sciendo

JAISCR, 2020, Vol. 10, No. 3, pp. 209 – 221
10.2478/jaiscr-2020-0014

# A NEW METHOD FOR AUTOMATIC DETERMINING OF THE DBSCAN PARAMETERS

Artur Starczewski[1,*], Piotr Goetzen[2], Meng Joo Er[3]

[1]*Department of Computer Engineering, Czestochowa University of Technology, al. Armii Krajowej 36, 42-200 Częstochowa, Poland*

[2]*Information Technology Institute, University of Social Sciences, 90-113 Łódź and Clark University, Worcester, MA 01610, USA*

[3]*School of Marine Electrical Engineering Dalian Maritime University, China*

[*]*E-mail: artur.starczewski@pcz.pl*

*Submitted: 10th August 2019; Accepted: 3rd March 2020*

**Abstract**

Clustering is an attractive technique used in many fields in order to deal with large scale data. Many clustering algorithms have been proposed so far. The most popular algorithms include density-based approaches. These kinds of algorithms can identify clusters of arbitrary shapes in datasets. The most common of them is the Density-Based Spatial Clustering of Applications with Noise (DBSCAN). The original DBSCAN algorithm has been widely applied in various applications and has many different modifications. However, there is a fundamental issue of the right choice of its two input parameters, i.e the $eps$ radius and the $MinPts$ density threshold. The choice of these parameters is especially difficult when the density variation within clusters is significant. In this paper, a new method that determines the right values of the parameters for different kinds of clusters is proposed. This method uses detection of sharp distance increases generated by a function which computes a distance between each element of a dataset and its $k$-th nearest neighbor. Experimental results have been obtained for several different datasets and they confirm a very good performance of the newly proposed method.

**Keywords**: clustering algorithms, DBSCAN, data mining

## 1 Introduction

Clustering algorithms discover naturally occurring structures in datasets. They group objects into meaningful clusters so that the elements of a cluster are similar, whereas they are dissimilar in different clusters. Nowadays, extensive collections of data pose a great challenge for clustering algorithms. Therefore, many new different clustering algorithms which can be applied in various areas, such as biology, spatial data analysis, business, and others are being intensively developed. It is worth considering that there is no single clustering algorithm which does the right data partitioning for all datasets. Moreover, the same algorithm can produce different results depending on applied input parameters. This problem is often resolved by using cluster validation, which is based on cluster validity indices, so several authors have proposed different validity indices e.g., [9, 23, 27, 30, 31]. Many researchers create new clustering algorithms [10, 11, 12, 13, 24, 33] or

a combined clustering algorithm with optimization and meta-heuristic algorithms [32, 2, 5, 21]. Generally, clustering algorithms can be divided into four categories: partitioning, hierarchical, grid-based and density-based clustering. Well-known partitioning algorithms include *K-means* or *Partitioning Around Medoids* (*PAM*) [3, 36]. The next clustering category called hierarchical is based on an agglomerative or divisive approach, e.g. the *Single-linkage*, *Complete-linkage*, *Average-linkage* or *Divisive ANAlysis Clustering* (*DIANA*)[19, 22]. On the other hand, the grid-based approach uses cells of a grid to analyze data elements. Such methods can be found in the *Statistical Information Grid-based* (*STING*) or *Wavelet-based Clustering* (*WaveCluster*) methods [20, 26, 34]. The last category is frequently represented by the *Density Based Spatial Clustering of Application with Noise* (*DBSCAN*) algorithm [8], which is used for various applications. This algorithm can discover clusters of an arbitrary shape and size, but requires two input parameters, i.e. the *eps* radius and the *MinPts* density threshold. Determination of these parameters is crucial to the correct performance of this clustering method.

In this paper, a new approach to determining the DBSCAN parameters is proposed. It is based on the detection of sharp distance increases generated by a function which computes distances between each element of a dataset and its $k$-th nearest neighbor. In the case of the *eps* parameter, the largest increases are used to choose a distance which can define the right value of the *eps* parameter. The choice of the *eps* value must be very precise, so several points are calculated on the chart of the sorted distances (see e.g. Figures4 and 5). On the figure, it can be observed that there is a place called the *knee*, where the largest increases in distances occur. This place is located in the upper region of the curve and can have a different size. So, these calculated points must be very precisely adjusted. This approach makes it possible to determine the right value of the *eps* parameter. The second parameter *MinPts* is also defined by the distances between the indicated points on the chart. The detailed description of the method for determining the *eps* and *MinPts* parameters is described in Section 3. This paper is organized as follows: In Section 2 related works about clustering algorithms are presented while Section 3 presents a short description of the *DBSCAN* and the new method for determining its parameters. Experimental results on datasets are illustrated in Section 4. Finally, Section 5 presents conclusions.

## 2 Related works

The DBSCAN density-based clustering algorithm is very popular and lots of algorithms are created on the basis of its modification and improvement, e.g. OPTICS [1], CLARANS [14], GMDBSCAN [35] or VDBSCAN [17]. It is worth noting that the problem of automatic choosing of input parameters of the DBSCAN algorithm is a great challenge. However, the methods used in order to determine these input parameters are only described in a few articles. For example, [15] proposes a hybrid DBSCAN algorithm combined with an optimization algorithm (Binary Differential Evolution) in order to choose the DBSCAN parameters. On the other hand, the method in [7] combines the grid partition technique and the DBSCAN algorithm. In article [28] is presented a combination of the Gaussian-Means and the DBSCAN to determine these input parameters. Then, [4] proposes the APSCAN which uses affinity propagation clustering to detect local densities and values of input parameters. Article [37] presents the I-DBSCAN algorithm to determine the *eps* and *MinPts*. The AGED algorithm [29] determines the *eps* of the DBSCAN based on local densities. Paper [16] proposes the Multi-verse optimizer algorithm which selects and improves optimizing of the DBSCAN parameters.

This study presents a new approach to automatic defining of the *eps* and *MinPts* parameters of the DBSCAN algorithm.

## 3 The new approach to determining the parameters of the DBSCAN

First, the description of the *DBSCAN* is presented, and next a new method for the determination of the input parameters is explained in detail.

