

Classifying Scaled-Turned-Shifted Objects with Optimal Pixel-to-Scale-Turn-Shift Standard Deviations Ratio in Training 2-Layer Perceptron on Scaled-Turned-Shifted 4800-Featured Objects under Normally Distributed Feature Distortion

Vadim V. Romanuke* (Professor, Polish Naval Academy, Gdynia, Poland)

Abstract – The problem of classifying diversely distorted objects is considered. The classifier is a 2-layer perceptron capable of classifying greater amounts of objects in a unit of time. This is an advantage of the 2-layer perceptron over more complex neural networks like the neocognitron, the convolutional neural network, and the deep learning neural networks. Distortion types are scaling, turning, and shifting. The object model is a monochrome 60×80 image of the enlarged English alphabet capital letter. Consequently, there are 26 classes of 4800-featured objects. Training sets have a parameter, which is the ratio of the pixel-to-scale-turn-shift standard deviations, which allows controlling normally distributed feature distortion. An optimal ratio is found, at which the performance of the 2-layer perceptron is still unsatisfactory. Then, the best classifier is further trained with additional 438 passes of training sets by increasing the training smoothness tenfold. This aids in decreasing the ultimate classification error percentage from 35.23 % down to 12.92 %. However, the expected practicable distortions are smaller, so the percentage corresponding to them becomes just 1.64 %, which means that only one object out of 61 is misclassified. Such a solution scheme is directly applied to other classification problems, where the number of features is a thousand or a few thousands by a few tens of classes.

Keywords – 2-layer perceptron, distortion, pixel-to-scale-turn-shift standard deviations ratio, scaled-turned-shifted objects.

I. THE PROBLEM OF CLASSIFYING SCALED-TURNED-SHIFTED OBJECTS

Watching and controlling diversely distorted objects (DDOs) is a commonplace event for contemporary monitoring and classification systems. For handling the problem of DDO classification, the process of DDO formation should be decomposed to constituents. For instance, an N -dimensional object by $N \in \{1, 2, 3\}$ may occur rotated (turned at a plane or space-solid angle), scaled (linearly and non-linearly) and mirror-like reflected, shifted towards each of the N dimensions. Essentially, scaled-turned-shifted objects (STSOs) may have N dimensions by $N \in \mathbb{N}$, when scaling, turning, and shifting distortions for $N > 3$ are watched only in

the object plane or space-solid projections. Otherwise, scale-turn-shift distortion is comprehensible only mathematically.

In contrary to the slow neocognitron and other complex feedforward neural networks [1], [2], the 2-layer perceptron (2LP) fits for swift STSO classification, which can handle the STSO classification problem if it has been successfully trained on STSOs [3], [4]. If the number of classes $N_{\text{class}} \in \mathbb{N} \setminus \{1\}$ is about 10 or less and the number of object features N_{feat} is not greater than a few hundreds, then the 2LP is trained on STSOs directly in three stages [5], [6]. In the first stage, the 2LP is trained on N_{class} pattern N_{feat} -featured objects, presented in the form of N -dimensional real-valued matrices

$\{\mathbf{B}_c = [b_j^{(c)}]_{\mathcal{L}}\}_{c=1}^{N_{\text{class}}}$ of the format $\mathcal{L} = \prod_{d=1}^N L_d$ and subscript J ,

with L_d features in its d -th dimension, $N_{\text{feat}} = \prod_{d=1}^N L_d$ (for

further consideration, let the object be named or identified with its matrix). In the second stage, the 2LP is trained on a training set (TS)

$$\left\{ \{\bar{\mathbf{B}}\}_{d=1}^R, \{\bar{\mathbf{A}}(p)\}_{p=1}^P \right\} \quad (1)$$

including $R \in \mathbb{N}$ replicas $\bar{\mathbf{B}} = [b_{wc}]_{N_{\text{feat}} \times N_{\text{class}}}$ and $P \in \mathbb{N}$ portions of $N_{\text{feat}} \times N_{\text{class}}$ matrices $\{\bar{\mathbf{A}}(p)\}_{p=1}^P$ of STSOs, where the c -th column of matrix $\bar{\mathbf{B}}$ corresponds to the c -th class pattern object, whose matrix \mathbf{B}_c has been reshaped into a $N_{\text{feat}} \times 1$ matrix-column, and the c -th column of matrix $\bar{\mathbf{A}}(p)$ corresponds to the c -th class STSO. In the third stage, the 2LP is re-trained on the single replica $\bar{\mathbf{B}} = [b_{wc}]_{N_{\text{feat}} \times N_{\text{class}}}$, and eventually the 2LP is ready to swiftly classify these little groups of small-format STSOs. Otherwise, when N_{class} is greater than 10 and N_{feat} is of the order of thousands or

* E-mail: romanukevadimv@gmail.com

higher, then the training of the 2LP is on TS (1) is utterly ineffective: the period of the training process (TP) becomes longer, and the TP converges slower and unreliably [7]. In article [3], a method was shown of how the problem of classifying STSOs could be swiftly solved with a 2LP trained with a modified TP, where the TP period was also prolonged for a small number of epochs and the modified TS was passed through the 2LP for a moderately great number of passes Q .

II. UPDATE OF TS (1) FOR TRAINING ON STSOs

In article [3], a 2LP was initialized and identified for classifying STSOs by the example of 26 classes' general totality of monochrome 60×80 images (M6080Is) by way of training with scaled-turned-shifted monochrome 60×80 images (STSM6080Is), whose matrices had been added to the matrices of the values of a normal variate with zero expectation and unit variance (NVZEUUV) [4], [7]–[9]. Thus STSM6080Is with normally distributed pixel distortion were formed (STSM6080I_{NDFD}), feeding the input of the 2LP as an updated TS. In general, it was proposed to train the 2LP on TS

$$\left\{ \{\bar{\mathbf{B}}\}_{d=1}^R, \{\tilde{\bar{\mathbf{A}}}(p)\}_{p=1}^P \right\} \quad (2)$$

instead of TS (1), where

$$\tilde{\bar{\mathbf{A}}}(p) = \bar{\mathbf{A}}(p) + \lambda(p) \cdot \mathbf{N} \quad \text{at } p = \overline{1, P} \quad (3)$$

at standard deviation (SD)

$$\lambda(p) < \lambda(p+1) \quad \text{for } p = \overline{1, P-1} \quad (4)$$

by a $N_{\text{feat.}} \times N_{\text{class}}$ matrix \mathbf{N} of values of NVZEUUV. In article [3], sequence (4) was taken ordinal:

$$\lambda(p) = \frac{\lambda_{\text{max}}}{P} \cdot p \quad \text{for } p = \overline{1, P} \quad (5)$$

by some $\lambda_{\text{max}} > 0$. However, a maximal SD $\lambda(P)$ or any SD sequence $\{\lambda(p)\}_{p=1}^P$ at all for addition (3) were not assigned strictly on the base of theory. They were assigned empirically, without substantiation, although the gain from applying the updated TS (2) for training the 2LP on STSM6080I_{NDFD} as a model of STSOs with a normally distributed feature distortion (NDFD) was obvious [3], [4], [7], [10], [11].

III. THE GOAL OF SUBSTANTIATION OF MAXIMAL SD $\lambda(P)$ FOR (3) INTO TS (2)

It is certain that the performance of the 2LP in classifying STSM6080Is within article [3] would have been higher if it were trained on STSM6080I_{NDFD} with a substantiated maximal SD $\lambda(P)$ for (3) into TS (2). In general, this will answer the question of how much of NDFD should be imported into matrices $\{\bar{\mathbf{A}}(p)\}_{p=1}^P$ for making TS (2) the most effective to

classify 4800-featured STSOs by $N_{\text{class}} = 26$ with a 2LP. Let the initialization of 2LP be denoted by

$$\mathcal{P}_0(N_{\text{feat.}}, N_{\text{HLS}}, N_{\text{class}}; f_{\text{HLTF}}, f_{\text{OLTF}}) \quad (6)$$

by 2LP hidden layer size (HLS, in neurons) N_{HLS} with hidden layer transfer function f_{HLTF} and output layer transfer function f_{OLTF} . The TP for 2LP (6) has integers R and P for TS (2), which is passed through the 2LP Q times, and along with λ_{max} there are positives η_{max} , μ_{max} and β_{max} , defining the maximal SD for scaling, turning, and shifting, correspondingly. SDs η_{max} , μ_{max} and β_{max} define the DDO extreme, and they have to be preset. SD λ_{max} may vary, defining how much of NDFD should be imported into matrices $\{\bar{\mathbf{A}}(p)\}_{p=1}^P$. Thus there is a pixel-to-scale-turn-shift standard deviations ratio (PSTSSDR), r_{PSTSSD} , which could be assigned as one of the statements:

$$\frac{\lambda_{\text{max}}}{\eta_{\text{max}}}, \frac{\lambda_{\text{max}}}{\mu_{\text{max}}}, \frac{\lambda_{\text{max}}}{\beta_{\text{max}}}.$$

