Multiobjective Genetic Algorithm for Outliers Detection

Lukasz Chomatek
Lodz University of Technology
Institute of Information Technology
Lodz, Poland
Email: lukasz.chomatek@p.lodz.pl

Agnieszka Duraj
Lodz University of Technology
Institute of Information Technology
Lodz, Poland
Email: agnieszka.duraj@p.lodz.pl

***Abstract*—The users of information systems often have to deal with outliers in their data. Such outliers can have negative (i.e. abnormal observations) or positive (i.e. detection of new features) impact on their work. Despite the fact, that several methods of outlier detection already exist, there is still a need to improve them. In this work we propose a method for evolutionary outlier detection. The novelty of our approach is a set of criteria, which are used to decide, whether to treat observation as an outlier or not. Conducted research revealed that our method performs very well on the selected problems.**

***Keywords*—multiobjective optimization; outlier detection; genetic algorithms**

I. Introduction

Currently, data collection and processing systems are commonly used not only by companies, corporations and administrations but also by private individuals. Decision support and business intelligence systems are gaining increasing importance. Therefore, it is important for scientists to provide the methods that will facilitate the process of data mining. An overview of data classification or data grouping algorithms can be found in [1], [2], [3].

The present study focuses on a branch of data mining, namely outlier detection. This issue plays an important role in data classification and grouping systems dealing with various data types. Research in this field is crucial, since the detection of an outlier object can affect the whole knowledge discovery process in a given database.

Outliers may have different origins. Firstly, an outlier may be caused by a damaged machine and erroneous measurement made by it. It may also indicate noise in the signal. An outlier can often result from human error. On the other hand, it may be understood as a data point that is distinct from the rest of the data. An outlier object is characterized by features unknown so far, or by different values of attributes. In classification or grouping tasks, an outlier object may be misclassified by the algorithm, because it may be interpreted as a new object, unknown so far in the database.

Numerous studies have offered a variety of definitions of an outlier. Many different outlier detection methods have also been proposed in the literature. Outlier detection algorithms are used in applications related to the identification of features of satellite images, for the detection of congestion or hacking into computer networks, diagnostics of machines, identification of new molecular structures in pharmaceutical research, medical diagnostics and so on.

978-1-5090-5795-5/17/$31.00

For the definition of the outlier as the object belonging to the other group, which does not come from erroneous or noisy data, we can assume the definition proposed by Hawkins [4] which states: For any object $x \in X$, if x has some abnormal characteristics as compared to the other objects in X, we may consider x as an outlier. Another definitions can be found in [5], [6].

In the present paper, assuming Hawkins's definition quoted above, the authors endeavor to detect outliers using genetic algorithms.

The genetic algorithm is a very popular iterative method for the heuristic optimization, which processes the population of candidate solutions. Each individual consists of a number of genes. Values stored in genes are used to compute the value of the fitness function, which is a measure of the individual's effectiveness. Each iteration of the genetic algorithm consists of three steps:

- selection
- crossover
- mutation

In the first operation, one has to find the individuals for reproduction, which entails the need to compute the value of the fitness function for each individual. The aim of crossover is to produce the individuals for the next iteration. During this operation, the genes from parents are mixed to form the new solutions. Mutation is the operator, the task of which is to inject a new pool of genes into the population. This process is performed for each offspring individually. With the small probability values of its genes are changed.

Real-valued fitness functions can be efficiently optimized with the use of the genetic algorithms. There are a lot of techniques designed to mitigate the influence of premature convergence and ensure the proper exploration of the search space [7], [8].

The paper is organized as follows: At first, we present the current methods applied to outlier detection. Then, our method for solving this problem with genetic algorithms is presented. In section IV, we present the results of our experiments. The

paper concludes with a summary and suggestions for further research.

II. Related Work

A. Deterministic Methods of Outlier Detection

A review of the literature (see for example [5], [9], [10]) shows that there are three basic approaches to outlier detection.