### 3.1 A short description of the DBSCAN algorithm

Let us denote a dataset by $X$, where point $p \in X$, the *eps* parameter (a radius) is usually determined by the user and it has a large influence on the right creation of clusters by this algorithm. The next parameter, i.e. the $MinPts$ is the minimal number of neighboring points belonging to the so-called *core point*. The following definitions (see [6] and [8]) will be helpful in determining the DBSCAN parameters.

**Definition 1**: The *eps-neighborhood* of point $p \in X$ is called $N_{eps}(p)$ and is defined as follows $N_{eps}(p) = \{q \in X | dist(p,q) \leq eps\}$, where $dist(p,q)$ is a distance function between $p$ and $q$.

**Definition 2**: $p$ is called the *core* if the number of points belonging to $N_{eps}(p)$ is greater or equal to the $MinPts$.

**Definition 3**: Point $q$ is *directly density-reachable* from point $p$ (for the given *eps* and the $MinPts$) if $p$ is the *core point* and $q$ belongs to $N_{eps}(p)$.

**Definition 4**: if point $q$ is *directly density-reachable* from point $p$ and the number of points belonging to $N_{eps}(q)$ is smaller than the $MinPts$, $q$ is called a *border point*.

**Definition 5**: Point $q$ is a *noise* if it is neither a *core point* nor a *border point*.

**Definition 6**: Point $q$ is *density-reachable* from point $p$ (for the given *eps* and the $MinPts$) if there is a chain of points $q_1, q_2, ..., q_n$ and $q_1 = p$, $q_n = q$, so that $q_{i+1}$ is *directly density-reachable* from $q_i$

**Definition 7**: Point q is *density-connected* to point $p$ (for the given *eps* and the $MinPts$) if there is point $o$ such that $q$ and $p$ are *density-reachable* from point $o$.

**Definition 8**: Cluster $C$ (for the given *eps* and the $MinPts$) is a non-empty subset of X and the following conditions are satisfied: first, $\forall p,q$: if $p \in C$ and $q$ is *density-reachable* from p, then $q \in C$, next $\forall p,q \in C$: $p$ is *density-connected* to $q$.

The DBSCAN algorithm creates clusters according to the following: at first, point $p$ is selected randomly if $|N_{eps}(p)| \geq MinPts$, than point $p$ will be the *core point* and a new cluster will be created. Next, the new cluster is expanded by the points which are *density-reachable* from $p$. This process is repeated until no cluster is found. On the other hand, if $|N_{eps}(p)| < MinPts$, then point $p$ will be a *noise*, but this point can be included in another cluster if it is *density-reachable* from some *core point*.

### 3.2 Automatic determination of the eps parameter

As mentioned above, the *eps* parameter plays a fundamental role in creating the right clusters by the DBSCAN algorithm.

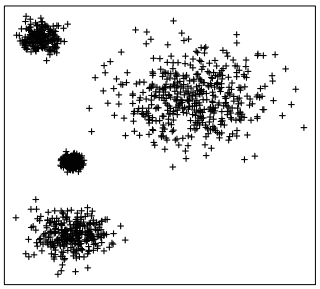


**Figure 1**. An example of a 2-dimensional dataset consisting of four clusters.

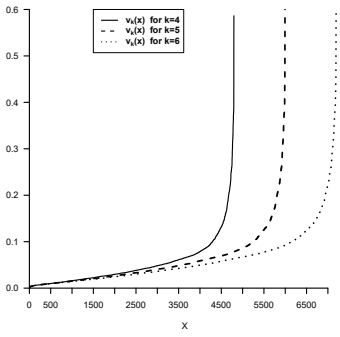



**Figure 2**. Sorted values of function $k_{dist}$ with respect to $k = 4$, $k = 5$ and $k = 6$ for a 2-dimensional dataset.

The most widely used method to calculate this parameter is based on a function which computes a distance between each element of a dataset and its $k$-th nearest neighbor. This function is often denoted by $k_{dist}$, and its $k$ parameter is equal to the $MinPts$. Figure 1 shows an example of a 2-dimensional dataset consisting of 1200 elements located in four clusters, i.e. 200, 250, 300 and 450 elements per cluster, respectively. For this dataset,

the $k_{dist}$ function is used. In order to better analyze the results generated by this function, three values of the $k$ parameter are used, i.e. $k = 4$, $k = 5$ and $k = 6$. Next, the distances are sorted in the ascending order and are presented in Figure 2. Sorted values of function $k_{dist}$ with respect to the $k$ parameter are denoted by $v_k(x)$. It can be observed that the number of calculated distances for $k = 6$ is much bigger than for $k = 4$ or $k = 4$. Moreover, there is a point range called the *knee* with a large change of distances. The fundamental issue is an appropriate determining of the *knee point*, which can be used to find out sharp changes of the distances and next to define the *eps* parameter of the DBSCAN algorithm. A sharp increase in the distances appears usually at the end of the *knee*. All elements of a dataset with higher distances than the value indicated by this *point* can be considered as noise. It is worth noting that when clusters of the dataset have a similar density there is only one *knee* for every value of parameter $k$ of the $k_{dist}$ function (see Figure 2). The *knee* is usually located at the end of the sorted distances and its size depends on the density of the clusters. As mentioned above, it is very difficult to determine the *knee point* correctly, because the width and slope of the *knee* can vary.

Let $V_{dist}$ denote a set of all distances generated by $k_{dist}$ function for a dataset. First, it is necessary to determine a *range* of points which precisely indicate the *knee*. Let us denote the beginning and end of the *knee* by $v_{start}$ and $v_{stop}$, respectively. The first parameter is defined as $v_{start} = |V_{dist}| - |X|$ and the other as $v_{stop} = |V_{dist}|$, where the $|V_{dist}|$ is the number of the elements of $V_{dist}$ and the $|X|$ is the size of dataset $X$. It can be noted that for parameter $k = 4$, $v_{start}$ equals $|V_{dist}| * 0.75$. The sorted distances of the $k_{dist}$ functions with $k = 4$ are presented in Figure 3 for the sample 2-dimensional dataset. It can be observed that there are $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$ points on the chart. $x_1$ and $x_2$ coordinates correspond to $v_{start}$ and $v_{stop}$, i.e. $x_1 = v_{start}$ and $x_2 = v_{stop}$ while $y_1$ and $y_2$ are equal to the values of the distances calculated by function $k_{dist}$. So, these two points simultaneously indicate the range of the *knee*. Next, line $A_1 * x + B_1$, which passes through points $p_1$ and $p_2$ is drawn. The $A_1$ and $B_1$ parameters are defined as follows

$$\begin{aligned} A_1 &= \frac{y_2 - y_1}{x_2 - x_1} \\ B_1 &= y_1 - A_1 * x_1. \end{aligned} \tag{1}$$

The line passing through points $p_1$ and $p_2$ is also presented in Figure 3.

