Fuzzy Set-Based Isolation Forest

Paweł Karczmarek
Department of Computer Science
Lublin University of Technology
Lublin, Poland
pawel.karczmarek@gmail.com

Adam Kiersztyn
Department of Computer Science
Lublin University of Technology
Lublin, Poland
adam.kiersztyn.pl@gmail.com

Witold Pedrycz
Department of Electrical & Computer Engineering
University of Alberta
Edmonton, Canada
Department of Electrical and Computer Engineering
King Abdulaziz University
Jeddah, Saudi Arabia
Systems Research Institute
Polish Academy of Sciences
Warsaw, Poland
wpedrycz@ualberta.ca

***Abstract*—One of the main challenges is the analysis of large data sets, in particular those containing various types of data, such as time, place, image, and those assuming categorical values. This type of data may contain numerous outliers. Despite the continuous development of data analysis, many methods can be effectively improved, in particular through the use of efficient solutions based on fuzzy set technologies. In this paper, we analyze the improvement of a well-known method, i.e. Isolation Forest, for which we introduce an innovative modification, referred to as the Fuzzy Set-Based Isolation Forest.**

***Index Terms*—isolation forest, membership value, fuzzy set-based isolation forest, outlier detection, anomaly score**

I. Introduction

Contemporary data analysis brings many critical problems for the development of societies and business organizations. One of them is the analysis of the integrity of large data sets, in particular the search for anomalies and the so-called outliers in these collections. These types of data irregularities can have different origins, e.g., they can be incorrectly entered ”manually” by users, added randomly as a result of erroneous implementations (e.g. through data format), or have significant variations in amplitude. Irregularities are not just mistakes. They may have their sources in, for example, extortion attempts or hacker attacks. In view of such a wide range of speeches, the use of effective methods, often oriented to a given subject field, seems to be a key task.

One of the important branches of applications is the analysis of transportation data. Modern logistics processes are oriented towards optimizing delivery costs and minimizing transportation time. These goals can be achieved on the basis of reliable analysis of historical data not affected by errors or, in a broader sense, by anomalies. Such data are particularly susceptible to errors, because human error often occurs during their acquisition. In addition, they are mixed (categorical data on types of shipments and cargoes, time series containing information on individual stages of the journey, and, finally, various quantitative data). On the one hand, they are characterized by significant repeatability; on the other, individual transports often differ in small details and it is difficult to determine whether a given historical record contains erroneous values, apart from obvious errors such as negative times or fields not filled with any data.

We recall the most important results regarding the detection of anomalies, reported in the literature. Classic approaches were based on k-nearest neighbor algorithm [1]–[3] or Isolation Forest [4], [5] (see the next section) and its enhancements [6]–[8] built upon binary search trees learned on samples of a dataset. Other methods incorporate kernels [9], support vector machines [10], autoencoders [11], self-organizing maps [12], or long-short term memory [13]. Also many approaches were based on DBSCAN algorithm [14]–[16], Fuzzy C-Means (FCM) [17]–[19], or fuzzy C-medoids [20]. Very comprehensive surveys are present in [21]–[24]. Here, it is worth to mention also a few works dedicated to anomaly or outlier detection in transportation. The methods are manifold embedding [25], spatial-region-based and perimeter-based metrics [26], cascaded clustering schemes [27], multi-channel singular spectrum analysis [28], sparse processing [29], FCM [30], or scene modeling [31].

The main objective of this work is to propose a new, extended and innovative version of the aforementioned Isolation Forest algorithm based on the analysis of the degrees of belonging (membership) of attributes of individual records to clusters (nodes) resulting from the division of trees on the basis of which the forest is built. The memberships are determined on the basis of distance from the so-called middle of the cluster, i.e. the average value of the attribute. The advantages of this approach include the fact that it is very intuitive, and, as experiments show, returns almost zero membership value for abnormal data. Moreover, the function constructed in such a way that returns the degree of anomaly does not require the use of a complicated standardization formula [4], [5]. Finally, the method's execution time does not exceed, and in some cases is even shorter than the classical method. Moreover, we are interested in an application of our version of Isolation

Funded by the National Science Centre, Poland under CHIST-ERA programme (Grant no. 2018/28/Z/ST6/00563).

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

Forest, namely, Fuzzy Set-Based Isolation Forest (FSBIF), to the problem of anomaly detection in transportation datasets.