The first of them comprises a group of methods which assume that errors and failures are separated from the correct data. The outlier is referred to as the feature of an object which did not previously appear in the analyzed database. These methods, analogous to unsupervised clustering, work mostly backwards. They require that the data be available before the pre-processing and be unchanged.

The second approach encompasses a group of methods that assume the occurrence of valid and invalid data in the database. As with the supervised classification, it is important that the methods belonging to this group can be used in real time. The classifier learns the model and qualifies new patterns to the appropriate classes. If the new pattern lies within the scope of valid values, it is qualified as a correct pattern, otherwise, it as an outlier. The classification algorithms belonging to this group require a good distribution of the data.

The third approach assumes the occurrence of correct data only, or a very small number of cases with abnormal patterns. Therefore, this assumption requires segregated data, although it learns only one class of data designated as valid. A noteworthy fact is that learning in this group of methods is incremental, since the new emerging patterns tune the model to improve the matching. The variable boundary area of the algorithm provides the ability to estimate the degree of uniqueness.

Each of these approaches applies algorithms associated with statistical methods, machine learning, and artificial intelligence. In most works, the algorithm is closely connected with the data under examination.

An overview of outlier detection methods was provided by Aggrawal [11], [5], Hawkins [4]. See also the works of Hodge [9] or Niu [10].

In the works of Barnett and Lewis [12], Rousseeuw and Leroy [13], Hawkinks [4], the detection of outliers is closely linked with the use of statistical methods. The authors often treat linear or logarithmic regression as a preliminary analysis of the data, detecting all outlying points [14], [4], [6], [15].

Undoubtedly, the most popular outlier detection algorithms are: the distance-based-outlier (DB-outlier for short) [16], [17], the Local outlier factor (LOF) [18], Connectivity-based Outlier Factor (COF) [19], and Density Based Spatial Clustering of Applications with Noise (DBSCAN) [20], local outlier [21], [6] See also [22], [23], [24] .

Duraj and Szczepaniak, formulated the definition of an outlier in linguistic summaries for both numerical and textual data [25], [26]. The authors used linguistic summaries of databases containing text and numeric attributes for the detection of outliers. The method employed Yagers standard summary based on fuzzy interval sets.

B. Outlier Detection with Use of Genetic Algorithms

However the authors prefer to use deterministic methods in the outlier detection process, there are some works that incorporate genetic algoritms and other heuristic methods for solving this task.

One of the first works on application of genetic algorithms for outlier detection was [27]. Authors proposed the algorithm, where genes represented the indices of observations that should be treated as outliers. They examined three fitness functions which were based on Least Squares, Cook's Distance [28] and Andrews/Pregibon ratio [29]. Authors decided to use the uniform order-based crossover where duplicated entries were removed in the offspring.

Tolvi [30] proposed another approach, where individuals were binary vectors. In this encoding, 1 on the $i-th$ place in the chromosome meant that the observation is an outlier. In this method there is no need to perform any other processing after the crossover and mutation. To compute the fitness function, authors proposed a linear regression model. What is more, they incorporated the genetic algorithm for the selection of variables with the method based on Bayesian Information Criterion [31]. The same encoding was widely used by many other authors [32], [33].

Detection of outliers is sometimes a result of clustering. Taloba et al. [34] utilized the Improved Genetic k-Means algorithm to perform clustering with the automatic detection of the outliers. Published results shown that the method performs very well.

The methods mentioned above were general. They can be easily applied for outlier detection in other datasets than ones presented in the original papers. On the other hand, authors proposed many other methods which are problem dependent, i.e.: time series [33], fuzzy sets [35], noise reduction [36].

C. Multiobjective Optimization in Genetic Algorithms

Dealing with multiple objectives in the optimization problem (MOP) can be solved in the several ways. One of the possible approaches is to choose real weight for each objective and the solve the single objective problem (SOP). In most problems when one of the objectives has a good value, values of the other objectives are worse than in other solutions. To choose the set of the best solution, the domination relation is proposed.