Let the post-training configuration of 2LP (6) be denoted by

$$\mathcal{P}(N_{\text{feat.}}, N_{\text{HLS}}, N_{\text{class}}; f_{\text{HLTF}}, f_{\text{OLTF}}; \eta_{\text{max}}, \mu_{\text{max}}, \beta_{\text{max}}, r_{\text{PSTSSD}}, R, P, Q). \quad (7)$$

2LP (7) performs with a classification errors number (CEN) $q(r_{\text{PSTSSD}})$ over STSOs as STSM6080Is. CEN can be evaluated while 2LP (7) is tested: its input is fed with b full batches (within a batch, every class has one representative) of N_{class} STSOs. Then, the classification error percentage (CEP), which is an indicator of the general 2LP performance, is calculated as follows:

$$p(r_{\text{PSTSSD}}) = \frac{q(r_{\text{PSTSSD}})}{b \cdot N_{\text{class}}} \cdot 100. \quad (8)$$

Hence, substantiation of the maximal SD $\lambda(P)$ for (3) into TS (2) is equivalent to solving the problem

$$r_{\text{PSTSSD}}^* \in \arg \left(\min_{r_{\text{PSTSSD}} \in [\tilde{r}_{\text{PSTSSD}}, \hat{r}_{\text{PSTSSD}}]} \{p(r_{\text{PSTSSD}})\} \right) \quad (9)$$

by the segment $[\tilde{r}_{\text{PSTSSD}}; \hat{r}_{\text{PSTSSD}}]$ containing all possible PSTSSDRs for TS (2) between its lower value $\tilde{r}_{\text{PSTSSD}}$ and the upper one \hat{r}_{PSTSSD} .

IV. GENERAL TOTALITY FOR FORMING TS (2) AND FOR FEEDING THE INPUT OF 2LP (7) WHILE TESTING

An M6080I, which has 4800 features in binary format, is described with its binary 60×80 matrix. The matrix space, where every element is a binary 60×80 matrix, is finite, containing altogether 2^{4800} elements. This space

$$G_{2^{4800}} = \left\{ \left\{ \mathbf{B}_c \right\}_{c=1}^{26}, \left\{ \tilde{\mathbf{B}}_m \right\}_{m=1}^{2^{4800}-26} \right\} \quad (10)$$

contains 26 pattern 4800-featured M6080Is $\left\{ \mathbf{B}_c \right\}_{c=1}^{26}$ depicting enlarged English alphabet capital letters (EEACLs) [3], [5], [12], [13], and the remaining $2^{4800} - 26$ matrices $\left\{ \tilde{\mathbf{B}}_m \right\}_{m=1}^{2^{4800}-26}$ with elements $\{0, 1\}$. And STSOs as STSM6080Is are among matrices $\left\{ \tilde{\mathbf{B}}_m \right\}_{m=1}^{2^{4800}-26}$, so (10) is a general totality whose elements will feed the input of 2LP (7) while it is tested. While 2LP (6) is trained, its input is fed with samples from general totality

$$E = G_{2^{4800}} \cup \tilde{G} \quad (11)$$

by infinite set \tilde{G} of 60×80 matrices $\tilde{\mathbf{A}}(p)$, where

$$\tilde{\mathbf{A}}(p) = \mathbf{A}(p) + \lambda(p) \cdot \mathbf{N}_1 \quad \text{at } p = \overline{1, P} \quad (12)$$

with $\mathbf{A}(p) \in G_{2^{4800}}$ and \mathbf{N}_1 , which is a matrix of the values of the above-mentioned NVZEUUV. In particular, general totality (11) contains STSM6080I_{NDP} for TS (2).

As a generalization, the space

$$G_{2^{N_{\text{feat}}}} = \left\{ \left\{ \mathbf{B}_c \right\}_{c=1}^{N_{\text{class}}}, \left\{ \tilde{\mathbf{B}}_m \right\}_{m=1}^{2^{N_{\text{feat}}} - N_{\text{class}}} \right\} \quad (13)$$

contains N_{class} pattern N_{feat} -featured objects $\left\{ \mathbf{B}_c \right\}_{c=1}^{N_{\text{class}}}$ and the remaining $2^{N_{\text{feat}}} - N_{\text{class}}$ matrices $\left\{ \tilde{\mathbf{B}}_m \right\}_{m=1}^{2^{N_{\text{feat}}} - N_{\text{class}}}$ with elements $\{0, 1\}$ by STSOs or DDOs among those $2^{N_{\text{feat}}} - N_{\text{class}}$ matrices $\left\{ \tilde{\mathbf{B}}_m \right\}_{m=1}^{2^{N_{\text{feat}}} - N_{\text{class}}}$ within general totality (13), whose elements are used for feeding the input of 2LP (7) while testing. Lastly, the space

$$E = G_{2^{N_{\text{feat}}}} \cup \tilde{G} \quad (14)$$

by infinite set \tilde{G} of matrices $\tilde{\mathbf{A}}(p)$ of the format

$$\mathcal{L} = \bigotimes_{d=1}^N L_d, \quad \text{where there is the operation of the type (12)}$$

with $\mathbf{A}(p) \in G_{2^{N_{\text{feat}}}}$, contains STSOs or DDOs with NDFD. Samples from general totality (14) feed the input of the 2LP while it is trained.

V. DESCRIPTION OF 2LP (6) CONFIGURATION FOR M6080IS AND A MATLAB FUNCTION FOR TRAINING ON TS (2)

The MATLAB environment possesses any means necessary to build and simulate the 2LP, regulating its parameters, properties or attributes far more operatively than when coded in C++ or C# [7], [12], [14], [15]. Before training on STSM6080I_{NDP} from TS (2), 2LP (6)

$$\mathcal{P}_0(4800, N_{\text{HLS}}, 26; f_{\text{HLTF}}, f_{\text{OLTF}}) \quad (15)$$

is initialized with MATLAB function “feedforwardnet” from the Neural Network Toolbox [4], [9], [13]. Integer N_{HLN} for 26 pattern 4800-featured M6080Is must be a matter of hundreds, so let it be $N_{\text{HLN}} = 300$. Transfer functions in 2LP (15) must squash its inputs $(-\infty; \infty)$ to the segment $[0; 1]$. Such a transfer within the Neural Network Toolbox is implemented with logarithmic sigmoid MATLAB transfer function “logsig” \mathcal{S} , which has the derivative due to its smooth “S-shape” [7], [16], [17]. Then 2LP (15)

$$\mathcal{P}_0(4800, 300, 26; \mathcal{S}, \mathcal{S}), \quad (16)$$

having been adapted with weight and bias learning rules by Neural Network Toolbox adapt function “adaptwb”, is going to be trained with MATLAB backpropagation training function “traingda” (see its usage in [3], [4], [7], [9], [12], [18], [19]). Function “traingda” will update

$$4800 \cdot 300 + 300 + 26 \cdot 300 + 26 = 1\,448\,126$$

weight and bias values of 2LP (16) according to the gradient descent with the adaptive learning rate [20], [21]. Let the usefulness of 2LP (16) during the training be measured with its performance function “mse” according to the sum of squared errors [8], [9], [22], [23]. Finally, having preset the minimum performance gradient to 10^{-6} , let the number of epochs be 5000 in order to prevent long-dragging convergence of TP and to shorten the ultimate TP period for each pass.

VI. MODELS OF STSM6080I AND STSM6080I_{NDP}

Although the process of forming DDOs is not dissociable, it should be divided into phases for mathematically stating each aspect of the complex distortion. In the case of STSOs, there are three phases in the following succession of completion: scaling, turning, shifting [3], [7].

