Hybridization of K Nearest Neighbors Classifier with Cuckoo Search Algorithm

Katarzyna Prokop

Faculty of Applied Mathematics, Silesian University of Technology, Kaszubska 23, 44100 Gliwice, POLAND

Abstract

Artificial intelligence methods are one of the most used algorithms in big data and Internet of Things solutions. Therefore, a very important aspect is to create new algorithms and improve existing ones. In this paper, the proposition of a hybrid method for classifying elements in certain datasets is presented. The proposed method joins properties of K Nearest Neighbors Algorithm (a classifier) and Cuckoo Search Algorithm (a heuristic algorithm). The proposed model was presented, tested, and discussed on a selected dataset of Iris Flowers. The effectiveness of the method is compared for different variants of input parameters to show the efficiency of the proposition.

Keywords

data classification, knn, heuristic, clustering

1. Introduction

In everyday life, people are constantly faced with the need to make choices. Depending on the method of selection, the decisions turn out to be more or less accurate in relation to a given problem. It is often justified to use certain mathematical techniques in the decision-making process for example to assess the probability that a choice will bring the expected result [1, 2, 3].

The main intention of the work is to propose a hybrid method for classifying an object into one of the existing groups from a specific data set. Such grouping of data is related to the concept of clustering, which is the ordering of database records into collections based on established guidelines [4].

The idea of clustering implies the separation of elements into clusters with similar characteristics. Simultaneously objects belonging to one group should be less similar to elements from other clusters. This process is unsupervised, based on measurements of the value of the similarity metric between items without making any other assumptions. However, the clustering process does not guarantee the ordering of the data while maintaining the actual relations between the elements because there are many possible divisions of the data set.

The aim of the hybrid classification method described in the paper is to assign an object to the appropriate group using a heuristic algorithm. The term "heuristic" is derived from the Greek word *heuresis* which means invention and from the word *heureka* which stands for "I have found". In ancient times heuristics were concerned with solving problems based on formulating appropriate hypotheses. They are defined as methods of searching solutions that have no guarantee of finding the optimal solution [5, 6]. Heuristics are used when conventional methods are too costly, e.g. due to the computational complexity or when the appropriate algorithm is not fully known. Therefore, it is possible that heuristics would solve the problem incorrectly. In the last years, heuristic algorithms were improved by applied parallel computing to decrease time complexities [7, 8] and in many applications like classifications [9, 10, 11] and clustering[12, 13, 14].

The effectiveness of the proposed method has been tested for different sets of input parameters.

2. Cuckoo Search Algorithm

Cuckoo Search Algorithm [15] is an example of a stochastic optimization method – in other words an optimization process whose essential element is the randomization of certain parameters [16]. Owing to the fact of using random values the solution is quickly obtained. Cuckoo Search Algorithm aim is to find an answer for a given problem by imitating the behavior of some species of cuckoos.

The cuckoo is a medium-sized migratory bird in the cuckoo *Cuculidae* family. It is representative of the nesting parasites which means animals that use the parents of other species to take care of their offspring. Cuckoos drop their eggs into foreign nests, where their progeny grows up [17]. Over the generations cuckoos have learned to adapt to their surroundings, making themselves similar to other bird species to increase the chance to survive in a foreign nest. This ability is called mimicry [17], which can be also defined by masking. Cuckoos become similar to the host species even before they hatch. The host is often unable to distinguish the cuckoos from his offspring

ICYRIME 2021 @ International Conference of Yearly Reports on Informatics Mathematics and Engineering, online, July 9, 2021

ktn.prokop@gmail.com (K. Prokop)

due to the mimicry, which results in him raising these individuals like his own hatchlings. However, when he discovers the intruder, throws it out of the nest without hesitation.

Every cuckoo in the algorithm is interpreted as a point in n-dimensional space. The individual is evaluated regarding the value of the fitness function based on taken coordinates. Thanks to that, the best cuckoos solving a given problem can be selected. All birds are in the nest, which is the set of the currently best-adapted points. Additionally, the movement of cuckoos takes place. It is set by a given equation of displacement. Another important element of the algorithm is the so-called "host decision", when the cuckoo can be thrown away from the nest and replaced with another, newly hatched bird. This operation depends on the comparison of the values of the fitness function for both individuals and the assumed probability detection of an intruder in the nest.

