between x and y. Hence, the similarity function can be written as follows:

$$R : xRy \iff S(x, y) \geq \xi_1. \tag{1}$$

This weaker binary relation states that x and y are deemed inseparable as long as their similarity degree $S(x, y)$ exceeds a similarity threshold $0 \leq \xi \leq 1$, and defines a similarity class where $\bar{R}(x) = \{y \in U | yRx\}$.

3 Outlier Generation in Multi-label Datasets

A pivotal issue in evaluating outlier detection algorithms is the accessibility of benchmark datasets. In many cases, synthetic datasets are more suitable than authentic data [20] since we often know in advance what to expect. However, synthetic data have the disadvantage of not having the realism of authentic data. The method proposed in this section generates synthetic multi-label datasets with anomalies. In this paper, we use existing datasets in the multi-label literature, and introduce some new objects labeled as outlier. Those objects already existing in the dataset were labeled as inliers.

The method starts by building two sets $C(x)$ and $D(x)$ for each object x in the dataset. The former consists of all objects that are similar to x taking into account the input features, while the latter is the set of identical objects to x by considering the labels. Our approach pursues the fact that insofar as these two sets are similar, an outlier could be built. The next step consists in building an object whose feature values are the result of a process of aggregating the information of the objects contained in $C(x)$, and its decision values are the set of labels to which most of the objects in $D(x)$ do not belong to. As a result, we would have an object in the dataset that would be very similar to a set of objects in terms of its predictive characteristics, and at the same time, very different in terms of its labels. Algorithm 1 formalizes this idea.

Algorithm 1 Outlier Generation in Multi-label Datasets

1: $UsedSet = \{\}$, $OutliersSet = \{\}$
2: For each object $\forall x_i \in U : x_i \notin Used$, compute its similarity class $C(x_i)$, and its equivalence class $D(x_i)$
3: Compute the similarity (δ) as done in [9] between the information granules $C(x_i)$ and $D(x_i)$ by using the Equation (2),

$$\delta_i = \frac{|C(x_i) \cap D(x_i)|}{0.5\,|C(x_i)| + 0.5\,|D(x_i)|} \tag{2}$$

4: **if** $\delta_i \geq \xi_2$ **then**
5: Build a outlier object $Out_i = [Out_{cond}, Out_{dec}]$, where Out_{cond} is derived from an features aggregation of all objects in $C(x_i)$, and $Out_{dec} = \{l_1, l_2, \ldots, l_k\}$ with $l_k = 0$ if most of the objects in $D(x_i)$ are labeled with that label, otherwise, $l_k = 1$
6: $UsedSet = UsedSet \cup x_i$
7: $OutliersSet = OutliersSet \cup Out_i$
8: **end if**

A similarity threshold (ξ_2) is established in order not to use in the construction of an outlier those objects that have a certain degree of anomaly. This is based on the criterion that the vicinity of a non-outlier object taking into account its condition and decision features must be similar.

It should be mentioned that, if the number of outliers is greater than the number of inliers, then inliers become noise and is not the purpose of our algorithm. We have to take into consideration that the number of outliers must always be considerably less than the number of inliers.

4 Outlier Detection in Multi-label Datasets

According to [2], a way to define outliers is to consider as such those points at which a function learned from the dataset results in an unusually large error. Since the learning process attempts to generalize the relation between inputs and outputs, it is expected a large error when processing objects having similar

inputs but very different outputs. In the same way, if an object is very similar to a subset of objects according to its predictive features, it is reasonable to assume that it is labeled in a similar way to the objects in the subset, otherwise, this inconsistency could be considered an anomaly.

The method proposed in this section is based on the above idea, which relies on the RST consistency. It first builds a prototype from a subset of objects that are similar to each other. The prototypes represent the typical characteristics of the objects of a category instead of necessary or sufficient conditions. Prototypes can be abstractions (e.g., the result of an aggregation process) of universe objects, or they can be some observed objects themselves.

For each $x \in \mathcal{U}$, a similarity class −all objects that are similar to x taking into account their predictive features− is built. Next, we derive a prototype for each similarity class such that each prototype includes both predictive and decision part. This process is performed by using an aggregation operator, which aggregates the predictive and decision information of the objects in the similarity class of x. The average operator can be used as the aggregation operator if the feature value is numeric, while the mode can be used if the value is nominal. As a result, the resulting prototype will have as decision values the most common labels of the objects in the similarity class.

