

# Use of the mini-model method in classification task on example of iris flower dataset

Marcin Pietrzykowski

Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, Szczecin, Poland

[mpietrzykowski@wi.zut.edu.pl](mailto:mpietrzykowski@wi.zut.edu.pl)

---

**Abstract:** *The paper presents use of the mini-models method in a classification task. The article briefly describes the method and compares it to the k-nearest neighbor algorithm. The algorithm concentrates only on local query data and uses a data samples only from local neighborhood of the query. The paper presents the results of experiment that compare the effectiveness of mini-models with selected methods of classification. The experiments were performed on well-known Iris Flower dataset and on other popular classification datasets.*

**Keywords:** *mini-model, local classification, k-nearest neighbors method, mathematical modeling*

---

## 1. Introduction

The paper presents use of the mini-models method in a classification task. Originally the method is a algorithm of local data modeling [1, 2, 3, 4, 5, 6, 7]. The concept of the method of mini-models was developed by A. Piegat. In contrast to most well known methods of modeling such as neural networks, neurofuzzy networks or polynomial approximation, the method does not create a global model if it is not necessary. Mini-models such as the method of k-nearest neighbors operate only on data from local neighborhood of the query. This two method are very similar in some way and the differences between them will be described later in this article. Originally the mini-models method consists of two groups of algorithms: algorithms for defining local neighborhood of the query point, algorithms for mathematical modeling on the mini-model area. The mini-model area can be defined as an area of a polytope placed in the general number of dimensions in the input space of the problem. It is a line segment in a 2D-space, polygon in 3D-space (irregular triangle, quadrilateral, etc.), polyhedron in 4D-space (simplex, hexahedron, etc.). In classification task, the query point is a data sample to be classified. The main aim of this paper it to check how the mini-models method works as a methods of modeling and if it has capability to achieve results comparable to other widely-used methods.

## 2. Mini-models as method of local modeling

Modeling methods can be divided into two main groups: local methods and global methods. Global methods use all available data, whilst local methods use only a part of samples from the entire dataset. Global methods include e.g.: neural networks, neurofuzzy networks, polynomial and linear approximation. Local methods are e.g.: k-nearest neighbors method, kernel density estimation.

In the modeling task we try to identify mathematical function  $y = f(x_1, x_2, \dots, x_n)$  which describes dependency between input and output variables. The approach of local modeling is a consequence of the fact that in the modeling process mostly we are interested in an answer for a specific query. Example of that question is: “*What does ‘y’ amounts to if ‘x<sub>1</sub>’ amount to 0.5, ‘x<sub>2</sub>’ to 0.8, ‘x<sub>2</sub>’ to 0.1 and etc?*”. When the scientist is modeling air pollution on the road it will ask question, such as: “*How large will the air pollution on road be when wind speed amounts to 5 m/s, temperature 2 meter above the ground amounts to 24 ° C, number of cars per hour amounts to 200?*”. The answer to the question requires only the data “*wind speed amounts about 5 m/s, temperature to 24 ° C and number of cars to 200*”. In the general case the query point is a set of independent variable-values. In the 2D-space example of a simple query can be: “*What does the unemployment rate in Poland amounts to, when the number of retired person equals 2000000?*”. The dependent variable  $y$  is here unemployment rate and the independent variable  $x$  is the engine power. The query point will have here the following form  $x_0 = 2000000y = ?$  or simply  $x_0 = 2000000$ . Mini-models operate on a data in the local neighborhood of the query point that actually is important for a scientist. The method requires relatively small number of data samples in the learning process. This fact has great importance in a situation of data deficiency which very often occurs in real problems.

Local methods are competitive to global methods which create smooth and regular functional surface. Global methods are able to visualize a shape of the system model over the whole area in a 2D or 3D space. However, this argument loses its importance in spaces with a greater number of dimensions, which cannot be clearly visualized. The mini-models method has possibility to build a global model in order to learn the value of a modeled variable across an entire domain. This can be done very simply by adding together mini-models for subsequent query points. We should remember that, the creation of a global model is a time-consuming process and when a new sample is acquired the global model becomes outdated and re-learning is required. Local modeling method does not construct global model and they are able to work in situations where new data points are continuously being received.