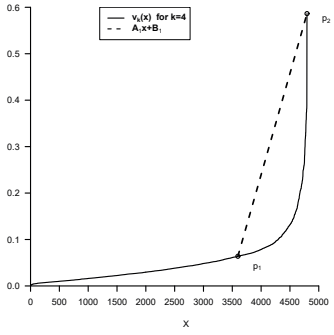



**Figure 3**. Sorted values of function $k_{dist}$ with respect to $k = 4$ and the line passing through points $p_1$ and $p_2$

Next, additional line $A_2 * x + B_2$ is created to find out the point corresponding to the abrupt increase of the distances. This line intersects halfway with line the $A_1 * x + B_1$ and its slope is equal to $-A_1$. Thus, parameters $A_2$ and $B_2$ of this line are expressed as follows

$$\begin{aligned} A_2 &= -A_1 \\ B_2 &= A_1 * (x_1 + x_2) + B_1, \end{aligned} \tag{2}$$

where $x_1$ and $x_2$ are the x-coordinates of points $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$, respectively. $A_2 * x + B_2$ line is presented in Figure 4. It can be observed that the line determines point $p_3(x_3, y_3)$ which is located in the upper part of the *knee*. There is a high probability there that $p_3(x_3, y_3)$ is located close to the *point* which can be used to calculate parameter *eps*.

In order to calculate this *point* more precisely, a new $A_3x + B_3$ line, tangent at point $p_3$ is drawn. So a temporary point $p_t(x_t, y_t)$ very closely located to point $p_3$ is indicated. Next, parameters $A_3$ and $B_3$ of the tangent line can be defined as follows

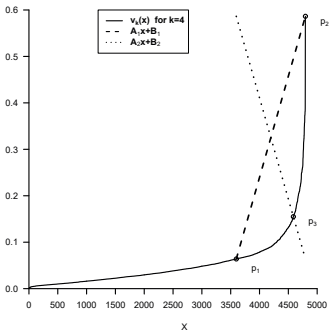



**Figure 4**. The straight line determining point $p_3$.

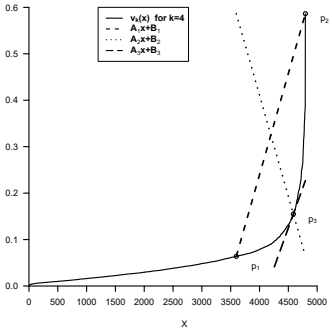



**Figure 5**. The tangent line at point $p_3$.

$$\begin{aligned} A_3 &= \frac{y_t - y_3}{x_t - x_3} \\ B_3 &= y_3 - A_3 * x_3. \end{aligned} \tag{3}$$

In Figure 5 is shown the tangent line at point $p_3$. Furthermore, difference $\Delta d(x)$ between the values of function $v_k(x)$ and the new line is determined for $x \in (x_3; x_2)$. The $x_3$ and $x_2$ values are x-coordinate of points $p_3(x_3, y_3)$ and $p_2(x_2, y_2)$, respectively. Thus, $\Delta d(x)$ can be defined as follows

$$\Delta d(x) = v_k(x) - (A_3 * x + B_3). \tag{4}$$

Let $M$ denote a set of all $\Delta d(x)$ increases calculated for $x \in (x_3; x_2)$. Next, the *average* value, i.e. the *arithmetic mean* from $M$ is calculated. In Figure 7 is presented point $p_a$ which corresponds to the *average* value from $M$. Thus, coordinate $y_a$ of the $p_a(x_a, y_a)$ point determines this *average* value, and the second coordinate $x_a$ indicates the number of the increase for $y_a$.

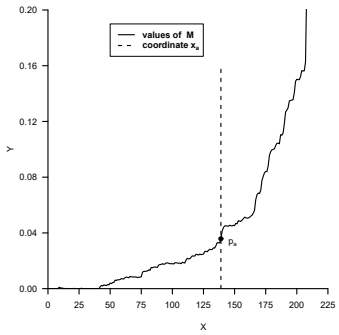



**Figure 6**. Values of $M$ and point $p_a$ which corresponds to the *average* value.

Based on coordinate $x_a$, the value of the *eps* parameter is expressed as follows

$$eps = v_k(x_3 + x_a), \tag{5}$$

where $x_3$ is the x-coordinate of point $p_3(x_3, y_3)$. As mentioned above the size of *knee* can be different and it depends on the density of clusters. If clusters have a similar density, the *knee* will be, e.g. as in Figure 5. Otherwise, the *knee* can be *wider* or the sorted distances can created several *knees*. This fact has an impact on the right value of the *eps* parameter. Consequently, an additional analysis of the *knee* properties is used. It is based on a comparison of the distances between points $p_1$, $p_2$ and $p_3$. It is defined as follows

$$d_p = \frac{d(p_2, p_3)}{d(p_1, p_3)}, \tag{6}$$

where $d(p_2, p_3)$ and $d(p_1, p_3)$ are the distances between points $p_2$, $p_3$ and $p_1$, $p_3$, respectively. In this approach a *bias* factor is experimentally selected and it equals 4. Such value of this factor makes it possible to find a considerable change of the distances, i.e. if the increase of the distances for $x \in (x_3; x_2)$ is significant, the value of factor $d_p$ will be greater than the value of the *bias*.

