ISSN 2278-3091
Volume 9, No.5, September - October 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse299952020.pdf
https://doi.org/10.30534/ijatcse/2020/299952020

GARP Method – an Approach to Increasing the Dataset for the Training of Artificial Neural Networks

Kostadin Yotov[1], Emil Hadzhikolev[2], Stanka Hadzhikoleva[3]
[1] University of Plovdiv "Paisii Hilendarski", Bulgaria, kostadin_yotov@abv.bg
[2] University of Plovdiv "Paisii Hilendarski", Bulgaria, hadjikolev@uni-plovdiv.bg
[3] University of Plovdiv "Paisii Hilendarski", Bulgaria, stankah@uni-plovdiv.bg

ABSTRACT

The use of various AI-based methods for analysis, assessment and prediction is becoming increasingly popular not only in business but also in people's daily lives. Particularly popular are the artificial neural networks that can be used with numerous free software tools and libraries. To perform effective training on a neural network, a sufficient training dataset is needed but it cannot always be provided. Such is the case with an experiment conducted by us for automated assessment of students' knowledge and skills. This motivated us to look for a way to generate additional reliable data for neural network training. The article presents the GARP (Generation of Additional Ray Points) algorithm which creates additional input-output samples based on already existing ones. It can be used in cases where the output samples are expected to be a linear function of the input data.

Key words: neural network training, training dataset generation, GARP algorithm, Generation of Additional Ray Points

1. INTRODUCTION

The search for effective methods for objective, fair and high-quality assessment of students is a goal that led us to the creation of models for multi-criteria assessment [1], [2] based on Bloom's taxonomy [3] and the concepts emerging from it for Higher Order Thinking Skills (HOTS) and Lower Order Thinking Skills (LOTS).

In our experiments for assessing students' knowledge and skills we have defined 4 main assessment components: practical LOTS (pLOTS), practical HOTS (pHOTS), theoretical LOTS (tLOTS), theoretical HOTS (tHOTS). The pHOTS assessment is obtained as a result of solving tasks or implementing practical projects. The remaining grades are formed through solving a test and assessed with points. The limits and values of the components' assessments, as well as the final assessment, use different assessment scales. The specific values in our experiment are presented in table 1.

What is special is that pHOTS is an assessment made by a practice teacher and can have a value – a real number, in the grading scale used. The assessments of the test components are integers, and the final score is also a number of the assessment scale used, but according to the requirements it must be an integer.

Table 1. Assessment components and their assessment scales

Assessment component	Definition	Assessment scale
Theoretical knowledge – tLOTS	x1	An integer in the interval [0, 30]
Practical knowledge – pLOTS	x2	An integer in the interval [0, 15]
Theoretical skills – tHOTS	x3	An integer in the interval [0, 15]
Practical skills – pHOTS	x4	A real number in the interval [2, 6]
Final assessment	finalGrade	An integer in the interval [2, 6]

The task of ***automating the process of assessing learners' knowledge and skills*** is becoming increasingly important when using learning management systems such as Moodle [4]. The formation of the final assessment in a multi-component assessment should not be a function of the total number of points, as most systems offer, instead it should involve a more complex procedure taking into account the different weights of the individual assessment components or a complex IF-THEN logic can be implemented with the tools of fuzzy logic [2], [5]. ***The main problems identified in our research*** are:

- ***The creation of a complex formula with multiple assessment components is not an easy task*** even for pedagogues with experience in this area. Proposing standard formulas is not a solution because the assessment components in the general case may be many, not just 4, as in our case. To integrate

multi-component assessment in an existing learning management system, an additional plug-in would have to be created if the system itself is open source and allows integration of user components.

- ***Deciding on a final integer assessment is not easy in borderline cases*** – when the calculated value is close to an intermediate position, such as 2.5, 3.5, 4.5 and 5.5 at a rating scale of 2 to 6. In such cases, the assessor could assess the student with a grade that may later seem to them unfair. Deviations from a fair value in automated assessment may be greater if the decision rules described by the mathematical model are not complete.

