BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES
DOI: 10.24425/bpasts.2024.152603

THIS IS AN EARLY ACCESS ARTICLE.
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

Impact of Data Particle Divide Depth Level on Effectiveness of Hypergeometrical Divide Classifier

Łukasz RYBAK[1] **and Janusz DUDCZYK**[1] *****

[1] Military University of Technology, Faculty of Electronics

Abstract.. Gravitational classifiers belong to the supervised machine learning area, and the basic element that they process is a data particle. So far, many algorithms have been presented in the world literature. They focus on creating a data particle and determining its two important parameters - a centroid and a mass. Hypergeometrical Divide is one of the latest algorithms in this group, which focuses on reducing the amount of processing data and keeping relevant information. A proportion of data to information depends on the data particle divide depth level. Its properties and application potential have been researched, and this article is the next step of the work. The aim of the research described in this article was to determine relation of the depth level value of data particle divide to the effectiveness of the Hypergeometrical Divide algorithm. The research was conducted on 7 real data sets with different characteristics, applying methods and measures of evaluating artificial intelligence algorithms described in the literature. 63 measurements were performed. As a result, the effectiveness of the Hypergeometrical Divide method was defined at each of available data particle divide depth levels for each of used databases.

Key words: data; information; artificial intelligence; machine learning; information processing; Hypergeometrical Divide; data particle; data gravitation

1. INTRODUCTION

The definition of artificial intelligence (AI) should be perceived as the ability of system thanks to which it can interpret collected data correctly, learn from them, adapt to current conditions, and use the processed information in the process of achieving assumed goals [1]. Analyzing this definition, it can be seen that the process of data processing by artificial intelligence algorithms implements an information hierarchy, which in literature on information theory is called the Data, Information, Knowledge and Wisdom (DIKW) pyramid. The DIKW pyramid describes the relations, including general transformation rules, between four levels of information processing [2]. Functioning of modern intelligent systems, which apply machine learning algorithms, is an example of practical application of this theory.

The history of the Hypergeometrical Divide (HypGD) algorithm, whose name was mentioned in the title of this article, started in 2022 [3]. The method published in the doctoral thesis of one of the authors of this publication was devoted to artificial intelligence algorithms [1] using a theory of data gravitation [4], which is based on Newton's law of universal gravitation [5]. From a high-level point of view, the HypGD mechanics combines the lazy learning strategy [6] used in the *k* Nearest Neighbors (*k*NN) classifier [7] with the idea of density-based clustering algorithms such as DBSCAN [8] or OPTICS [9]. As a result, the Hypergeometrical Divide, based on the density distribution of multidimensional feature space, creates a generalized description of its decision regions. The result of creation is a small, easy to manage, reference database without detailed information about the source data, used in the pattern recognition stage based on a minimum distance between objects in the feature space [3]. A significant feature of HypGD in the context of practical applications is the lack of requirement to select parameter values depending on the characteristics of feature space. Moreover, it does not require a learning process leading to build a model, which distinguishes it from the Support Vector Machine (SVM) [10] and the Decision Trees (DTs) [11] algorithms. The presented features implicate that the Hypergeometrical Divide is dedicated to the following applications:

- requiring rapid pattern recognition at the expense of its accuracy;
- in which a training data set is quickly changed and dynamically adapted to a current purpose of pattern recognition process;
- with a high risk of revealing an inference mechanism or even the reference database.

In the context of information theory the overall idea of HypGD method is to reduce the amount of training data, simultaneously keeping relevant information in the pattern recognition process.

*e-mail: janusz.dudczyk@wat.edu.pl

The subplot of the abovementioned dissertation [3] was the impact of data particle divide depth level on the effectiveness of the algorithm. This was a starting point for conducting in-depth research and analyses presented in this publication.

Hypergeometrical Divide is a method belonging to the group of supervised machine learning algorithms [3, 12] that focuses on creating data particles whose, according to the theory, inherent parameters are the center point and mass [4]. An essential issue related to the processing of data particles is the process of determining the mentioned parameters. These are activities whose results may have a direct impact on the effectiveness of gravitation classifier. In the world literature much attention has been paid to determining the center point of data particle [13]. A simple and effective strategy is to construct it based on the average values of particular attributes of processed database [4]. In a few research publications several approaches of defining the value of data particle mass have been proposed over the last years as well [4, 13, 14].

Mentioned algorithm extends the previously published methods, which have been still applied to realize classification process only in two dimensional data sets. Motivation to develop the Hypergeometrical Divide algorithm was determined by a significant disadvantage of previously published data particle geometrical divide methods - the possibility to apply them only in the classification process of data sets whose elements are placed in the feature space of dimension $\mathbb{R}^2$. The Hypergeometrical Divide approach put an end to that limitation and enables to create a data particle by its geometrical divide in data sets whose objects belong to the $\mathbb{R}^{2+}$ dimension feature space. An important issue in using the HypGD method is selection of data particle divide depth level, which became a main subject of this article research problem [3].