The mathematical model of the cuckoo search algorithm can be presented in the steps described below.

The algorithm starts by generating an initial cuckoo population. It is assumed that the cuckoo is the point of n coordinates (depending on the size of the problem, i.e. the number of variables of the fitness function $f(\cdot)$):

$$x = (x_1, ..., x_n), \tag{1}$$

where $x_1, ..., x_n$a re chosen e.g. randomly from a given interval. Then each individual is displaced using Lévy's movement. The flight of the cuckoos is modeled as the following function:

$$l(x_i; \mu, c) = \sqrt{\frac{c}{2\pi}} \frac{e^{-\frac{c}{2(x_i-\mu)}}}{(x_i-\mu)^{1.5}}, \tag{2}$$

where x_i is the appropriate coordinate and μ and c are the initially determined coefficients. Therefore, the Cuckoo Search Algorithm will depend on these parameters. In addition, the value of displacement depends on the fixed step $\alpha > 0$. Then the value $\alpha \cdot l(x; \mu; c)$ be added to every coordinate to move the cuckoo. In nature, many animal species use this type of movement strategy when they do not remember or have no information about where the food can be found.

Another operation that goes into the algorithm is "host decision". Detection of intruder birds takes place with a certain fixed probability $p \in (0, 1)$. Removal of a bird takes place if and only if the individual y which is compared with individual x is better suited to the fitness function. The decision process can be written as the following function $g(\cdot, \cdot)$:

$$g(x, y) = \begin{cases} 0, & \text{for } \lambda \leq p \text{ and } f(y) < f(x) \\ 1, & \text{for } \lambda \leq p \text{ and } f(y) > f(x) \text{ or for } \lambda > p \end{cases} \tag{3}$$

where 0 means throwing away the intruder from the nest, 1 saving the cuckoo, λ is a random value in the range (0, 1), generated separately for each individual and the minimum of the function $f(\cdot)$ is searched. If the function $f(\cdot)$ were to be maximised, the function $g(\cdot, \cdot)$would take the form:

$$g(x, y) = \begin{cases} 0, & \text{for } \lambda \leq p \text{ and } f(y) > f(x) \\ 1, & \text{for } \lambda \leq p \text{ and } f(y) < f(x) \text{ or for } \lambda > p \end{cases}. \tag{4}$$

When the final set of the best solutions is known and all the actions characteristic of the algorithm were performed a certain number of times (iteratively), the last step is to find the best individual. This is done by comparing the values of the fitness function for all the cuckoos in the nest.

The structure of the algorithm is presented in pseudocode 1.

Algorithm 1 Cuckoo Search Algorithm.

```
Input: cuckoos number k, iterations number t,
       range of arguments, c, μ, α
Output: the best cuckoo x
Create an initial cuckoo generation of k
  individuals;
i := 0;
for i < t do
    j := 0;
    for cuckoos do
        Displace the cuckoo according to the
          equation (2);
        Decide to throw or save the cuckoo
          according to the function (3);
        j++;
    end
    i++;
end
Find the best cuckoo x;
return x;
```

3. K Nearest Neighbors Algorithm

The K Nearest Neighbors Algorithm is a simple classifier, which is based on finding k elements in a given database as similar as possible to the tested element, i.e. its so-called neighbors [18]. Then it is assumed that the neighbors by "voice of the majority" will settle to which group the new record will belong. The result of the classification is the group that contains the largest number of neighbors. The similarity of records is determined by measuring the distance between individual elements, which can be treated as vectors. For this measurement, the Euclidean or the Manhattan metric can be used.

Suppose that a and b are records with n characteristics, between which the distance is measured. The elements of the dataset are interpreted as vectors. Then the Euclidean distance function can be defined as:

$$d(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}, \tag{5}$$

and the Manhattan distance function can be described by the formula:

$$d(a,b) = \sum_{i=1}^{n}|a_i - b_i|. \tag{6}$$

These functions satisfy the condition of the identity of indiscernibles, symmetry, and triangle inequality thus represent metrics.