We say that the solution $f(x_1) \in \mathbb{R}^n$ dominates $f(x_2) \in \mathbb{R}^n$ iff

$$\forall_{i \in 1,\ldots,n} f_i(x_i) \leq f_i(x_2) \tag{1}$$

and

$$\exists_{i \in 1,\ldots,n} f_i(x_1) < f_i(x_2) \tag{2}$$

where $f_i(x)$ is the i-th value of the fitness function f, and n is the number of objectives.

A good study of genetic algorithms dedicated for solving MOP are in [37]. One of the most popular approaches in this domain is SPEA2 [38], which ranks the solutions by the number of other individuals they dominate and utilize the concept of archive to protect the best solutions. Deb et

al. [39] proposed NSGA-II algorithm, which also favors non-dominated solutions and ensures that individuals are genetically diversified. PESA-II algorithm [40] not only uses the archive, but divides the search space into some number of hypercubes. In each region of the search space, the independent optimization process is performed.

III. Proposed Method

As it was mentioned in the previous section, both deterministic and genetic algorithms are parametrized. It is a very time consuming task to tune the algorithms correctly and it is unusual to find the comprehensive set of parameters, which is applicable for all the problems. Due to the numerous variants of algorithms intended to identify outliers we decided to treat them equally. Our approach is to get some number of measures that identify outliers and choose Pareto-optimal solutions with respect to this set of measures.

A. Encoding and Phenotype

In our approach we used the same encoding as introduced in [30]. Each chromosome consists of l binary genes, where l is the number of samples in the dataset. The value of each gene can be 1 or 0, depending on the corresponding sample is an outlier or not.

To obtain the phenotype, we need to take all the samples from the dataset, where the corresponding gene is set to 1.

B. Mutation and Crossover

During the experiments we used single point crossover with the constant probability p_c. It means that for the chromosomes selected for the reproduction we chose a locus, after that the parts of the chromosomes were exchanged between the parents.

In the mutation we flipped the values of genes in the offspring chromosomes. Each gene could have been changed with the constant probability p_m.

C. Objective Function

Properly built objective function has the major influence to the performance of the optimization algorithm. In our experiments we used three types of measures as the components of the fitness function:

- average distance from the nearest neighbours
- distance from the centroid
- overall number of outliers

Let us denote as x the set of the samples identified as outliers. Let x' be a set of samples that are not outliers.

One of the objectives in the fitness function is the average distance of the samples in x from the k nearest samples in x'. To obtain such a value we need to:

1) for each sample $s \in x$ we calculate the distances from the samples in x'. We sort the samples and take the k-th value.
2) add the values obtained in the previous step and divide them between the number of elements in x.

We denote such objective as $d_k(x)$.

The second type of objective is the average distance from the samples in x from the centroid of the x' dataset. We compute the centroid as the average position of the vectors in x':

$$c(x') = \frac{\sum_{s' \in x'} s'}{|x'|} \tag{3}$$

The actual value of this objective is obtained as follows:

$$dc(x) = \frac{\sum_{s \in x} s - c(x')}{|x|} \tag{4}$$

where $|x|$ is the number of elements in the set x.

The last type of objective is the number of the identified outliers (denoted as $no(x)$) as it is supposed that the number of outliers should be much smaller than the number of samples in the whole dataset.

In our experiments we used the fitness function composed of four objectives: d_1, d_2, d_3, dc and no.

IV. Experimental Results

We performed our experiments with use of two algorithms: NSGA-II and PESA-2. At the beginning of this section we describe in details the simulation environment and the chosen dataset. Then we present the results of the experiments on the limited and complete dataset.

The results obtained during the optimization with the multi-objective algorithm can contain more than one chromosome. In this case we decided to introduce the accuracy factor, which is a quotient of the number of solutions where the given sample was identified as an outlier to the number of all Pareto-optimal solutions.