In this case, the value of the *eps* parameter is increased because there is a big change of the distances there. So, the modification of the *eps* parameter is expressed as below

$$eps = \begin{cases} v_k(x_3 + b) & for \quad d_p \geq bias \\ v_k(x_3 + x_a) & for \quad d_p < bias \end{cases}, \tag{7}$$

where

$$b = (x_a + x_n)/2. \quad (8)$$

$x_n$ is the number of elements of $M$. This proposed method allows for calculating the correct value of the $eps$ parameter for a different size of the *knee* based on the $v_k(x)$ function. It worth noting that parameter $A_1$ defines the slope of line $A_2 * x + B_2$ and it also determines the location of point $p_3$. Moreover, the start of the *knee* region is defined by $p_1$, where coordinate $x_1$ equals $v_{start}$.

### 3.3 Determination of the MinPts parameter

The $MinPts$ parameter is also very difficult to choose because it decides about the size of clusters and also affects the number of so-called noise data. Moreover, if the $MinPts$ has a high value, the number of clusters is small, but the size of the $V_{dist}$ collection can be quite large. On the other hand, when this parameter is too small, the clustering algorithm can create a lot of small clusters. Generally, the choice of this parameter is often realized individually depending on a dataset, but in many cases, the $MinPts$ equals 4 or 5. Such value of this parameter ensures a good compromise between the size of clusters and the amount of noise data in most datasets. However, this paper proposes a new approach to the selection of this parameter. This method uses the $d_p$ factor to calculate $MinPts$ and is expressed as follows

$$MinPts = \begin{cases} round(d_p + 0.5) & for\ dim(X) == 2 \\ round(d_p - 0.5) & for\ dim(X) > 2 \end{cases}, \quad (9)$$

where the $dim(X)$ function defines the dimensions of dataset $X$. If $dim(X)$ equals 2, the value of $d_p$ is rounded up, and otherwise, it is rounded down. The key issue is the calculation of factor $d_p$, so first, the $k_{dist}$ function must compute the distances of the dataset. In the case of calculating the $MinPts$ parameter, $k$ equals 2. Thus, for this value of parameter $k$ of the $k_{dist}$ function, factor $d_p$ is determined and the $MinPts$ parameter is estimated by formula 9. Next, the $eps$ parameter can be defined for the calculated value of $MinPts$ (see Section 3.2). In the next Section, the results of the experimental studies are presented to confirm the effectiveness of this new approach.

## 4 Experimental results

In this Section, several experiments have been conducted on 2-dimensional and 3-dimensional artificial datasets using the *DBSCAN* algorithm.

**Table 1**. A detailed description of the 2-dimensional artificial datasets

| Datasets | No. of elements | Clusters |
|---|---|---|
| *Data* 1 | 1050 | 3 |
| *Data* 2 | 700 | 6 |
| *Data* 3 | 700 | 3 |
| *Data* 4 | 900 | 4 |
| *Data* 5 | 500 | 4 |
| *Data* 6 | 700 | 2 |

**Table 2**. A detailed description of the 3-dimensional artificial datasets

| Datasets | No. of elements | Clusters |
|---|---|---|
| *Data* 1 | 900 | 3 |
| *Data* 2 | 1100 | 4 |
| *Data* 3 | 1300 | 5 |
| *Data* 4 | 1800 | 7 |

**Table 3**. The $eps$ and $MinPts$ values of the DBSCAN algorithm used in the artificial datasets

| Datasets | $eps$ | $MinPts$ |
|---|---|---|
| *Data* 1 | 0.36 | 7 |
| *Data* 2 | 0.23 | 4 |
| *Data* 3 | 0.21 | 4 |
| *Data* 4 | 0.18 | 5 |
| *Data* 5 | 0.22 | 6 |
| *Data* 6 | 0.27 | 7 |
| *Data* 7 | 0.55 | 4 |
| *Data* 8 | 0.48 | 6 |
| *Data* 9 | 0.42 | 4 |
| *Data* 10 | 0.49 | 4 |

The new approach to the automatic determination of this algorithm parameters is used. In Table 3, there are the $eps$ and $MinPts$ parameters of the DBSCAN algorithm used to cluster these datasets. It is worth noting that the artificial datasets include clusters of various sizes and shapes. Moreover, in all the