The K Nearest Neighbors Algorithm is an example of a lazy classification method [18]. It does not gather any information about a given problem. A solution is selected in real-time, directly after passing along a vector to classification.

Let $x^i = (x_1^i, ..., x_n^i)$ be the i^{th} record with n features for $i = 1, ..., m$ where m is the number of all vectors in the dataset. It is clear that $m \geq k$. An element given to be classified is described by $y = (y_1, ..., y_n)$. The distance measure d was assumed. Then the K Nearest Neighbors Algorithm may be represented by the pseudocode 2.

Algorithm 2 K Nearest Neighbors Algorithm.

Input: dataset with m vectors x^i, unclassified vector y, neighbors number k
Output: y group

```
i := 1;
for i ≤ m do
    Calculate the distance from y to x^i using measure d;
    i++;
end
Sort the records in the database in ascending order relative to the calculated distances;
j := 1;
for j ≤ k do
    Make a note of the assigned group for x^j;
    j++;
end
Select the most frequent group;
return y group;
```

Depending on the established distance measure and the number of neighbors, it is possible to obtain different classification results. Testing various variants for a given database allows us to determine the configuration that will assign the class with the highest probability of correctness.

4. Hybrid classification method

Heuristic algorithms can be used in the process of data clustering. The idea of this concept is based on combining the classic classifier with the heuristic. This paper presents a hybrid classification method that combines the K Nearest Neighbors Algorithm with the Cuckoo Search Algorithm.

Let $y = y_1, ..., y_n$ be a n-dimensional vector of unknown group. The task of the method is to assign a vector to the appropriate cluster. The idea is that cuckoos, which are points in $n$dimensional space, can be identified with records of the database with n features. Then the initial population is generated by assigning one specific element from the dataset to a single cuckoo. A population is therefore a set of a size equal to the size m of the considered database, and the cuckoo coordinates will reflect the values of records features. As a consequence the i^{th} cuckoo storing values from the i^{th} record where $i = 1, ..., m$ can be represented as

$$x^i = (x_1^i, ..., x_n^i). \tag{7}$$

The cuckoos are displaced by Levy's movement according to the formula (2). This is followed by a decision process called "host decision", described as a function (3). In this case, the f fitness function is the distance function (Euclidean, described by the formula (5) or Manhattan, described by the function (6)). Its role is to measure the distance between the cuckoo and the considered vector y The result of the "host decision" is to save the best specimens in the nest, i.e. records with the shortest distance from y. Such refinement of the population occurs a certain number of times iteratively. The next step is to find the best cuckoo in a final population.

The above operations are performed exactly k times to get k of the best individuals, one from each population. These cuckoos represent the k nearest neighbors of the K Nearest Neighbors Algorithm. The last step is to select an appropriate cluster for the y vector based on information about groups stored by neighbors. It is decided by the "majority vote", which means that the vector y is assigned to the most frequently appearing cluster.

The concept of using the Cuckoo Search Algorithm in classification is presented by the pseudocode 3.

5. Experiments

The hybrid classification method was tested in the decision-making process, the purpose of which was to

Algorithm 3 Application of the cuckoo algorithm in classification.

Input: database of m records x^i, unclassified vector y, number of nearest neighbors k, number of iterations t, c, μ, α
Output: y group

```
Create the initial cuckoo generation of m records;
Create a table classes of length k;
h := 0;
for h < k do
    i := 0;
    for i < t do
        j := 0;
        for cuckoos do
            Displace a cuckoo according to the equation (2);
            Decide to throw or save the cuckoo according to the function (3);
            j++;
        end
        i++;
    end
    Find the best cuckoo x;
    Assign to the classes[h] a group of the cuckoo x;
    h++;
end
Select the most frequently repeated class in the table classes;
return y group;
```

match the corresponding class to the object from the database. A dataset of iris flowers was used for this task. These data were shared in 1936 by the British biologist and statistician Ronald Fisher [19]. The set contains 150 records, each of which consists of 5 attributes that determine the length of the plot of the flower cup, the width of the plot, the length of the petal and its width, as well as the name of the species. Therefore, the first four attributes are expressed by numerical values, while the species name is presented in words. The collection contains data on three species of irises: *Iris setosa, Iris virginica, Iris versicolor.*