The structure of the paper is as follows. In Section II, we recall a concept of Isolation Forest approach. In Section III proposed is its enhancement based on the membership concept. Section IV covers the results of numerical experiments with artificial and transportation databases. Finally, conclusions and future work directions are discussed in Section V.

II. Isolation Forest

Isolation Forest [4], [5] is generally built through two general steps. The first is training on a basis of binary search trees building. The trees are constructed based on samples of the overall dataset D. The second step is scoring which is realized on a basis of searching these binary search trees. The arguments here are all the records contained in D. Assume that r is a number of all the records in D, each of them has q attributes. Let the number of binary search trees be b and the number of samples generating these trees be n. Then, obviously, the samples are x_i, $i = 1, 2, \ldots, n$. Such a binary decision tree is built in the following way. On a basis of the subsample set X, the attribute B is randomly chosen, and its value is also chosen in a random way. This is a threshold (cutoff) value t dividing the set of this attribute's values onto two subsets related to two nodes of the root. It is worth to note the randomness of the threshold. Next, again, for the subsets, attributes and their values are randomly found with preserving of the filter values obtained at the earlier divisions.

The second stage of Isolation Forest is devoted to scoring the anomaly value. Each of the elements of the original dataset D is an input to the tree searching algorithm and the final anomaly score is [4], [5]:

$$s\left(x\right) = 2^{-E(x)/C(n)} \tag{1}$$

where

$$C\left(x\right) = 2H\left(x-1\right) - 2\left(x-1\right)/x \tag{2}$$

where

$$H\left(x\right) = \ln\left(x\right) + 0.5772156649 \tag{3}$$

and $E\left(x\right)$ is a sum of all the search lengths when tracing all the binary search trees while this sum is enlarged by the value of C with argument being the number of elements in the reached node if the maximal depth of a tree is reached. Here, one has to set a maximal depth of a tree which is suggested to be a ceil of $\log_2 n$. $C\left(x\right)$ plays the role of normalizing function, where $H\left(x\right)$ is a harmonic operator, see [32], where the consideration of searching binary trees was presented.

III. Enhancement of Isolation Forest

In response to such a rather complicated process of building a tree and finding the value of anomalies, we propose the following modification of the algorithm. In the first stage of the above method, we always determine the average value of m from all the values of the chosen attribute located in the filtered cluster. This value is saved in a memory with the corresponding node. Then, in the second stage, the degree of membership to the cluster constructed in such a manner is calculated as

$$p\left(x\right) = 1 - d\left(x, m\right)/d\left(m, m_L\right) \tag{4}$$

when $x < m$ and

$$p\left(x\right) = 1 - d\left(x, m\right)/d\left(m, m_H\right) \tag{5}$$

otherwise. This construction is taken directly from the concept of the triangular membership function, where m is a modal value. Here, m_L and m_H are the lowest and highest boundary of a cluster, respectively. It is worth noting that when constructing the binary search tree, at the beginning these values are the minimal or maximal of a set. After the consecutive divisions of subsamples the values are respectively higher or lower, if the filters coming from the divisions appear.

The final anomaly score is the sum of all memberships at each node and after searching all the trees the average value is found. Of course, when some node is empty, the membership is zero at this node.

An example of a process of membership finding is presented for a simplified subset of data in Fig. 1. The point for which an anomaly score is calculated is $(x, y) = (0.7, 0.6)$. The first division of a set occurs at the vertical line (0.5). Next, the yellow set's center (#1) is at point $x = 0.8$. Then, the membership is $p = 2/3$. The center (#2) of the red set is at $y = 0.3$. Then, $p = 1/4$. In the green area the point (x, y) is a singleton, so the membership is not calculated (as it is obviously 1). The final anomaly score is $2/3 + 1/4 = 11/12$.

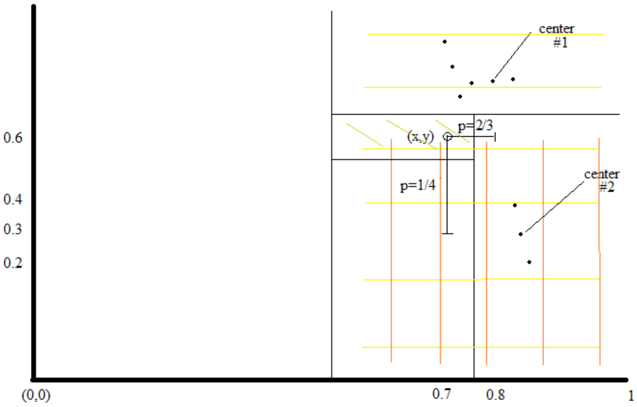

Fig. 1. The process of membership finding.