One possible approach to solving these problems is the use of artificial intelligence (AI) algorithms which are trained through information provided by the teacher for real assessments. Moreover, when there are available assessments for the individual assessment components, the assessor makes a relatively fair and objective assessment, according to their subjective view. Without formulas set by the assessor, on the basis of training using the existing data from old assessments only, appropriate artificial intelligence algorithms can make a final assessment based on the assessments assigned to the assessment components. ***When AI algorithms are used,*** there are difficulties, too, for example:

- ***Selection of an appropriate AI algorithm and training method.*** This is a technical problem in the implementation of an application for assessment automation. There are many libraries and systems for working with AI algorithms in which the process of testing different algorithms can be automated [6], [7], [8], [9], [10].
- ***Distrust of the assessor and students to assessments made by a software application***. This problem can also be described as technical. It is a matter of software implementation for an assessment system to require the assessor to confirm the automatically assigned assessments.
- ***Insufficient number of samples for training the AI algorithms***. In our experiments, the data from 130 actual assessment were not sufficient for training and high-quality assessment of AI. As a result, we received an unacceptably high number of final assessments which differed from the assessments made by the assessor.

One possible solution to the latter problem is the ***GARP (Generation of Additional Ray Points) method*** presented in this article ***for algorithmically increasing the number of training samples*** used in the AI algorithms. It is ***applicable in cases where the function for obtaining the final result on multiple input components is assumed to be linear***. In the case of assessment, the method is applicable when the logic by which the assessor makes an assessment is linear, i.e. of the type $L = \sum_{i=1}^{n} w_i\, x_i + c$, where n is the number of assessment components, x_i – the values of the components, w_i are their weights, and c is a constant. Let us remind that this formula is obtained automatically as a result of training an AI algorithm and can remain invisible to the average user.

2. MATHEMATICAL MODEL OF THE GARP METHOD

Let us consider the more general case in which the assessment components are $\boldsymbol{n}$ in number. They can then be presented by n dimensions $x_1, x_2, \dots, x_n$ in an ***n-dimensional Euclidean space of factors*** $\boldsymbol{FS^n \subset R^n}$. An arbitrary ***point*** $\boldsymbol{P_p(x_{1p}, x_{2p}, \dots, x_{np})}$ ***of*** $\boldsymbol{FS^n}$ is presented as a combination of values of the respective factors. We denote ***the space of the input-output samples as*** $\boldsymbol{FSL^{n+1} = (FS^n \times L) \subset R^{n+1}}$, where $L \subset R$ is the set of possible output values. ***A point in*** $\boldsymbol{FSL^{n+1}}$ ***is of type*** $\boldsymbol{N_p(P_p, L_p) = N_p(x_{1p}, x_{2p}, \dots, x_{np}, L_p)}$. Defining a more general class of tasks, we not only solve the specific problem with the small number of input-output samples in the assessment of students, but also for other object areas where there are multiple components on whose characteristics a supposedly linear result function must be applied.

Main problem: Given is a set of input-output samples $\boldsymbol{A_0L_0 \subset FSL^{n+1}}$ ***where*** $\boldsymbol{A_0 \subset FS^n}$, $\boldsymbol{and\ L_0 \subset L}$. ***Let*** $\boldsymbol{f}$ ***be a linear function:***

$$\boldsymbol{f: A_0 \rightarrow L_0.}$$

Create an algorithm that generates many additional samples $\boldsymbol{A_1L_1 \subset FSL^{n+1}}$, $\boldsymbol{A_1 \subset FS^n, and\ L_1 \subset L}$, ***such that with the same function f:***