## 2.1. The difference between local and global model

There are a significant difference between a global model and a mini-model (which is local and space-constrained). Both models can use the same method of approximation e.g. linear approximation for 2D-space problem. The mini-model method uses only a part of samples from the entire dataset. Samples on which mini-model is learned, belong to the neighborhood of a query point and create the mini-model area. Sometimes for specified datasets the mini-model area can be extended to the whole system domain, but this situation is very rare and possible only when a modeled data fits to the mini-model in an entire system domain. In contrast the global model is identified on the basis of all available data. The next difference is that global method builds global model which is usually smooth and continuous. It is also possible to build a global model based by fusion of many local mini-models if it is necessary. This can be done by a set of mini-models that were identified for following query points uniformly distributed in the whole system space. The surface build by fusion of mini-models is not smooth and continuous. The differences are illustrated in Fig. 1.

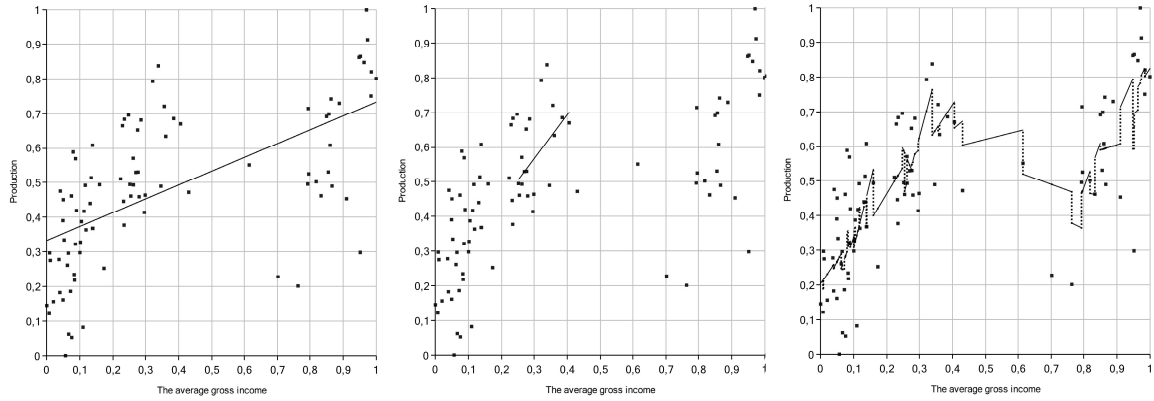


Figure 1. The difference between the linear global model (left), the linear mini-model (center), and the global model built with use of a set of linear mini-models (right)

### 3. Mini-models method

As has been mentioned above the mini-models method consists of two groups of algorithms:

- algorithms for defining local neighborhood of the query point,
- algorithms for mathematical modeling (or classification) on the mini model area.

The mini-model area can be defined as an area of a polytope placed in the general number of dimensions in the input space of the problem. It is a line segment in a 2D-space, polygon in 3D-space (irregular triangle, quadrilateral, etc.), polyhedron in 4D-space (simplex, hexahedron, etc.). In a  $n$ -dimensional space the mini-model area will be an  $n-1$ -dimensional convex polytope. In the initial researches concerning 2D- and 3D-space, the local neighborhood of the query point was created by manipulation of line segment or polygon vertices. The mini-models apart from a good accuracy and other advantages have a lot of drawbacks which reveal when the method is scaled to higher dimensional spaces. The most serious disadvantage is “a curse of dimensionality” because the number of vertices grows exponentially when the number of dimensions is increasing. For example the mini-model area that is based on hypercube has  $2^n$  vertices. Other disadvantages are:

- problem of coplanarity of vertices belonging to the same polytope face,
- problem of manipulating the face as a group of points,
- problem of including or excluding points within the mini-model area.

#### 3.1. Mini-models area embedded in the spherical coordination system

Disadvantages of mini-models described above can be omitted by the face manipulation in the coordinate system based on hyper-sphere [8]. For easy of understanding, the article describes an algorithm for 3D-space problem which uses polar coordination system. For the problem in higher dimensional space the algorithm can be easily extended by replacing polar coordination system for coordination system based on hyper-sphere [9, 10, 11].

The query point  $Q$  will be the pole of a spherical coordinate system. All training points  $p_i$  are transformed into this system and are defined by radius  $r$  (distance from pole) and polar angle  $\varphi$  (measured from polar axis determined by OX axis):

$$\begin{aligned} P &= \{p_1, p_2, \dots, p_n\} \\ p_i &= \{x_{1i}, x_{2i}, y_i\} = \{r_i, \varphi_i, y_i\} \end{aligned} \quad (1)$$

The transformation occur only in the input space. The output variable remains untouched and is used in a process of the mathematical model calculation or in classification process. Face  $F_j$  of the geometric shape is defined by the point  $G_j$  belonging to the face, which is called *face generation point*. The plane which contains polyhedron face  $F_j$  will be called simply *face*. There is an assumption that the plane is orthogonal to the vector  $\overrightarrow{QG_j}$ . The geometric shape has  $J$  faces (for triangle  $J = 3$ , for quadrilateral  $J = 4$ ) and each face is defined as:

$$F_j = \left\{ G_j, p_i : \varphi_{ij} < \frac{\pi}{2} \wedge r_i = \frac{r_j}{\cos \varphi_{ij}} \right\} \quad (2)$$

where:  $\varphi_{ij}$  – is an angles value between vectors  $\overrightarrow{Qp_i}$  and  $\overrightarrow{QG_j}$ . in 2D-space:  $\varphi_{ij} = |\varphi_i - \varphi_j|$ .

The face in fact divides all the space into two half-planes. The first half-plane consists of data points which may be included into the area of mini-model. The set of points which may be included by face  $F_j$  is defined as:

$$I_j = \left\{ p_i : \varphi_{ij} \geq \frac{\pi}{2} \cup \left( \varphi_{ij} < \frac{\pi}{2} \wedge r_i \leq \frac{r_j}{\cos \varphi_{ij}} \right) \right\} \quad (3)$$

The second half-plane consists of data points which are certainly excluded from the mini-model area. The set of points certainly excluded by face  $F_j$  is defined as:

$$E_j = \left\{ p_i : \varphi_{ij} < \frac{\pi}{2} \wedge r_i > \frac{r_j}{\cos \varphi_{ij}} \right\} \quad (4)$$

Every face divides the plane in such way. Intersection of half-planes (which include points) of all faces shows which points are included into the mini-model area. Set of points  $Z$  included in the mini-model area we denoted as:

$$Z = I_1 \cap I_2 \cap \dots \cap I_J \quad (5)$$

How a plane divide 2D-space into two half-planes is described on Fig. 2. Data points marked by triangles are certainly excluded from mini-model area whilst points marked by squares are possibly included into the figure. Whether the point will be included in the mini-model area also depends on its position in relation to other faces. Only points inside the triangle will be included in the mini-model area, other points will be excluded by another faces. On such defined mini-model area, any method of classification or mathematical modeling can be used.

The first part of the process of the mini-model learning (finding optimal mini-model area) is heuristic. It usually involves changing location of points  $G_j$  in random way. Every polytope created by this manipulation is a potential mini-model area. The model which will be taken for further consideration should fulfill initial criteria of the number of samples within specified range. Unfortunately, there is no simple rule for choosing the optimal values of the range for a particular problem. This parameter should be choose empirically. The query point should be in is in the center of the model but not always it is possible. How to choose optimal mini-model area will be described later in this article.