For the sake of order, we supplement the text with an overview illustration of the binary search tree in the classic Isolation Forest on a basis of divisions coming from the above example, see Fig. 2. Here, the trace length is 2 and this is the argument of the function E. Moreover, pseudocodes of training the binary search trees and anomaly scoring are presented in Algorithm 1 and Algorithm 2 code lines.

IV. Experimental Results

Here, we discuss the results of numerical experiments with Isolation Forest and its fuzzy set-based counterpart. We

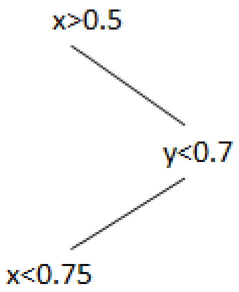

Fig. 2. Binary tree searching.

Algorithm 1 Algorithm for training

```
Input: $D$ - dataset, $r$ - number of all the records in $D$, $q$ – number of attributes, $b$ - number of binary search trees, $n$ - number of samples generating trees
Output: a set of binary trees out
height limit $l = ceiling(\log_2 n)$
for $i = 1$ to $b$ do do
    for $j = 1$ to $n$ do do
        randomly choose $x_j$ from the set $D$
    end for
    build binary search tree $T$ [4] on a basis of samples $x_1, \dots, x_n$
    for each node $w$ in $T$ do do
        $w_{memb}$ = average value of the distances of the points belonging to the node on a basis of (4) or (5)
    end for
end for
return return forest of binary trees
```

Algorithm 2 Algorithm for scoring

```
Input: $x$ - an element of $D$, $l$ - path length, $r, n, w_{memb}$ for each node (see alg. 1)
Output: $s$ - anomaly score of $x$ out
$s = 0$
for $i = 1$ to $r$ do do
    for $j = 1$ to $n$ do do
        randomly choose $x_j$ from the set $D$
        traverse the $j$th tree $T$
        for each traversed node $w$ in nodes of $T$ do do
            $s$ += $w_{memb}$
        end for
    end for
end for
return score $s$
```

consider two datasets, namely (i) an artificially generated two-dimensional dataset containing $100,200$ points, and (ii) the data coming from the set New York City Taxi Trip Data [33] containing records of New York taxi travels including geographical coordinates. We have used $737,462$ non-empty records here. In all the series of experiments, the number of decision trees is 100, the number of samples of the dataset building the decision trees is 128, and the maximal depth is set to 9, as recommended in [4], [5].

A. Artificial Dataset

The artificial dataset was created to place the most of the points inside selected geometrical figures while the rest of the points are located outside of them. Therefore, it is easy to observe the efficiency of the two compared methods. The methods clearly differ in performance. Isolation Forest (see, Fig. 3) tends to suspect points located between figures as the outliers or anomalies. However, all the points at the plot are of relatively similar color. Finally, the boundary points of the figures are evaluated as abnormal. The Fuzzy Set-Based method is performing well for the outside points (see, Fig. 4). However, it classifies the points located between the figures as close to normal. However, the boundary points of the figures are not classified as abnormal.

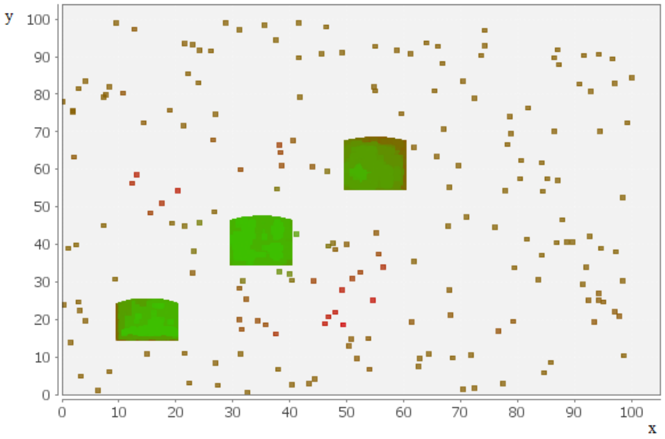

Fig. 3. The results of Isolation Forest (artificial dataset).