A. Environment and Data

As an environment for simulations we considered 3 different solutions:

- HeuristicLab [41], implemented in .NET, supports many different heuristic methods, but only NSGA-II is provided for MOP.
- MOEAFramework [42], implemented in Java, supports different genetic algorithms for MOP.
- jMetal [43], [44], implemented in Java, offers almost same features as MOEAFramework

After the brief analysis of these frameworks, we decided to use jMetal due to the ease of implementation of new problems, such as outlier detection. We have found only one drawback in this environment - there is no direct method to decide how the initial population is generated. To do it, one has to either inherit from the concrete algorithm and rewrite whole `execute` method, or propose a new solution type (that probably duplicates the existing one), where one can decide how variables are created. The most general solution is to implement a template method pattern for the algorithm execution and move the initialization to the specialized method, but it needs a changes in all algorithms provided by jMetal.

In our simulations we used well known Computer Hardware Dataset[45]. Each sample contains eight attributes which describe machine cycle time, amount of memory, number

TABLE I
ATTRIBUTES IN COMPUTER HARDWARE DATASET

Attribute	Abbr.	Min	Max
machine cycle time [ns]	MYCT	17	1500
min. main memory [kB]	MMIN	64	32000
max. main memory [kB]	MMAX	64	64000
cache memory [kB]	CACH	0	256
min. channels [units]	CHMIN	0	52
max. channels [units]	CHMAX	0	176
published performance	PRP	6	1150
estimated performance	ERP	15	1238

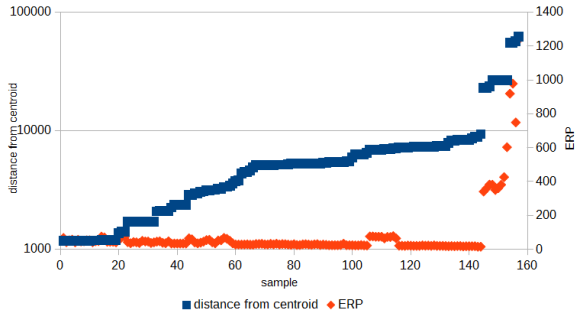

Fig. 1. Distance of samples from the centroid of the dataset

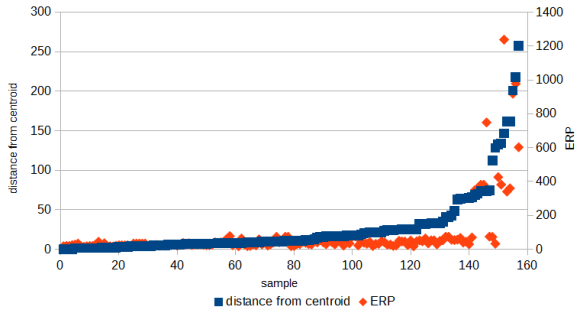

Fig. 2. Distance of samples (limited feature set) from the centroid of the dataset

channels and performance. Detailed desription of the attributes is shown in table I.

For our experiments we used a limited dataset, containing only samples that have ERP lower than 100 or greater than 300. It was intended to treat the hardware with lower ERP as normal observations (144 samples) and these with higher ERP as outliers (13 samples). We performed two experiments: one on the full feature set and the second with features regarding the cache memory and channels.

At the beginning we decided to check, whether our estimation was correct. We calculated the centroid of the chosen dataset and observed that samples with greater ERP values are farther from the centroid (Fig. 1). On the other hand, for the limited feature set, there exist some samples that are far from the centroid and have low ERP. (Fig. 2).