To test the method, the program was created. Its task was to randomly select 30 records to be checked, which is 20% of all objects in the database. For testing, these objects were stripped of the fifth attribute (species name), which was remembered by the program to later determine whether the match using the hybrid classification method was successful. The test set was subjected to the method so that each record was matched with a class (species of the iris). On the output, the program returns information about the obtained acc correctness of the method, according to the following formula:

$$acc = \frac{n_c}{n_t} \cdot 100\%, \tag{8}$$

where n_c is the number of correct matches and n_t is the number of all tested items. The final results of the obtained correctness have been rounded to two decimal places.

Table 1
Arithmetic mean of the acc coefficient for $t = 100, \alpha = 0.5, c = 0.2, \mu = 0.2$.

k	Euclidean distance	Manhattan distance
1	100.00%	86.00%
2	100.00%	74.00%
3	100.00%	82.67%
4	100.00%	89.33%
5	100.00%	86.00%
6	100.00%	83.33%
7	100.00%	85.33%
8	100.00%	77.33%
9	100.00%	80.67%
10	100.00%	83.33%
11	100.00%	86.67%
12	100.00%	90.00%
13	100.00%	90.00%
14	100.00%	84.00%
15	100.00%	85.33%

The experiments were performed for eight different sets of input parameters t, α, c, μ. Depending on the used distance function (the tested variants are the Euclidean distance (5) and the Manhattan distance (6)) and the number of k nearest neighbors of the K Nearest Neighbors Algorithm, the effectiveness was compared. The program was launched five times for each of the integers k in the range [1, 15] using the Euclidean distance, and then the same was repeated using the Manhattan distance. The arithmetic mean of the correctness coefficient for the obtained effects (expressed by the formula (8)) was calculated for each value of k and the used type of the distance function. The results are presented in the form of a table for each parameter set, respectively.

The first set of input parameters was $t = 100, \alpha = 0.2, c = 0.2, \mu = 0.2$ regardless of the value of the k parameter, the Euclidean distance was 100% effective in matching the iris species to the tested record. In the case of the Manhattan distance, the obtained values of the acc coefficient were slightly lower, but the arithmetic mean for each value of k did not fall below 74%. The maximum value was 90%. Detailed results are presented in the table 1.

Then α was increased to value 0.7, leaving the rest of the parameters unchanged. The program efficiency was

Table 2
Arithmetic mean of the acc coefficient for $t = 100, \alpha = 0.7, c = 0.2, \mu = 0.2$.

k	Euclidean distance	Manhattan distance
1	100.00%	77.33%
2	100.00%	78.00%
3	100.00%	72.00%
4	100.00%	76.67%
5	100.00%	84.00%
6	100.00%	76.67%
7	100.00%	85.33%
8	100.00%	80.00%
9	100.00%	80.00%
10	100.00%	79.33%
11	100.00%	83.33%
12	100.00%	78.00%
13	100.00%	85.33%
14	100.00%	86.00%
15	100.00%	75.33%

again flawless for the Euclidean distance. For the Manhattan distance, the arithmetic means of the acc coefficient assumed values from 72% to 86%, which is presented in the table 2. It follows that for a higher value of the α parameter, grouping using the Manhattan distance produced slightly worse results.

In the next step, the effectiveness of the method was tested for the number of iterations $t = 100$ and the coefficients $\alpha = 0.2, c = 0.5, \mu = 0.5$. Concerning the first test, the c and μ parameters have been changed. The use of the Euclidean distance provided 100% of the correctness of the obtained results for all k values. For the Manhattan distance, the program achieved average effectiveness ranging between 76.66% and 89.33%, which can be read from the table 3. There was no significant increase in the correctness of the matches acc to the change in the values of the input parameters.