B. Taxi Database

Next dataset being analyzed is the publicly available New York City Taxi Trip Data [32]. The results presented in Fig. 5 show that the most isolated points according to Isolation Forest method are relatively close to New York while Fuzzy Set-Based Isolation Forest marks the points far from NY (located in Asia) as the 1000 most isolated, see Fig. 6. They are not seen by Isolation Forest. The next two figures, namely Fig. 7 and Fig. 8, present the results with lower levels of anomaly scoring. All the points located outside of the blue dots should be more or less isolated. Classic Isolation Forest marks them as abnormal. However, many of them are less abnormal (marked

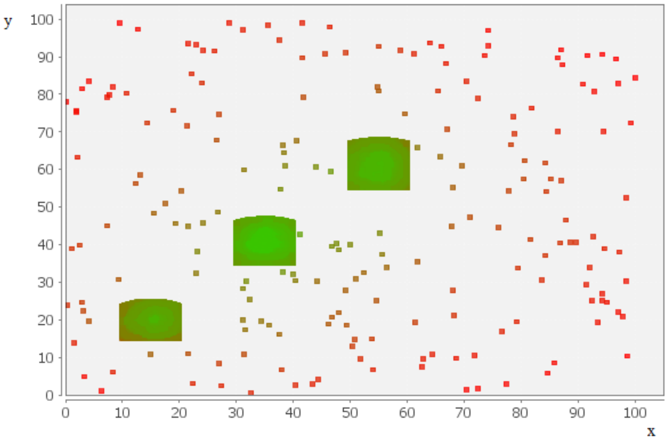

Fig. 4. The results of Fuzzy Set-Based Isolation Forest (artificial dataset).

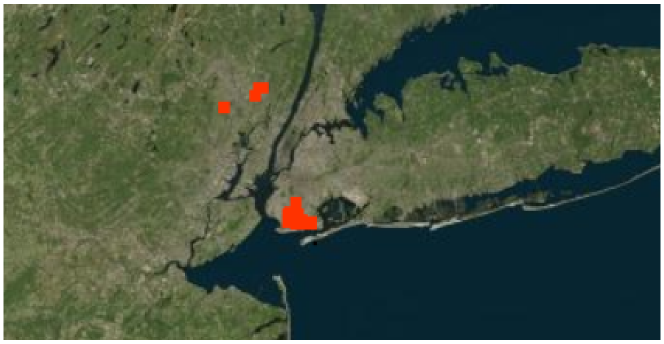

Fig. 5. Top 1000 of the most isolated points according to Isolation Forest method.

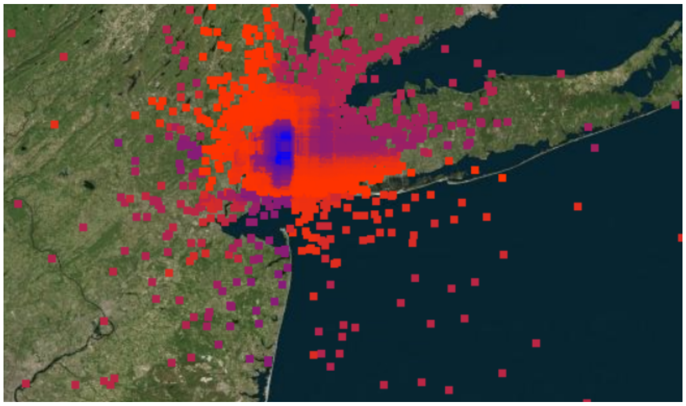

Fig. 7. The results for the neighborhood of New York (Isolation Forest).

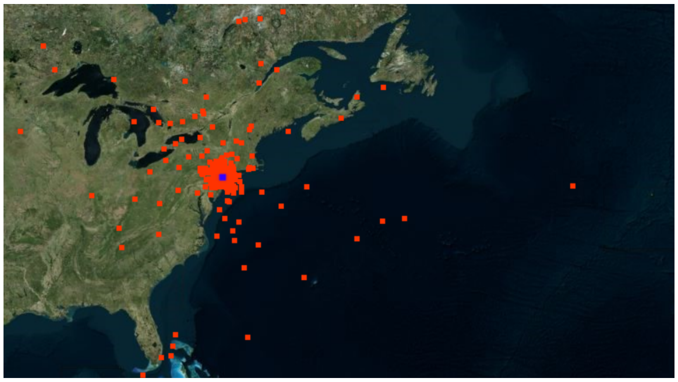

Fig. 8. The results for neighborhood of New York according to Fuzzy Set-Based Isolation Forest.

purple). Fuzzy Set-Based IF classifies them more clearly as anomalies, see Fig. 8.