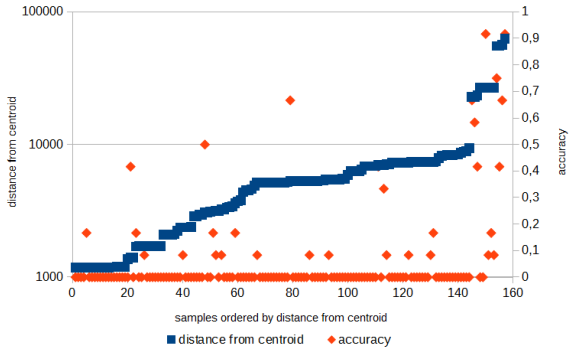

Fig. 3. Results for NSGA-II algorithm with respect to accuracy parameter

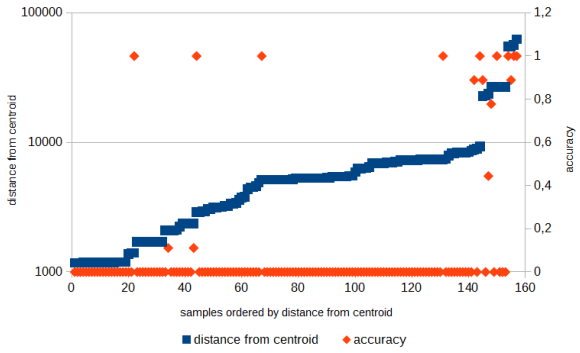

Fig. 4. Results for PESA-2 algorithm with respect to accuracy parameter

B. *Finding Outliers Injected to the Data*

In our experiment we used the following parameters of the algorithms:

- crossover probability - 90%,
- mutation probability - 2%,
- population size - 40,
- maximum evaluations of the fitness function - 7000 for PESA2 and 4000 for NSGA-II,
- archive size - 10,
- number of bisections - 5

The first five parameters were needed by both examined algorithms. The last parameter was required only in PESA-2 algorithm.

The examples of obtained results are shown on Fig. 3 and Fig. 4. For NSGA-II, there are more identified outliers, but some of them have very low accuracy values. This is caused by the fact, that there were many nondominated solutions and only few of them marked such samples as outliers. Observations that are far from the centroid have usually higher accuracy.

Results obtained for PESA-2 algorithm are slightly better. In this case there was a lower number of Pareto-optimal solutions than in the previous case. What is more, samples with the highest accuracy are gathered in the right part of the chart, what was expected.

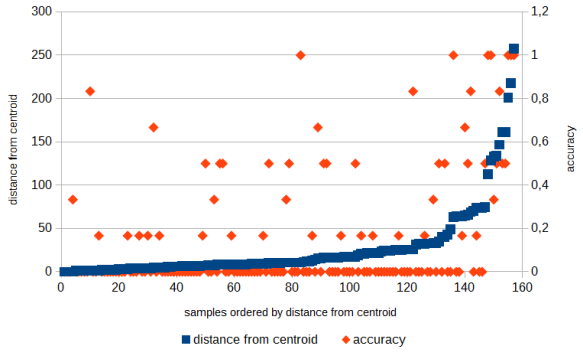

Fig. 5. Distance from centroid (limited features) for NSGA-II algorithm with respect to accuracy parameter

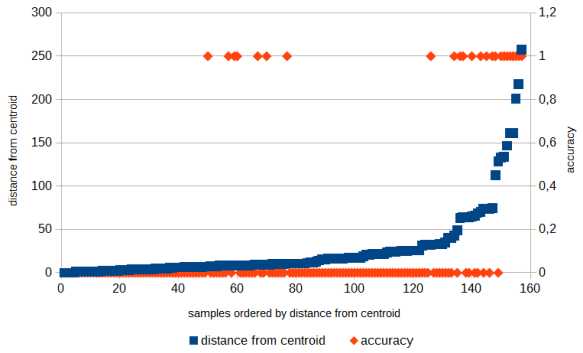

Fig. 6. Distance from centroid (limited features) for PESA-2 algorithm with respect to accuracy parameter

C. Finding Outliers - Reduced Feature Set

The next experiment was performed on the same dataset, but only three features were taken into account: CACH, CHMIN, CHMAX. The only difference in the parameters of genetic algorithms was the number of maximum evaluations of the fitness function changed to 5000 for NSGA-II and 2000 for PESA-2.