The Hypergeometrical Divide algorithm has already passed its practical exam [15]. The article [15] evaluated the potential of its application in the task of specific emitter identification (SEI) [16, 17] belonging to field of electronic intelligence (ELINT) [18, 19]. At that time a research was carried out to recognize the belonging of particular pulses to one of six radar copies of the same type. This is a key task performed by modern mobile ELINT systems, in which increasingly often the sensor recognizing the radar signals along with the limited amount of reference data are carried on an unmanned aerial vehicle (UAV) [20]. While conducting an operational activities, UAVs are an object of interest for a foe intelligence, therefore the resources carried on its board should contain the most generalized information, which could deliver a minimal value in a case of such platform interception. Moreover, the ELINT activities applying UAVs are often carried out in Emissions Control (EMCON) conditions, in which radio transmission resources are rigorously managed and significantly limited, in order to avoid detection, localization and data leakage [21]. Therefore, taking into consideration the dynamically changing targets and the reference data during the reconnaissance activities, this type of systems require usage of small and easily manageable reference databases and the pattern recognition methods, in which the relearning process is unnecessary. Due to the fact that the described systems record many pulses in a short time, another important issue is usage of algorithms, which limit the number of comparisons made in the decision-making process at the expense of an acceptable decrease in its quality. The results showed in [15] revealed that the Hypergeometrical Divide method is characterized by good performance in the process of specific emission sources identification. However, despite its demonstrated advantages, the approach is not free from weaknesses. Previous publications have shown that the main problem of the Hypergeometrical Divide algorithm is the need to manually define the depth level value of data particle divide [3]. It was stated that the development of approaches or rules dedicated to determining the value of mentioned parameter, maximizing the effectiveness of this classifier, may constitute a significant contribution to the development of data particle creation algorithms by its geometrical divide [3, 15].

Currently, when analyzing the abovementioned problem, it was recognized that before automating the process of defining the value of the data particle divide depth level, an in-depth analysis of its impact on the effectiveness of the Hypergeometrical Divide algorithm should be performed. This became the purpose of this article and was directly included in the title of this publication. The results of this research may be an important step towards the development of algorithms that enable automatic selection of the value of data particle divide depth level. The research described in this article was carried out on 7 data sets related to various areas of reality. Within research works 63 experiments were carried out. They showed changes in the effectiveness of the Hypergeometrical Divide algorithm on particular data sets, depending on the used data particle divide depth level. One of the main conclusions refers to the fact that not in every case there is a need to perform divide at the maximum available depth level for an individual data set because there is an iteration of divide, after which no subsequent iteration brings a significant improvement in the effectiveness of the tested algorithm.

2. APPLIED METHODS AND MATERIALS

2.1. Hypergeometrical Divide theoretical details

The Hypergeometrical Divide algorithm, belonging to a pattern recognition approaches set, was proposed in [3]. It is used in the gravitational model-based classification process [4]. Its idea is to manipulate the affiliation of atomic data particles to particular data particles. The atomic data particle should be identified as an elementary object of the feature space, which is processed by the gravitational algorithm and created on the basis of a single record of the analyzed database. Such data particle cannot be divided, what is literally pointed out in its name [4]. It is implemented by iterative dividing of existing data particles. The number of divide cycles is equal to the value of data particle divide depth level selected by the user. The impact of this parameter value on the efficiency of the classifier is the main issue of this article. It is important to emphasize that the result of divide are two new data particles. Implementation of this process changes the masses of data particles and the location of their central points. The next link in the chain of changes are the data gravity forces which determine the relations between the existing data particles and the classified sample. The above-mentioned factors morph the decision boundaries and modify the number of elements processed in the decision-making process. This directly affects the properties of classifier.

Divide of data particle in an *n*-dimensional feature space (*n* >= 3) begins with determining the vectors defining the geometric center **c** and the data particle center of mass **μ**. Assuming that the values of the *i*-th in the *n*-element attributes set of data particle being process constitute the set **F***ᵢ*, then the vector defining the data particle geometric center **c** expresses the Eq. (1) [3].

$$\mathbf{c} = \begin{bmatrix} min(\mathbf{F}_1) + \dfrac{max(\mathbf{F}_1) - min(\mathbf{F}_1)}{2}, \\ \dots, \\ min(\mathbf{F}_n) + \dfrac{max(\mathbf{F}_n) - min(\mathbf{F}_n)}{2} \end{bmatrix} \tag{1}$$

Knowing that there is a relationship showed in Eq. (2).

$$\mathbf{F}_i = \{f_{ji}, \dots, f_{mi}\} \tag{2}$$

Then the vector describing the data particle center of mass **μ** is given by Eq. (3) [3].

$$\boldsymbol{\mu} = \left[\left(\sum_{j=1}^{|\mathbf{F}_1|} f_{j1}\right) \cdot \frac{1}{|\mathbf{F}_1|}, \dots, \left(\sum_{j=1}^{|\mathbf{F}_n|} f_{jn}\right) \cdot \frac{1}{|\mathbf{F}_n|}\right] \tag{3}$$

Then, the condition is verified whether the vectors expressing the geometric center **c** and the center of mass **μ** are not identical. If **c** ≡ **μ**, the process is terminated. In this moment, in the feature space, there are data particles created in the previous iteration of divide. If **c** ≠ **μ**, the next step of the algorithm is performed - determining the normal vector **n** of the searched hyperplane (Eq. (4)), which will divide the data particle [3].

$$\mathbf{n} = \begin{bmatrix} min(\mathbf{F}_1) + \dfrac{max(\mathbf{F}_1) - min(\mathbf{F}_1)}{2} - \left(\sum_{j=1}^{|\mathbf{F}_1|} f_{j1}\right) \cdot \dfrac{1}{|\mathbf{F}_1|}, \\ \dots, \\ min(\mathbf{F}_n) + \dfrac{max(\mathbf{F}_n) - min(\mathbf{F}_n)}{2} - \left(\sum_{j=1}^{|\mathbf{F}_n|} f_{jn}\right) \cdot \dfrac{1}{|\mathbf{F}_n|} \end{bmatrix} \tag{4}$$

Knowing that there is a relationship showed in Eq. (5).

$$\mathbf{n} = \{a_i, \dots, a_n\} \tag{5}$$

In the next step of data particle divide, taking into account the assumption of the Hypergeometrical Divide method that the data particle center of mass **μ** belongs to the hyperplane dividing this data particle, the value of arbitrary constant a_0 is determined, which is expressed in Eq. (6) [3].

$$a_0 = \sum_{i=1}^{n}\left(a_i \cdot \left(\left(\sum_{j=1}^{|\mathbf{F}_i|} f_{ji}\right) \cdot \frac{1}{|\mathbf{F}_i|}\right)\right) \tag{6}$$

Having all the components of the equation of the hyperplane dividing the data particle, the last step of divide is to check the position of each atomic component of the data particle **p**. Knowing that the relation presented in Eq. (7) is true [3].

$$\mathbf{p} = \{p_i, \dots, p_n\} \tag{7}$$

The process of assigning an atomic data particle **p** to one of the two newly created data particles ***P****A* or ***P****B* is expressed by Eq. (8) [3].

$$\mathbf{p} \in \begin{cases} \boldsymbol{P}_A, if \left(\sum_{i=1}^{n} a_i p_i\right) - a_0 \geq 0 \\ \boldsymbol{P}_B, if \left(\sum_{i=1}^{n} a_i p_i\right) - a_0 < 0 \end{cases} \tag{8}$$

A measurable added value of the Hypergeometrical Divide algorithm training phase is a reduction in the amount of data processed at the classification phase. Denoting the number of classes in the data set as *c* and the hypergeometrical divide depth level as *d*, the number of elements created by the HypGD for the classification process (*n_clf*) is expressed by the equation (9).

$$n\_clf_{c,d} = c \cdot 2^d \tag{9}$$

Assuming that:

- each of the *n*-elements in the data set refers to exactly one atomic data particle (ADP),
- each ADP belongs to only one data particle,
- *d* << *n* and *c* << *n* (in practice),

then, analyzing the computational complexity of the Hypergeometrical Divide algorithm, during which only the dominant component is preserved, it can be concluded that the computational complexity of the HypGD training phase is asymptotically linear *O(n)*.

2.2. Configuration details of examined algorithm

In sum, this approach focuses on creating new data particles by dividing existing data particle using their geometric properties in multidimensional feature space [3, 22]. As already mentioned in the previous chapter, an important issue in using this algorithm is the process of determining the depth level of data particle divide *d* [3, 15]. In the research carried out, the maximum value of the *d* parameter was determined for each data set. As far as the philosophy of examined method is concerned, it is known that for each data set these values may be different [3, 14, 15]. The established maximum depth levels of data particle divide were presented in Table 1.

Table 1. The maximum depth level of data particle divide *d* for particular data sets (source: own elaboration)

Data set	Maximum Divide Depth Level (*d*)
banknote_authentication	5
iris	3
magic_gamma_telescope	8
occupancy	7
parkinsons	3
sonar	4
wifi_localization	5

As mentioned in Chapter 1. Introduction - the Hypergeometrical Divide algorithm is used to create a data particle [3, 15, 22]. It is known from the theory of data gravitation that during data particles processing it is also necessary to determine its two parameters - center point and mass value [3, 4, 13]. In the conducted research, the center point was determined based on the strategy of average value of particular attributes in the context of all elements of specific data particle. However, three approaches were used to determine the data particle mass value. Two of them were presented and described in detail in [13] - Stochastic Learning Algorithm (SLA) and Batch-update Learning Algorithm (BLA). They improve some properties of the popular Centroid-Based Classifier (CBC), whose popularity is due to its simple theoretical foundation and linear computational complexity in the training phase [13]. In the training phase, the SLA algorithm iteratively corrects the value of mass coefficient for individual data particles to obtain the best possible match to the entire training set [13]. In this research the Stochastic Learning Algorithm was configured as follows:

- max. iterations number *maxIters* = 50;
- mass value update factor $\xi = 0.0001$;
- expected error level $\varepsilon = 0.00$.

In turn, the second algorithm proposed in [13] - Batch-update Learning Algorithm - corrects the weight coefficients of particular data particles after completing the classification process of all samples included in the training data set. The update factor of the data particle mass value in the Batch-update Learning Algorithm was set to $\xi = 0.0001$. The last approach used to define the data particle mass values was the *n*-Mass Model. According to its philosophy, the value of the mass of a data particle is equal to the size of its base class [3, 4, 13, 14]. In this research, a fourth variant was used as well, which ignores the mass of data particles. Therefore, at each level of data particle divide, four results were obtained for each data set, on the basis of which the average value was calculated, describing the final quality of classification process.

2.3. Evaluation method and quality metrics

The method and quality measures selected to evaluate the classification process have already been used in publications whose topics fall within the field of artificial intelligence. In these studies, one of the most popular methods used for data sampling was used to estimate the actual effectiveness of the classifier and possible tuning of its parameters [23]. The method described was *k*-fold cross-validation. Its use enabled to eliminate the phenomenon of predictive model overfitting in the evaluation process. Moreover, thanks to its application, it was also possible to examine the generalization ability of the tested algorithm. The use of *k*-fold cross-validation required determining the value of *k* parameter [24]. In these studies, it was assumed that $k = 10$, what ensured a slight difference in the values of measures [25].