Finally, we compute the degree of anomaly of the x object by using the proximity to its associated prototype regarding the decision part. In other words, this degree will be determined by computing the distance between the set of labels associated with the object and its prototype. In our case, we used Hamming set distance [5]. Algorithm 2 formalizes this idea.

Algorithm 2 Outlier Detection in Multi-label Datasets

1: For each object $x_i \in U$, compute its similarity class $C(x_i)$ using the similarity relation defined in Equation 1, $x_i \notin C(x_i)$
2: Build a prototype $P_i = [P_{cond}, P_{dec}]$, where P_{cond} is derived from an features aggregation of all objects in $C(x_i)$, and $P_{dec} = \{l_1, l_2, \ldots, l_k\}$ with $l_k = 1$ if most of the objects in $C(x_i)$ are labeled with that label, otherwise, $l_k = 0$
3: Compute the anomaly degree of x_i from $HammingDistance(x_{i_{dec}}, P_{dec})$

The degree of anomaly obtained for each object in the dataset could be used to discern between weak outliers (noise) and strong outliers. A high degree indicates a significant difference in the behavior of an object, so that it would be considered a strong outlier. The advantage of this method is that it does not depend on the classification method used, which allows us to detect the outliers before any learning process is performed.

5 Results and Discussion

In this section, we carry some numerical simulations to evaluate the performance of the method of outlier detection proposed in this work. The first step consists in

creating a group of datasets with outliers using the method proposed in Section 3. With this goal in mind, we adopt 10 multi-label datasets taken from the well-known RUMDR repository [7]. In these problems (see Table 1), the number of objects ranges from 207 to 10,491, the number of features goes from 72 to 635, and the number of labels from 6 to 400. Also, the last column of the table shows the number of outliers inserted in each dataset.

Table 1: Characterization of datasets used for simulations.

Name	# Objects	# Nominal features	# Numerical features	# Labels	# Outliers
birds	708	2	258	19	63
emotions	597	0	72	6	4
genbase	679	1186	0	27	17
GnegativePseAAC	1397	0	441	8	5
GpositivePseAAC	521	0	441	4	2
HumanPseAAC	3131	0	441	14	25
PlantPseAAC	985	0	441	12	7
scene	2410	0	294	6	3
VirusPseAAC	213	0	441	6	6
yeast	2495	0	103	14	78

5.1 Performance of the outlier detection method

According to [2], if an anomaly detection method is able to achieve a significant difference between the degree of anomaly of the objects labeled as normal (inliers), and those labeled as anomaly (outliers), we can confirm the quality of the method. From this point on, we conducted the experimental analysis. The results shown in this section were obtained by establishing the values of 0.95 and 0.90 for the ξ_1 and ξ_2 similarity thresholds, respectively. These values have been arbitrarily selected, so other alternatives are also possible.

Figure 1 portrays the average anomaly degree achieved for each object labeled as inliers, and outliers in each dataset. The results show how the proposed method for all the study cases is able to distinguish to a great extent between an inlier and outlier object. Since, the method in most cases assigns a value close to 0 to inliers, and close to 1 to outliers.

Table 2 shows a comparison of the performance of the proposed method against two algorithms reported in the literature: *Exact k-Nearest Neighbor Score* and *Average k-Nearest Neighbor Score* [2]. Both were adapted to the multi-label problem, and were selected because they also provide a score of anomaly for each object in a dataset. The second and third columns show the average of the anomaly degrees observed in the inliers and outliers, respectively. In addition, the last column in Table 2 shows the difference between both average values. The greater this difference, the better the performance, since it achieves a greater distinction between inlier and outlier objects.