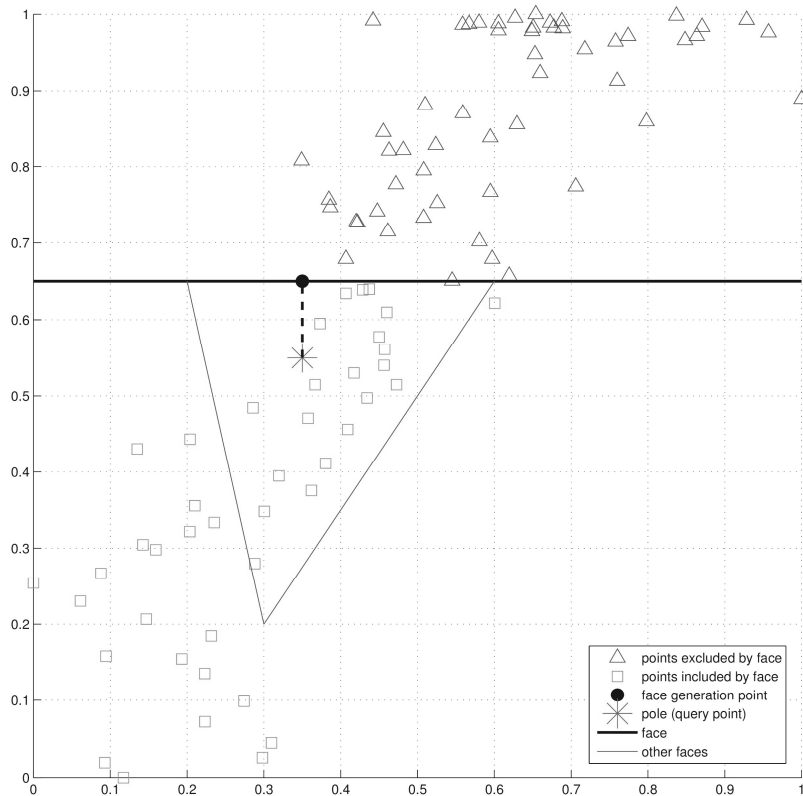


Figure 2. Example of triangle-shaped mini-model area based on polar coordination system in 2D-space

By using algorithm described above manipulation of a face is relatively simple and is reduced to manipulation of two parameters (in 2D-space) which are equivalent to the face generation point coordinates  $(r; \varphi)$ . In description of polytope based on its vertices, manipulation of one face requires translation of all vertices, keeping in mind the fact they have to be coplanar. Thus, manipulation of a single vertex location in a space greater than 2D is impossible. Manipulation by faces frees us from that problem. Inclusion of point within figure can be computed by computing angle between vector created by point  $p_i$  and vector created by face generation point and by using simple trigonometric calculation based on cosine. The most important advantage is that the algorithm could be very easily extended to higher dimensional space, only by adding coordinates to face generation point.

### 3.2. Mini-models in classification task

How it was above-mentioned the first part of the mini-model learning is heuristic. It has such implication that for a particular query point, many local neighborhoods are possible. In the modeling task any method of approximation can be used e.g.: polynomial approximation, mean value, fuzzy reasoning or simple neural network. Mini-model accuracy is measured with use of leave-one-out cross validation method. The method takes one point from original dataset and uses it as a query point and the remaining points as learning data. Then the mini-model calculates numeric answer for the query point, and the error committed by the model. After that, the point is returned to the whole set and another point is taken as a query point. This is repeated for each data point: each point in the original dataset is used once as the query point. After finishing the calculation the mean error committed by the mini-model is calculated. The model which commits the smallest error is chosen as an optimal one.

In classification task, query point is a data sample to be classified. The similar algorithm to the procedure described above is possible, and any method of classification can be used on mini-model area. For the purpose of this article, the very simple variant of the method has been chosen, the method the mini-models method was programmed that it should contains (if it is possible) data samples which belongs to the same class. The quasi-optimal mini-model area is that which is highly homogeneous in the meaning of samples belonging to the same class and the classified sample is in the center of the model but not always it is possible. The model should fulfill initial criteria of the number of samples within specified range.

#### 4. Differences and similarities between $k$ -nearest neighbors and mini-models

The mini-models method in some ways is similar to the method of  $k$ -nearest neighbors [12]. Both algorithms are local data modeling methods and both use data samples from local neighborhood of the query point. The  $k$ NN method is evaluated by many scientists as very effective and some of them are of the opinion that other methods are not necessary [13]. During calculation of an answer for a query point in  $k$ NN only  $k$  nearest samples are taken into account. The base version of the method uses Euclidian metric to identify nearest samples. In the classic  $k$ NN method the model answer is calculated as a mean value of a target function values or a weighted (by distance) mean value. All  $k$  nearest points are encircled in the area of circle in 3D-space problem, sphere in 4D-space problem and in hyper-sphere in general number of dimensions. In contrast in the mini-models method, points are encircled in the area of any polygon in 3D-space problem, polyhedron in 4D-space problem and polytope in general number of dimensions. Number of samples in the mini-models method is not constant but should be placed within specified range. It has such implication that for a particular sample which is modeled or classified, many local neighborhoods are possible, which can be obtained in the learning process. Fig 3. and Fig 4. show the differences between methods.