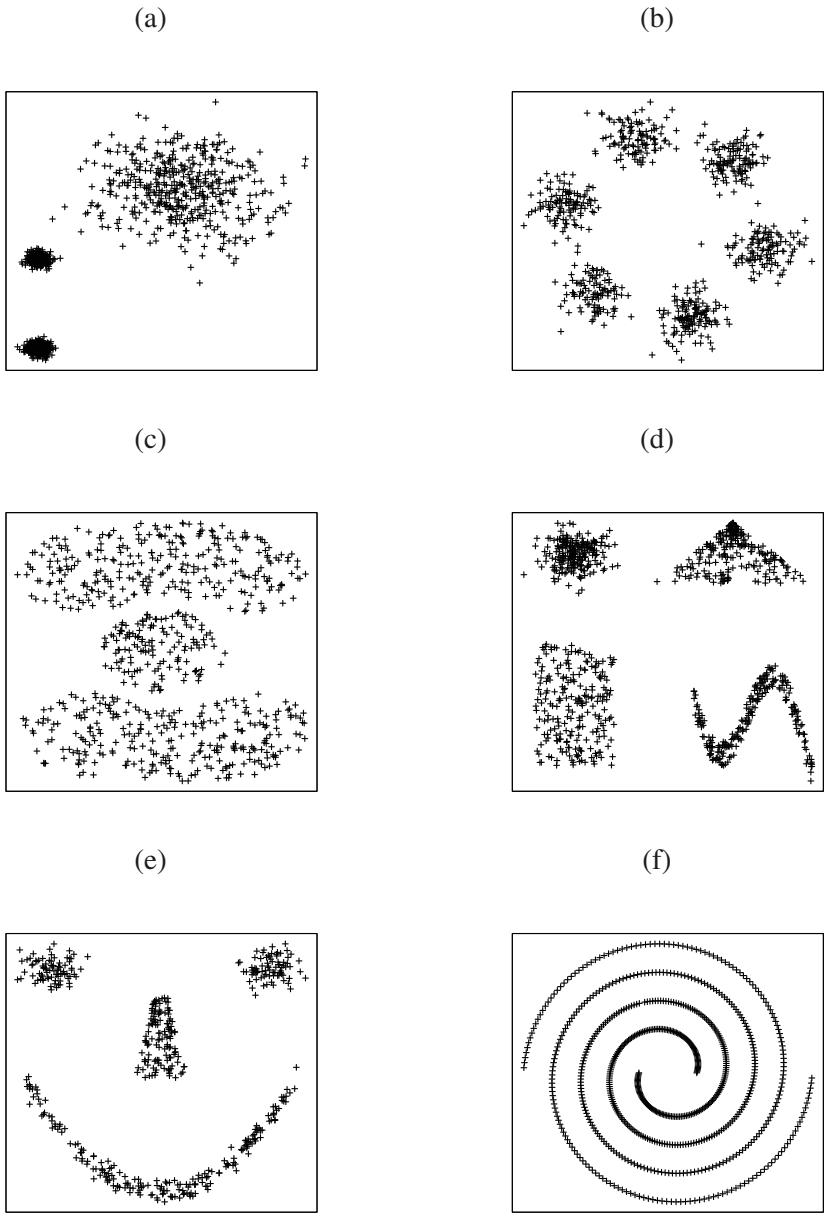



**Figure 7**. Examples of 2-dimensional artificial datasets: (a) *Data* 1, (b) *Data* 2, (c) *Data* 3, (d) *Data* 4, (e) *Data* 5, and (f) *Data* 6.

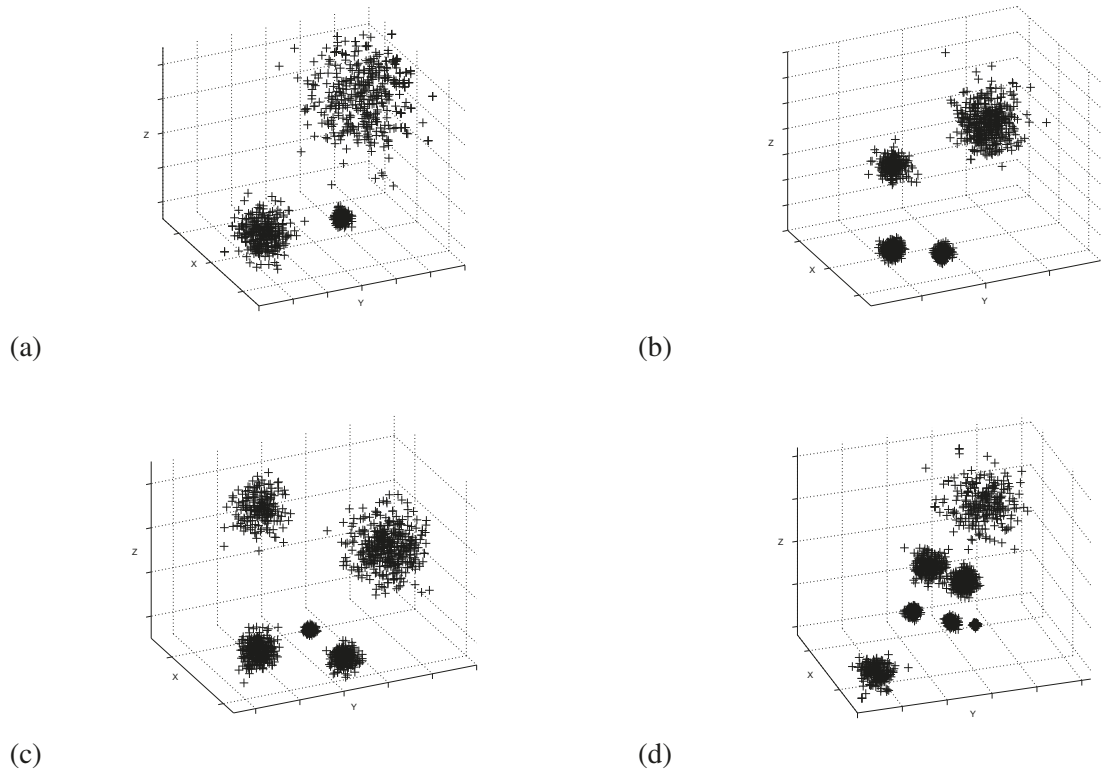



**Figure 8**. Examples of 3-dimensional artificial datasets: (a) *Data* 7, (b) *Data* 8, (c) *Data* 9, and (d) *Data* 10.

conducted experiments, the evaluation of the accuracy of clusters generated by the DBSCAN algorithm is realized by visual inspection. The original DBSCAN is difficult to use for multidimensional data, but new modifications of the DBSCAN algorithm have been also proposed to solve this problem, e.g [25].

## 4.1 Datasets

In the conducted experiments six 2-dimensional and four 3-dimensional datasets are used. Several data come from the *R* package and the other are generated by functions of the *Scliab* environment. The new approach to the automatic determination of this algorithm parameters is used. The artificial datasets include clusters of various sizes and shapes. The artificial data are called *Data* 1, *Data* 2, *Data* 3, *Data* 4, *Data* 5 and *Data* 6 for 2-dimensional datasets and *Data* 7, *Data* 8, *Data* 9 and *Data* 10 for 3-dimensional datasets. These datasets consist of various number of clusters, i.e. from 2 to 7 clusters. The scatter plot of these data is presented in Figures 7 and 8. It can be observed in the figures that the distances between the clusters are very different and some clusters are quite close. For instance, in *Data* 4 the elements create the Gaussian, square, triangle and wave shapes, *Data* 5 consists of 2 Gaussian eyes, a trapezoid nose and a parabola mouth, and *Data* 6 is the so-called spirals problem, where points are on two entangled spirals. Moreover, the sizes of the clusters are different and they contain a different number of elements. Tables 1 and 2 show a detailed description of these datasets.