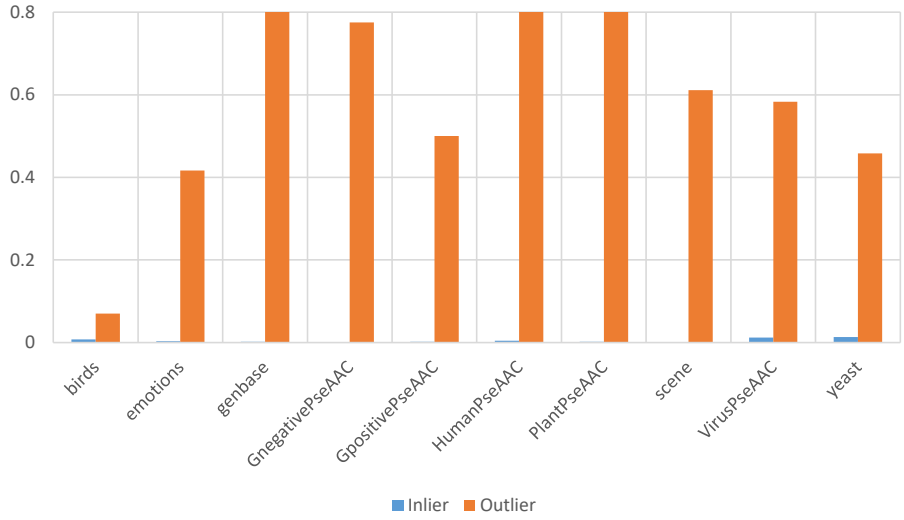


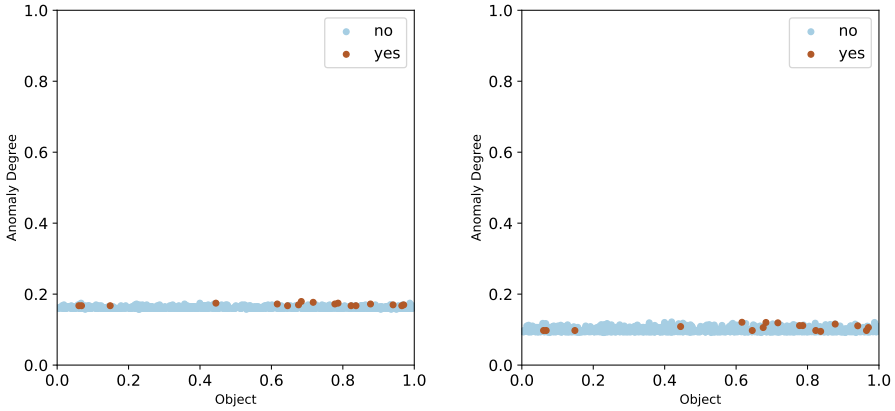
Fig. 1: Average anomaly degree observed for object labeled as inliers and outliers in each dataset adopted for simulation.

Table 2: Comparison against other methods in the literature. Boldface denotes the largest partition obtained between inliers and outliers.

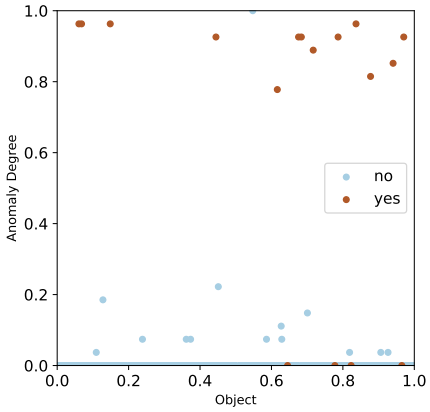
	Inliers Average	Outliers Average	Difference
AverageKnnScore	0.2876294	0.2956309	0.0080015
ExactKnnScore	0.3055845	0.3285543	0.0229698
Proposal	0.0046806	0.5993862	**0.5947056**

The results suggest that our method is more effective in detecting outliers since it obtains a higher difference (i.e. over 0.59) than the other methods when discerning between an anomaly and a regular pattern. The reason for this is that these methods do not consider the relationship between the features and the labels in an object. This relation allows for more accurate results, even where there are objects that are isolated or in dense regions.

Figure 2 illustrates how the objects are distributed according to the anomaly degree computed by using the previous outlier detection methods. For each object in the dataset, we assign a random value between $[0, 1]$ to identify it on the x-axis, and then associate it with a degree of anomaly (i.e. the y-axis). In this way, the two colors in the plot represent whether the object is an outlier or not. Overall, this plot confirms the superiority of our proposal, since it achieves an outstanding partition between the objects that are outlier, and those that are not. In other words, most of the outliers (i.e, those objects labeled as "yes") have a high associated anomaly degree, and the opposite occurs in the case of the inliers (i.e, those objects labeled as "no").