In the fourth test, the α parameter was increased to 0.7 again, leaving the other parameters unchanged compared to the previous test. The results are shown in the table 4. The arithmetic mean of the acc coefficient for the Euclidean distance in all cases was 100%. For the Manhattan distance, this value varied between 78.67% and 91.33%. The latter number represents the highest average performance achieved so far using the Manhattan distance. Compared to the previous test, the modification of the α parameter positively influenced the operation of the program.

The next four tests were performed for the number of iterations t= 250. At the beginning, the following parameter values were adopted: $\alpha = 0.05, c = 0.7, \mu = 0.33$. The results were placed in the table 5. Again, the method was 100% effective for the Euclidean distance. The arithmetic mean of the acc coefficient for the Manhattan distance was the smallest 78% and the largest 87.33%. A much smaller range of the obtained results was observed compared to the previous tests.

Table 3
Arithmetic mean of the acc coefficient for $t = 100, \alpha = 0.2, c = 0.5, \mu = 0.5$.

k	Euclidean distance	Manhattan distance
1	100.00%	83.33%
2	100.00%	76.66%
3	100.00%	81.34%
4	100.00%	83.33%
5	100.00%	87.33%
6	100.00%	80.00%
7	100.00%	88.67%
8	100.00%	82.00%
9	100.00%	87.33%
10	100.00%	89.33%
11	100.00%	87.33%
12	100.00%	85.33%
13	100.00%	85.33%
14	100.00%	85.33%
15	100.00%	83.33%

Table 4
Arithmetic mean of the acc coefficient for $t = 100, \alpha = 0.7, c = 0.5, \mu = 0.5$.

k	Euclidean distance	Manhattan distance
1	100.00%	78.67%
2	100.00%	84.00%
3	100.00%	84.67%
4	100.00%	91.33%
5	100.00%	88.67%
6	100.00%	81.33%
7	100.00%	86.67%
8	100.00%	85.33%
9	100.00%	88.00%
10	100.00%	86.00%
11	100.00%	86.67%
12	100.00%	86.67%
13	100.00%	87.33%
14	100.00%	86.67%
15	100.00%	88.00%

In the sixth variant, the parameter set from the fifth test was modified by increasing the α parameter to 0.75. The results are presented in the table 6. The program unmistakably allocated the species to the irises using the Euclidean distance for all the considered k values. Using the Manhattan distance, an average acc between 74% and 91.33% was obtained. Such a high value did not occur in the previous test, when $\alpha = 0.05$, but compared to that variant of parameters, the range of the arithmetic mean of the acc coefficient in this sample is larger.

In the penultimate test, the α arameter was returned to

Table 5
Arithmetic mean of the acc coefficient for $t = 250, \alpha = 0.05, c = 0.7, \mu = 0.33$.

k	Euclidean distance	Manhattan distance
1	100.00%	86.67%
2	100.00%	86.00%
3	100.00%	78.00%
4	100.00%	84.00%
5	100.00%	83.33%
6	100.00%	87.33%
7	100.00%	82.00%
8	100.00%	83.33%
9	100.00%	84.00%
10	100.00%	82.00%
11	100.00%	82.00%
12	100.00%	87.33%
13	100.00%	85.33%
14	100.00%	82.00%
15	100.00%	86.00%

Table 6
Arithmetic mean of the acc coefficient for $t = 250, \alpha = 0.75, c = 0.7, \mu = 0.33$.

k	Euclidean distance	Manhattan distance
1	100.00%	80.00%
2	100.00%	74.00%
3	100.00%	80.67%
4	100.00%	80.00%
5	100.00%	84.67%
6	100.00%	82.00%
7	100.00%	84.00%
8	100.00%	84.67%
9	100.00%	84.67%
10	100.00%	80.67%
11	100.00%	86.00%
12	100.00%	81.34%
13	100.00%	91.33%
14	100.00%	79.33%
15	100.00%	83.33%

Table 7
Arithmetic mean of the acc coefficient for $t = 250, \alpha = 0.05, c = 1, \mu = 1$.

k	Euclidean distance	Manhattan distance
1	100.00%	80.00%
2	100.00%	77.33%
3	100.00%	83.33%
4	100.00%	84.00%
5	100.00%	84.00%
6	100.00%	75.33%
7	100.00%	88.67%
8	100.00%	88.00%
9	100.00%	88.66%
10	100.00%	84.67%
11	100.00%	84.00%
12	100.00%	84.00%
13	100.00%	87.33%
14	100.00%	82.67%
15	100.00%	87.33%