The correlations for the Isolation Forest (IF) and Fuzzy Set-Based IF (FSBIF) are presented at Table I. It follows that the rankings of both methods are relatively convergent. The values of both methods are almost perfectly negatively correlated, i.e. as one increases, the other decreases. This is in line with expectations because these methods have inverted scales.

Comparing ranks (sorting based on the degree of isolation - the higher the rank, the lower the isolation, i.e., rank 1 is for the most isolated point), it turns out that in the case of standard ranking for the Isolation Forest method, all 121 points with the rank less or equal 100 are distant from Times Square on Manhattan at least 10 miles but not more than 50 miles. Whereas for the Fuzzy Set-Based Isolation Forest method all points (100) with a rank of at most 100 are at least 1000 miles away from the Times Square. This clearly shows the superiority of the proposed method over the compared.

For points distant from Times Square by at least 1000 miles, the rank distribution is shown in Fig. 9 and Fig. 10 for the Isolation Forest and Fuzzy Set-Based Isolation Forest methods, respectively. As one can observe, all these points received the highest degrees of isolation in the case of FSBIF approach,

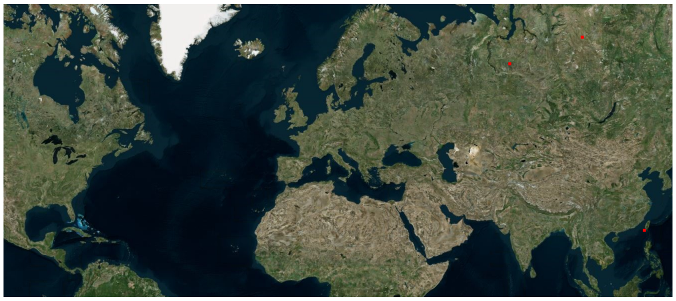

Fig. 6. Top 1000 of the most isolated points according to the Fuzzy Set-Based Isolation Forest.

TABLE I
CORRELATIONS FOR THE TAXI DATASET

	I	**II**	**III**	**IV**	**V**	**VI**
An. score IF (I)	1	-0.13	-0.97	-0.85	-0.2	0.04
An. score FSBIF(II)	-0.13	1	0.13	0.2	-0.12	-0.96
Rank (IF) (III)	-0.97	0.13	1	0.82	0.3	-0.04
Rank (FSBIF) (IV)	-0.85	0.2	0.82	1	-0.29	-0.08
Diff. in rank (V)	-0.2	-0.12	0.3	0.29	1	0.08
Dist. T. Sq. (VI)	0.04	-0.96	-0.04	-0.08	0.08	1

while for the IF method relatively low degrees.

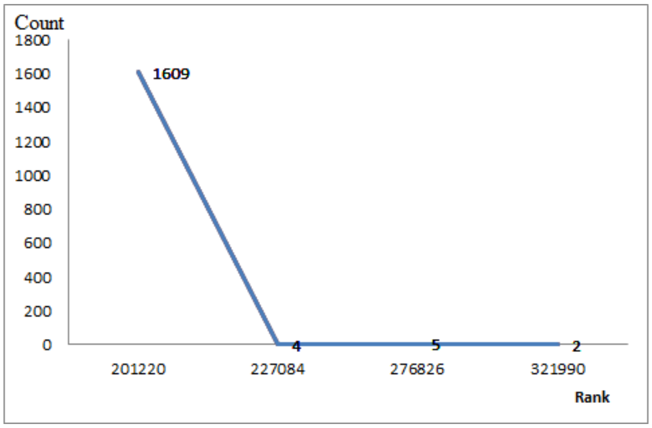

Fig. 9. Ranking distribution for points distant from Times Square at least 1000 miles (Isolation Forest).