(a) ExactKnnScore (b) AverageKnnScore



(c) Proposal

Fig. 2: Object distribution obtained for the *genbase* dataset from the anomaly degree estimated by each method. In this plot, universe objects labeled as "yes" are outliers while those labeled as "no" are inliers.

5.2 How do outliers affect multi-label classifiers?

In the literature, it is frequently mentioned that the presence of outliers affects the performance of a classifier, but there are few studies verifying such claim [1]. As part of this study, we evaluated the effect of outliers on multi-label classifiers. To do this, we estimated the *Hamming Loss* (HL) value [12, 22] by using a 10-fold cross validation scheme. The HL metric is probably the most used performance metric in multi-label scenarios. We considered three classifiers implemented in MULAN [28]: ML-kNN[32], RAkEL [29], and BP-MLL [31]. Those are considered state-of-the-art classifiers for multi-label classification.

Figure 3 shows the average HL values achieved by each method through the use of those datasets with outliers. It should be noted that for HL, small values show better results. The results show an increase of the HL values in those datasets that have anomalies. This confirms the sensibility of these classifiers to the presence of this type of objects. Similarly, the results indicate that BP-MLL seems to be slightly more vulnerable in these cases.

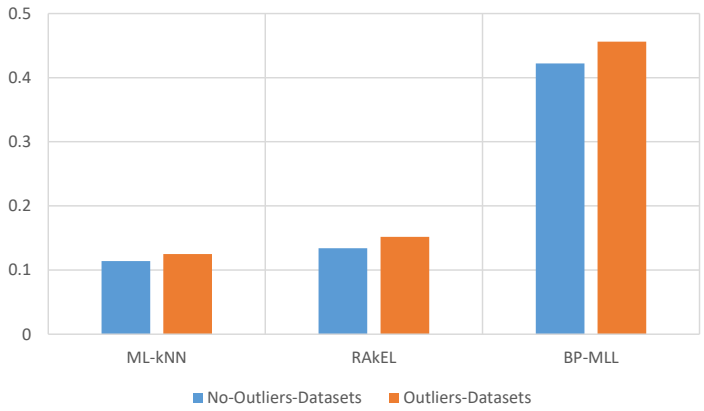


Fig. 3: Average HL values achieved by each classification model for multi-label datasets with and without outliers.

6 Conclusions

In this paper, we extended outlier detection to multi-label problems using the Pawlak's rough set theory. The method proposed is able to estimate the degree of anomaly of an object with respect to the others composing the dataset. As an additional contribution, we proposed a method for generating anomalies in a multi-label dataset, which allows for the validation of our method and other techniques to detect outliers in this type of problem.

The experimental study shows the superiority of our method over others existing in the literature, since it is more effective in distinguishing if an object is an outlier or not. Furthermore, we confirmed that the presence of these anomalies causes the performance of existing multi-label classifiers to decrease. The main advantage of the proposed methods is that they do not depend on any particular classification model. There are however two main issues to be mentioned. Firstly, the concept of outlier we used in this paper is based on the inconsistency, that is just a type of uncertainty. Secondly, one might wonder to what extent our outlier detection method is biased by the way we generate those outliers. Whichever the case might be, our research is a necessary step into overcoming of outliers in problems concerning multi-label pattern classification.

Moreover, as anomaly detection is of great interest in areas such as the financial and security sector, where it is essential not only to be able to detect anomalies, but also to understand what is considered an *outliers*, we find it interesting as future work to include the use of Explainable AI techniques.