During the first phase, the c -th class pattern 4800-featured M6080I $\mathbf{B}_c = \left[b_{uv}^{(c)} \right]_{60 \times 80}$ is scaled with the scaling map σ into a c -th class EEACL as $V \times H$ matrix $\mathbf{Z}_c(p)$ by SD $\eta(p)$ and scale coefficient $\zeta(\eta(p))$, where

$$\begin{aligned} \sigma[\mathbf{B}_c, \zeta(\eta(p))] &= \mathbf{Z}_c(p), \quad \eta(p) = \frac{\eta_{\text{max}}}{P} \cdot p, \\ \zeta(\eta(p)) &= \eta(p)\xi(p) + 1 \quad \text{at } p = \overline{1, P} \end{aligned} \quad (17)$$

with the value $\xi(p)$ of NVZEUUV $\Xi(p)$, and the map σ in (17) is supported with MATLAB function “imresize” [7], [24], [25]. If it occurs that $\zeta(\eta(p)) \leq 0$ then NVZEUUV $\Xi(p)$ is re-raffled until $\zeta(\eta(p)) > 0$. M6080I \mathbf{B}_c is enlarged $\zeta(\eta(p))$ times through (17) if $\zeta(\eta(p)) > 1$, and \mathbf{B}_c is reduced $\frac{1}{\zeta(\eta(p))}$ times through (17) if $\zeta(\eta(p)) < 1$. M6080I \mathbf{B}_c remains unchanged if $\zeta(\eta(p)) = 1$.

During the second phase, the scaled M6080I as $V \times H$ matrix $\mathbf{Z}_c(p)$ is turned by an angle $\rho(p)$ in degrees by SD $\mu(p)$ to the turned-scaled M6080I as $V \times H$ matrix $\mathbf{T}_c(p)$, where

$$\begin{aligned} \mathbf{T}_c(p) &= 1 - \tau [1 - \mathbf{Z}_c(p), \rho(p)], \\ \mu(p) &= \frac{\mu_{\max}}{P} \cdot p, \\ \rho(p) &= \frac{180}{\pi} \cdot \mu(p) \theta(p) \quad \text{at } p = \overline{1, P} \end{aligned} \quad (18)$$

$$\begin{aligned} I &= \left\{ \overline{\{1, N_V\}}, \overline{\{61 + N_V, V\}} \right\}, \quad J = \left\{ \overline{\{1, N_H\}}, \overline{\{81 + N_H, H\}} \right\}, \\ N_V &= \Omega\left(\frac{V}{2}\right) - 30 + \left(\frac{1 + \text{sign} \psi_V}{2} \cdot \text{sign} |\psi_V|\right) \cdot \text{sign} \left[\frac{V}{2} - \Omega\left(\frac{V}{2}\right) \right], \\ N_H &= \Omega\left(\frac{H}{2}\right) - 40 + \left(\frac{1 + \text{sign} \psi_H}{2} \cdot \text{sign} |\psi_H|\right) \cdot \text{sign} \left[\frac{H}{2} - \Omega\left(\frac{H}{2}\right) \right], \end{aligned}$$

and $\Omega(\alpha)$ is a function, returning the integer part of number α , calculated by values $\{\psi_V, \psi_H\}$ of two independent NVZEUVs. These NVZEUVs are raffled every time when function $\Omega(\alpha)$ is applied. If $\zeta(\eta(p)) < 1$ then $\mathbf{T}_c(p)$ is padded from the left for N_{left} columns of ones and from the

$$\begin{aligned} N_{\text{left}} &= \Omega\left(\frac{80 - H}{2}\right) + \left(\frac{1 + \text{sign} \psi_H}{2} \cdot \text{sign} |\psi_H|\right) \cdot \text{sign} \left[\frac{H}{2} - \Omega\left(\frac{H}{2}\right) \right], \quad N_{\text{right}} = 80 - H - N_{\text{left}}, \\ N_{\text{top}} &= \Omega\left(\frac{60 - V}{2}\right) + \left(\frac{1 + \text{sign} \psi_V}{2} \cdot \text{sign} |\psi_V|\right) \cdot \text{sign} \left[\frac{V}{2} - \Omega\left(\frac{V}{2}\right) \right], \quad N_{\text{bottom}} = 60 - V - N_{\text{top}}. \end{aligned}$$

The c -th class representative $\mathbf{R}_c(p) = [r_{uv}^{(c)}(p)]_{60 \times 80}$ is shifted in two subphases [3], [4], [7], [12]. It is horizontally

$$\begin{aligned} X[\beta(p)] &= \varphi(8\beta(p) \cdot \gamma_1(p)) \cdot \frac{1 - \text{sign}(|\varphi(8\beta(p) \cdot \gamma_1(p))| - 80)}{2} + 80 \cdot \frac{1 + \text{sign}(|\varphi(8\beta(p) \cdot \gamma_1(p))| - 80)}{2}, \\ Y[\beta(p)] &= \varphi(6\beta(p) \cdot \gamma_2(p)) \cdot \frac{1 - \text{sign}(|\varphi(6\beta(p) \cdot \gamma_2(p))| - 60)}{2} + 60 \cdot \frac{1 + \text{sign}(|\varphi(6\beta(p) \cdot \gamma_2(p))| - 60)}{2}, \end{aligned}$$

by SD

$$\beta(p) = \frac{\beta_{\max}}{P} \cdot p \quad \text{for } p = \overline{1, P} \quad (19)$$

and function $\varphi(\alpha)$, which rounds α to the nearest integer less than or equal to α , calculated by values $\{\gamma_1(p), \gamma_2(p)\}$ of two independent NVZEUVs. After the horizontal shift and before the vertical one, matrix $\mathbf{R}_c(p) = [r_{uv}^{(c)}(p)]_{60 \times 80}$ is changed into matrix $\mathbf{S}_c(p) = [s_{uv}^{(c)}(p)]_{60 \times 80}$. For $X[\beta(p)] > 0$ the elements of $\mathbf{S}_c(p)$ are

with the value $\theta(p)$ of NVZEUV $\Theta(p)$. Map τ in (18), supported with MATLAB function “imrotate”, turns the input negative $1 - \mathbf{Z}_c(p)$ in a counterclockwise direction if $\rho(p) > 0$, and for $\rho(p) < 0$ it is turned clockwise, while for $\rho(p) = 0$ the matrix $\mathbf{Z}_c(p)$ remains unchanged [13], [26]. Before shifting, $\mathbf{T}_c(p)$ is re-formatted into an element $\mathbf{R}_c(p) \in G_{2^{4800}}$, representing a c -th class EEACL. If $\zeta(\eta(p)) > 1$ then the lines with numbers I and columns with numbers J in $\mathbf{T}_c(p)$ are discarded, where [7]

right for N_{right} columns of ones [3], [7], [24], and it is padded from the top for N_{top} lines of ones and from the bottom for N_{bottom} lines of ones for contouring the turned-scaled image $\mathbf{T}_c(p)$ with the background white colour [7], where

shifted by $X[\beta(p)]$ pixels and vertically shifted by $Y[\beta(p)]$ pixels, where

$$\begin{aligned} s_{uv}^{(c)}(p) &= 1 \quad \text{for } v = \overline{1, X[\beta(p)]} \quad \text{and } s_{uv}^{(c)}(p) = r_{uv}^{(c)}(p) \\ &\quad \text{at } t = v - X[\beta(p)] \quad \text{for } v = \overline{X[\beta(p)] + 1, 80} \quad \forall u = \overline{1, 60}. \end{aligned}$$

For $X[\beta(p)] < 0$ these elements are

$$\begin{aligned} s_{uv}^{(c)}(p) &= r_{uv}^{(c)}(p) \quad \text{at } t = v - X[\beta(p)] \quad \text{for } v = \overline{1, 80 + X[\beta(p)]} \\ &\quad \text{and } s_{uv}^{(c)}(p) = 1 \quad \text{for } v = \overline{81 + X[\beta(p)], 80} \quad \forall u = \overline{1, 60}. \end{aligned}$$

For $X[\beta(p)] = 0$ the c -th class turned-scaled EEACL

$\mathbf{R}_c(p)$ is not shifted horizontally:

$$s_{uv}^{(c)}(p) = r_{uv}^{(c)}(p) \quad \forall u = \overline{1, 60} \text{ and } \forall v = \overline{1, 80}.$$

The second subphase in shifting the plane M6080I is the vertical shifting, where matrix $\mathbf{S}_c(p) = [s_{uv}^{(c)}(p)]_{60 \times 80}$ is changed into matrix $\mathbf{A}_c(p) = [a_{uv}^{(c)}(p)]_{60 \times 80}$. For $Y[\beta(p)] > 0$

$$a_{uv}^{(c)}(p) = s_{rv}^{(c)}(p) \text{ at } r = u + Y[\beta(p)] \text{ for } u = \overline{1, 60 - Y[\beta(p)]}$$

$$\text{and } a_{uv}^{(c)}(p) = 1 \text{ for } u = \overline{61 - Y[\beta(p)], 60} \quad \forall v = \overline{1, 80}.$$

For $Y[\beta(p)] < 0$

$$a_{uv}^{(c)}(p) = 1 \text{ for } u = \overline{1, -Y[\beta(p)]} \text{ and } a_{uv}^{(c)}(p) = s_{rv}^{(c)}(p)$$

$$\text{at } r = u + Y[\beta(p)] \text{ for } u = \overline{-Y[\beta(p)] + 1, 60} \quad \forall v = \overline{1, 80}.$$

For $Y[\beta(p)] = 0$ the c -th class turned-scaled-and-shifted-horizontally EEACL $S_c(p)$ is not shifted vertically:

$$a_{uv}^{(c)}(p) = s_{uv}^{(c)}(p) \quad \forall u = \overline{1, 60} \text{ and } \forall v = \overline{1, 80}.$$

Hereby the third phase of STSO formation as STSM6080I is completed.