Obtained results are shown on Fig. 5 and Fig. 6. As one can see, PESA-2 gives significantly better results than SPEA-II. In the results for SPEA-II, there are many samples with low accuracy, which appear in the whole dataset. On the other hand, hardware with high estimated performance (far from the centroid) usually has the accuracy not lower than 0.5. For PESA-2 (Fig. 6) we obtained only one Pareto-optimal solution, so that accuracy can have only 2 values. In the previous experiment, usually there were more nondominated solutions. As one can see, the number of outliers in the hardware closer to the centroid is lower, than in the results for NSGA-II.

V. Conclusion

In this article we proposed a method for the detection of outliers with use of the genetic algorithms. During the conducted research this approach performed very well. In the experiment with some data representing abnormal observations injected to the dataset both tested algorithms - NSGA-II and PESA-2 found about 70% of outliers and abnormal observations. During the second experiment most of the data marked as outliers was close to the border of the dataset. In both experiments PESA-2 performed slightly better than NSGA-II as it found outliers with the higher accuracy than NSGA-II. Our future plans on this research field is to incorporate knowledge about the dataset in the genetic operators.

Acknowledgment

The database used in experiments was taken from the UCI Machine Learning Repository[45].

References

[1] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.
[2] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.
[3] P.-N. Tan *et al.*, *Introduction to data mining*. Pearson Education India, 2006.
[4] D. M. Hawkins, *Identification of outliers*. Springer, 1980, vol. 11.
[5] C. C. Aggarwal, "Outlier detection in categorical, text and mixed attribute data," in *Outlier Analysis*. Springer, 2013, pp. 199–223.
[6] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognition Letters*, vol. 24, no. 9, pp. 1641–1650, 2003.
[7] H. M. Pandey, A. Chaudhary, and D. Mehrotra, "A comparative review of approaches to prevent premature convergence in ga," *Applied Soft Computing*, vol. 24, pp. 1047–1077, 2014.
[8] M. Rocha and J. Neves, "Preventing premature convergence to local optima in genetic algorithms via random offspring generation," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 1999, pp. 127–136.
[9] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.
[10] Z. Niu, S. Shi, J. Sun, and X. He, "A survey of outlier detection methodologies and their applications," in *International Conference on Artificial Intelligence and Computational Intelligence*. Springer, 2011, pp. 380–387.
[11] C. C. Aggarwal and P. S. Yu, *Outlier detection for high dimensional data*. ACM Sigmod Record, vol. 30, no. 2.
[12] V. Barnett and T. Lewis, *Outliers in statistical data*. Chichester: John Wiley, 1995. 584p, 1964.
[13] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John wiley & sons, 2005, vol. 589.
[14] A. Goel, H. Xu, and S. M. Shatz, "A multi-state bayesian network for shill verification in online auctions." in *SEKE*, 2010, pp. 279–285.
[15] S. Hekimoglu, R. C. Erenoglu, and J. Kalina, "Outlier detection by means of robust regression estimators for use in engineering science," *Journal of Zhejiang University Science A*, vol. 10, no. 6, pp. 909–921, 2009.
[16] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: algorithms and applications," *The VLDB JournalThe International Journal on Very Large Data Bases*, vol. 8, no. 3-4, pp. 237–253, 2000.
[17] E. M. Knox and R. T. Ng, "Algorithms for mining distancebased outliers in large datasets," in *Proceedings of the International Conference on Very Large Data Bases*. Citeseer, 1998, pp. 392–403.
[18] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *ACM sigmod record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.
[19] W. Jin, A. K. Tung, and J. Han, "Mining top-n local outliers in large databases," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001, pp. 293–298.
[20] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.
[21] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Loop: local outlier probabilities," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 1649–1652.