References

1. Acuña, E., Rodriguez, C.: On detection of outliers and their effect in supervised classification. University of Puerto Rico at Mayaguez **15** (2004)
2. Aggarwal, C.C.: Outlier analysis. In: Data mining. pp. 237–263. Springer (2015)
3. Barnet, V., Lewis, T.: Outliers in statistical data. 1994
4. Basharat, A., Gritai, A., Shah, M.: Learning object motion patterns for anomaly detection and improved object detection. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2008)
5. Bookstein, A., Kulyukin, V.A., Raita, T.: Generalized hamming distance. Information Retrieval **5**(4), 353–375 (2002)
6. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data. pp. 93–104 (2000)
7. Charte, F., Charte, D., Rivera, A., del Jesus, M.J., Herrera, F.: R ultimate multilabel dataset repository. In: International Conference on Hybrid Artificial Intelligence Systems. pp. 487–499. Springer (2016)
8. Chen, Y., Miao, D., Zhang, H.: Neighborhood outlier detection. Expert Systems with Applications **37**(12), 8745–8749 (2010)
9. Filiberto, Y., Bello, R., Caballero, Y., Frias, M.: An analysis about the measure quality of similarity and its applications in machine learning. In: Fourth International Workshop on Knowledge Discovery, Knowledge Management and Decision Support. Atlantis Press (2013)
10. Gebhardt, J., Goldstein, M., Shafait, F., Dengel, A.: Document authentication using printing technique features and unsupervised anomaly detection. In: 2013 12th International Conference on Document Analysis and Recognition. pp. 479–483. IEEE (2013)
11. Hawkins, D.M.: Identification of outliers, vol. 11. Springer (1980)
12. Herrera, F., Charte, F., Rivera, A.J., Del Jesus, M.J.: Multilabel classification. In: Multilabel Classification, pp. 17–31. Springer (2016)
13. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM computing surveys (CSUR) **31**(3), 264–323 (1999)
14. Jiang, F., Chen, Y.M.: Outlier detection based on granular computing and rough set theory. Applied intelligence **42**(2), 303–322 (2015)
15. Jiang, F., Sui, Y., Cao, C.: Outlier detection using rough set theory. In: International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing. pp. 79–87. Springer (2005)
16. Jiang, F., Sui, Y., Cao, C.: A rough set approach to outlier detection. International Journal of General Systems **37**(5), 519–536 (2008)
17. Johnson, T., Kwok, I., Ng, R.T.: Fast computation of 2-dimensional depth contours. In: KDD. pp. 224–228. Citeseer (1998)
18. Knorr, E.M., Ng, R.T., Tucakov, V.: Distance-based outliers: algorithms and applications. The VLDB Journal **8**(3-4), 237–253 (2000)

19. Kovács, L., Vass, D., Vidács, A.: Improving quality of service parameter prediction with preliminary outlier detection and elimination. In: Proceedings of the second international workshop on inter-domain performance and simulation (IPS 2004), Budapest. vol. 2004, pp. 194–199 (2004)
20. Lundin, E., Kvarnström, H., Jonsson, E.: A synthetic fraud data generation methodology. In: International Conference on Information and Communications Security. pp. 265–277. Springer (2002)
21. Pawlak, Z.: Rough sets. International journal of computer & information sciences **11**(5), 341–356 (1982)
22. Pereira, R.B., Plastino, A., Zadrozny, B., Merschmann, L.H.: Correlation analysis of performance measures for multi-label classification. Information Processing & Management **54**(3), 359–369 (2018)
23. Porwal, U., Mukund, S.: Credit card fraud detection in e-commerce: An outlier detection approach. arXiv preprint arXiv:1811.02196 (2018)
24. Ramakrishnan, J., Shaabani, E., Li, C., Sustik, M.A.: Anomaly detection for an e-commerce pricing system. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1917–1926 (2019)
25. Rousseeuw, P.J., Leroy, A.M.: Robust regression and outlier detection, vol. 589. John wiley & sons (2005)
26. Shaari, F., Bakar, A.A., Hamdan, A.R.: Outlier detection based on rough sets theory. Intelligent Data Analysis **13**(2), 191–206 (2009)
27. Slowinski, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. IEEE Transactions on knowledge and Data Engineering **12**(2), 331–336 (2000)
28. Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I.: Mulan: A java library for multi-label learning. Journal of Machine Learning Research **12**(Jul), 2411–2414 (2011)
29. Tsoumakas, G., Vlahavas, I.: Random k-labelsets: An ensemble method for multilabel classification. In: European conference on machine learning. pp. 406–417. Springer (2007)
30. Wilson, D.R., Martinez, T.R.: Improved heterogeneous distance functions. Journal of artificial intelligence research **6**, 1–34 (1997)
31. Zhang, M.L., Zhou, Z.H.: Multilabel neural networks with applications to functional genomics and text categorization. IEEE transactions on Knowledge and Data Engineering **18**(10), 1338–1351 (2006)
32. Zhang, M.L., Zhou, Z.H.: Ml-knn: A lazy learning approach to multi-label learning. Pattern recognition **40**(7), 2038–2048 (2007)