## 4.2 Experiments

The 2-dimensional and 3-dimensional artificial datasets are used to evaluate the performance of the newly proposed method defining the parameters of the DBSCAN algorithm. At first, in these experiments, the $MinPts$ parameter is determined according to formula 9. Next, parameter $k$ of the $k_{dist}$ function equals $MinPts$ and the steps described in Section 3.2 are made in order to determine the correct $eps$ parameter. When the $eps$ and $MinPts$ parameters are identified, the DBSCAN algorithm is used to cluster artificial datasets. Moreover, a visual inspection of the results is made to evaluate this new method, i.e. Figures 9 and 10 show the data clus-

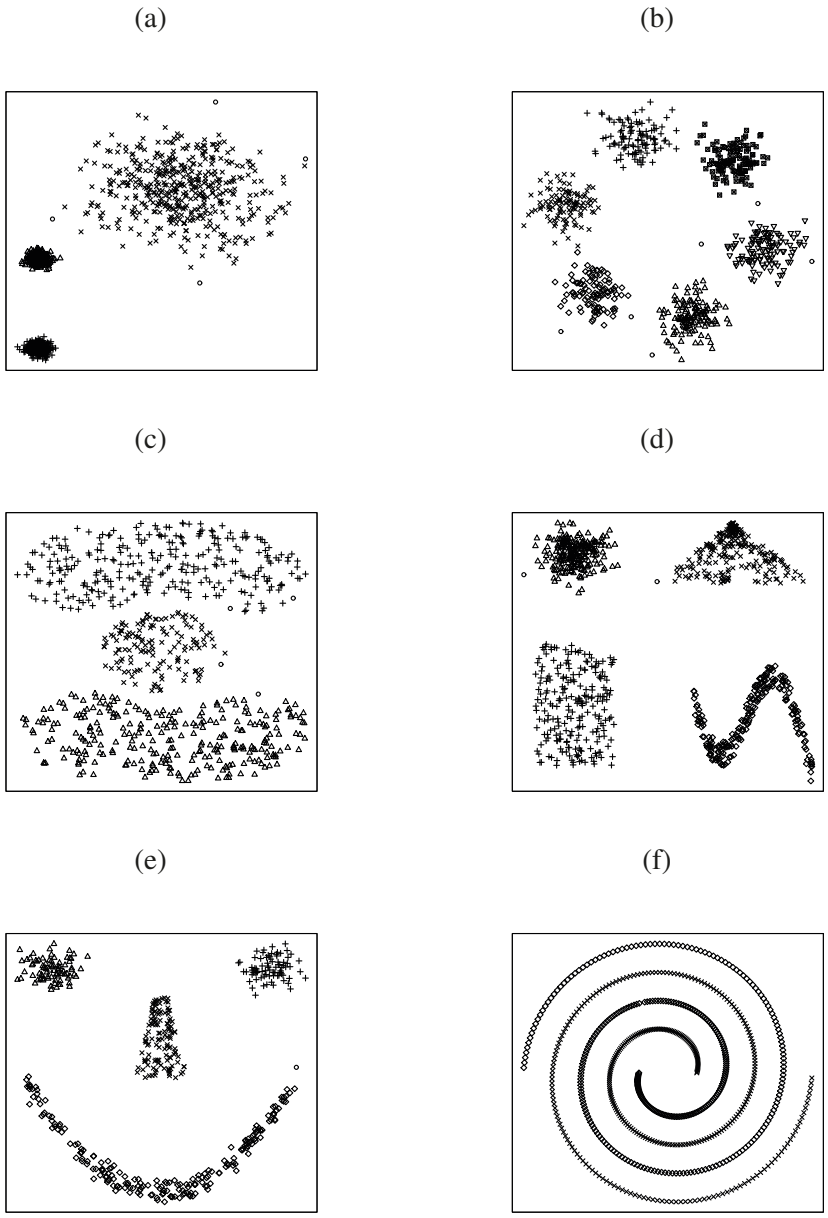
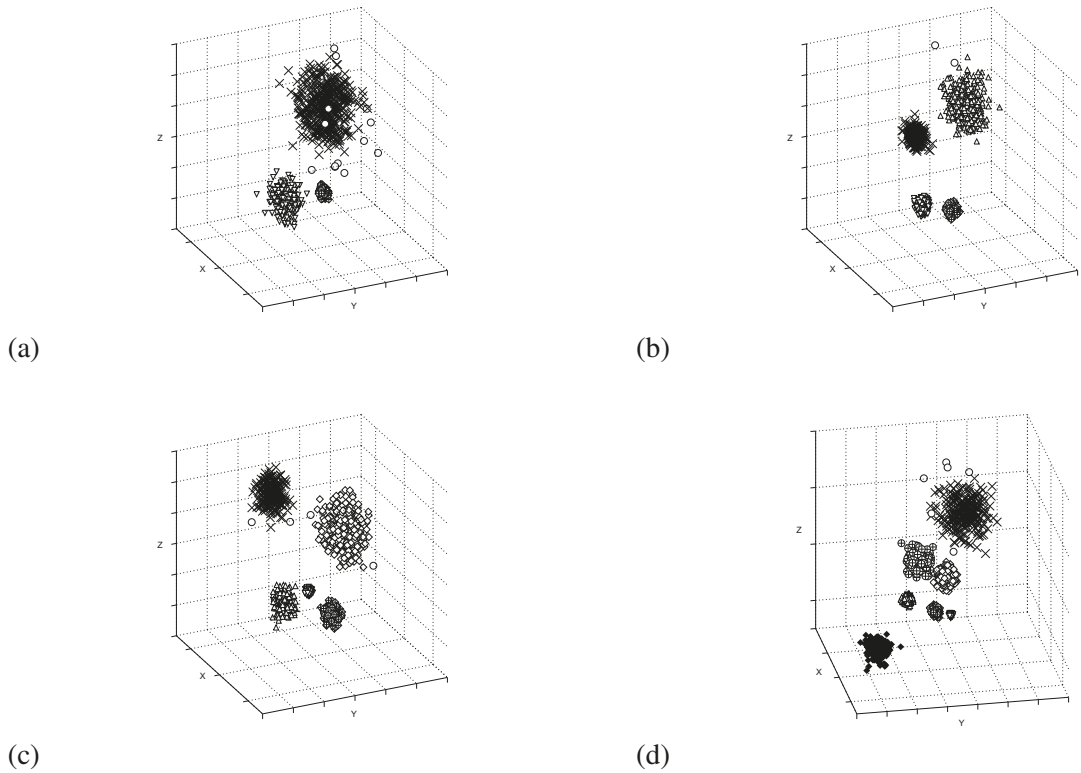