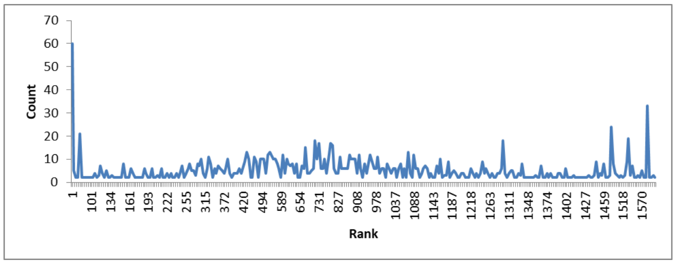

Fig. 10. Ranking distribution for points distant from Times Square at least 1000 miles (Fuzzy Set-Based Isolation Forest).

C. *Execution Times*

It is worth to mention the execution times of the method IF and FSBIF. The experiments were conducted using standard computer 64-bit architecture with 2.4GHz processor, 16GB RAM, and implemented with C++ language. It is easy to see from Table II that beside the effectiveness the FSBIF is also fast.

TABLE II
EXECUTION TIMES OF IF AND FSBIF

Method	Artificial dataset	Taxi Database [33]
Isolation Forest	11.023 s	55.997 s
Fuzzy Set-Based IF	1.152 s	7.298 s

V. CONCLUSIONS AND FUTURE WORK

In the study, we have proposed a novel and efficient enhancement of the well-known Isolation Forest method used to find anomalies and outliers in the datasets. The proposed approach is based on the distance from the center of analyzed (during the binary search tree tracing) cluster. Such way of finding anomaly scores is intuitive and fast. The series of experiments have shown the potential of the method. Therefore, it is worth to investigate it further.

With regard to the future directions of the study, it is worth to consider using fuzzy clustering to obtain the properties of the nodes of a binary search tree or engage more sophisticated approaches using Granular Computing method to analyze the attributes of the dataset records in a global and comprehensive way. Moreover, we are going to comprehensively compare our approach with other propositions and find the possibilities of merging it with other methods presented in the recent literature. Finally, we are going to test Fuzzy Set-Based version of Isolation Forest with other than artificial datasets.

ACKNOWLEDGMENT

Funded by the National Science Centre, Poland under CHIST-ERA programme (Grant no. 2018/28/Z/ST6/00563).

REFERENCES

[1] F. Angiulli and C. Pizzuti, “Fast outlier detection in high dimensional spaces,” in: Principles of Data Mining and Knowledge Discovery. Lecture Notes in Computer Science, vol. 2431, pp. 15–26, September 2002.

[2] E. B. Knorr, R. T. Ng, and V. Tucakov, “Distance-based outliers: Algorithms and applications,” VLDB Int. J. Very Large Data Bases, vol. 8, no. 3–4, pp. 237—253, February 2000.

[3] S. Ramaswamy, R. Rastogi, and K. Shim, “Efficient algorithms for mining outliers from large data sets,” in: Proceedings of the 2000 ACM SIGMOD Int. Conf. on Management of Data, 2000, pp. 427–438.

[4] F. T. Liu, K .M. Ting, and Z.-H. Zhou, “Isolation forest,” in: 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 413–422.

[5] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation-based anomaly detection,” ACM Trans. Knowl. Discov. Data (TKDD), vol. 6, no. 1, article no. 3, March 2012.

[6] J. Liu, J. Tian, Z. Cai, Y. Zhou, R. Luo, and R. Wang, “A hybrid semi-supervised approach for financial fraud detection,” in: 2017 International Conference on Machine Learning and Cybernetics (ICMLC), Ningbo, 2017, pp. 217–222.

[7] P. Karczmarek, A. Kiersztyn, W. Pedrycz, and E. Al, “K-means-based isolation forest,” Knowl.-Based Syst. 105659, February 2020.

[8] A. Mensi A and M. Bicego, “A novel anomaly score for isolation forests,” in: Image Analysis and Processing – ICIAP 2019. Lecture Notes in Computer Science, vol. 11751. Springer, Cham, September 2019, pp. 152–163.

[9] L. Zhang, J. Lin, and R. Karim, “Adaptive kernel density-based anomaly detection for nonlinear systems,” Knowl.-Based Syst., vol. 139, pp. 50–63, January 2018.

[10] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” Neural Comput., vol. 13, no. 7, pp. 1443—1471, July 2001.

[11] C. Zhou and R.C. Paffenroth, “Anomaly detection with robust deep autoencoders,” in: KDD ’17 Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, 2017, pp. 665–674.

[12] E. de la Hoz, E. de la Hoz, A. Ortiz, J. Ortega, and A. Martínez-Álvarez, “Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps,” Knowl.-Based Syst., vol. 71, pp. 322–338, November 2014.