Table 8
Arithmetic mean of the acc coefficient for $t = 250, \alpha = 0.75, c = 1, \mu = 1$.

k	Euclidean distance	Manhattan distance
1	100.00%	72.67%
2	100.00%	70.67%
3	100.00%	85.33%
4	100.00%	78.67%
5	100.00%	80.00%
6	100.00%	83.33%
7	100.00%	84.67%
8	100.00%	83.33%
9	100.00%	81.33%
10	100.00%	81.33%
11	100.00%	82.00%
12	100.00%	78.67%
13	100.00%	87.33%
14	100.00%	80.67%
15	100.00%	86.00%

0.05, while c and μ where assumed to be $c = \mu = 1$. The result of the program is shown in the table 7. Using the Euclidean distance again ensured that the method was 100% effective, while for the Manhattan distance the mean of the acc ranged from 75.33% to 88.67%. Modifications to c and μ have not been observed to significantly improve program operation.

The last test was carried out for the number of iterations t= 250, while for the remaining parameters the values were $c = \mu = 1$, as in the previous test, and the α parameter was increased to 0.75. Table 8 presents the results obtained during this experiment. The program matched all records with the species correctly when it used the Euclidean distance. Using the Manhattan distance produced slightly worse results. The efficiency of the program using this distance function, depending on the different values of k,ranged between 70.67% and 87.33%. A fairly low value of the coefficient for k= 2 can be noticed. This is the lowest average value of acc obtained during method testing. Modifying the α parameter did not bring any significant benefits in terms of the method's efficiency.

6. Conclusions

The conducted tests showed that the hybrid classification method solves the problem of matching the appropriate

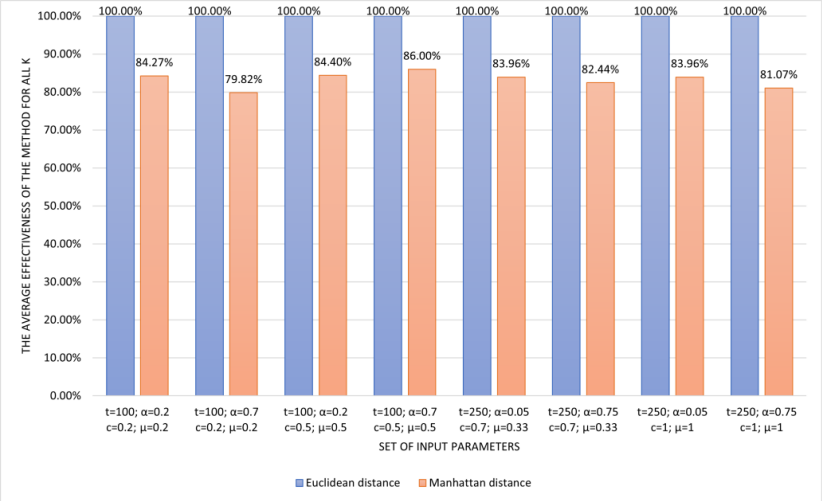

Figure 1: Arithmetic mean of the method effectiveness depending on the variant of input parameters.

species for the iris without mistakes when the Euclidean distance is used in the K Nearest Neighbors Algorithm. When using the Manhattan distance function, the user can expect most records to be classified correctly, but there are few mistakes. The lowest mean value of the acc correctness coefficient obtained was 70.67%, which roughly means that 21 valid matches were made per 30 of the records under test. For the Manhattan distance several times one of the highest effectiveness of the program was noted for k=13. However, the differences are not large enough to clearly state that this is the rule. It was not observed that any of the tested values of k significantly improved the effectiveness of the method, but in most cases, the use of the k parameter with the value from the set 1, 2, 3 resulted in a decrease in the program effectiveness. The graph 1 shows the arithmetic mean of the obtained results for eight trials with specific input parameters.