**Figure 9**. Results of the *DBSCAN* clustering algorithm for 2-dimensional datasets: (a) *Data* 1, (b) *Data* 2, (c) *Data* 3, (d) *Data* 4, (e) *Data* 5, and (f) *Data* 6




(a) (b)

(c) (d)

**Figure 10**. Results of the *DBSCAN* clustering algorithm for 3-dimensional datasets: (a) *Data* 7, (b) *Data* 8, (c) *Data* 9, and (d) *Data* 10

tered by the *DBSCAN* algorithm. It can be observed that each cluster is signed with different symbols, but the *noise* data is always presented as a circle. Even though the differences of distances and shapes between the clusters are significant, the elements of the datasets are correctly classified by the *DBSCAN*. Moreover, the number of data elements classified as noise in all the datasets is small.

## 5 Conclusion

In this paper, a new approach is proposed to calculate the *eps* and *MinPts* parameters of the DBSCAN algorithm. It is based on the $k_{dist}$ function calculating distances between points of a dataset and their *k*th nearest neighbors. As mentioned above, the determination of the *MinPts* parameter is very difficult, so it is often chosen empirically depending on the datasets being investigated. In the method presented, the size of the *knee* is studied to correctly calculate this parameter, and so the value of the *MinPts* parameter is defined by Equation 9. In the case of parameter *eps*, the fundamental issue is to correctly determine the sharp increases of the distances, so at first, the *knee* must be precisely specified in the sorted distances. Next, it is defined that the *point* which corresponds to sharp increases in the distances. Based on this *point* and on the size of *knee* the correct value of parameter *eps* is calculated. In the conducted experiments, several 2-dimensional and 3-dimensional datasets were used. There were a number of clusters, sizes and shapes varied within a wide range there. From the perspective of the conducted experiments, this automatic way to compute the *eps* and the *MinPts* parameters is very useful. All the presented results confirm very a high efficiency of the newly proposed approach.

## Acknowledgements

The paper is financed under the program of the Polish Minister of Science and Higher Education under the name "Regional Initiative of Excellence" in the years 2019-2022; project number 020/RID/2018/19; the amount of financing PLN 12,000,000.00.

## References

[1] Ankerst M., Breunig M, Kriegel H.P, Sandler J.: OPTICS: Ordering Points to Identify the Clustering Structure. Proceedings of the Int. Conf. on Management of Data, pp.49-60, (1999).

[2] Babu G.P., Murty M.N.: Simulated annealing for selecting optimal initial seeds in the k-means algorithm. Indian Journal of Pure and Applied Mathematics, Vol 25, pp.85-94 (1994).

[3] Bradley P., Fayyad U.: Refining initial points for k-means clustering. In Proceedings of the fifteenth international conference on knowledge discovery and data mining, New York, AAAI Press, pp. 9-15 (1998).

[4] Chen X., Liu W., Qui H, Lai J: APSCAN: A parameter free algorithm for clustering. Pattern Recognition Letters, Vol. 32, pp.973-986 (2011).

[5] Chen J.: Hybrid clustering algorithm based on pso with the multidimensional asynchronism and stochastic disturbance method. Journal of Theoretical and Applied Information Technology, Vol.46, pp.434-440 (2012).

[6] Chen Y., Tang S., Bouguila N., Wang C., Du J., Li H.: A Fast Clustering Algorithm based on pruning unnecessary distance computations in DBSCAN for High-Dimensional Data. Pattern Recognition Vol.83, pp.375-387 (2018)

[7] Darong H., Peng W.: Grid-based dbscan algorithm with referential parameters. Physics Procedia, Vol.24, Part B, pp.1166-1170 (2012).

[8] Ester M., Kriegel H.P, Sander J., Xu X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceeding of 2nd International Conference on Knowledge Discovery and Data Mining, pp.226-231 (1996).

[9] Fränti P., Rezaei M., Zhao Q.: Centroid index: Cluster level similarity measure. Pattern Recognition, Vol.47, Issue 9, pp.3034-3045 (2014).

[10] Gabryel M.: The Bag-of-Words Method with Different Types of Image Features and Dictionary Analysis. Journal of Universal Computer Science 24(4), pp.357-371 (2018).

[11] Gabryel M.: Data Analysis Algorithm for Click Fraud Recognition. Communications in Computer and Information Science, Vol.920, pp.437-446 (2018).

[12] Gabryel M., Damaševičius R., Przybyszewski K.: Application of the Bag-of-Words Algorithm in Classification the Quality of Sales Leads. Lecture Notes in Computer Science, Vol. 10841, pp.615-622 (2018).

[13] Hruschka E.R., de Castro L.N., Campello R.J.: Evolutionary algorithms for clustering gene-expression data, In: Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on Data Mining, pp.403-406, IEEE (2004).

[14] Jain A.K., Murty M.N, Flynn P.J: Data Clustering: A Review. ACM Computing Surveys, Vol.31, No.3, pp.264-323 (1999).

[15] Karami A., Johansson R.: Choosing DBSCAN Parameters Automatically using Differential Evolution. International Journal of Computer Applications, Vol.91, pp.1-11 (2014).

[16] Lai W., Zhou M., Hu F., Bian K., Song Q.: A New DBSCAN Parameters Determination Method Based on Improved MVO. IEEE Access, Vol.7 (2019).

[17] Liu Z., Zhou D., Wu N.: Varied Density Based Spatial Clustering of Application with Noise. In proceedings of IEEE Conference ICSSSM, pp.528-531 (2007).

[18] Luchi D., Rodrigues A.L., Varejao F.M.: Sampling approaches for applying DBSCAN to large datasets. Pattern Recognition Letters, Vol.117, pp.90-96 (2019).