The obtained classification results were saved in the form of a four-element confusion matrix. It consisted of the following values: true positive (TP), true negative (TN), false positive (FP) and false negative (FN) [26]. Based on the matrix organized this way, the values of two quality measures of predictive model were determined - precision and recall [27, 28]. Taking into account the fact that the abovementioned measures are determined for a single class, macro variants of these measures were used to examine the overall effectiveness of the classifier in each data set. According to the definition, $PRECISION_{macro}$ and $RECALL_{macro}$ are average values calculated on the basis of *PRECISION* and *RECALL* for each of *n*-classes [29, 30]. Equation (10) and Eq. (11) describe $PRECISION_{macro}$ and $RECALL_{macro}$, respectively.

$$PRECISION_{macro} = \left(\sum_{i=1}^{n} \frac{TP_i}{TP_i + FP_i}\right) \cdot \frac{1}{n} \tag{10}$$

$$RECALL_{macro} = \left(\sum_{i=1}^{n} \frac{TP_i}{TP_i + FN_i}\right) \cdot \frac{1}{n} \tag{11}$$

Using domain knowledge, the characterized measures were reduced to a single F_{macro} value, which is expressed by the Eq. (12) [29, 30].

$$F_{macro} = \frac{2 \cdot PRECISION_{macro} \cdot RECALL_{macro}}{PRECISION_{macro} + RECALL_{macro}} \tag{12}$$

2.4. Details of data sets used in the research

As mentioned in the introduction of this article, 7 data sets were used in the research. Each of them concerns a different problem occurring in a real environment. Issues related to particular databases include:

- confirming the authenticity of banknotes based on image entropy and features extracted from digital images with application wavelet transform [31, 32];
- distinguishing the type of Iris plant based on the analysis of photos [33];
- discovering of high energy gamma particles on the images of hadronic showers recorded by Cherenkov gamma telescope [34, 35];
- detection of room occupancy, based on the analysis of: temperature, humidity, light and CO_2, which were recorded once a minute [36, 37];
- distinguishing healthy patients from those with Parkinson's disease based on the analysis of their voice recordings [38, 39];
- distinguishing sonar signals reflected from a metal cylinder from signals reflected from a quasi-cylindrical rock [40];
- smartphone location, based on the analysis of the strength of WiFi signals [41].

In the Table 2. the numbers of samples belonging to particular classes in each of applied data sets were presented.

Table 2. Number of samples in classes for particular data sets (source: [31-42])

Data set	Number of samples in classes
banknote_authentication	762 : 610
iris	50 : 50 : 50
magic_gamma_telescope	12332 : 6688
occupancy	15810 : 4750
parkinsons	48 : 147
sonar	111 : 97

wifi_localization	500 : 500 : 500 : 500

All described datasets are available in the public repository of the University of California, Irvine (UCI) [42].

3. RESULTS

In accordance to the aim of article, the results reveal the impact of data particle divide depth level on the effectiveness of the Hypergeometrical Divide method. This chapter presents the average results obtained by the tested algorithm on 7 data sets at all data particle depth levels available to them. The results obtained using the well-known lazy learning *k* Nearest Neighbors algorithm, which based on a distance between the objects in a feature space, were used as the reference values. The outcomes of eager learning approaches - the SVM and the Decision Trees, were taken into consideration as well. A total of 63 measurements were performed. Figures 1-7 show the results obtained for the described data sets, in the following order: banknote_authentication, iris, magic_gamma_telescope, occupancy, parkinsons, sonar, wifi_localization.

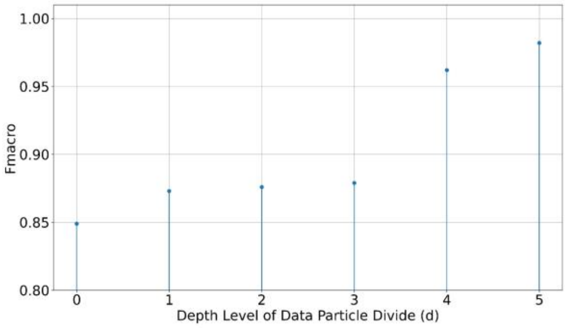

Fig.1. Average value of the F_{macro} measure obtained by Hypergeometrical Divide at each of the available data particle divide depth levels on the banknote_authentication dataset (source: own elaboration)

Figure 1 shows that in the case of the banknote_authentication dataset, the Hypergeometrical Divide algorithm without performing data particle divide obtained a measure value of F_{macro} = 0.849. The first three levels of data particle divide – d = 1, d = 2 and d = 3 – brought an increase of F_{macro} measure value by: 0.024, 0.003, 0.003. The largest leap of F_{macro} value occurred after dividing data particles at the fourth depth level (d = 4) and amounts to 0.082. Performing the last available for this data set divide of existing data particles at depth level d = 5, brought an increase in the F_{macro} value by 0.021. Finally, the Hypergeometrical Divide approach on the banknote_authentication dataset, performing 64 comparisons in the classification phase, obtained a value of F_{macro} = 0.982, whereas the *k*NN algorithm F_{macro} = 0.993, using 1234 comparisons. The classification quality with the SVM approach was F_{macro} = 0.999 and with the Decision Trees algorithm F_{macro} = 0.983.

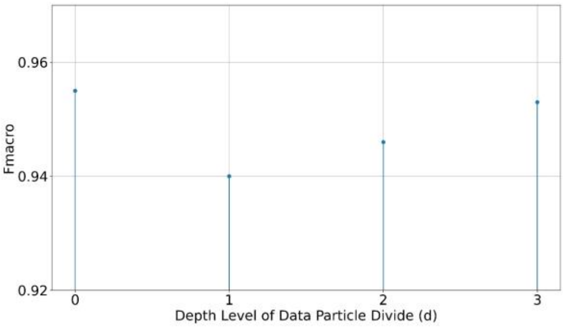

Fig.2. Average value of the F_{macro} measure obtained by Hypergeometrical Divide at each of the available data particle divide depth levels on the iris dataset (source: own elaboration)

Figure 2 visualizes the results obtained on one of the two smallest of the analyzed data sets - iris. It can be observed that the change in the effectiveness of the tested algorithm depending on the level of data particle divide depth used is small. Without divide (depth level d = 0), the Hypergeometrical Divide method obtained F_{macro} = 0.955. After dividing data particle at depth level d = 1, the F_{macro} value decreased to 0.940. The quality of the classification performed on the data set after dividing the data particles at the next depth level d = 2 was described by a higher value than in the case of depth level d = 1, which amounted to F_{macro} = 0.946. After dividing the data particle at the last available depth level for this data set, the algorithm again reached an increase of the F_{macro} value, obtaining a result of F_{macro} = 0.953 with 24 operations in the prediction phase. For comparison, the quality of *k*NN method was lower and amounted to F_{macro} = 0.945, applying 135 operations. However, the eager learning algorithms: the SVM and the Decision Trees obtained F_{macro} = 0.942 and F_{macro} = 0.934, respectively.

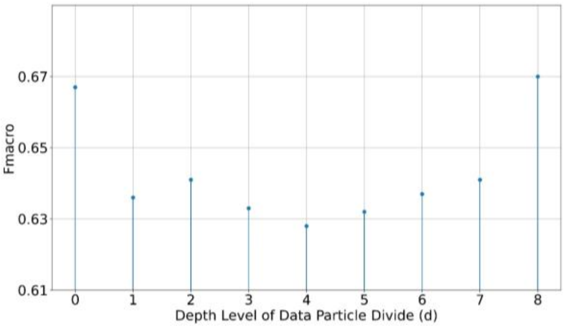

Fig.3. Average value of the F_{macro} measure obtained by Hypergeometrical Divide at each of the available data particle divide depth levels on the magic_gamma_telescope dataset (source: own elaboration)

In Fig. 3 it can be observed that for the magic_gamma_telescope data set, the Hypergeometrical Divide algorithm without application of dividing a data particle obtained the value F_{macro} = 0.667. The first divide of data particle (d = 1) determined the decrease of the F_{macro} value by 0.031. After implementing next divide of data particle d = 2, an increase in the value of measure used by 0.005 was followed. The quality of the classification process after each of the two subsequent divides of data particles - at levels d = 3 and d = 4 - decreased consecutively by 0.009 and 0.005. After each subsequent available depth level of existing data particles divide (d = 5, d = 6, d = 7, d = 8), the value of the F_{macro} measure increased consecutively by 0.005, 0.004, 0.005 and 0.029. The definitive quality of classification process carried out using the Hypergeometrical Divide algorithm on the magic_gamma_telescope data set was described by measure F_{macro} = 0.670, performing 512 comparisons. However, the result obtained using the *k*NN method, which was F_{macro} = 0.810 for 17118 comparison operations in training phase, was taken as the reference value. On this dataset, the SVM reached the value of F_{macro} = 0.837, and the Decision Trees obtained F_{macro} = 0.806.

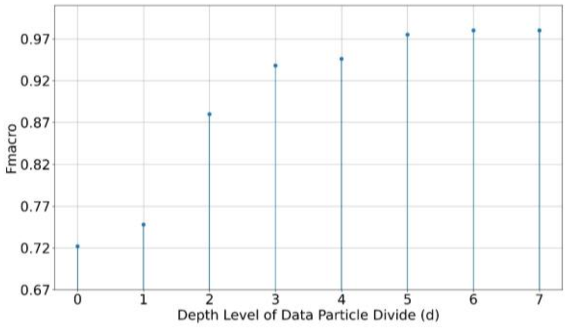

Fig.4. Average value of the F_{macro} measure obtained by Hypergeometrical Divide at each of the available data particle divide depth levels on the occupancy dataset (source: own elaboration)

Figure 4 visualizes the results obtained on the largest of the analyzed data sets - the occupancy data set. It could be seen that divide of data particle at subsequent depth levels never reduces the quality of the classification process. Without divide the data particle, the mentioned quality was described by F_{macro} = 0.722. After divide the data particle at depth level d = 1, the F_{macro} value increased to 0.748. However, the greatest increase in the value of the measure used can be observed for the parameter d = 2. Then the F_{macro} measure assumed the value of 0.880. The divide of data particle at three subsequent depth levels d = 3, d = 4 and d = 5 improved the results to F_{macro} = 0.938, F_{macro} = 0.946 and F_{macro} = 0.975, respectively. For the d = 6 parameter, the quality of the classification process increased slightly, reaching the level of 0.980, using 128 comparisons. The F_{macro} value did not change after the data particle divide at the last available depth level d = 7. For comparison, the quality of *k*NN algorithm, performing 18504 operations, amounted to F_{macro} = 0.987. The classification quality using the SVM approach was F_{macro} = 0.982, and applying the DTs algorithm F_{macro} = 0.985.

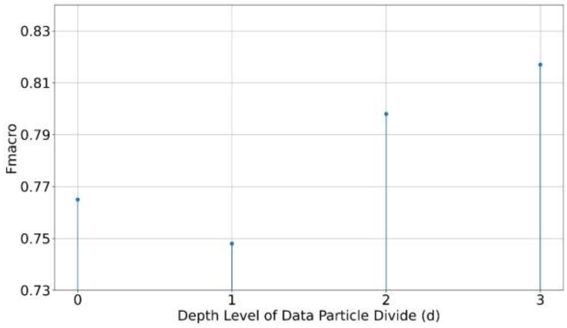

Fig.5. Average value of the F_{macro} measure obtained by Hypergeometrical Divide at each of the available data particle divide depth levels on the parkinsons dataset (source: own elaboration)

Figure 5 shows the results obtained on the parkinsons data set. It can be observed that without dividing the data particle (d = 0), the Hypergeometrical Divide method obtained the value of F_{macro} = 0.765. After the divide was carried out at the depth level d = 1, the F_{macro} value decreased and amounted to F_{macro} = 0.748. The quality of the classification performed on the data set after dividing the data particles at the d = 2 depth level was described by a higher value than at the d = 1 level and was equal to F_{macro} = 0.798. Performing the last available for this dataset divide of the existing data particles, at a depth level d = 3, resulted in an increase in the F_{macro} value, which amounted to F_{macro} = 0.817, applying 16 comparisons in the prediction phase. The *k*NN algorithm obtained F_{macro} = 0.924, performing 175 compare operations. On the other hand, the SVM classifier achieved F_{macro} = 0.765, and the Decision Trees algorithm F_{macro} = 0.858.

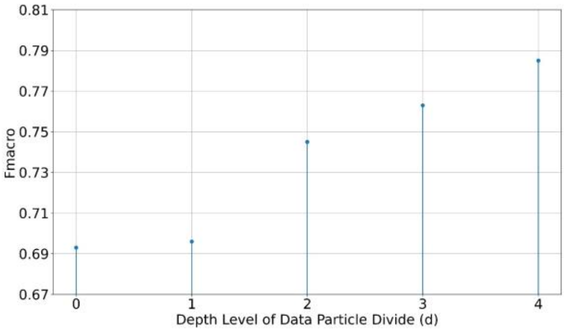

Fig.6. Average value of the F_{macro} measure obtained by Hypergeometrical Divide at each of the available data particle divide depth levels on the sonar dataset (source: own elaboration)

Analyzing Fig. 6 it can be observed that on the sonar data set, each iteration of data particle divide increased the effectiveness of the Hypergeometrical Divide algorithm. Without dividing the data particles, the quality of the classification process was described by F_{macro} = 0.693. The first iteration of data particle divide resulted in an increase in the value of the F_{macro} measure by 0.003. After the divide was carried out at the next depth level d = 2, an increase in the quality of the classification process by 0.049 was obtained. Performing pattern recognition after another data particle divide d = 3 resulted in another increase in the value of the quality measure used by 0.018. After performing the last divide possible for this data set at the level of d = 4, the value of the F_{macro} measure increased by 0.022 and finally reached the level of 0.785, based on 32 compare operations. For comparison, the reference quality obtained using the *k*NN algorithm was higher and amounted to F_{macro} = 0.826, performing 187 comparisons in the classification phase. Whereas the SVM and the Decision Trees algorithms obtained F_{macro} = 0.828 and F_{macro} = 0.747, respectively.

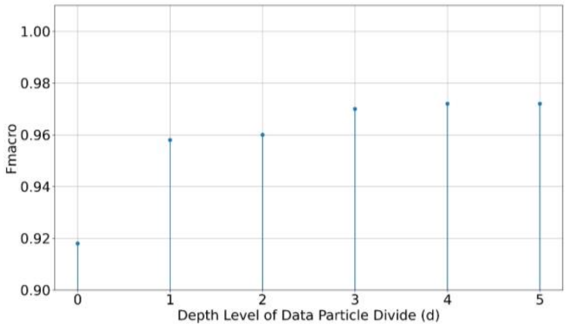

Fig.7. Average value of the F_{macro} measure obtained by Hypergeometrical Divide at each of the available data particle divide depth levels on the wifi_localization dataset (source: own elaboration)

Based on Fig. 7 it can be seen that on the real wifi_localization data set, without the data particle divide (d = 0), the classification process quality described by the F_{macro} measure was at the level of 0.918. After the data particle divide at the d = 1 level, the greatest improvement in results can be seen. Then the value of F_{macro} = 0.958 was reached. Another iteration at depth level d = 2 resulted in a slight increase in the value of the measure used, leading to F_{macro} = 0.960. The data particle divide with the parameter d = 3 resulted in a greater improvement in the quality of the classification process than in the previous iteration of divide, up to the level of F_{macro} = 0.970, which in subsequent loops of divide process (d = 4 and d = 5) finally stopped at the level of F_{macro} = 0.972 with 64 comparisons in the prediction phase. However, the *k*NN classifier obtained the result F_{macro} = 0.980, applying 1800 compare operations, the SVM achieved F_{macro} = 0.981, and the last of the evaluated algorithms - Decision Trees obtained the value of measure F_{macro} = 0.970.

4. SUMMARY

Based on the conducted research a relation between data particle divide depth level and effectiveness of Hypergeometrical Divide algorithm was defined. Therefore, the aim of paper was achieved - the impact of data particle divide depth level on the effectiveness of the examined method was revealed.

The first time an added value of the Hypergeometrical Divide algorithm training phase on the number of objects reduction, applied in the classification process, were clearly explained. The mentioned case has not been considered in world literature so far. This relationship was expressed by equation (9) in the Chapter 2.

Analyzing Fig. 1-7, it can be concluded that the maximum value of the data particle divide depth level for each data set may be different.

By analyzing Fig. 4 and Fig. 7 it can be concluded that in the case of selected data sets, in the process of dividing data particles, there is an iteration after which each subsequent divide does not determine a significant change in the effectiveness of the Hypergeometrical Divide algorithm.

Another conclusion related to the above one which is the fact that using the highest available level of data particle divide depth is not necessary to achieve its almost maximum efficiency for the Hypergeometrical Divide classifier. This is important in the context of maximizing the effectiveness of the classifier while minimizing the data processing time. Moreover, the analysis of Fig. 4 and Fig. 7 allows us to conclude that mentioned level of data particle divide depth, from which there is no significant change in the effectiveness of the Hypergeometrical Divide method, is different for each data set. Therefore, there is no universal value for the depth level of data particle divide that is optimal in the considered criteria.

Analyzing Fig. 2, Fig. 3 and Fig. 5 it can be concluded that not each iteration of data particle divide results in an increase of the classifier effectiveness. Moreover, there are datasets in which dividing the data particle even at the maximum available divide depth level does not improve the effectiveness of the Hypergeometrical Divide approach.

Based on the analysis of the results, it can be concluded that the Hypergeometrical Divide algorithm obtained an average value of the F_{macro} measure lower than the k Nearest Neighbors algorithm. However, the HypGD performed an average of 120 compare operations in the classification process, while the *k*NN algorithm performed 5593 comparisons, which gives a difference of two orders of magnitude with an advantage for the Hypergeometrical Divide algorithm.

Another conclusion arising from the analysis of the results is the Hypergeometrical Divide algorithm obtained an average value of the F_{macro} measure lower than the Support Vector Machine and Decision Trees, which are the eager learning algorithms, applying an earlier prepared model in the classification process.

In conclusion, the article fills the gap in knowledge regarding the properties of algorithm, which found application in Specific Emitter Identification based on the analysis of many pulses. The paper is the next step in research and development work on automating the parameterization of the Hypergeometrical Divide algorithm.

The direction of further research in this area, which may positively affect the usability of the Hypergeometrical Divide algorithm, may be development of a method of automatic defining the data particle divide depth level.

Another significant direction for further research may be a comparison of the Hypergeometrical Divide algorithm properties with artificial neural networks, especially deep learning methods.

ACKNOWLEDGEMENTS

We would like to thank Marlena Zalewska-Rybak for language proofreading.

REFERENCES

[1] A. Kaplan and M. Haenlein. "Siri, Siri, in My Hand: Who's the Fairest in the Land? On the Interpretations, Illustrations, and Implications of Artificial Intelligence," *Business Horizons*, vol. 62, no. 1, pp. 15-25, 2019. doi: 10.1016/j.bushor.2018.08.004.

[2] N. Shedroff. "Information Interaction Design: A Unified Field Theory of Design," in *Information Design*, 1st ed., R. Jacobson, Ed. Cambridge, MA, USA: MIT Press, 2000, pp. 267-292.

[3] Ł. Rybak. "Geometrical Division of Data Particle in Classification of Multidimensional Data Sets," Doctoral Thesis, Lodz University of Technology, Lodz, Poland, September 2022.

[4] L. Peng, Y. Chen, B. Yang and Z. Chen. “A Novel Classification Method Based on Data Gravitation,” in *Proceedings of the 2005 International Conference on Neural Networks and Brain*, 2005, pp. 667–672. doi: 10.1109/ICNNB.2005.1614719.

[5] I. Newton. *Matematyczne Zasady Filozofii Naturalnej*, S. Brzezowski, Translator, Cracow, Poland: Cracow Copernicus Center Press, 2015.

[6] G.I. Webb. “Lazy Learning,” in *Encyclopedia of Machine Learning and Data Mining*. C. Sammut and G. Webb, Eds. Boston, MA, USA: Springer, 2016, pp. 1-2. doi: 10.1007/978-1-4899-7502-7_449-1.

[7] E. Fix and J. L. Hodges. “Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties,” *International Statistical Review*, vol. 57, no. 3, pp. 238–247, 1989. doi: 10.2307/1403797.

[8] M. Ester, H.-P. Kriegel, J. Sander and X. Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231. doi: 10.5555/3001460.3001507.

[9] M. Ankerst, M.M. Breunig, H.-P. Kriegel and J. Sander. “OPTICS: ordering points to identify the clustering structure,” *SIGMOD Record*, vol. 28, no. 2, pp. 49–60, 1999. doi: 10.1145/304181.304187.

[10] C. Cortes and V. Vapnik. “Support-Vector Networks,” *Machine Learning*, vol. 20, pp. 273–297, 1995. doi: 10.1007/BF00994018.

[11] L. Breiman, J. Friedman, R. Olshen and C. Stone. “Classification and Regression Trees,” *Biometrics*, vol. 40, no. 3, 1984. doi: 10.2307/2530946.

[12] S. Theodoridis and K. Koutroumbas. “Chapter 1 – Introduction,” in *Pattern Recognition*. 4th ed., S. Theodoridis and K. Koutroumbas, Eds. Orlando, FL, USA: Academic Press, 2009, pp. 1-12. doi: 10.1016/B978-1-59749-272-0.50003-7.

[13] C. Liu, W. Wang, G. Tu, Y. Xiang, S. Wang and F. Lv. “A new Centroid-Based Classification Model for Text Categorization,” *Knowledge-Based Systems*, vol. 136, pp. 15–26, 2017. doi: 10.1016/j.knosys.2017.08.020.

[14] Ł. Rybak and J. Dudczyk. “Various Approaches to Modelling of the Mass Using the Size of the Class in the Centroid Based Classification,” *Elektronika – Konstrukcje, Technologie, Zastosowania*, vol. 60, no. 6, pp. 62–65, 2019. doi: 10.15199/13.2019.6.13.

[15] J. Dudczyk and Ł. Rybak. “Application of Data Particle Geometrical Divide Algorithms in the Process of Radar Signal Recognition,” *Sensors*, vol. 23, no. 19, 2023. doi: 10.3390/s23198183.

[16] Y. Zhao, X. Wang, Z. Lin and Z. Huang. “Multi-Classifier Fusion for Open-Set Specific Emitter Identification,” *Remote Sensing*, vol. 14, no. 9, 2022, doi: 10.3390/rs14092226.

[17] C. Wang, Y. Wang, Y. Zhang, H. Xu and Z. Zhang. “Open-Set Specific Emitter Identification Based on Prototypical Networks and Extreme Value Theory,” *Applied Sciences*, vol. 13, no. 6, 2023. doi: 10.3390/app13063878.

[18] R. G. Wiley. *Electronic Intelligence: The Interception of Radar Signals*, Dedham, MA, USA: Artech House Publishers, 1985.

[19] A. Alparslan and K. Yegin. "A Fast ELINT Receiver Design," in *Proceedings of the 13th European Radar Conference*. (EuRAD), 2016, pp. 217-220.

[20] V. Gautam and V. Shishodia. “The E-Intelligence System,” arXiv, 2022. doi: 10.48550/arXiv.2201.02590.

[21] B. Nguyen and R. Rom. “Communication Services Under EMCON,” *SIGCOMM Computer Communication Review*, vol. 16, no. 3, pp. 275–281, 1986. doi: 10.1145/1013812.18203.

[22] D.-C. Li, Q.-S. Shi, Y.-S. Lin and L.-S. Lin. “A Boundary-Information-Based Oversampling Approach to Improve Learning Performance for Imbalanced Datasets,” *Entropy*, vol. 24, no. 3, 2022. doi: 10.3390/e24030322.

[23] D. Berrar. “Cross-Validation,” in *Encyclopedia of Bioinformatics and Computational Biology*. 1st ed., S. Ranganathan, M. Gribskov, K. Nakai and Ch. Schönbach, Eds. Elsevier, 2019, pp. 542-545. doi: 10.1016/b978-0-12-809633-8.20349-x.

[24] R. O. Duda, P. E. Hart and D. G. Stork. “Introduction,” in *Pattern Classification*, 2nd ed., New York, NY, USA: Wiley-Interscience, 2000, pp. 1–19.

[25] I. K. Nti, O. Nyarko-Boateng and J. Aning, "Performance of Machine Learning Algorithms with Different K Values in K-fold Cross-Validation," *International Journal of Information Technology and Computer Science*, vol. 13, no. 6, 2021. pp. 61-71 doi: 10.5815/ijitcs.2021.06.05.

[26] M. Hossin, M. Sulaiman, A. Mustapha, N. Mustapha and R. Rahmat. “A Hybrid Evaluation Metric for Optimizing Classifier,” in *Proceedings of the 3rd Conference on Data Mining and Optimization*. (DMO), 2011, pp. 165-170. doi: 10.1109/DMO.2011.5976522.

[27] A. Kent, M. Berry, F. U. Luehrs and J. W. Perry. „Machine literature searching VIII. Operational criteria for designing information retrieval systems,” *Journal of the Association for Information Science and Technology*, vol. 6, no. 2, pp. 93-101, 1955. doi: 10.1002/asi.5090060209.

[28] Y. Jiang, W. Li and L. Liu. “R-CenterNet+: Anchor-Free Detector for Ship Detection in SAR Images,” *Sensors*, vol. 21, no. 17, 2021. doi: 10.3390/s21175693.

[29] Z. C. Lipton, C. Elkan and B. Narayanaswamy. “Optimal Thresholding of Classifiers to Maximize F1 Measure,” in *Machine Learning and Knowledge Discovery in Databases*. 1st ed., T. Calders, F. Esposito, E. Hüllermeier and R. Meo, Eds. Berlin, Heidelberg, Germany: Springer, 2014, pp. 225-239. doi: 10.1007/978-3-662-44851-9_15.

[30] P. Flach and M. Kull. “Precision-Recall-Gain Curves: PR Analysis Done Right,” in *Proceeding of the 28th International Conference on Neural Information Processing Systems*. (NIPS 2015), 2015, pp. 838–846. doi: 10.5555/2969239.2969333.

[31] V. Lohweg. “banknote authentication.” UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/dataset/267/banknote+authentication. (Accessed: 12. Nov. 2023). doi: 10.24432/C55P57.

[32] N. Ghasem Abadi. “Machine Learning-based Authentication of Banknotes: A Comprehensive Analysis,” *Big Data and Computing Visions*, vol. 4, no. 1, pp. 22–30, 2024. doi: 10.22105/bdcv.2024.197120.

[33] R. A. Fisher. “iris.” UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/dataset/53/iris. (Accessed: 12. Nov. 2023). doi: 10.24432/C56C76.

[34] R. Bock. “MAGIC Gamma Telescope.” UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/dataset/159/magic+gamma+telescope. (Accessed: 12. Nov. 2023). doi: 10.24432/C52C8B.

[35] D. Heck, J. Knapp, J. N. Capdevielle, G. Schatz and T. Thouw. “CORSIKA: A Monte Carlo Code to Simulate Extensive Air Showers,” Technical Note, FZKA 6019, Forschungszentrum, Karlsruhe, Germany, 1998. doi: 10.5445/IR/270043064.

[36] L. Candanedo. “Occupancy Detection.” UCI Machine Learning Repository. [Online] Available: https://archive.ics.uci.edu/dataset/357/occupancy+detection. (Accessed: 12. Nov. 2023). doi: 10.24432/C5X01N.

[37] L. Candanedo and V. Feldheim. “Accurate Occupancy Detection of an Office Room From Light, Temperature, Humidity and CO2 Measurements Using Statistical Learning Models,” *Energy and Buildings*, vol. 112, pp. 28-39, 2016. doi: 10.1016/J.ENBUILD.2015.11.071.

[38] M. Little. “Parkinsons.” UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/dataset/174/parkinsons. (Accessed: 12. Nov. 2023). doi: 10.24432/C59C74.

[39] M. A. Little, P. E. McSharry, E. J. Hunter, J. Spielman and L. O. Ramig. “Suitability of Dysphonia Measurements for Telemonitoring of Parkinson's Disease,” in *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 4, pp. 1015-1022, 2009. doi: 10.1109/TBME.2008.2005954.

[40] T. Sejnowski and R. Gorman. “Connectionist Bench (Sonar, Mines vs. Rocks).” UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/dataset/151/connectionist+bench+sonar+mines+vs+rocks. (Accessed: 12 Nov. 2023). doi: 10.24432/C5T01Q.

[41] R. Bhatt. “Wireless Indoor Localization.” UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/dataset/422/wireless+indoor+localization. (Accessed: 12 Nov. 2023). doi: 10.24432/C51880.

[42] M. Kelly, R. Longjohn and K. Nottingham. “The UCI Machine Learning Repository.” [Online]. Available: https://archive.ics.uci.edu. (Accessed: 12. Nov. 2023).