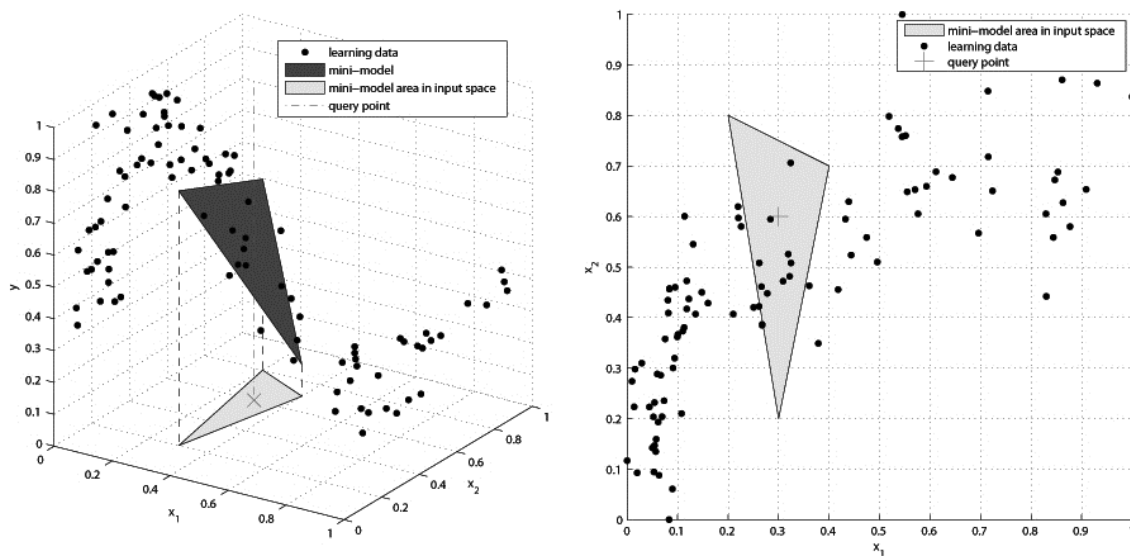


Figure 3. Example of a mini-model in 3D space (left) and the mini-model area in the input space (right)

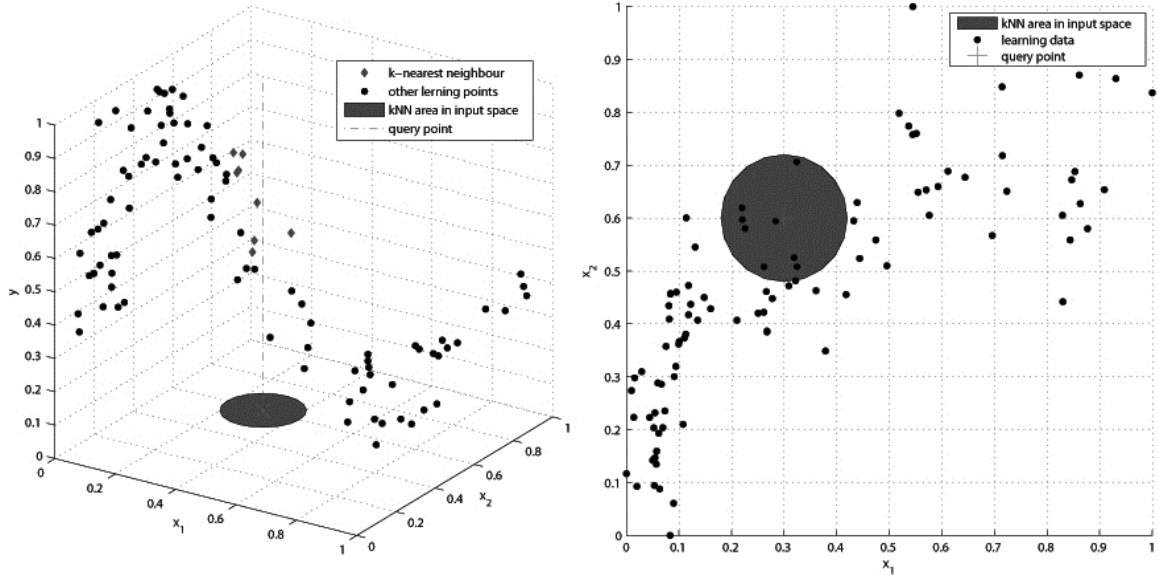


Figure 4. Example of a  $k$ NN method in 3D space (left) and the  $k$ NN area in the input space (right)

## 5. Result of experiments

The paper presents the results of experiments that compare the effectiveness of mini-models with selected methods on the well-known multivariate Iris dataset [14]. The dataset consists of 50 samples from each of three types of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters. One of the class is linearly separable from the other two. The remaining two class have significant overlap. Mini-models were extended to work in 5D-space in order to classify Iris Flower samples. The second part of (1) was changed to:

$$\begin{aligned} p_i &= \{x_{1i}, x_{2i}, x_{3i}, x_{4i}, y_i\} = \{r_i, \varphi_{1i}, \varphi_{2i}, \varphi_{3i}, y_i\} \\ y &\in \{class1, class2, class3\} \end{aligned} \quad (6)$$

where:  $\varphi_{1i}, \varphi_{2i} \in [0; \pi)$  and  $\varphi_{3i} \in [0; 2\pi)$ .

Also an angles value between vectors  $\overrightarrow{Qp_i}$  and  $\overrightarrow{QG_j}$  ( $\varphi_{ij}$ ) used in (2) – (4) needs to be extended to 4D-space. The angles value can be computed using dot product for Cartesian coordinates and after translation to spherical coordinates and simplification the formula has following form.

$$\begin{aligned} \varphi_{ij} &= \arccos(\cos \varphi_{1i} \cos \varphi_{1j} + \sin \varphi_{1i} \sin \varphi_{1j} \cdot \\ &(\cos \varphi_{2i} \cos \varphi_{2j} + \sin \varphi_{2i} \sin \varphi_{2j} \cos(\varphi_{3i} - \varphi_{3j}))) \end{aligned} \quad (7)$$

The result was presented in Table 1.

Table 1. Classification accuracy on the Iris dataset

Algorithm	test	source
mini-models method	97.3	This work
13-NN	96.7	[13]
13-NN weighted by distance	96.7	[13]
13-NN weighted by distance and attribute	96.8	[13]
SVM	97.3	[15]
Naive Bayes	97.3	[16]
NEFCLASS	96.7	[17]

Moreover author performed experiments with mini-models on other two popular benchmark datasets from the UCI Machine Learning repository [18]: Wisconsin Brest Cancer and Diabetes. The results are presents respectively on Table 2 and 3. Experiments were performed with a polytope based on hyper-cube as the mini-models area in the starting position. During further calculation the shape of the polytope became irregular in order to fit classified data. The accuracy was checked with leave-one-out cross validation method. The author performed experiments only with mini-models and presented the other algorithm results as reported either by their authors or by the comparison projects.

Table 2. Classification accuracy on the Wisconsin Brest Cancer dataset

algorithm	test	source
mini-models method	96.8	This work
5-NN	97.4	[13]
5-NN weighted by distance	97.3	[13]
5-NN weighted by distance and attribute	97.6	[13]
C4.5	94.7	[16]
LVQ	96.6	[16]
FDA	96.7	[16]
CART	93.5	[16]
LDA	96.0	[16]
MLP+BP	96.7	[16]
SVM lin, opt C	97.1	[16]

Table 3. Classification accuracy on the Diabetes dataset

algorithm	test	source
mini-models method	75.8	This work
13-NN	76.2	[13]
13-NN weighted by distance	76.6	[13]
13-NN weighted by distance and attribute	77.0	[13]
SVM, Gauss, C, sigma opt	77.4	[16]
RBF	75.7	[16]
LVQ	76.0	[16]
CART	74.7	[16]
C4.5	73.0	[16]
MLP+BP	75.2	[13]



## 6. Conclusion

Experiments were made only on three dataset and the method needs to be tested on a larger number of cases. However in the article the author proofs that the mini-models method which was designed and developed as the method of modeling has a potential good classification accuracy. There are algorithms that performs slightly better, however difference is little. Unfortunately the method has a drawback because the method is partially heuristic and same time results may vary. Authors in future research should move towards improving method in a classification tasks. In the article only the simplest variant of the method was tested. In the next research author plan to check more complex variants of the method e.g. where simple neural network will be used as a method of classification on the mini-model area. Experiments on larger group of datasets are also necessary.