[13] P. Malhotra, L. Vig, G. Shroff, G., and P. Agarwal, “Long short term memory networks for anomaly detection in time series,” in: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2015, pp. 89–94.

[14] R. Scitovski and K. Sabo, “DBSCAN-like clustering method for various data densities,” Pattern Anal. Appl., https://doi.org/10.1007/s10044-019-00809-z, April 2019.

[15] Z. Wu and J. Huang, “Application of DBSCAN cluster algorithm in anormaly detection,” Netw. Comput. Secur., vol. 8, pp. 43—46, 2007.

[16] J. Li and X. Hu, “Efficient mixed clustering algorithm and its application in anomaly detection,” J. Comput. Appl., vol. 7, pp. 1916–1918, 2010.

[17] H. Izakian and W. Pedrycz, “Anomaly detection in time series data using a fuzzy c-means clustering,” in: 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), Edmonton, AB, 2013, pp. 1513–1518.

[18] H. Izakian, W. Pedrycz, and I. Jamal, "Clustering spatiotemporal data: An augmented fuzzy c-means," IEEE Trans. Fuzzy Syst., vol. 21, no. 5, 855–868, October 2013.

[19] H. Izakian and W. Pedrycz, "Anomaly detection and characterization in spatial time series data: A cluster-centric approach," IEEE Trans. Fuzzy Syst., vol. 22, no. 6, pp. 1612–1624, December 2014.

[20] P. D'Urso and R. Massari, "Fuzzy clustering of mixed data," Inf. Sci., vol. 505, pp. 513–534, December 2019.

[21] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," Data Min. Knowl. Discov., vol. 29, no. 3, pp. 626—688, May 2015.

[22] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Comput. Surv. (CSUR), vol. 41, no. 3, pp. 1–72, July 2009.

[23] H. Fanaee-T and J. Gama, "Tensor-based anomaly detection: An interdisciplinary survey," Knowl.-Based Syst., vol. 98, pp. 130–147, April 2016.

[24] R. A. A. Habeeb, F. Nasaruddin, A. Gani, I. A. T. Hashem, E. Ahmed, and M. Imran, "Real-time big data processing for anomaly detection: A survey," Int. J. Inf. Manag., vol. 45, pp. 289–307, April 2019.

[25] A. Agovic, A. Banerjee, A. R. Ganguly, and V. Protopopescu, "Anomaly detection in transportation corridors using manifold embedding," in: A. R. Ganguly, J. Gama, O. A. Omitaomu, M. Gaber, R. R. Vatsavai, Eds., Knowledge Discovery from Sensor Data, CRC Press, 2008, pp. 81–105.

[26] Q. Lu, F. Chen, and K. Hancock, "On path anomaly detection in a large transportation network," Comp., Environ. Urb. Syst., vol. 33, no. 6, pp. 448–462, November 2009.

[27] A. Tsiligkaridis and I. C. Paschalidis, "Anomaly detection in transportation networks using machine learning techniques," in: 2017 IEEE MIT Undergraduate Research Technology Conference (URTC), Cambridge, MA, 2017, pp. 1–4.

[28] M. H. Hassan, A. Tizghadam, and A. Leon-Garcia, "Spatio-temporal anomaly detection in intelligent transportation systems," Proced. Comput. Sci., vol. 151, pp. 852–857, 2019.

[29] M. Wilbur, A. Dubey, B. Leão, and S. Bhattacharjee, "A decentralized approach for real time anomaly detection in transportation networks," in: 2019 IEEE International Conference on Smart Computing (SMARTCOMP), Washington, DC, USA, 2019, pp. 274–282.

[30] G. Lin, L. Xin, H. Feng, and L. Ying, "A new outlier detection algorithm and its application in intelligent transportation system," in: 2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference, Chongqing, 2014, pp. 442–445.

[31] E. Kwon, S. Noh, M. Jeon, and D. Shim, "Scene modeling-based anomaly detection for intelligent transport system," in: 2013 4th International Conference on Intelligent Systems, Modelling and Simulation, Bangkok, 2013, pp. 252–257.

[32] R. Preiss, Data structures and algorithms with object-oriented design patterns in Java, Wiley, 1999.

[33] B. Donovan, D. Work, "New York City taxi trip data (2010-2013)," University of Illinois at Urbana-Champaign, 2016.