[19] Murtagh F.: A survey of recent advances in hierarchical clustering algorithms. Computer Journal, Vol.26, Issue 4, pp.354-359 (1983).

[20] Patrikainen A., Meila M.: Comparing Subspace Clusterings. IEEE Transactions on Knowledge and Data Engineering, Vol.18, Issue 7, pp.902-916 (2006).

[21] Pei Z., Xia Hua X., Han J.. The clustering algorithm based on particle swarm optimization algorithm. In Proceedings of the 2008 International Conference on Intelligent Computation Technology and Automation, Washington, USA. Vol.1, pp.148-151, (2008).

[22] Rohlf F.: Single-link clustering algorithms. In: P.R Krishnaiah and L.N. Kanal (Eds.), Handbook of Statistics, Vol.2, pp.267-284 (1982).

[23] Sameh A.S., Asoke K.N.: Development of assessment criteria for clustering algorithms. Pattern Analysis and Applications, Vol.12, Issue 1, pp.79-98 (2009).

[24] Serdah AM., Ashour WM.: Clustering Large-scale Data Based on Modified Affinity Propagation Algorithm. Journal of Artificial Intelligence and Soft Computing Research, Volume 6, Issue 1, pp.23-33, DOI:10.1515/jaiscr-2016-0003 (2016)

[25] Shah G.H.: An improved dbscan, a density based clustering algorithm with parameter selection for high dimensional data sets. In Nirma University International Engineering,(NUiCONE), pp.1-6 (2012).

[26] Sheikholeslam G., Chatterjee S., Zhang A.: WaveCluster: a wavelet-based clustering approach for spatial data in very large databases. The International Journal on Very Large Data Bases, Vol.8 Issue 3-4, pp.289-304 (2000).

[27] Shieh H-L.: Robust validity index for a modified subtractive clustering algorithm. Applied Soft Computing, Vol.22, pp.47-59 (2014).

[28] Smiti A., Elouedi Z.: Dbscan-gm: An improved clustering method based on gaussian means and dbscan techniques. In 16th International Conference on Intelligent Engineering Systems (INES), pp. 573-578, (2012).

[29] Soni N., Ganatra A.: AGED (Automatic Generation of Eps for DBSCAN. Int. J. of Computer Science and Information Security, Vol.14, No.5, pp.536-559, (2016).

[30] Starczewski A.: A new validity index for crisp clusters. Pattern Analysis and Applications, Vol.20, Issue 3, pp.687-700 (2017).

[31] Starczewski A., Krzyżak A.: A Modification of the Silhouette Index for the Improvement of Cluster Validity Assessment. Lecture Notes in Computer Science, Vol.9693, pp.114-124 (2016).

[32] Tsekouras G.E: A simple and effective algorithm for implementing particle swarm optimization in rbf networks design using input-output fuzzy clustering. Neurocomputing, Vol.108, pp.36-44, (2013).

[33] Viswanath P., Suresh Babu V.S.: Rough-dbscan: A fast hybrid density based clustering method for large data sets. Pattern Recognition Letters, Vol.30 Issue 16, pp.1477-1488 (2009).

[34] Wang W., Yang J., Muntz R.: STING: A Statistical Information Grid Approach to Spatial Data Mining. VLDB '97 Proceedings of the 23rd International Conference on Very Large Data Bases, pp.186-195 (1997).

[35] Xue-yong L., Guo-hong G., Jia-xia S.: A new intrusion detection method based on improved dbscan. In International Conference on Information Engineering (ICIE), Vol.2, pp.117-120 (2010).

[36] Zalik K.R.: An efficient k-means clustering algorithm. Pattern Recognition Letters, Vol.29, Issue 9, pp.1385-1391 (2008).

[37] Zhou H., Wang P., Li H.: Research on adaptive parameters determination in DBSCAN algorithm. J. of Information and Computational Science, Vol.9, No.7, pp.1967-1973 (2012).

**Artur Starczewski** received the M.Sc. degree in electrical engineering from Częstochowa University of Technology, Poland. In 2000, he received his Ph.D. degree in computer science from the AGH University of Science and Technology, Cracow, Poland. He is an Assistant Professor in the Department of Computer Engineering, Częstochowa University of Technology. His research interests include data clustering, data mining, and pattern recognition. He has authored many research papers on fuzzy systems and clustering algorithms.

**Piotr Goetzen** received the Ph.D. in computer chemistry from Université de Neuchâtel, Switzerland. Since his graduation he has been interested in computer science, especially computer networks, operating systems and security of IT systems. Dr. Goetzen leads the Department of Computer Networks at University of Social Sciences, Highly certified (CCNA, CCNP, CCDA, CCDP, ITIL, Microsoft, Linux) Dr Goetzen has been the IT Trainer for more than 20 years. He also works in a security department of one of the global IT Corporations. He is pursuing the research of security of IT systems. He has also been involved in several international projects. Active Erasmus teacher.

Professor **Er Meng Joo** is currently a Full Professor in the School of Marine Electrical Engineering, Dalian Maritime University, China. He has authored five books entitled "Dynamic Fuzzy Neural Networks: Architectures, Algorithms and Applications" and "Engineering Mathematics with Real-World Applications" published by McGraw Hill in 2003 and 2005 respectively, and "Theory and Novel Applications of Machine Learning" published by In-Tech in 2009, "New Trends in Technology: Control, Management, Computational Intelligence and Network Systems" and "New Trends in Technology: Devices, Computer, Communication and Industrial Systems", both published by SCI-YO, 18 book chapters and more than 500 refereed journal and conference papers in his research areas of interest.

Professor Er was bestowed the Web of Science Top 1 % Best Cited Paper and the Elsevier Top 20 Best Cited Paper Award in 2007 and 2008 respectively. In recognition of the significant and impactful contributions to Singapore's development by his research projects, Professor Er won the Institution of Engineers, Singapore (IES) Prestigious Engineering Achievement Award twice (2011 and 2015). He is also the only dual winner in Singapore IES Prestigious Publication Award in Application (1996) and IES Prestigious Publication Award in Theory (2001). Recently, he was bestowed the Amity Researcher Award 2018 for his outstanding and significant contributions in Robotics and Automation.