## References

- [1] Piegat A., Wąsikowska B., Korzeń M.: *Application of the self-learning, 3-point mini-model for modelling of unemployment rate in Poland*. Studia Informatica, nr 27, University of Szczecin, 2010, pp. 59-69, [in Polish].
- [2] Piegat A., Wąsikowska B., Korzeń M.: *Differences between the method of mini-models and the k-nearest neighbors an example of modeling unemployment rate in Poland*. Information Systems in management IX-Business Intelligence and Knowledge Management. WULS Press, Warsaw, 2011, pp. 34-43.
- [3] Pietrzykowski M.: *The use of linear and nonlinear mini-models in process of data modelling in a 2D-space*. Nowe Trendy w Naukach Inżynieryjnych. CREATIVETIME, Kraków, 2011, pp. 100-108.
- [4] Pietrzykowski M.: *Comparison of effectiveness of linear mini-models with some methods of modelling*. Młodzi Naukowcy dla Polskiej Nauki. CREATIVETIME, Kraków, 2011, pp. 113-123.
- [5] Pietrzykowski M.: *Effectiveness of mini-models method when data modelling within a 2D-space in an information deficiency situation*. Journal of Theoretical and Applied Computer Science, vol. 6, no. 3, 2012, pp. 21-27.
- [6] Pluciński M.: *Mini-models - Local Regression Models for the Function Approximation Learning*. In Rutkowski L. et. al. Proceedings of ICAISC 2012, Part II, LNCS 7268, Springer-Verlag Berlin Heidelberg, 2012, pp. 160-167.
- [7] Pluciński M.: *Nonlinear ellipsoidal mini-models - application for the function approximation task*. Przegląd Elektrotechniczny (Electrical Review), R. 88 NR 10b/2012, pp. 247 - 251.
- [8] Pietrzykowski M.: *Mini-models working in 3D space based on polar coordinate system*. Nowe trendy w naukach inżynieryjnych 4. Tom II, CREATIVETIME, Kraków 2013, pp. 117-125.
- [9] Bronshtein I., Semendyayev K., Musiol G., Muhlig H.: *Handbook of Mathematics*. Springer, 2007. ISBN 9783540721215.
- [10] Polyanin A., Manzhirov A.: *Handbook of Mathematics for Engineers and Scientists*. Taylor & Francis, 2010. ISBN 9781584885023.
- [11] Moon P., Spencer D.: *Field theory handbook: including coordinate systems, differential equations, and their solutions*. Springer-Verlag, 1988. ISBN 9780387027326.
- [12] Fix E., Hodges J. L., *Discriminatory analysis, nonparametric discrimination: Consistency properties*, Randolph Field, Texas, 1951, pp. 1-21.
- [13] Kordos M., Blachnik M., Strzempa D.: *Do We Need Whatever More than k-NN?*, in Proceedings of 10-th International Conference on Artificial Intelligence and Soft Computing, Zakopane, 2010.
- [14] Fisher R.A.: *The use of multiple measurements in taxonomic problems*. Annals of Eugenics, vol. 7 (2), 1936, pp. 179-188.

- [15] Ben-Hur A., Horn D., Siegelmann H.T., Vapnik V.: *Support vector clustering*. Journal of Machine Learning Research, vol. 2, 2001, pp. 125-137.
- [16] Ster B., Dobnikar A.: *Neural Networks in medical diagnosis: Comparison with other methods*, EANN '96, 1996, pp. 427-430.
- [17] Zarndt F.: *A comprehensive case study: An examination of machine learning and connectionists algorithms*, MSc Thesis, Department of Computer Science, Brigham Young University, 1995.
- [18] UCI Machine Learning Repository: <http://archive.ics.uci.edu/ml/>