$$\boldsymbol{f: A_1 \rightarrow L_1.}$$

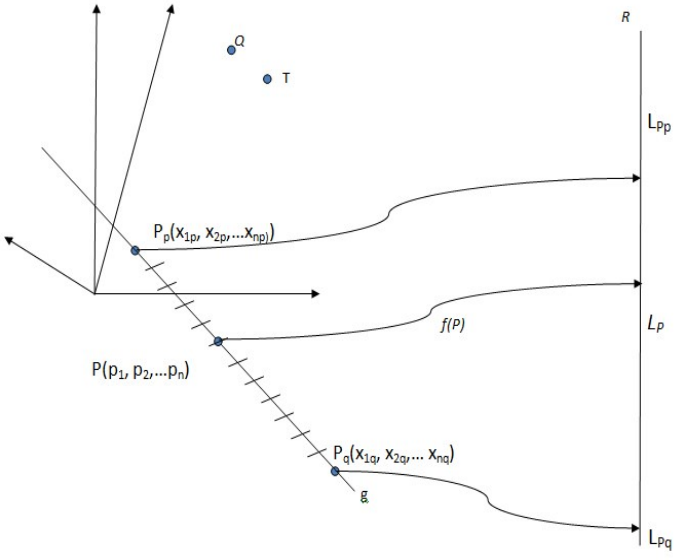

Figure 1. Main elements in the space of input-output samples

Let $P_p(x_{1p}, x_{2p}, \dots, x_{np})$ and $P_q(x_{1q}, x_{2q}, \dots, x_{nq})$ be two arbitrary points, such as $P_p,\ P_q \in \boldsymbol{A_0L_0}$ (fig. 1).
The canonical equation of the straight line defined by them is:

$$(1)\ g: \frac{x_1 - x_{1p}}{x_{1q} - x_{1p}} = \frac{x_2 - x_{2p}}{x_{2q} - x_{2p}} = \cdots = \frac{x_n - x_{np}}{x_{nq} - x_{np}} = const \in R,$$

where $(x_1, x_2, \dots, x_n)$ are the coordinates of an arbitrary point on g.

Then, if the point $P(p_1, p_2, \dots p_n) \in g$ is located between $P_p(x_{1p}, x_{2p}, \dots, x_{np})$ and $P_q(x_{1q}, x_{2q}, \dots, x_{nq})$, then for its coordinates the scalar parametric equations are satisfied:

$$(2)\ p_i = x_{ip} + \lambda(x_{iq} - x_{ip}), \forall\ i = 1,2, \dots n, \lambda \in (0,1)$$

Statement 1.

Let $P_p(x_{1p}, x_{2p}, \dots, x_{np})$ and $P_q(x_{1q}, x_{2q}, \dots, x_{nq})$ are two arbitrary points in the set of factors FS^n, and g is the straight line in R^n, defined by them:

$$g: p_i = x_{ip} + \lambda(x_{iq} - x_{ip}), \forall i = 1, 2, \dots n, \lambda \in R.$$

Let the linear estimation function f have values at the points P_p and P_q, respectively:

$$f(P_p) = L_p, f(P_q) = L_q.$$

Then, every point

$$P(p_1, p_2, \dots p_n) \in g \text{ for which } \lambda \in (0, 1),$$

determines the input-output pattern

$$N_p[P, f(P) = L] \in FSL^{n+1}, \text{ for which } L \in [min(L_p, L_q), max((L_p, L_q)].$$

Proof:

We will present the proof of the case in which $L_p < L_q$.
The case $L_p \geq L_q$ is proved in an analogous way.

If we take the point $O(0,0, \dots 0)$ as the center of the coordinate system in FS^n, then the coordinates of each point in the space of factors are in fact also the coordinates of its radius-vector. Therefore, the assessment function defines an correspondence

$$f: FS^n \to R,$$

where for each point $A(a_1, a_2, \dots a_n) \in FS^n$ is satisfied:

$$(3)\ f(A) = f(\overrightarrow{OA}) = f(a_1, a_2, \dots a_n) = L_a \in R.$$

On the other hand, it follows from the equation of lines that the point $P \in g$, and has a radius-vector:

$$(4)\ \overrightarrow{OP} = \overrightarrow{OP_p} + \lambda\overrightarrow{P_pP_q} = \overrightarrow{OP_p} + \lambda(\overrightarrow{OP_q} - \overrightarrow{OP_p})$$

Thus, from (3), (4) and the fact that the assessment function is linear, follows that:

$$L = f(P) = f(P_p) + \lambda\left(f(P_q) - f(P_p)\right) = L_p + \lambda(L_q - L_p)$$

.