References

[1] U. Ahmed, J. C.-W. Lin, G. Srivastava, P. Fournier-Viger, A transaction classification model of federated learning, in: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer, 2021, pp. 509–518.

[2] M. Alrasheedi, A. Mardani, A. R. Mishra, P. Rani, N. Loganathan, An extended framework to evaluate sustainable suppliers in manufacturing companies using a new pythagorean fuzzy entropy-swara-waspas decision-making approach, Journal of Enterprise Information Management (2021).

[3] D. Połap, M. Włodarczyk-Sielicka, N. Wawrzyniak, Automatic ship classification for a riverside monitoring system using a cascade of artificial intelligence techniques including penalties and rewards, ISA transactions (2021).

[4] P. Kulczycki, M. Charytanowicz, Kompletny algorytm gradientowej klasteryzacji, Sterowanie i automatyzacja: aktualne problemy i ich rozwiazania (2008) 312–321.

[5] A. V. Aho, J. E. Hopcroft, J. D. Ullman, Algorytmy i struktury danych, Helion, 2003.

[6] M. Woźniak, D. Połap, G. Borowik, C. Napoli, A first attempt to cloud-based user verification in distributed system, 2015, p. 226 – 231. doi:10.1109/APCASE.2015.47.

[7] D. Połap, K. Kęsik, M. Woźniak, R. Damaševičius, Parallel technique for the metaheuristic algorithms using devoted local search and manipulating the solutions space, Applied Sciences 8 (2018) 293.

[8] M. Woźniak, D. Połap, C. Napoli, E. Tramontana, Graphic object feature extraction system based on cuckoo search algorithm, Expert Systems with Applications 66 (2016) 20 – 31. doi:10.1016/j.eswa.2016.08.068.

[9] G. De Magistris, S. Russo, S. J. Roma, P. and, C. Napoli, An explainable fake news detector based on named entity recognition and stance classification applied to covid-19, Information (Switzerland) 13 (2022). doi:10.3390/info13030137.

[10] D. Połap, M. Woźniak, R. Damaševičius, R. Maskeliūnas, Bio-inspired voice evalua-

tion mechanism, Applied Soft Computing 80 (2019) 342–357.

[11] G. Lo Sciuto, G. Capizzi, S. Coco, R. Shikler, Geometric shape optimization of organic solar cells for efficiency enhancement by neural networks, Lecture Notes in Mechanical Engineering (2017) 789 – 796. doi:10.1007/978-3-319-45781-9_79.

[12] G. Capizzi, G. Lo Sciuto, M. Woźniak, R. Damaševičius, A clustering based system for automated oil spill detection by satellite remote sensing, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9693 (2016) 613 – 623. doi:10.1007/978-3-319-39384-1_54.

[13] G. C. Cardarilli, L. Di Nunzio, R. Fazzolari, M. Panella, M. Re, A. Rosato, S. Span, A parallel hardware implementation for 2-d hierarchical clustering based on fuzzy logic, IEEE Transactions on Circuits and Systems II: Express Briefs 68 (2020) 1428–1432.

[14] N. Brandizzi, V. Bianco, G. Castro, S. Russo, A. Wajda, Automatic rgb inference based on facial emotion recognition, volume 3092, 2021, p. 66 – 74.

[15] X.-S. Yang, S. Deb, Cuckoo search via lévy flights, in: 2009 World congress on nature & biologically inspired computing (NaBIC), Ieee, 2009, pp. 210–214.

[16] G. Capizzi, C. Napoli, S. Russo, M. Woźniak, Lessening stress and anxiety-related behaviors by means of ai-driven drones for aromatherapy, volume 2594, 2020, p. 7 – 12.

[17] J. Sudyka, Genetyczne zagadki specjalizacji u kukułki cuculus canorus, Ornis Polonica 53 (2012).

[18] Q. Kuang, L. Zhao, A practical gpu based knn algorithm, in: Proceedings. The 2009 International Symposium on Computer Science and Computational Technology (ISCSCI 2009), Citeseer, 2009, p. 151.

[19] R. A. Fisher, The use of multiple measurements in taxonomic problems, Annals of eugenics 7 (1936) 179–188.