Thereupon the three phases of STSO completion, each full portion of 60×80 matrices $\{\mathbf{A}_c(p)\}_{c=1}^{26}$ is reshaped into 4800×26 matrix $\bar{\mathbf{A}}(p)$ for $p = \overline{1, P}$. Eventually, a p -th portion of the full batch of STSM6080I_{NDFD} for TS (2) is formed due to (3) by ordinal SD sequence (5).

VII. TRAINING ON TS (2) AND TESTING THE TRAINED 2LP (7) ON STSM6080IS FROM GENERAL TOTALITY (10) TO EVALUATE FUNCTION (8)

The initialized and configured 2LP (16) must be trained under parameters

$$\{\eta_{\max}, \mu_{\max}, \beta_{\max}, r_{\text{PSTSSD}}, R, P, Q\}$$

to define TS (2) and to configure the TP. Here, only PSTSSDR is changeable, and the remaining six parameters are to be preset. Based on experience [3], [7], let these parameters be

$$\eta_{\max} = 0.2, \quad \mu_{\max} = 0.2, \quad \beta_{\max} = 1,$$

$$R = 2, \quad P = 8, \quad Q = 220. \quad (20)$$

Also for the sake of definiteness, let PSTSSDR be assigned via SD of the most severe distortion in terms of influence:

$$r_{\text{PSTSSD}} = \frac{\lambda_{\max}}{\beta_{\max}} = \lambda_{\max}. \quad (21)$$

And since for $\lambda_{\max} \notin [0.01; 1]$ NDFD is either deficient or

surplus, then $[\tilde{r}_{\text{PSTSSD}}; \hat{r}_{\text{PSTSSD}}] = [0.01; 1]$.

Consequently, 2LP (7)

$$\mathcal{P}(4800, 300, 26; \mathcal{S}, \mathcal{S}; 0.2, 0.2, 1, r_{\text{PSTSSD}}, 2, 8, 220) \quad (22)$$

must be tested for $r_{\text{PSTSSD}} \in [0.01; 1]$ to evaluate function (8), which will allow solving the problem (9)

$$r_{\text{PSTSSD}}^* \in \arg \left(\min_{r_{\text{PSTSSD}} \in [0.01; 1]} \{p(r_{\text{PSTSSD}})\} \right). \quad (23)$$

While 2LP (22) is tested, its input is fed with $b = 400$ full batches of STSM6080Is, formed by some SDs

$$\eta \in [0; \eta_{\max}] = [0; 0.2], \quad \mu \in [0; \mu_{\max}] = [0; 0.2],$$

$$\beta \in [0; \beta_{\max}] = [0; 1]. \quad (24)$$

SDs in (24) will be changed by increasing them simultaneously, generating CEN $q(r_{\text{PSTSSD}}, \eta, \mu, \beta)$. Letting the shift SD be leading again [3], [7], the averaged CEN for (8) is

$$q(r_{\text{PSTSSD}}) = \int_0^1 q(r_{\text{PSTSSD}}, \eta, \mu, \beta) d\beta \approx$$

$$\approx \frac{1}{M+1} \sum_{j=0}^M q \left(r_{\text{PSTSSD}}, \frac{j}{5M}, \frac{j}{5M}, \frac{j}{M} \right) \quad (25)$$

for $(M+1)$ -pointed-equidistantly subsets

$$\left\{ \frac{j}{5M} \right\}_{j=0}^M \subset [0; 0.2] = [0; \eta_{\max}],$$

$$\left\{ \frac{j}{5M} \right\}_{j=0}^M \subset [0; 0.2] = [0; \mu_{\max}],$$

$$\left\{ \frac{j}{M} \right\}_{j=0}^M \subset [0; 1] = [0; \beta_{\max}].$$

For them, it is sufficient to take $M = 10$ and with (25) CEP (8) is

$$p(r_{\text{PSTSSD}}) = \frac{1}{1144} \cdot \sum_{j=0}^{10} q(r_{\text{PSTSSD}}, 0.02j, 0.02j, 0.2j). \quad (26)$$

Indisputably, for 2LP training, 2LP testing, and eventually solving the problem (23), the segment of PSTSSDR $[0.01; 1]$ must be sampled. A successive sweep of 2LP (22) on the segment of PSTSSDR $[0.01; 1]$ may be fulfilled on a subset

$$\left\{ \{0.01 + 0.01i\}_{i=0}^9, \{0.2 + 0.1i\}_{i=0}^8 \right\} \subset [\tilde{r}_{\text{PSTSSD}}; \hat{r}_{\text{PSTSSD}}] =$$

$$= [0.01; 1]. \quad (27)$$

The purpose of sweeping 2LP (22) through 19 points in (27) is to narrow the segment of PSTSSDR, where the solution of problem (23) would be enclosed.

VIII. SOLUTION AND VERIFICATION OF PROBLEM (23)

For an initial statistical evaluation, each point of function (8) is the average of 40 2LPs. Figure 1 displays that, in the first approximation with $\{0.01, 0.1\}$ -discretization of abscissa axis subset by (27), there is a view of the problem (23) solution that shall be enclosed within the subsegment $[0.01; 0.1] \subset [0.01; 1]$. For more accurate re-evaluation, let each point of function (8) be the average of 100 2LPs. Re-evaluating the function (8) on the 0.01-sampled subsegment

$$\{0.01 + 0.01i\}_{i=0}^9 \subset [0.01; 0.1] \quad (28)$$

shows (Fig. 2) that point $r_{\text{PSTSSD}}^* = 0.08$ is a 0.01-solution of problem (23), providing 0.01-minimal CEP

$$p(r_{\text{PSTSSD}}^*) = p(0.08) \approx 10.32 \quad (29)$$

for the performance of 2LP

$$\mathcal{P}(4800, 300, 26; \mathcal{E}, \mathcal{E}; 0.2, 0.2, 1, 0.08, 2, 8, 220). \quad (30)$$

Here a δ -optimal solution of problem (23) and a δ -minimal CEP imply δ -discretization of the abscissa axis. However, the difference between

$$p(0.06) \approx 10.2833$$

in Fig. 1 and CEP (29) is close to the statistical accuracy [4], [27] of the evaluation of function (8) (see Fig. 3 and Fig. 4). This is why there is no sense in trying to search for a δ -optimal solution at some $\delta \in (0; 0.01)$.

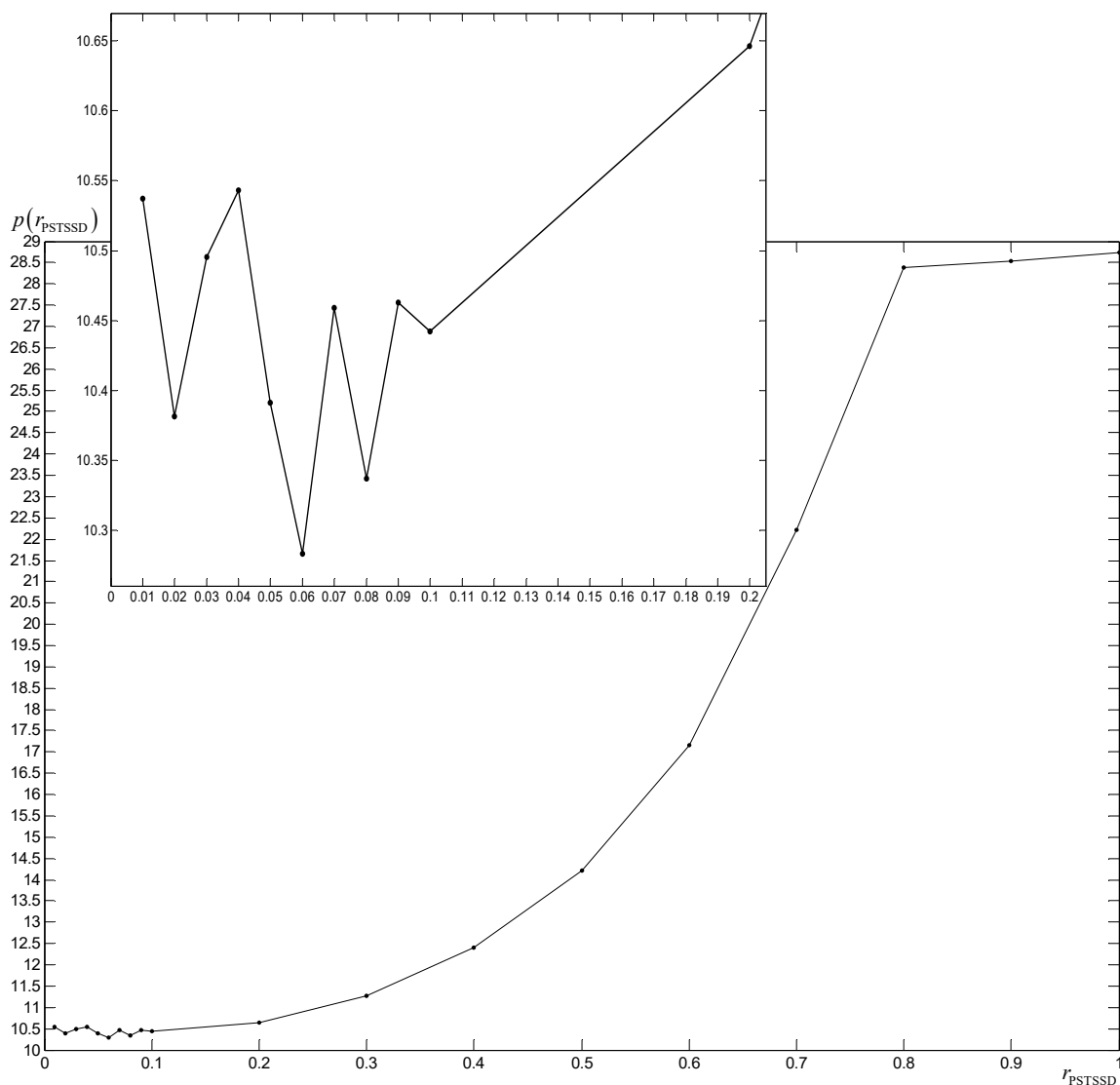


Fig. 1. An evaluation of function (8) for 2LP (22) on the 19-pointed subset (27). Each point is the average of 40 2LPs. A view of solving problem (23) is seen.

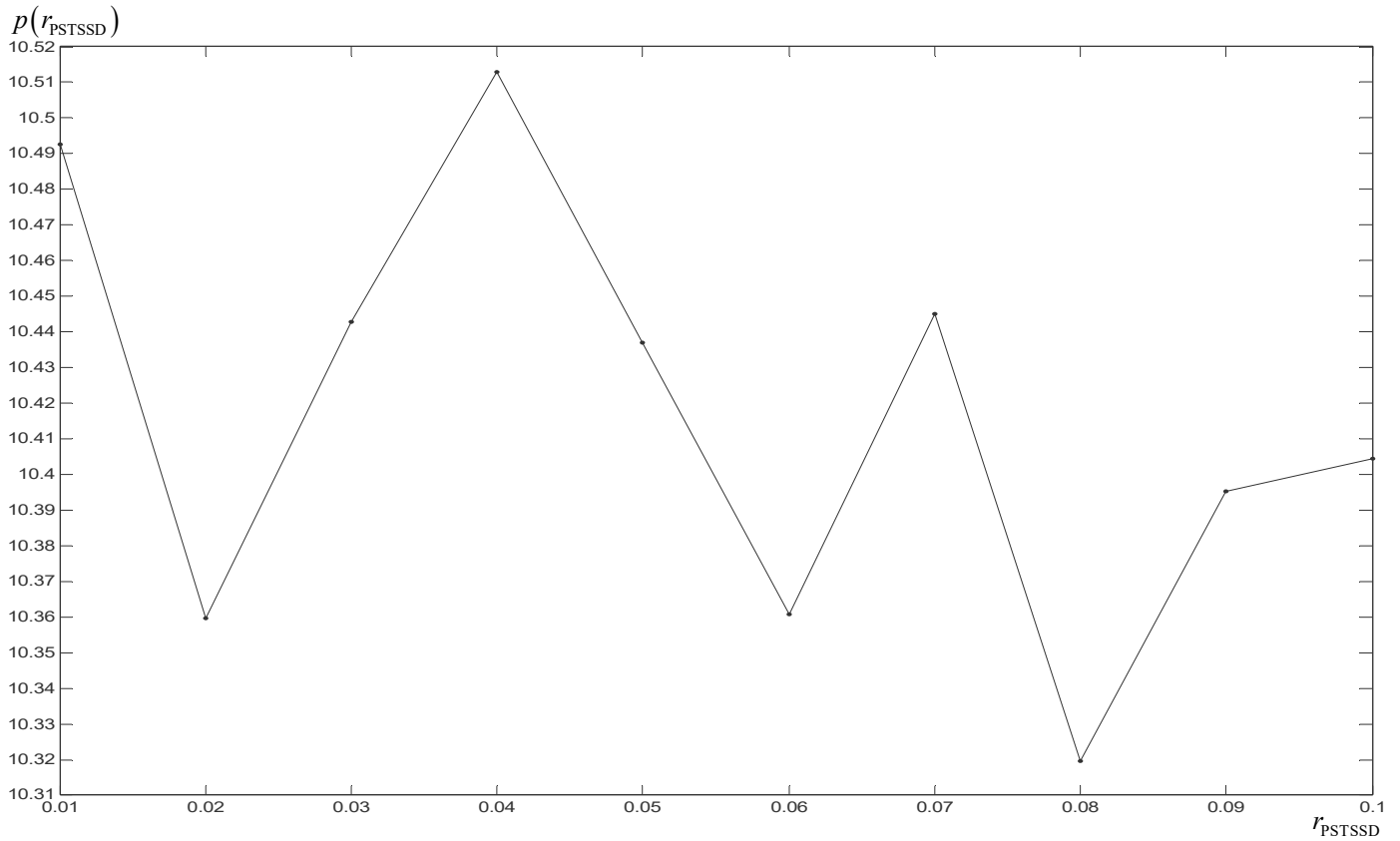


Fig. 2. A 2.5-times multiple re-evaluation of function (8) for 2LP (22) on the 10-pointed subset (28). A solution is seen clearer, but contrary to the view in Fig. 1.

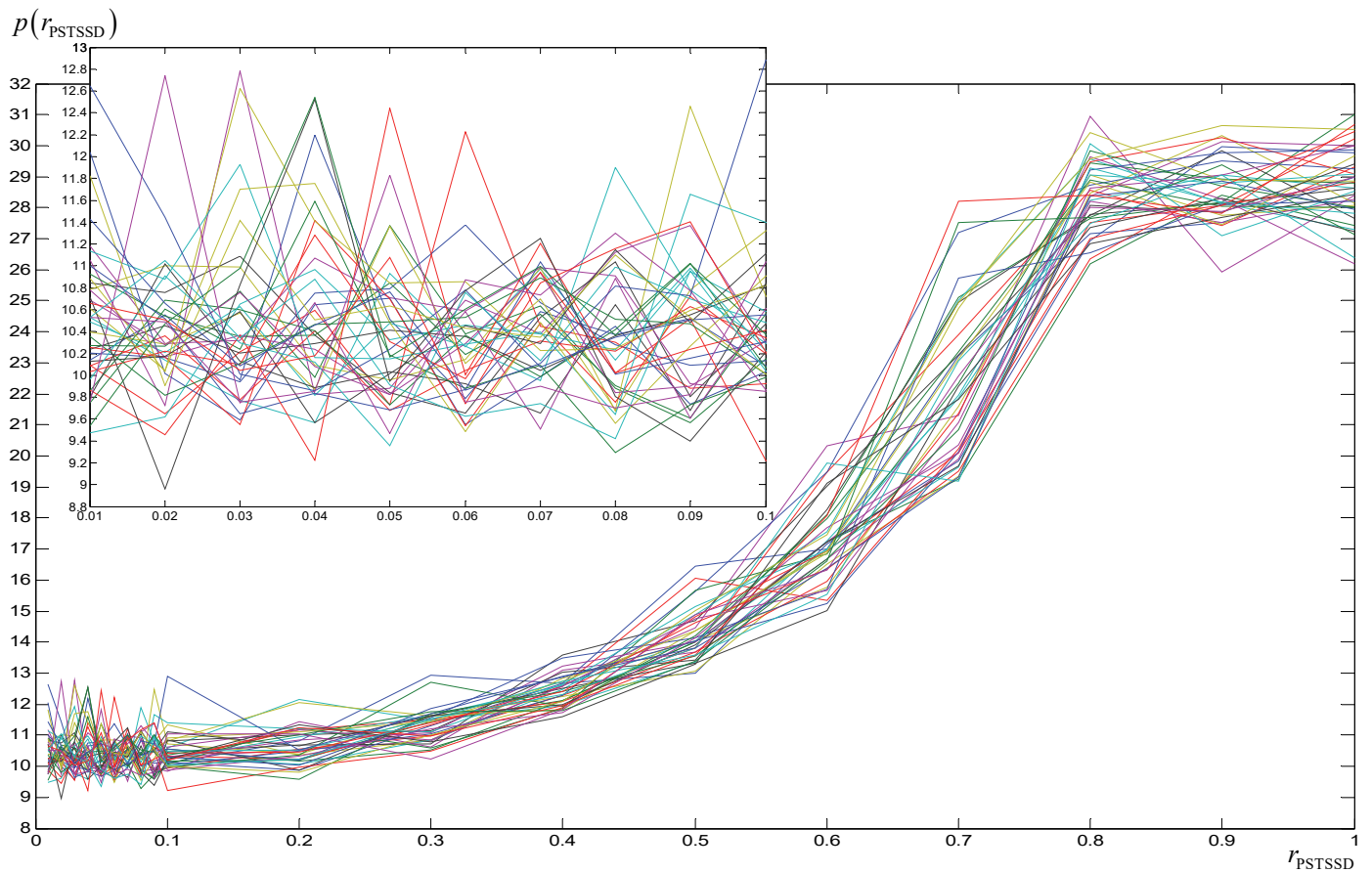


Fig. 3. A bunch of 40 functions (8) for 2LP (22) on the 19-pointed subset (27) showing how CEP is badly scattered (that implies the statistical accuracy is about the magnitude of the CEP scattering, at each PSTSSDR point). The scattering is worse at the top PSTSSDR points, and it appears the least about the solution.

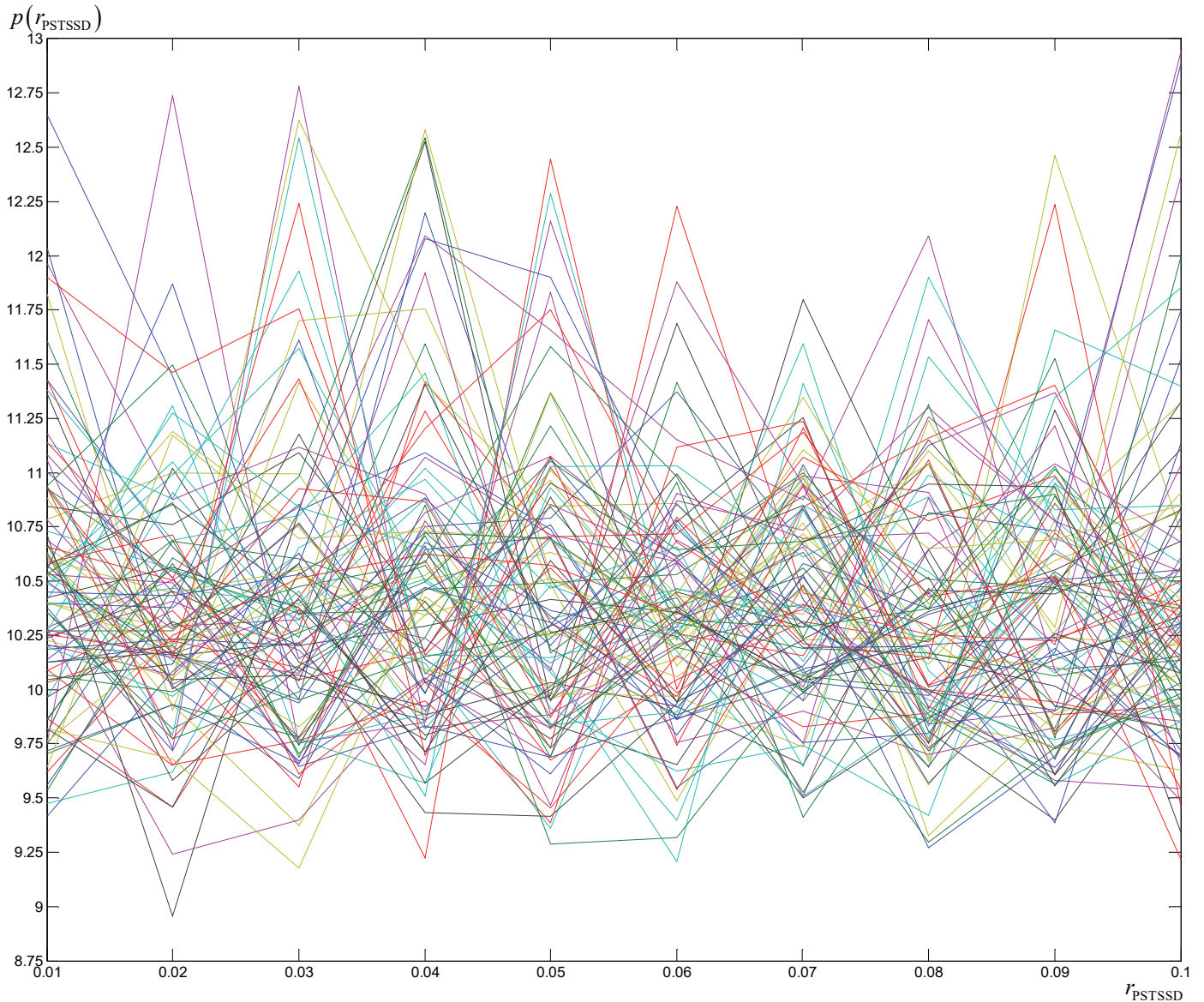


Fig. 4. A bunch of 100 functions (8) for 2LP (22) on the 10-pointed subset (28) showing how CEP is badly scattered (that implies statistical accuracy).

It is worth noting that, in the polylines of Fig. 3, CEP is scattered less by

$$r_{\text{PSTSSD}} \in \{0.06, 0.07, 0.08\}.$$

The scattering is least at point $r_{\text{PSTSSD}} = 0.07$, which appears the most stable PSTSSDR in Fig. 4 as well. However, yielding to $r_{\text{PSTSSD}} = 0.08$ on average (see Fig. 2), point $r_{\text{PSTSSD}} = 0.07$ cannot pretend to be another solution. PSTSSDR $r_{\text{PSTSSD}} = 0.06$ loses out because there are more CEPs scattered up.

Before verification, the best one of 100 2LPs (30) must be trained further until its performance becomes unimprovable [5], [8], [28]. The best 2LP (30) performs at an average CEP, which is 9.31 %. However, at the maximal distortions, when

$$\eta = \eta_{\text{max}} = 0.2, \mu = \mu_{\text{max}} = 0.2, \beta = \beta_{\text{max}} = 1, \quad (31)$$

CEP is 35.23 %. This maximal-SD CEP is decreased to

12.92 % via additional 438 passes of TS (2) for the best 2LP (30) by increasing the smoothness of the distortions with $P = 80$ (see article [29] for how TP is smoothed for improving the performance of 2LP).

A disclosure of the best CEP across the increasing distortions tied to shifting denoted by $p_{0.08}^*(\beta)$ is seen in Fig. 5. The distortions by SDs (31) appear ultimate, so the expected practicable SDs are smaller. They are about up to $\beta = 0.7$ by the corresponding $\eta = 0.14$ and $\mu = 0.14$, where CEP is 1.64 %. Such a CEP is tolerable.

Figure 6 visualizes STSOs by maximal SDs in (31), which form STSM6080Is of EEACLs successfully classified (the left-side images) with the best 2LP (30) further-trained [4], [28] with the 438 additional passes (438AP further-trained 2LP). Successful classification under lesser SDs is shown below. Those STSM6080Is classified incorrectly appear very much distorted (the right-side images) even under twice as small SDs. The most influential distortion is the shifting [12].

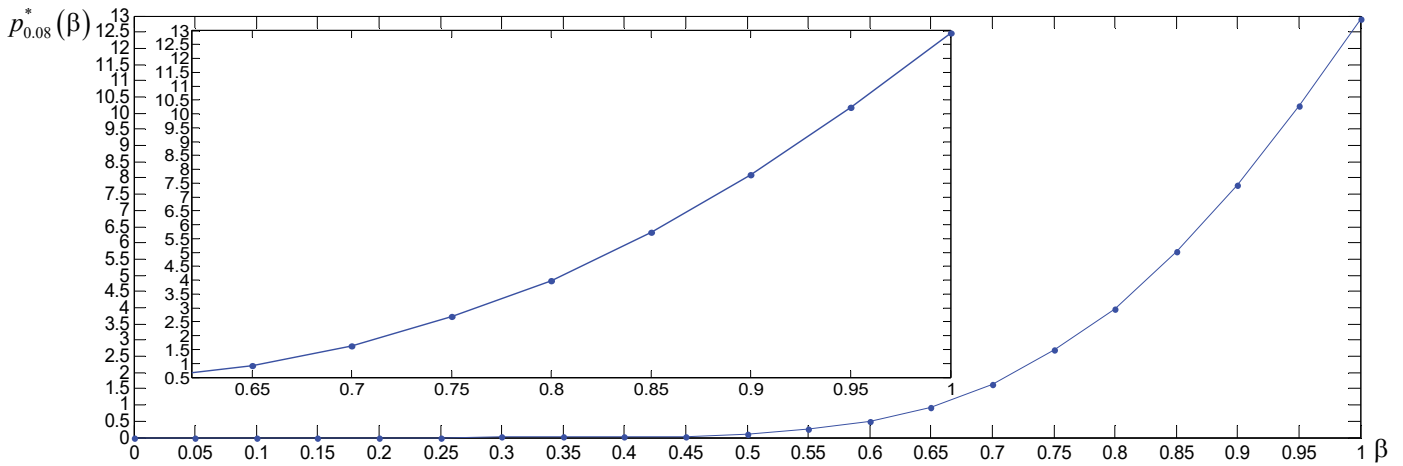


Fig. 5. A disclosure of the best CEP across the increasing distortions (here, it is shifting). At expected practicable SDs, only one object of 61 ones is misclassified.

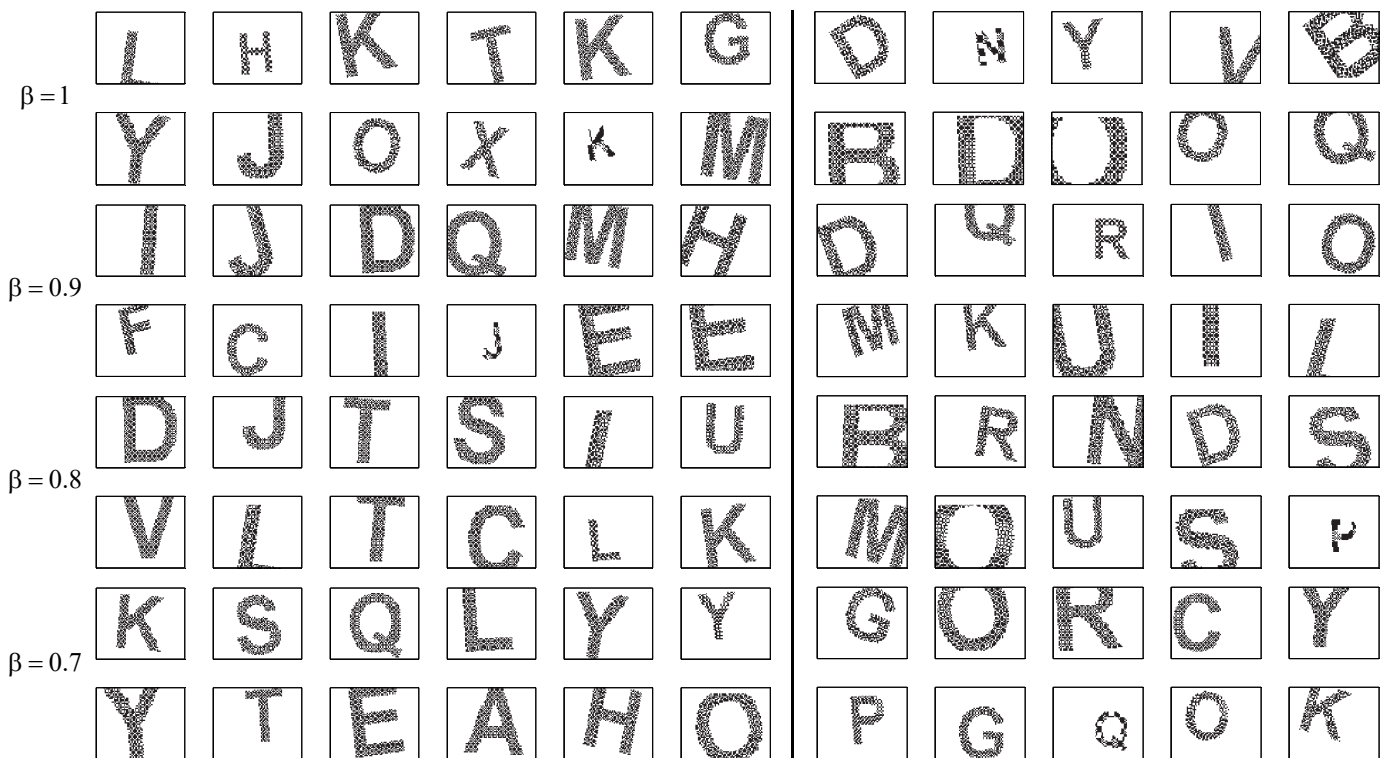


Fig. 6. STSM6080Is of EEACLs classified incorrectly (the right-side images) and classified successfully (the left-side images) by the best 438AP further-trained 2LP. The misclassified images appear so much distorted that the classifier cannot properly “see” them. Seemingly, bad shifting has the worst influence on the classifier.

The 438AP further-trained 2LP can be further-trained more, with probably a greater P . This is the core of the verification of the solution of problem (23), which allows obtaining a higher performance classifier from an optimal PSTSSDR point. Nevertheless, the increment of P has its limit, after which the performance is not improvable [29].

IX. SOLUTION PROPAGATION AND CONCLUSION

An optimal PSTSSDR in training 2LP on STSOs under NDFD provides a prepared classifier whose performance is improved by further-training. STSM6080I_{NDFD} in TP along with searching for the solution of problem (23) is an example of how to achieve high-performance classification of STSOs (DDOs). This is quite implementable owing to the fact that

NDFD effaces some distinguishers of DDOs, which allows applying 2LP to accurately classify DDOs. The solution of problem (23) is to be used for other classification problems, where the number of features is a thousand or a few thousands by a few tens of classes [7], [12], [28], [30]. For DDOs that are visualized, an optimal 2LP will be effective for classifying simpler images, rather than recognizing scenes. However, DDOs need not necessarily be distorted images. They can be series of medical, technical, social data, where the number of features is a matter of hundreds. For such problems, the effectiveness of the 2LP is of the same priority than that of more complex neural networks (like the neocognitron, the convolutional neural network, and the deep learning neural networks). This is explained by the fact that roughly the same

CEP (only one object out of 61 is misclassified) is reached, although the heavier and swift 2LP stands out against the lighter and slower deep learners [7], [31].

REFERENCES

- [1] K. Fukushima, "Artificial vision by multi-layered neural networks: Neocognitron and its advances," *Neural Networks*, vol. 37, pp. 103–119, Jan. 2013. <https://doi.org/10.1016/j.neunet.2012.09.016>
- [2] S. Kim, Y. Choi, and M. Lee, "Deep learning with support vector data description," *Neurocomputing*, vol. 165, pp. 111–117, Oct. 2015. <https://doi.org/10.1016/j.neucom.2014.09.086>
- [3] V. V. Romanuke, "Two-layer perceptron for classifying scaled-turned-shifted objects by 26 classes general totality of monochrome 60-by-80-images via training with pixel-distorted scaled-turned-shifted images," *Information processing systems*, iss. 7 (132), pp. 98–107, 2015.
- [4] V. V. Romanuke, "Optimal Pixel-to-Shift Standard Deviation Ratio for Training 2-Layer Perceptron on Shifted 60 × 80 Images with Pixel Distortion in Classifying Shifting-Distorted Objects," *Applied Computer Systems*, vol. 19, no. 1, Jan. 2016. <https://doi.org/10.1515/accs-2016-0008>
- [5] V. V. Romanuke, "Boosting ensembles of heavy two-layer perceptrons for increasing classification accuracy in recognizing shifted-turned-scaled flat images with binary features," *Journal of Information and Organizational Sciences*, vol. 39, no. 1, pp. 75–84, 2015.
- [6] V. V. Romanuke, "Accuracy improvement in wear state discontinuous tracking model regarding statistical data inaccuracies and shifts with boosting mini-ensemble of two-layer perceptrons," *Problems of tribology*, no. 4, pp. 55–58, 2014.
- [7] V. V. Romanuke, "Two-layer perceptron for classifying flat scaled-turned-shifted objects by additional feature distortions in training," *Journal of Uncertain Systems*, vol. 9, no. 4, pp. 286–305, 2015.
- [8] K. Hagiwara, T. Hayasaka, N. Toda, S. Usui, and K. Kuno, "Upper bound of the expected training error of neural network regression for a Gaussian noise sequence," *Neural Networks*, vol. 14, no. 10, pp. 1419–1429, Dec. 2001. [https://doi.org/10.1016/s0893-6080\(01\)00122-8](https://doi.org/10.1016/s0893-6080(01)00122-8)
- [9] V. Romanuke, "Setting the Hidden Layer Neuron Number in Feedforward Neural Network for an Image Recognition Problem under Gaussian Noise of Distortion," *Computer and Information Science*, vol. 6, no. 2, Mar. 2013. <https://doi.org/10.5539/cis.v6n2p38>
- [10] C.-H. Yoo, S.-W. Kim, J.-Y. Jung, and S.-J. Ko, "High-dimensional feature extraction using bit-plane decomposition of local binary patterns for robust face recognition," *Journal of Visual Communication and Image Representation*, vol. 45, pp. 11–19, May 2017. <https://doi.org/10.1016/j.jvcir.2017.02.009>
- [11] C. Zhu and Y. Peng, "Discriminative latent semantic feature learning for pedestrian detection," *Neurocomputing*, vol. 238, pp. 126–138, May 2017. <https://doi.org/10.1016/j.neucom.2017.01.043>
- [12] V. V. Romanuke, "An attempt for 2-layer perceptron high performance in classifying shifted monochrome 60-by-80-images via training with pixel-distorted shifted images on the pattern of 26 alphabet letters," *Radio Electronics, Computer Science, Control*, no. 2, pp. 112–118, 2013.
- [13] V. V. Romanuke, "A 2-layer perceptron performance improvement in classifying 26 turned monochrome 60-by-80-images via training with pixel-distorted turned images," *Research Bulletin of the National Technical University of Ukraine "Kyiv Polytechnic Institute"*, no. 5, pp. 55–62, 2014.
- [14] A. Y. Alanis, J. D. Rios, J. Rivera, N. Arana-Daniel, and C. Lopez-Franco, "Real-time discrete neural control applied to a Linear Induction Motor," *Neurocomputing*, vol. 164, pp. 240–251, Sep. 2015. <https://doi.org/10.1016/j.neucom.2015.02.065>
- [15] A. B. Asghar and X. Liu, "Estimation of wind turbine power coefficient by adaptive neuro-fuzzy methodology," *Neurocomputing*, vol. 238, pp. 227–233, May 2017. <https://doi.org/10.1016/j.neucom.2017.01.058>
- [16] D. Costarelli and R. Spigler, "Approximation results for neural network operators activated by sigmoidal functions," *Neural Networks*, vol. 44, pp. 101–106, Aug. 2013. <https://doi.org/10.1016/j.neunet.2013.03.015>
- [17] Z. Chen, F. Cao, and J. Hu, "Approximation by network operators with logistic activation functions," *Applied Mathematics and Computation*, vol. 256, pp. 565–571, Apr. 2015. <https://doi.org/10.1016/j.amc.2015.01.049>
- [18] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 4, pp. 525–533, Jan. 1993. [https://doi.org/10.1016/s0893-6080\(05\)80056-5](https://doi.org/10.1016/s0893-6080(05)80056-5)
- [19] T. Kathirvalavakumar and S. Jeyaseeli Subavathi, "Neighborhood based modified backpropagation algorithm using adaptive learning parameters for training feedforward neural networks," *Neurocomputing*, vol. 72, no. 16–18, pp. 3915–3921, Oct. 2009. <https://doi.org/10.1016/j.neucom.2009.04.010>
- [20] A. Nied, S. I. Seleme, G. G. Parma, and B. R. Menezes, "On-line neural training algorithm with sliding mode control and adaptive learning rate," *Neurocomputing*, vol. 70, no. 16–18, pp. 2687–2691, Oct. 2007. <https://doi.org/10.1016/j.neucom.2006.07.019>
- [21] S. J. Yoo, J. B. Park, and Y. H. Choi, "Indirect adaptive control of nonlinear dynamic systems using self recurrent wavelet neural networks via adaptive learning rates," *Information Sciences*, vol. 177, no. 15, pp. 3074–3098, Aug. 2007. <https://doi.org/10.1016/j.ins.2007.02.009>
- [22] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networks—a review," *Pattern Recognition*, vol. 35, no. 10, pp. 2279–2301, Oct. 2002. [https://doi.org/10.1016/s0031-3203\(01\)00178-9](https://doi.org/10.1016/s0031-3203(01)00178-9)
- [23] C. Yu, M. T. Manry, J. Li, and P. Lakshmi Narasimha, "An efficient hidden layer training method for the multilayer perceptron," *Neurocomputing*, vol. 70, no. 1–3, pp. 525–535, Dec. 2006. <https://doi.org/10.1016/j.neucom.2005.11.008>
- [24] V. V. Romanuke, "Pixel-to-scale standard deviations ratio optimization for two-layer perceptron training on pixel-distorted scaled 60-by-80-images in scaled objects classification problem," *Visnyk of Kremenchuk National University of Mykhaylo Ostrogradskyy*, iss. 2 (85), pp. 96–105, 2014.
- [25] V. V. Romanuke, "Classification error percentage decrement of two-layer perceptron for classifying scaled objects on the pattern of monochrome 60-by-80-images of 26 alphabet letters by training with pixel-distorted scaled images," *Scientific bulletin of Chernivtsi National University of Yuriy Fedkovich. Series: Computer systems and components*, vol. 4, iss. 3, pp. 53–64, 2013.
- [26] V. V. Romanuke, "Optimal hidden layer neurons number in two-layer perceptron and pixel-to-turn standard deviations ratio for its training on pixel-distorted turned 60 × 80 images in turned objects classification problem," *Visnyk of Kremenchuk National University of Mykhaylo Ostrogradskyy*, iss. 5 (94), pp. 86–93, 2015.
- [27] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, *Probability & Statistics for Engineers & Scientists (9th ed.)*. Boston, Massachusetts: Prentice Hall, 2012.
- [28] V. V. Romanuke, "Optimal Training Parameters and Hidden Layer Neuron Number of Two-Layer Perceptron for Generalised Scaled Object Classification Problem," *Information Technology and Management Science*, vol. 18, no. 1, Jan. 2015. <https://doi.org/10.1515/itms-2015-0007>
- [29] V. V. Romanuke, "Dependence of performance of feed-forward neuronet with single hidden layer of neurons against its training smoothness on noised replicas of pattern alphabet," *Herald of Khmelnytskyi national university. Technical sciences*, no. 1, pp. 201–206, 2013.
- [30] M. J. Kochenderfer, C. Amato, G. Chowdhary, J. P. How, H. J. Davison Reynolds, J. R. Thornton, P. A. Torres-Carrasquillo, N. K. Üre, and J. Vian, *Decision Making Under Uncertainty: Theory and Application*. Cambridge, Massachusetts, London, England: The MIT Press, 2015.
- [31] H. R. Tavakoli, A. Borji, J. Laaksonen, and E. Rahtu, "Exploiting inter-image similarity and ensemble of extreme learners for fixation prediction using deep features," *Neurocomputing*, vol. 244, pp. 10–18, Jun. 2017. <https://doi.org/10.1016/j.neucom.2017.03.018>

Vadim V. Romanuke was born in 1979. The higher education was received in 2001. In 2006, he obtained the degree of Candidate of Technical Sciences in Mathematical Modelling and Computational Methods. His candidate dissertation suggested a way of increasing the interference noise immunity of data transferred over radio systems. Mr. Romanuke obtained his degree of Doctor of Technical Sciences in Mathematical Modelling and Computational Methods in 2014. His Doctor-of-Science dissertation solved the problem of increasing the efficiency of the identification of models for multistage technical control and run-in under multivariate uncertainties of their parameters and relationships. In 2016, he received the status of Full Professor. Mr. Romanuke is a Professor at the Faculty of Navigation and Naval Weapons at the Polish Naval Academy. His research interests concern decision-making, game theory, statistical approximation, and control engineering based on statistical correspondence. Vadim Romanuke has good programming skills in MATLAB. For practical implementations, Mr. Romanuke uses Python. Also, he directs a branch of fitting statistical approximators at the Centre of Parallel Computations managed by Khmelnytskyi National University (Ukraine). Address for correspondence: 69 Śmidowicza Street, Gdynia, Poland, 81–127.