Since $\lambda \in (0,1), and\ (L_q - L_p) > 0$ for $\forall\ \lambda_1 \geq \lambda_2$ it is obvious that:

$$L(\lambda_1) \geq L(\lambda_2),$$

i. e. the function $L(\lambda)\ is\ increasing\ in\ (0,1)$, with functional values

$$L_p \leq L(\lambda) \leq L_q.$$

Statement 1 tells us three important things:

1. ***On each line in R^n, defined by two n-tuple of factors $P_p(x_{1p}, x_{2p}, \dots, x_{np})$ and $P_q(x_{1q}, x_{2q}, \dots, x_{nq})$ we can choose new points for which to calculate corresponding assessments, using the same assessment function.***
2. ***If we form a series of factors on the line, which with respect to the parameter can be considered between P_p and P_q, the assessments of the factors from this series have values between the assessments of P_p and P_q.***
3. ***If $(P_1; L_{P1})$ and $(P_2; L_{P2})$ are two input-output samples $P_1(x_{11}, x_{21}, \dots, x_{n1})$ and $P_2(x_{12}, x_{22}, \dots, x_{n2})$, then the elements of each additional pattern formed by these two points are represented as an ordered pair***

$$(Q_j\ ; L_{Qj}) = (q_{1j}, q_{2j}, \dots q_{nj}\ ;\ L_{Qj}), \text{ where}$$

$$Q_j(q_{1j}, q_{2j}, \dots q_{nj}), q_{ij} = x_{i1} + \lambda_j(x_{i2} - x_{i1}),$$
$$i = 1, 2, \dots n, j = 1, 2, \dots m, \text{ and}$$

$$L_{Qj} = L_{P1} + \lambda_j(L_{P2} - L_{P1}), j = 1, 2, \dots m$$

Given the specific application of GARP, we will set one condition which in practice has axiomatic value: the increase of a student's knowledge must be related to a proportional increase of his / her grade. In terms of the GARP method, this means that for every two points $P_i, P_j \in g$ and their corresponding assessments $L_i, L_j \in R$, the ratio between the lengths (norms) of the vectors $\overrightarrow{P_iP_j}$ and $\overrightarrow{L_iL_j}$ must be a constant:

$$\frac{\|\overrightarrow{P_iP_j}\|}{\|\overrightarrow{L_iL_j}\|} = const$$

But then for every two points $P_i, P_j \in g$, for which it is satisfied:

$$\lim_{P_i \to P_j} \|\overrightarrow{P_iP_j}\| = 0,$$

for corresponding assessments $L_i, L_j \in R$ will be valid:

$$\lim_{P_i \to P_j} \|\overrightarrow{L_iL_j}\| = \lim_{P_i \to P_j} \frac{\|\overrightarrow{P_iP_j}\|}{C} = 0, C = const,$$

i.e. students with infinitely close knowledge receive infinitely similar assessments, namely:

$$(5)\ \lim_{P_i \to P_j} L_i = L_j.$$

Thus, it follows from Statement 1 and Equation (5) that for every two points $P_p(x_{1p}, x_{2p}, \dots, x_{np})$ and $P_q(x_{1q}, x_{2q}, \dots, x_{nq})$, a series of m points can be formed in FS^n

$$(6.1)\ \{P_{pqj}\}_{j=1}^{m} :\ P_{pq1}, P_{pq2}, \dots P_{pqm}$$

with coordinates:

$$(6.2)\ \{(x_{1p} + \delta_{1j}, x_{2p} + \delta_{2j}, \dots, x_{np} + \delta_{nj})\}_{j=1}^{m},$$

in which

$$(6.3)\ \{\delta_{ij}\}_{j=1}^{m} = \left\{\frac{j(x_{iq} - x_{ip})}{m+1}\right\}_{j=1}^{m}, i = 1,2, \dots n.$$

Let's introduce the following notations:

$$(6.4)\ \lambda_j = \frac{j}{m+1}, j = 1,2, \dots m$$

Then, due to the nature of the coordinates of the points in the sequence $\{P_{pqj}\}_{j=1}^{m}$ follows that they belong to the line g, defined by $P_p(x_{1p}, x_{2p}, \dots, x_{np})$ and $P_q(x_{1q}, x_{2q}, \dots, x_{nq})$, and from the fact that

$$\lambda_j \in (0,1), j = 1,2, \dots m,$$

it follows, that the series $\{\delta_{ij}\}_{j=1}^{m}$ satisfies the conditions of Statement 1. Therefore, all n-tuples in $\{P_{pqj}\}_{j=1}^{m}$ have assessments for the same assessment function with a value defined by Statement 1:

$$(7)\ L_{pqj} = L_p + \frac{j(L_q - L_p)}{m+1}, j = 1,2, \dots m, \text{ while}$$

$$min(L_p, L_q) \le L_{pqj} \le max(L_p, L_q), j = 1,2, \dots m.$$

What we have to show is that the intermediate points satisfy the requirements of the axiomatic condition for reciprocal change of the assessment in the corresponding change of knowledge.

Let us consider two points in the series of intermediate points $\{P_{pqj}\}_{j=1}^{m}$. For each two points $P_{pqi}(x_{1p} + \delta_{1i}, x_{2p} + \delta_{2i}, \dots, x_{np} + \delta_{ni})$ and $P_{pqj}(x_{1p} + \delta_{1j}, x_{2p} + \delta_{2j}, \dots, x_{np} + \delta_{nj})$, the length of the vector $\overrightarrow{P_{pqi}P_{pqj}}$, according to the Euclidean metric is:

$$(8)\ \|\overrightarrow{P_{pqi}P_{pqj}}\| = \sqrt{\sum_{k=1}^{n}(\delta_{kj} - \delta_{ki})^2}.$$

From (6.3) it follows that

$$\sum_{k=1}^{n}(\delta_{kj} - \delta_{ki})^2 = \sum_{k=1}^{n}\left(\frac{(j-i)(x_{kq} - x_{kp})}{m+1}\right)^2.$$

Then

$$(8')\ \|\overrightarrow{P_{pqi}P_{pqj}}\| = \frac{(j-i)}{m+1}\sqrt{\sum_{k=1}^{n}(x_{kq} - x_{kp})^2}, \text{ i.e.}$$

$$(8'')\ \|\overrightarrow{P_{pqi}P_{pqj}}\| = \frac{(j-i)}{m+1}\|\overrightarrow{P_pP_q}\|.$$

For the assessments of these two points, taking into account the conclusion made from Statement 1 and the chosen type of λ_j (6.4), follows that:

$$\|\overrightarrow{L_{pqi}L_{pqj}}\| = L_{pqj} - L_{pqi} = L_p + \frac{(j-1)(L_q - L_p)}{m+1} - L_p - \frac{(i-1)(L_q - L_p)}{m+1}$$

$$\text{i.e. } (9)\ \|\overrightarrow{L_{pqi}L_{pqj}}\| = \frac{(j-i)}{m+1}\|\overrightarrow{L_pL_q}\|.$$

Then from $(8'')$ and (9) follows:

$$(10)\ \frac{\|\overrightarrow{P_{pqi}P_{pqj}}\|}{\|\overrightarrow{L_{pqi}L_{pqj}}\|} = \frac{\|\overrightarrow{P_pP_q}\|}{\|\overrightarrow{L_pL_q}\|} = C.$$

In ***summary***, it can be stated that ***from the condition for proportional increase of the grade, corresponding to the increase of the students' knowledge*** $\frac{\|\overrightarrow{P_pP_q}\|}{\|\overrightarrow{L_pL_q}\|} = C$ ***and Statement 1*** it follows that:

By lines in R^n***, defined by two points*** $P_p(x_{1p}, x_{2p}, \dots, x_{np})$ ***and*** $P_q(x_{1q}, x_{2q}, \dots, x_{nq})$***, and their respective assessments*** L_p ***and*** L_q ***obtained by an unknown linear assessment function, we can choose new points*** $\{P_{pqj}\}_{j=1}^{m}$ ***according to formulas*** $(6.1), (6.2), (6.3)$ ***and determine their***

corresponding assessments $\{L_{pqj}\}_{j=1}^{m}$ ***by the formula (7).***

3. GARP ALGORITHM

Based on the described mathematical model, in the MATLAB [10] environment we have implemented the ***GARP algorithm*** through which additional samples are generated on the basis of existing input-output samples. ***The main steps*** in the algorithm are:

1. Input data for the algorithm are identified – a set of s input-output samples of the type $\{N_k(P_k, L_k)\}_{k=1}^{s}$.
2. The set of input-output samples is arranged as follows:
   - The one that is closest to the center of the coordinate system in FS^n is selected for the first element, i.e. the one whose radius-vector has the smallest length $\|\overrightarrow{OP_1}\| = min\{\|\overrightarrow{OP_k}\|\}_{k=1}^{s}$;
   - Each subsequent element in the ordered set is determined by taking the nearest neighbor among the set of unordered elements as the last ordered element.
3. For each two neighbors $P_p(x_{1p}, x_{2p}, \dots, x_{np})$ and $P_q(x_{1q}, x_{2q}, \dots, x_{nq})$ from the ordered series of n-tuple factors, an additional series of new training samples of the type $N_{pqj}(P_{pqj}, L_{pqj})$ is generated using formulas $(6.1), (6.2), (6.3), (7)$.

This algorithm creates a list of adjacent points in FS^n. They define a "broken line" (first-degree spline), on which additional points and their corresponding assessments are generated. Although the points and assessments created in this way are not evenly distributed in space, in many cases they are sufficient to improve the process of neural network training.
To test the effectiveness of the GARP method, two groups of experiments were performed with assessments of 130 students used to form 130 input-output samples.

In the first group of experiments, only the available 130 samples were used. A relatively efficient neural network was obtained with 75 neurons in the hidden layer. A neural network was constructed and trained, which gave error levels of 1e-5. However, when tested with additional examples that are not close to those used in the neural network training, the assessments obtained were inaccurate.

The GARP method was used in the second experiment. Using 50 randomly selected input-output samples, with the help of GARP we generated 7350 additional samples (data sets). The neural network was trained, tested and validated only on those 50 basic and the 7350 input-output samples generated from them. After that, we tested the trained network with the 80 samples not used by GARP.

An efficient network with an error of 1e-7 was obtained when using 100 neurons in the hidden layer. As a result, there were discrepancies between only 2 of the assessments made by the teacher and the neural network. Details on the experiments are presented in [1].

Possible improvements and extensions of the GARP algorithm aimed at a more even compaction of the space of input-output samples FSL^{n+1}, are:

- Generation of secondary samples determined not only by two, but also by more adjacent points;
- Generation of third-level samples between the generated secondary samples;
- Generating samples not only between close neighbors, etc.

The creation of new algorithms for space compaction by input-output samples FSL^{n+1}, research and solution of possible issues are the subject of future research.

4. CONCLUSION

In practice, it is often necessary to conduct training on artificial intelligence systems, where we do not have a sufficient amount of data. In terms of the artificial neural networks theory, we need a sufficient number or input-output samples. Both the human brain and the artificial neuron systems need data to perform a training process, data to determine its validity, as well as accurate data to test their effectiveness. However, when the data we have is insufficient, we will encounter challenges not only with training, but also for validation and testing, which in turn will raise reasonable doubts about the qualities of the modeled neural networks. Using the GARP method, for an unknown function of a linear nature $L_{P_k} = f(P_k), P_k(x_{1k}, x_{2k}, .. x_{nk})$ even with a small sample of basic examples, we were able to create a large number of additional samples for effective training of an artificial neural network approximating the function f.

ACKNOWLEDGEMENT

The work is funded by the SP19-FMI-012 project at the Research Fund of the University of Plovdiv "Paisii Hilendarski".

REFERENCES

1. E. Hadzhikolev, K. Yotov, M. Trankov and S. Hadzhikoleva. **Use of Neural Networks in Assessing Knowledge and Skills of University Students**, *Proceedings of ICERI2019 Conference*, 11th-13th November 2019, Seville, Spain, ISBN: 978-84-09-14755-7, pp. 7474-7484.
2. E. Hadzhikolev, S. Hazdhikoleva, K. Yotov and D. Orozova. **Models for Multicomponent Fuzzy Evaluation, with a Focus on the Assessment of**

Higher-Order Thinking Skills, *Tem Journal* vol.9, No.4, 2020, ISSN: 2217-8309 (in print).

3. B. Bloom, M. Engelhart, E. Furst, et. al., **Taxonomy of educational objectives: The classification of educational goals**. *Handbook I: Cognitive domain*. New York: David McKay Company, 1956.
4. **Learning Management System Moodle**. Retrieved from https://www.moodle.com.
5. M. Vasileva. *Fuzzy sets. Theory and practice*, Shumen, Bulgaria: NVU Vasil Levski, 2008.
6. **Scikit-learn, Machine Learning in Python**. Retrieved from https://scikit-learn.org.
7. **TensorFlow. An end-to-end open source machine learning platform.** Retrieved from https://www.tensorflow.org/.
8. **Keras: the Python deep learning API**. Retrieved from https://keras.io/.
9. **Orange Data Mining.** Retrieved from https://orange.biolab.si/.
10. **Matlab.** Retrieved from https://www.mathworks.com/products/matlab.html.