[22] G. H. Orair, C. H. Teixeira, W. Meira Jr, Y. Wang, and S. Parthasarathy, "Distance-based outlier detection: consolidation and renewed bearing," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 1469–1480, 2010.

[23] V. Kreinovich, L. Longpré, P. Patangay, S. Ferson, and L. Ginzburg, "Outlier detection under interval uncertainty: algorithmic solvability and computational complexity," *Reliable Computing*, vol. 11, no. 1, pp. 59–76, 2005.

[24] E. Schubert, A. Zimek, and H.-P. Kriegel, "Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection," *Data Mining and Knowledge Discovery*, vol. 28, no. 1, pp. 190–237, 2014.

[25] A. Duraj, P. S. Szczepaniak, and J. Ochelska-Mierzejewska, "Detection of outlier information using linguistic summarization," pp. 101–113, 2016.

[26] A. Duraj and P. S. Szczepaniak, "Information outliers and their detection," in *Information Studies and the Quest for Transdisciplinarity*. World Scientific Publishing Company, 2017, pp. 413–437.

[27] K. D. Crawford and R. L. Wainwright, "Applying genetic algorithms to outlier detection." in *ICGA*, 1995, pp. 546–550.

[28] R. D. Cook, "Detection of influential observation in linear regression," *Technometrics*, vol. 19, no. 1, pp. 15–18, 1977.

[29] D. F. Andrews and D. Pregibon, "Finding the outliers that matter," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 85–93, 1978.

[30] J. Tolvi, "Genetic algorithms for outlier detection and variable selection in linear regression models," *Soft Computing*, vol. 8, no. 8, pp. 527–533, 2004.

[31] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[32] Ö. G. Alma, K. Serdar, and U. Aybars, "Genetic algorithm based outlier detection using bayesian information criterion in multiple regression models having multicollinearity problems," *Gazi University Journal of Science*, vol. 22, no. 3, pp. 141–148, 2009.

[33] D. Cucina, A. di Salvatore, and M. K. Protopapas, "Outliers detection in multivariate time series using genetic algorithms," *Chemometrics and Intelligent Laboratory Systems*, vol. 132, pp. 103–110, 2014.

[34] A. I. Taloba, M. Marghny, and R. M. A. El-Aziz, "Outlier detection using improved genetic k-means," *International Journal of Computer Applications*, 2014.

[35] C.-C. Lin and A.-P. Chen, "Fuzzy discriminant analysis with outlier detection by genetic algorithm," *Computers & Operations Research*, vol. 31, no. 6, pp. 877–888, 2004.

[36] P. Górski and L. Morzyński, "Active noise reduction algorithm based on notch filter and genetic algorithm," *Archives of Acoustics*, vol. 38, no. 2, pp. 185–190, 2013.

[37] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992–1007, 2006.

[38] E. Zitzler, M. Laumanns, L. Thiele *et al.*, "Spea2: Improving the strength pareto evolutionary algorithm," 2001.

[39] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[40] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "Pesa-ii: Region-based selection in evolutionary multiobjective optimization," in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*. Morgan Kaufmann Publishers Inc., 2001, pp. 283–290.

[41] A. Elyasaf and M. Sipper, "Software review: the heuristiclab framework," *Genetic Programming and Evolvable Machines*, vol. 15, no. 2, pp. 215–218, 2014. [Online]. Available: http://dx.doi.org/10.1007/s10710-014-9214-4"

[42] D. Hadka, "Moea framework: a free and open source java framework for multiobjective optimization," 2012.

[43] J. J. Durillo, A. J. Nebro, and E. Alba, "The jmetal framework for multi-objective optimization: Design and architecture," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, 2010, pp. 1–8.

[44] J. J. Durillo and A. J. Nebro, "jmetal: A java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, no. 10, pp. 760–771, 2011.

[45] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml