IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 1, 2021, accepted March 15, 2021, date of publication March 23, 2021, date of current version April 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3068127

Neural Networks Regularization With Graph-Based Local Resampling

ALEX D. ASSIS[1,2], LUIZ C. B. TORRES[3], LOURENÇO R. G. ARAÚJO[2], VÍTOR M. HANRIOT[4], AND ANTONIO P. BRAGA[4]

[1]Department of Economics, Universidade Federal de Juiz de Fora (UFJF), Governador Valadares 35010-180, Brazil
[2]Graduate Program in Electrical Engineering, Universidade Federal de Minas Gerais (UFMG), Belo Horizonte 31270-901, Brazil
[3]Department of Computing and Systems, Universidade Federal de Ouro Preto (UFOP), João Monlevade 35931-008, Brazil
[4] Department of Electronics Engineering, Universidade Federal de Minas Gerais (UFMG), Belo Horizonte 31270-901, Brazil

Corresponding author: Alex D. Assis (assis.alex.d@gmail.com)

This work was supported in part by the National Research Council (CNPq), in part by the Coordination for Higher Education Staff Development (CAPES), in part by the Research Support Foundation of the State of Minas Gerais (FAPEMIG), and in part by the Pró-Reitoria de Pesquisa (PRPq) da Universidade Federal de Minas Gerais (UFMG) and Universidade Federal de Juiz de Fora (UFJF).

ABSTRACT This paper presents the concept of Graph-based Local Resampling of perceptron-like neural networks with random projections (RN-ELM) which aims at regularization of the yielded model. The addition of synthetic noise to the learning set finds some similarity with data augmentation approaches that are currently adopted in many deep learning strategies. With the graph-based approach, however, it is possible to direct resample in the margin region instead of exhaustively cover the whole input space. The goal is to train neural networks with added noise in the margin region, located by structural information extracted from a planar graph. The so-called structural vectors, which are the training set vertices near the class boundary, are obtained from the structural information using Gabriel Graph. Synthetic samples are added to the learning set around the geometric vectors, improving generalization performance. A mathematical formulation that shows that the addition of synthetic samples has the same effect as the Tikhonov regularization is presented. Friedman and pos-hoc Nemenyi tests indicate that outcomes from the proposed method are statistically equivalent to the ones obtained by objective-function regularization, implying that both methods yield smoother solutions, reducing the effects of overfitting.

INDEX TERMS Classifier, neural network, regularization, training with noise.

I. INTRODUCTION

Many efforts have been made in the last decades in order to represent the learning problem of Single Hidden Layer Feed-forward Networks (SLFN) [1]–[3] with convex formulations and to avoid the burden of iterative gradient descent learning on complex objective functions. Higher dimensional random projections to the hidden layer is an approach to convexification [4]. Random projection methods are based on Cover's Theorem principles [5] to assure that the projected data results on a linear problem that can be treated under a convex optimization perspective. Such an approach became popular in recent years under the framework of Extreme Learning Machine (ELM) [6], [7], [8], [9] a two-layer perceptron with large random expansions in the hidden layer.

The associate editor coordinating the review of this manuscript and approving it for publication was Kok-Lim Alvin Yau.

A large number of hidden neurons may, however, lead to neural networks with a far higher capacity than the required to solve the problem [10]. The network then becomes overspecialized on the training samples, which may result on overfitting, and lose its generalization ability. Regularization methods aim at solving the overfitting problem by smoothing the separation surface, thus leading to an improved performance on unknown data. A regularized surface can be achieved by combining the objective function with a penalization term, as shown in (1) [11].

$$\tilde{E} = E + \lambda\Omega \tag{1}$$

where E is the training set error, λ is a regularization parameter and Ω is a model complexity penalty function.

Functions E and Ω have conflicting behavior [12], so trading them off by selecting a proper value of λ is essential to achieve a more general model. Regularization methods

have been proposed and applied to control the smoothness of the approximating function in many application problems. A common choice for the smoothness function is the norm of the weight vectors [13] as presented in (2), known as Tikhonov Regularization, L2 penalty and Ridge Regression [14].

$$J = \frac{1}{2}\sum_{i=1}^{N}(\mathbf{y}_i - f(\mathbf{x}_i, \mathbf{w}))^2 + \lambda \sum_{j=1}^{L} \mathbf{w}_j^2 \qquad (2)$$

where N is the number of samples of the training set and L is the number of hidden layer neurons of the network.

Since ELMs tend to be oversized, works in the literature apply regularization to control the smoothness of the resulting approximation function. Deng [15] proposed the Regularized Extreme Learning Machine (ELM-REG), which implements (2). The Optimally Pruned ELM proposed by Miche [16] achieves regularization by employing multiresponse sparse regression and leave-one-out cross validation to remove the least relevant neurons. The Tikhonov Regularized OP-ELM [17], combines L1 and L2 penalties in order to generate a regularized separation surface. The $L_{2,1}$-norm based Online Sequential Extreme Learning Machine proposed by Preeti [18] creates an iterative bi-objective optimization algorithm to solve $L_{2,1}$ norm-based minimization problem, dealing with real time sequential data. These approaches to regularization require that λ is provided prior to training.

Silvestre *et al.* [19] proposed a method for parameter-free regularization of extreme learning machines, that uses only an affinity matrix obtained from the training samples, which leads to the same Tikhonov Regularization effect. Araujo [20] proposed a method for automated parameter selection, based on the linear separability of projected data, that requires neither user defined parameter nor cross validation. Both approaches rely on the quality and representability of training samples in order to be effective.

Another regularization approach that focuses neither on the dataset structure nor on pre-setting hyperparameters was given by Bishop [21], which consists of adding noise to the training set and is shown to be equivalent to Tikhonov Regularization. Training with noise has been applied to the training of deep neural networks, under the framework of data augmentation and dropout regularization [14], [22] and also as noise injection in hidden units to yield stochastic behavior that exploits a probabilistic formulation for optimization [23]. Furthermore, this approach has also been used for generative adversarial networks (GAN) applied to machinery fault diagnosis, so as to add synthetic samples based on the distribution of the original samples, avoiding training with imbalanced data [24]. The most widely used data augmentation strategy for deep neural networks consists of randomly applying operations such as rotations, cropping and mirroring to images. While leading to improvements in performance, this strategy has been further explored with more specific approaches: dataset augmentation in feature space is explored by DeVries and Taylor [25], a multiphase blending method is proposed by Quan *et al.* [26] and Lemley *et al.* [27] use a GAN to generate samples with features that improve performance. Adding local noise to border regions can also be viewed as analogous to boosting, a robust machine learning approach that combines multiple weak classifiers and assigns higher relevance to those patterns located near the border regions [28]. Boosting has already been applied to ELMs learning, especially in problems involving imbalanced data [29].

Resampling over an unconstrained input space can be prohibitive, particularly in higher dimensions which is mostly the case of current applications. So, local resampling in the margin region may yield regularization effects without the need to exhaustly cover the whole input space. The margin region can be identified by considering the geometry of the dataset given by proximity graphs such as the Gabriel Graph (GG) [30]. Torres *et al.* [31] proposed a geometric approach that uses GG to build a large margin classifier based on the edges between points of different classes, defining a boundary region. These edges can be used in order to define the border region for local resampling.

The proposed method explores both the geometric information of the dataset and the regularization effect obtained when training with noise. Structural information is extracted from the GG in the form of structural vectors (SVs), which are vectors that share edges with vertices from the opposite class [31], [32]. The SVs are then used to generate synthetic noise samples in the separation region in order to smooth the decision surface. It is shown, that the generated noise samples lead to a Tikhonov regularization effect. Those noise samples are added to the training set and used to train an ELM.

The proposed method, Regularization with Noise of Extreme Learning Machines (RN-ELM), is compared to the standard ELM algorithm and to ELM-REG in 18 real-world datasets. The datasets differ in size, dimension, class overlap and imbalance. The values of the norm of the weights and accuracy are used to evaluate the models. Statistical tests have shown a significant difference between RN-ELM and the standard ELM, which indicates that local resampling in the border region yields the expected regularization effect. Furthermore, no significant statistical difference was also observed between RN-ELM and ELM-REG, which reinforces the expected regularization behavior of the model training with the resampling approach. It is also formally shown in this paper that local resampling in convex networks is equivalent to Tikhonov regularization. As a combination of different concepts, such as training with noise, data augmentation and graph-based margin resampling, this paper adds up to the formal proofs of regularization a new perspective to neural networks training.

This paper is organized as follows: section II presents a review of relevant literature, in section III the proposed method is explained, section IV details the experimental setup, section V presents the results, and in section VI is the conclusion of the work.

II. THEORETICAL BACKGROUND

A. GRAPH-BASED STRUCTURAL INFORMATION

Computational Geometry methods allow dataset patterns to be represented by a planar structure. One example is the Gabriel Graph (GG) [30]; a planar connected graph, built from geometric information of a dataset $\mathbf{x} \in \mathbb{R}^m$, defined by a set of vertices $\mathbf{V} = \{\mathbf{x}_i\}_{i=1}^N$ and a set of edges $\mathbf{A} = \{(\mathbf{x}_i, \mathbf{x}_j) \,|i \neq j\}$, which satisfy inequality (3):

$$\delta(\mathbf{x}_i, \mathbf{x}_j)^2 \leq [\delta(\mathbf{x}_i, \mathbf{x}_n)^2 + \delta(\mathbf{x}_j, \mathbf{x}_n)^2], \forall \mathbf{x}_n \in \mathbf{V}, \quad i \neq j \neq n \tag{3}$$

where $\delta(.)$ is the euclidean distance between vectors.

Edge (x_i, x_j) defined by inequality (3) is represented in Fig. 1, whereas Fig. 2 shows two edges that do not satisfy the inequality.

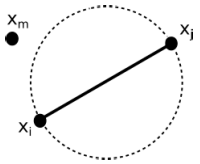


FIGURE 1. Edge that belongs to the GG.

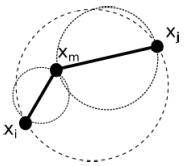


FIGURE 2. Edge that does not belong to the GG.

Given GG edges of a particular dataset, those connecting samples from different classes can be considered for generating synthetic samples in the border region. Fig. 3 depicts the GG of a two moons problem and synthetic samples added to the border region.

The expected value $E[y|\mathbf{x}]$ tends to the separation region where the effects of labels from opposite classes in the error function are balanced. According to Geman et. al [33], the function $f(\mathbf{x}, \mathbf{w})$ that approximates $E[y|\mathbf{x}]$ minimizes the general approximation error. Local resampling with equally probable labels from opposite classes in the region tends to approximate $f(\mathbf{x}, \mathbf{w})$ to $E[y|\mathbf{x}]$, thus minimizing the approximation error. Once synthetic samples are added to the border region, models with a large number of neurons are less likely to present overfitting, since these new samples smooth the separation surface.

B. EXTREME LEARNING MACHINE AND REGULARIZATION

ELM is a learning algorithm for SLFN [6] that is based on the random projections approach. ELM is easily implemented: bias and weights of the hidden layers are randomly assigned and the weights of the output layer are determined by a generalized inverse matrix.

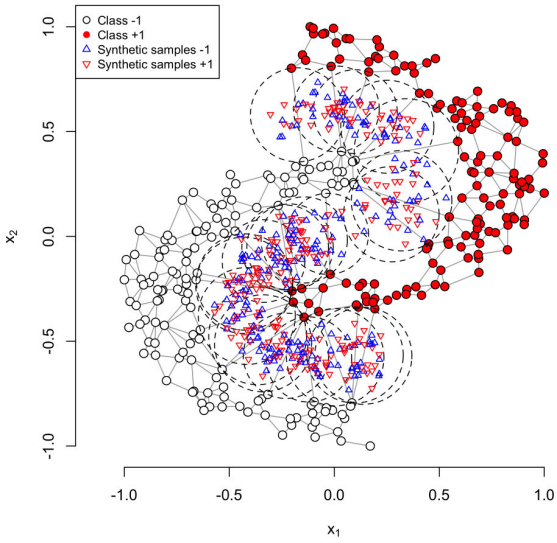


FIGURE 3. Gabriel graph of the two moons dataset together with synthetic samples generated in the hyperspheres of the separation surface.

Given a set of N distinct samples $(\mathbf{x}_i, y_i)$ where $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{im}]^T \in \mathbb{R}^m$ and $y_i \in \mathbb{R}$ for $i = \{1, \ldots, N\}$, the network output is given by (4)

$$\hat{\mathbf{y}}_i = \sum_{j=1}^{L} w_j g_j(\mathbf{x}_i) = \sum_{j=1}^{L} w_j g(\mathbf{v}_j \mathbf{x}_i + b_j) \quad i = 1, \ldots, N \tag{4}$$

where L is the number of hidden layer neurons, $g(.)$ is the activation function, $\mathbf{v}_j = [v_{j1}, v_{j2}, \ldots, v_{jm}]^T$ is a weight that connects the input and the hidden layer and w_j is a weight that connects the hidden layer and the output. Finally, b_j is the bias term of the j-th hidden neuron. For a SLFN with L hidden layer neurons, which is capable of approximating a function from N samples, there exists $\mathbf{w_j}$, $\mathbf{v}_j$ and b_j such that (5) is satisfied.

$$\mathbf{Hw} = \mathbf{y} \tag{5}$$

where

$$\mathbf{H} = \begin{bmatrix} g(\mathbf{v}_1 \mathbf{x}_1 + b_1) & \ldots & g(\mathbf{v}_L \mathbf{x}_1 + b_L) \\ \ldots & \ldots & \ldots \\ g(\mathbf{v}_1 \mathbf{x}_N + b_1) & \ldots & g(\mathbf{v}_L \mathbf{x}_N + b_L) \end{bmatrix}_{N \times L} \tag{6}$$

The output weights matrix $\mathbf{w}$ is calculated using the Moore-Penrose [34] pseudoinverse:

$$\mathbf{w} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y} = \mathbf{H}^+ \mathbf{y} \tag{7}$$

Regularization can be employed to smooth the effects of overfitting in oversized networks, as in ELM-REG [35]. Equation (8) shows the expressions for obtaining the weight

matrix for N smaller than L and vice-versa.

$$\mathbf{w} = \begin{cases} \mathbf{H}^T \left(\dfrac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{y}, & \text{if } N < L \\ \left(\dfrac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{y}, & \text{if } N \gg L \end{cases} \tag{8}$$

Regularization effects can also be achieved by resampling as shown by Bishop [21]. In his original work he demonstrates that the addition of small amplitude synthetic samples leads to a penalty term which is equivalent to the regularized sum-of-squares error. Hence, the method proposed in this paper is based on the same principle, however with a GG-based local resampling in the border region.

III. PROPOSED METHOD

Structural information is extracted from a planar graph (GG), as defined in section II-A [36]. The addition of synthetic patterns to the training set can be seen on Fig. 4. In this example, two classes sampled from Gaussian distributions are represented as empty and filled circles. Synthetic samples are triangles and upside down triangles. The geometric vectors and mean points are indicated by thick black circles and a "x" mark, respectively.

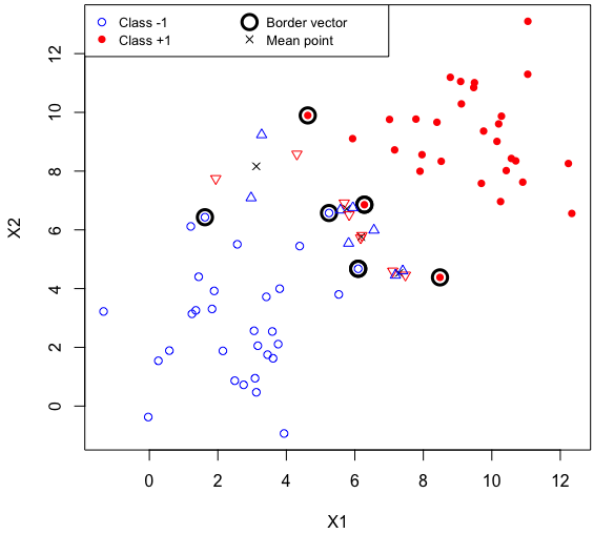


FIGURE 4. Addition of synthetic samples around the structural vectors mean.

The general expression for the sum of squared errors (SSE) is given in (9).

$$J_e = \sum_{i=1}^{N_1} (y_i - f(\mathbf{h}_i, \mathbf{w}))^2 \tag{9}$$

where N_1 is the number of samples in the training set. Once the synthetic samples are added, the error function can be rewritten as in (10).

$$J_e = \sum_{i=1}^{N_1} (y_i - f(\mathbf{h}_i, \mathbf{w}))^2 + \sum_{k=1}^{N_2} (v_k - f(\mathbf{r}_k + \epsilon_k, \mathbf{w}))^2 \tag{10}$$

where N_2 is the number of samples added to the training set, $(\mathbf{r}_k + \epsilon_k)$ is the k-th synthetic sample, composed of $\mathbf{r}_k$ selected from the geometric vectors, ϵ_k is a random noise and v_k is the label of the sampled pattern, which is the same as $\mathbf{r}_k$.

The additional term in (10) shifts the solution to the direction of the synthetic samples added $(\mathbf{r}_k + \epsilon_k)$, so the larger N_2 is, the more influential the synthetic samples become.

Considering that conditions of linear separability are met by ELM, function $f(\mathbf{h}_i, \mathbf{w}) = \mathbf{w}^T \mathbf{h}_i$, and consequently $f(\mathbf{r}_k + \epsilon_k, \mathbf{w}) = \mathbf{w}^T(\mathbf{r}_k + \epsilon_k)$ is considered, leading to (11).

$$J_e = \frac{1}{2} \sum_{i=1}^{N_1} (y_i - \mathbf{w}^T \mathbf{h}_i)^2 + \frac{1}{2} \sum_{k=1}^{N_2} (v_k - (\mathbf{w}^T \mathbf{r}_k + \mathbf{w}^T \epsilon_k))^2 \tag{11}$$

So, if $\hat{y}_i = \mathbf{w}^T \mathbf{h}_i$, $\hat{v}_k = \mathbf{w}^T \mathbf{r}_k$ and $\hat{u}_k = \mathbf{w}^T \epsilon_k$, (11) can be rewritten as (12).

$$J_e = \frac{1}{2} \sum_{i=1}^{N_1} (y_i - \hat{y}_i)^2 + \frac{1}{2} \sum_{k=1}^{N_2} (v_k - (\hat{v}_k + \hat{u}_k))^2 \tag{12}$$

Output weights that describe the separation hyperplane are obtained by deriving (11) and equating it to zero.

$$\begin{aligned} \frac{\partial J_e}{\partial w_j} = &- \sum_{i=1}^{N_1} y_i h_{ij} + \sum_{i=1}^{N_1} \hat{y}_i h_{ij} - \sum_{k=1}^{N_2} v_k r_{kj} \\ &- \sum_{k=1}^{N_2} v_k \epsilon_{kj} + \sum_{k=1}^{N_2} \hat{v}_k r_{kj} + \sum_{k=1}^{N_2} \hat{u}_k \epsilon_{kj} \\ &+ \sum_{k=1}^{N_2} \hat{u}_k r_{kj} + \sum_{k=1}^{N_2} \hat{v}_k \epsilon_{kj} = 0 \end{aligned} \tag{13}$$

Now (13) can be rewritten as (14).

$$\begin{aligned} \sum_{i=1}^{N_1} \hat{y}_i h_{ij} + \sum_{k=1}^{N_2} \hat{v}_k r_{kj} + \sum_{k=1}^{N_2} \hat{u}_k \epsilon_{kj} + \sum_{k=1}^{N_2} \hat{u}_k r_{kj} + \sum_{k=1}^{N_2} \hat{v}_k \epsilon_{kj} \\ = \sum_{i=1}^{N_1} y_i h_{ij} + \sum_{k=1}^{N_2} v_k r_{kj} + \sum_{k=1}^{N_2} v_k \epsilon_{kj} \end{aligned} \tag{14}$$

which is further developed into (15).

$$\begin{aligned} \sum_{i=1}^{N_1} \mathbf{w}^T \mathbf{h}_i h_{ij} + \sum_{k=1}^{N_2} \mathbf{w}^T \mathbf{r}_k r_{kj} + \sum_{k=1}^{N_2} \mathbf{w}^T \epsilon_k \epsilon_{kj} \\ + \sum_{k=1}^{N_2} \mathbf{w}^T \epsilon_k r_{kj} + \sum_{k=1}^{N_2} \mathbf{w}^T \mathbf{r}_k \ \epsilon_{kj} \\ = \sum_{i=1}^{N_1} y_i h_{ij} + \sum_{k=1}^{N_2} v_k r_{kj} + \sum_{k=1}^{N_2} v_k \epsilon_{kj} \end{aligned} \tag{15}$$

Finally, (15) is represented in its matrix form (16), which leads (16) to (17) and (18).

$$\begin{aligned}&\mathbf{H}^T\mathbf{H}\mathbf{w}+\mathbf{R}^T\mathbf{R}\mathbf{w}+\mathbf{E}^T\mathbf{E}\mathbf{w}+\mathbf{R}^T\mathbf{E}\mathbf{w}+\mathbf{E}^T\mathbf{R}\mathbf{w}\\&\quad=\mathbf{H}^T\mathbf{y}+\mathbf{R}^T\mathbf{v}+\mathbf{E}^T\mathbf{v}\end{aligned}\tag{16}$$

$$\begin{aligned}&(\mathbf{H}^T\mathbf{H}+\mathbf{R}^T\mathbf{R}+\mathbf{E}^T\mathbf{E}+\mathbf{R}^T\mathbf{E}+\mathbf{E}^T\mathbf{R})\mathbf{w}\\&\quad=\mathbf{H}^T\mathbf{y}+(\mathbf{R}^T+\mathbf{E}^T)\mathbf{v}\end{aligned}\tag{17}$$

$$\begin{aligned}\mathbf{w}&=(\mathbf{H}^T\mathbf{H}+\mathbf{R}^T\mathbf{R}+\mathbf{E}^T\mathbf{E}+\mathbf{R}^T\mathbf{E}+\mathbf{E}^T\mathbf{R})^{-1}\\&\quad(\mathbf{H}^T\mathbf{y}+(\mathbf{R}^T+\mathbf{E}^T)\mathbf{v})\end{aligned}\tag{18}$$

Training set and target values are represented by matrix $\mathbf{H}$ and vector $\mathbf{y}$.

$$\mathbf{H}=\begin{bmatrix}h_{11}&h_{12}&\dots&h_{1L}\\h_{21}&h_{22}&\dots&h_{2L}\\\vdots&\vdots&\ddots&\dots\\h_{N_11}&h_{N_12}&\dots&h_{N_1L}\end{bmatrix}\quad\mathbf{y}=\begin{bmatrix}y_1\\y_2\\\dots\\y_{N_1}\end{bmatrix}\tag{19}$$

Data noise samples are represented by matrix $\mathbf{P}$, which is the sum of matrices $\mathbf{R}$ (composed of the reference vectors) and $\mathbf{E}$ (random noise). Target values for these samples are given in matrix $\mathbf{v}$.

$$\mathbf{P}=\mathbf{R}+\mathbf{E}\tag{20}$$

$$\mathbf{R}=\begin{bmatrix}r_{11}&r_{12}&\dots&r_{1L}\\r_{21}&r_{22}&\dots&r_{2L}\\\vdots&\vdots&\ddots&\dots\\r_{N_21}&r_{N_22}&\dots&r_{N_2L}\end{bmatrix}\tag{21}$$

$$\mathbf{E}=\begin{bmatrix}\epsilon_{11}&\epsilon_{12}&\dots&\epsilon_{1L}\\\epsilon_{21}&\epsilon_{22}&\dots&\epsilon_{2L}\\\vdots&\vdots&\ddots&\dots\\\epsilon_{N_21}&\epsilon_{N_22}&\dots&\epsilon_{N_2L}\end{bmatrix}\tag{22}$$

$$\mathbf{v}=\begin{bmatrix}v_1\\v_2\\\dots\\v_{N_2}\end{bmatrix}\tag{23}$$

Matrix Λ corresponding to the regularization term is expressed in (24).

$$\Lambda=\mathbf{R}^T\mathbf{R}+\mathbf{E}^T\mathbf{E}+\mathbf{R}^T\mathbf{E}+\mathbf{E}^T\mathbf{R}\tag{24}$$

Finally, (24) is taken to (18), leading to the new weight update equation shown in (25).

$$\mathbf{w}=(\mathbf{H}^T\mathbf{H}+\Lambda)^{-1}(\mathbf{H}^T\mathbf{y}+(\mathbf{R}^T+\mathbf{E}^T)\mathbf{v})\tag{25}$$

Thus, (25) shows the new least squares weight update equation when local resampling is applied. As can be observed in (24), the regularization term Λ is composed by resampling terms $\mathbf{E}$ and $\mathbf{R}$, leading to smoothing effects of the separation surface. In the absence of noise, $\mathbf{E}$ and $\mathbf{R}$ are null and (25) is reduced to the standard least squares (7).

Suppose $N=N_p+N_n$, N_p the number of positive labeled samples and N_n the number of negative labeled samples.

$$\sum_{k=1}^{N}(r_k+\epsilon_k)v_k=\sum_{k=1}^{N_p}(r_{pk}+\epsilon_{pk})(+1)+\sum_{k=1}^{N_n}(r_{nk}+\epsilon_{nk})(-1)\tag{26}$$

$r_{pk}=r_{nk}=r$ is constant.

$$\sum_{k=1}^{N}(r_k+\epsilon_k)v_k=N_pr+\sum_{k=1}^{N_p}\epsilon_{pk}-N_nr-\sum_{k=1}^{N_n}\epsilon_{nk}\tag{27}$$

If the sampled set is balanced, *i.e.*, $N_p=N_r=M$:

$$\sum_{k=1}^{N}(r_k+\epsilon_k)v_k=\sum_{k=1}^{M}\epsilon_{pk}-\sum_{k=1}^{M}\epsilon_{nk}=\sum_{k=1}^{M}(\epsilon_{pk}-\epsilon_{nk})\tag{28}$$

If the synthetic samples are generated according to gaussian distributions, $\epsilon_p\sim\mathcal{N}(\mu=0,\sigma^2)$ and $\epsilon_n\sim\mathcal{N}(\mu=0,\sigma^2)$, $\epsilon_d=(\epsilon_p-\epsilon_n)\sim\mathcal{N}(\mu=0,2\sigma^2)$,

$$\sum_{k=1}^{N}(r_k+\epsilon_k)v_k=\sum_{k=1}^{M}\epsilon_{dk}\tag{29}$$

Finally, according to the Law of Large Numbers [37], since variables ϵ_d are indepedent identically distributed, when $M\to\infty$, $\frac{1}{M}\sum_{k=1}^{M}\epsilon_{dk}$ converges in probability to μ,

$$\sum_{k=1}^{N}(r_k+\epsilon_k)v_k=\sum_{k=1}^{M}\epsilon_{dk}=M\mu=M(0)=0\tag{30}$$

The result shown in (30) proves that, for a sufficiently large amount of generated samples, the term $(\mathbf{R}^T+\mathbf{E}^T)\mathbf{v}$ in (25) is equal to zero and (25) can be rewritten as (31).

$$\mathbf{w}=(\mathbf{H}^T\mathbf{H}+\Lambda)^{-1}(\mathbf{H}^T\mathbf{y})\tag{31}$$

Analyzing (31), it can be seen that the addition of sufficient synthetic samples is equivalent to Tikhonov's Regularization. In order to avoid the need for asymptotically large number of samples, symmetry can be achieved as long as each sample generates a mirrored one, with the same label, in relation to r, as defined by (32) and (33).

$$\mathbf{P}'=\mathbf{R}-\mathbf{E}\tag{32}$$

$$\mathbf{v}'=\mathbf{v}\tag{33}$$

For this synthetic set, (34) holds.

$$\mathbf{P}^T\mathbf{v}+\mathbf{P}'^T\mathbf{v}'=0\tag{34}$$

IV. EXPERIMENTS

The experiments were performed on binary classification problems. The first one was carried out on the two moons synthetic problem for visualization purposes. The second one was accomplished on real world datasets and compared to the standard regularized ELM [35].

A. TWO-DIMENSIONAL PROBLEM

In order to compare the standard ELM with RN-ELM, both methods were applied to the two-moons dataset. ELM separation surface with 500 hidden layer neurons results on overfitting, as shown in Fig. 5, whereas the RN-ELM separation surface with the same number of hidden neurons after regularization with local resampling is shown in Fig. 6. The addition of noise samples to the training set leads to regularization and to smaller norm values of the weights.

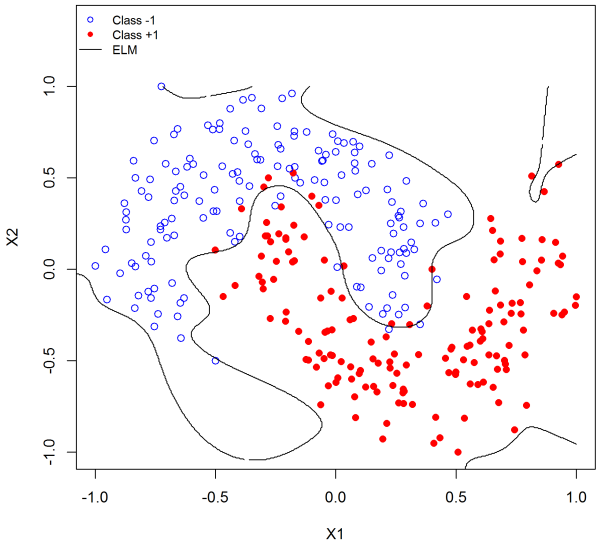


FIGURE 5. ELM separation surface for the two moons dataset with 500 hidden layer neurons results on overfitting.

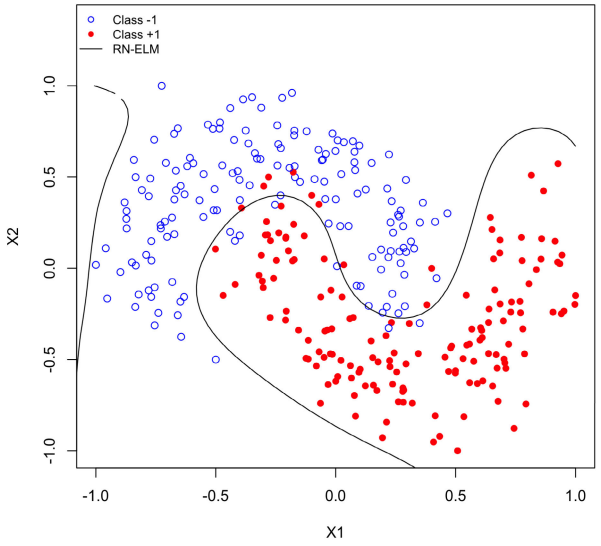


FIGURE 6. RN-ELM separation surface of Fig 5 with the same number of hidden neurons after regularization with local resampling.

B. MULTIDIMENSIONAL DATASET

The performance of the proposed regularization algorithm, RN-ELM, was evaluated on 18 datasets with different numbers of variables and sizes. Ten datasets were obtained on the UCI Repository [38] (Audit Data (aud), Australian Credit Approval (aca), Wisconsin Diagnostic Breast Cancer (bcr), Diabetic Retinopathy (drp), Ionosphere (ion), Parkinsons (pks), Pima Indians Diabetes (pid), QSAR biodegradation (qsr), Sonar (snr), Statlog Heart (sth)), six datasets were obtained on the KEEL Repository [39] (Appendicitis (apd), Bupa (bpa), Ecoli1 (ec1), Haberman (hbm), Monk2 (mk2), Breast Cancer Wisconsin Original (wcs)), and, finally, Golub (glb) [40] and Hess (hes) [41]. All datasets are binary classification problems. Instances of the datasets containing missing values were discarded. Table 1 summarizes dataset sizes and number of variables.

TABLE 1. Dataset characteristics.

dataset	number of variables	number of samples
apd	8	106
aud	18	776
aca	15	690
bcr	31	569
bpa	7	345
drp	20	1151
ec1	8	336
glb	72	51
hbm	4	306
hes	31	133
ion	35	351
mk2	7	432
pks	23	195
pid	9	768
qsr	42	1055
snr	61	208
sth	14	270
wcs	10	683

TABLE 2. Friedman test results (p-value) for different numbers of hidden layer neurons (L).

L	p-value
10	0.0004144
30	0.03974
100	4.604e-05
500	1.58e-05
1000	3.843e-05

The performance of the proposed method (RN-ELM) was compared to two other ELM learning approaches. Input data were standardized to zero mean and unity standard deviation and the output was assigned to $+1$ and -1 labels. The hyperbolic tangent function was used on hidden neurons and initial weights sampled from an uniform distribution within the interval $[-0.5, 0.5]$. The numbers of neurons on the hidden layer were 10, 30, 100, 500, and 1000 [19]. The dataset was randomly split into training and test sets with 70% to 30% ratio, respectively. The number of samples for resampling was obtained by 10-fold cross-validation within the range $n = \{1 \ldots 10\}$. Local resampling was obtained from a normal distribution with standard deviation defined by (35), which guarantees that all data is within three standard deviations from the mean.

$$\sigma^2 = \left(\frac{D}{6}\right)^2 \tag{35}$$

where D is the distance between border vertices of opposite classes.

TABLE 3. Accuracy of the test set (mean ± standard deviation). Results are shown according to the model, number of neurons in the hidden layer (L) and dataset.

L	ELM	ELM-REG	RN-ELM	L	ELM	ELM-REG	RN-ELM
apd				**aud**			
10	87.08 ± 4.62	86.88 ± 5.94	85.10 ± 5.90	10	77.95 ± 3.05	77.73 ± 3.68	76.52 ± 3.21
30	79.90 ± 6.07	86.56 ± 5.51	83.65 ± 5.04	30	79.41 ± 2.30	80.36 ± 2.36	79.21 ± 3.08
100	56.46 ± 7.95	87.40 ± 5.53	83.23 ± 7.59	100	78.28 ± 2.78	81.25 ± 2.37	80.04 ± 2.34
500	62.81 ± 9.19	85.94 ± 5.67	75.62 ± 9.12	500	57.10 ± 3.49	81.55 ± 2.41	80.17 ± 2.14
1000	64.48 ± 10.33	86.98 ± 4.93	65.31 ± 10.77	1000	56.71 ± 3.48	80.78 ± 2.15	79.24 ± 1.92
aca				**bcr**			
10	82.91 ± 3.24	82.96 ± 3.13	82.37 ± 3.19	10	93.14 ± 2.25	92.87 ± 2.62	93.08 ± 2.52
30	85.97 ± 2.38	86.18 ± 2.46	86.18 ± 2.88	30	95.87 ± 1.57	96.04 ± 1.51	96.55 ± 1.28
100	84.94 ± 1.83	86.96 ± 1.99	86.84 ± 2.16	100	96.04 ± 1.53	96.69 ± 1.33	96.71 ± 1.17
500	51.76 ± 3.37	86.78 ± 2.54	86.73 ± 2.33	500	86.61 ± 2.42	97.43 ± 0.93	97.19 ± 1.05
1000	56.18 ± 4.78	86.92 ± 2.41	86.86 ± 1.98	1000	94.76 ± 1.59	97.68 ± 1.05	97.25 ± 1.05
bpa				**drp**			
10	67.98 ± 4.25	67.82 ± 4.40	66.19 ± 4.54	10	62.40 ± 2.94	62.30 ± 3.08	62.49 ± 3.29
30	69.17 ± 4.74	70.06 ± 4.78	68.81 ± 4.69	30	68.58 ± 1.90	68.71 ± 1.98	66.84 ± 2.18
100	64.23 ± 4.69	68.78 ± 4.47	68.78 ± 4.52	100	70.64 ± 2.33	70.77 ± 2.43	69.06 ± 2.53
500	54.04 ± 5.96	69.81 ± 4.46	70.83 ± 3.82	500	60.92 ± 2.40	72.20 ± 2.28	70.15 ± 2.64
1000	54.90 ± 5.57	70.26 ± 4.02	70.03 ± 4.80	1000	53.42 ± 2.93	72.60 ± 1.76	70.68 ± 2.05
ec1				**glb**			
10	88.05 ± 3.60	86.93 ± 3.60	86.53 ± 2.72	10	69.09 ± 10.57	72.42 ± 8.85	68.79 ± 10.46
30	90.63 ± 2.35	90.43 ± 2.17	88.55 ± 3.13	30	67.27 ± 10.09	74.55 ± 8.90	72.12 ± 9.53
100	86.93 ± 3.10	90.43 ± 2.51	89.67 ± 3.11	100	67.27 ± 11.47	77.27 ± 8.52	75.15 ± 9.45
500	69.11 ± 5.08	90.40 ± 2.58	89.11 ± 2.96	500	72.08 ± 9.40	75.97 ± 11.40	74.03 ± 8.74
1000	72.05 ± 5.40	89.67 ± 2.61	88.94 ± 2.86	1000	75.61 ± 7.50	78.48 ± 7.25	74.85 ± 8.75
hbm				**hes**			
10	75.04 ± 3.27	75.58 ± 3.62	75.94 ± 3.35	10	71.33 ± 6.32	71.17 ± 7.87	68.58 ± 8.30
30	72.43 ± 3.76	73.26 ± 4.28	74.38 ± 4.12	30	74.58 ± 6.98	76.75 ± 7.29	76.92 ± 6.81
100	67.68 ± 4.83	73.48 ± 4.34	74.86 ± 4.58	100	57.08 ± 7.25	80.00 ± 5.49	80.08 ± 5.63
500	64.31 ± 4.60	72.75 ± 4.21	74.49 ± 3.99	500	68.83 ± 7.30	81.50 ± 6.58	82.75 ± 6.27
1000	64.49 ± 3.35	74.46 ± 4.15	75.14 ± 4.16	1000	70.43 ± 6.45	83.62 ± 5.41	84.31 ± 4.91
ion				**mk2**			
10	79.08 ± 4.12	78.67 ± 4.34	77.46 ± 4.28	10	80.46 ± 3.02	80.13 ± 3.49	79.08 ± 3.23
30	84.67 ± 3.20	85.02 ± 3.33	84.70 ± 3.19	30	84.31 ± 4.05	84.00 ± 4.48	78.33 ± 3.29
100	82.92 ± 4.05	88.13 ± 3.15	87.87 ± 2.60	100	94.18 ± 2.48	93.92 ± 2.34	80.13 ± 3.99
500	81.94 ± 4.21	87.52 ± 2.77	88.13 ± 2.43	500	87.92 ± 4.29	95.82 ± 1.53	90.23 ± 3.75
1000	84.16 ± 2.56	88.48 ± 2.95	88.83 ± 2.69	1000	90.49 ± 2.48	95.77 ± 1.19	91.77 ± 3.04
pks				**pid**			
10	76.50 ± 6.43	75.65 ± 6.19	75.31 ± 5.90	10	74.80 ± 2.70	74.94 ± 2.88	74.19 ± 2.93
30	83.90 ± 5.30	84.80 ± 5.57	80.68 ± 5.39	30	75.72 ± 2.25	76.38 ± 2.26	76.32 ± 2.10
100	79.38 ± 6.54	89.10 ± 4.29	85.08 ± 4.57	100	73.84 ± 3.16	77.28 ± 2.79	76.94 ± 2.65
500	87.06 ± 4.23	92.20 ± 3.52	85.71 ± 4.07	500	56.88 ± 3.26	75.91 ± 2.56	75.70 ± 2.67
1000	90.11 ± 4.25	92.77 ± 3.27	88.93 ± 4.60	1000	59.87 ± 2.57	76.20 ± 2.10	75.86 ± 2.19
snr				**qsr**			
10	68.76 ± 7.87	67.80 ± 7.57	67.80 ± 8.13	10	73.71 ± 3.19	73.54 ± 3.33	73.60 ± 3.57
30	72.96 ± 4.80	73.23 ± 5.85	74.03 ± 5.89	30	81.99 ± 2.58	81.76 ± 2.75	79.95 ± 2.67
100	67.85 ± 5.59	75.48 ± 5.80	73.98 ± 4.44	100	85.33 ± 1.66	85.46 ± 1.62	83.10 ± 1.98
500	79.35 ± 3.43	79.19 ± 3.87	76.29 ± 4.43	500	74.57 ± 2.61	87.35 ± 1.62	84.33 ± 2.07
1000	81.99 ± 4.87	81.13 ± 5.48	78.28 ± 5.29	1000	58.87 ± 5.17	87.36 ± 1.42	85.81 ± 1.68
sth				**wcs**			
10	79.22 ± 4.10	78.72 ± 4.08	78.52 ± 4.51	10	96.62 ± 1.10	96.47 ± 1.17	95.82 ± 1.22
30	81.73 ± 4.68	83.21 ± 4.17	83.33 ± 3.99	30	96.80 ± 1.25	96.72 ± 1.20	96.11 ± 1.31
100	74.24 ± 5.37	83.00 ± 4.84	83.09 ± 4.46	100	95.51 ± 1.49	96.86 ± 1.09	96.02 ± 1.37
500	74.28 ± 3.64	83.17 ± 3.56	82.76 ± 4.08	500	88.68 ± 2.62	96.70 ± 1.04	95.87 ± 1.48
1000	75.23 ± 4.34	82.65 ± 4.17	83.28 ± 3.86	1000	91.04 ± 1.92	96.93 ± 1.13	96.00 ± 1.12

For ELM-REG the regularization parameter is selected within the range $C = \{2^{-24}, \ldots, 2^{15}\}$ [19] with 10-fold cross-validation [35]. For each dataset, three different ELM training methods were used (ELM, ELM-REG, and RN-ELM). Overall performance was assessed by comparing mean accuracy (Table 3) and weight norm $||\mathbf{w}||$ (Table 4). For each network configuration, average values were obtained in 30 trials. Finally, Friedman and pos-hoc Nemenyi tests were adopted for comparing multiple models and multiple domains [19], [42].

V. RESULTS

For 18 datasets of real world classifications problems, RN-ELM was compared with the standard ELM and ELM-REG. As expected, regularization on networks with a small number of hidden neurons (10 or 30) did not lead to better results, so that standard ELM accuracy was within one standard deviation of the other two methods and even performed better for *apd*, *bcr*, *bpa*, *ec1*, *ion*, *hes*, *mk2*, *pks*, *qsr*, *snr*, *sth*, and *wcs* datasets. However, when the number of hidden neurons is greater than 100, as expected, regularization plays

TABLE 4. Model weight norms (mean ± standard deviation). Norms are presented according to the model, number of neurons in the hidden layer (L) and dataset.

L	ELM	ELM-REG	RN-ELM	L	ELM	ELM-REG	RN-ELM
apd				**aud**			
10	5.71 ± 2.46	3.22 ± 0.96	2.65 ± 0.19	10	4.07 ± 0.16	4.01 ± 0.15	3.91 ± 0.13
30	28.23 ± 10.57	4.74 ± 0.73	4.53 ± 0.11	30	6.96 ± 0.15	6.77 ± 0.14	6.70 ± 0.12
100	913.81 ± 337.86	8.16 ± 0.15	8.18 ± 0.18	100	13.79 ± 0.42	12.25 ± 0.11	12.25 ± 0.11
500	127.27 ± 52.97	18.27 ± 0.10	37.57 ± 38.35	500	37505.15 ± 5535.61	27.37 ± 0.13	27.37 ± 0.13
1000	72.32 ± 29.68	25.82 ± 0.13	93.75 ± 44.66	1000	20508.92 ± 4048.61	38.73 ± 0.11	38.73 ± 0.10
aca				**bcr**			
10	3.69 ± 0.15	3.67 ± 0.16	3.53 ± 0.15	10	5.20 ± 0.11	5.18 ± 0.10	5.13 ± 0.10
30	6.25 ± 0.10	6.17 ± 0.11	6.12 ± 0.10	30	8.90 ± 0.11	8.87 ± 0.11	8.84 ± 0.11
100	11.50 ± 0.13	11.20 ± 0.11	11.19 ± 0.11	100	16.15 ± 0.13	16.07 ± 0.13	16.06 ± 0.13
500	447.38 ± 226.90	25.00 ± 0.13	24.99 ± 0.13	500	36.60 ± 0.18	35.92 ± 0.10	35.91 ± 0.10
1000	91.87 ± 37.98	35.27 ± 0.13	35.27 ± 0.13	1000	50.85 ± 0.14	50.77 ± 0.14	50.77 ± 0.14
bpa				**drp**			
10	3.65 ± 0.51	3.21 ± 0.50	2.41 ± 0.15	10	4.23 ± 0.11	4.21 ± 0.11	4.10 ± 0.11
30	8.61 ± 1.30	5.39 ± 1.29	4.19 ± 0.12	30	7.36 ± 0.13	7.27 ± 0.15	7.06 ± 0.13
100	59.13 ± 9.38	8.78 ± 1.86	7.64 ± 0.13	100	13.59 ± 0.17	13.05 ± 0.20	12.88 ± 0.13
500	884.34 ± 278.99	18.54 ± 5.03	17.09 ± 0.10	500	73.07 ± 5.33	28.91 ± 0.14	28.86 ± 0.13
1000	454.34 ± 153.56	24.20 ± 0.12	24.18 ± 0.10	1000	368.93 ± 31.32	40.80 ± 0.09	40.78 ± 0.10
ec1				**glb**			
10	4.00 ± 1.02	3.40 ± 0.66	2.75 ± 0.19	10	6.56 ± 0.11	6.51 ± 0.11	6.50 ± 0.11
30	9.74 ± 2.71	5.83 ± 1.34	4.53 ± 0.15	30	11.53 ± 0.20	11.31 ± 0.18	11.28 ± 0.15
100	93.38 ± 19.78	8.70 ± 1.18	8.26 ± 0.12	100	20.72 ± 0.13	20.63 ± 0.12	20.63 ± 0.12
500	1112.86 ± 260.47	18.40 ± 0.30	18.27 ± 0.12	500	46.10 ± 0.11	46.10 ± 0.11	46.10 ± 0.11
1000	513.86 ± 133.58	26.13 ± 0.68	25.86 ± 0.16	1000	65.18 ± 0.13	65.18 ± 0.13	65.18 ± 0.13
hbm				**hes**			
10	8.68 ± 3.62	3.89 ± 2.56	1.86 ± 0.12	10	5.17 ± 0.13	5.12 ± 0.13	5.10 ± 0.12
30	117.83 ± 46.32	8.47 ± 12.19	3.16 ± 0.12	30	8.93 ± 0.14	8.82 ± 0.14	8.79 ± 0.14
100	59692.60 ± 38473.75	6.42 ± 1.21	5.78 ± 0.14	100	24.02 ± 3.54	16.07 ± 0.15	16.06 ± 0.15
500	67824.21 ± 14660.07	13.26 ± 1.02	12.90 ± 0.12	500	35.94 ± 0.14	35.88 ± 0.13	35.88 ± 0.13
1000	38131.04 ± 6231.65	18.28 ± 0.14	18.26 ± 0.13	1000	50.86 ± 0.12	50.84 ± 0.12	50.84 ± 0.12
ion				**mk2**			
10	5.52 ± 0.13	5.50 ± 0.13	5.44 ± 0.12	10	3.36 ± 0.49	2.93 ± 0.42	2.46 ± 0.13
30	9.46 ± 0.14	9.40 ± 0.14	9.36 ± 0.13	30	8.53 ± 2.62	7.32 ± 2.36	4.18 ± 0.17
100	17.51 ± 0.12	17.13 ± 0.10	17.12 ± 0.09	100	31.60 ± 4.72	20.95 ± 8.19	7.70 ± 0.15
500	38.46 ± 0.17	38.19 ± 0.14	38.18 ± 0.14	500	91.83 ± 11.95	19.23 ± 3.07	38.20 ± 18.44
1000	54.08 ± 0.13	54.01 ± 0.13	54.01 ± 0.13	1000	47.00 ± 4.93	24.44 ± 0.64	37.36 ± 19.09
pks				**pid**			
10	4.63 ± 0.13	4.55 ± 0.14	4.42 ± 0.11	10	3.13 ± 0.16	3.01 ± 0.17	2.76 ± 0.13
30	8.16 ± 0.21	7.91 ± 0.27	7.63 ± 0.14	30	5.27 ± 0.25	4.86 ± 0.19	4.72 ± 0.11
100	18.40 ± 1.42	14.11 ± 0.30	13.86 ± 0.11	100	11.42 ± 0.58	8.68 ± 0.13	8.64 ± 0.13
500	31.23 ± 0.15	30.97 ± 0.15	31.29 ± 0.39	500	454.02 ± 57.82	19.38 ± 0.13	19.38 ± 0.13
1000	43.86 ± 0.15	43.79 ± 0.15	43.91 ± 0.17	1000	129.10 ± 9.32	27.36 ± 0.14	27.35 ± 0.14
qsr				**snr**			
10	6.00 ± 0.13	5.98 ± 0.13	5.95 ± 0.13	10	7.17 ± 0.13	7.15 ± 0.13	7.14 ± 0.13
30	10.32 ± 0.14	10.31 ± 0.14	10.27 ± 0.13	30	12.38 ± 0.11	12.35 ± 0.12	12.33 ± 0.11
100	18.74 ± 0.14	18.70 ± 0.14	18.67 ± 0.14	100	22.74 ± 0.13	22.54 ± 0.12	22.53 ± 0.12
500	43.15 ± 0.21	41.81 ± 0.13	41.80 ± 0.13	500	50.47 ± 0.13	50.46 ± 0.13	50.45 ± 0.13
1000	96.89 ± 28.06	59.18 ± 0.14	59.17 ± 0.14	1000	71.27 ± 0.13	71.27 ± 0.13	71.26 ± 0.13
sth				**wcs**			
10	3.56 ± 0.12	3.49 ± 0.15	3.41 ± 0.13	10	3.09 ± 0.13	2.96 ± 0.15	2.95 ± 0.14
30	6.16 ± 0.14	5.95 ± 0.12	5.91 ± 0.12	30	5.30 ± 0.15	5.02 ± 0.14	5.01 ± 0.15
100	12.32 ± 0.33	10.84 ± 0.14	10.83 ± 0.14	100	11.12 ± 0.47	9.20 ± 0.25	9.14 ± 0.11
500	24.68 ± 0.15	24.14 ± 0.12	24.14 ± 0.12	500	35.80 ± 4.47	20.41 ± 0.14	20.39 ± 0.13
1000	34.32 ± 0.12	34.18 ± 0.12	34.17 ± 0.12	1000	32.10 ± 0.74	28.88 ± 0.12	28.88 ± 0.12

a major role, as for most datasets ELM-REG and RN-ELM had better results than ELM. Furthermore, it can be seen that the proposed method led to similar results when compared to ELM-REG, and both were within one standard deviation from each other, except for the *apd*, *mk2*, and *pks* datasets. The only dataset for which regularization did not improve results, with outcomes mostly within one standard deviation, was *snr*. The results of obtained mean accuracies are presented in Table 3.

For most datasets, especially for a large number of hidden neurons ($L = 500$ and $L = 1000$), RN-ELM and ELM-REG were both capable of reducing the norm of the weight vector (used as a measure of network complexity), indicating that the network outputs are smoothed. These results can be seen on Table 4. The proposed method (RN-ELM) has achieved performance similar to that of ELM-REG in terms of mean accuracy and norm of the weight vector, which indicates that local resampling also leads to regularization.

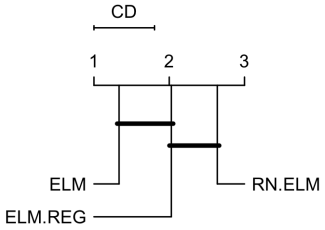


FIGURE 7. Critical difference diagram of the model with 10 neurons in the hidden layer.

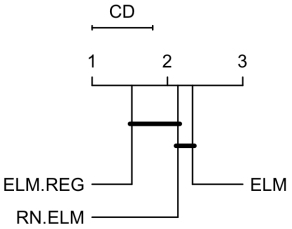


FIGURE 8. Critical difference diagram of the model with 30 neurons in the hidden layer.

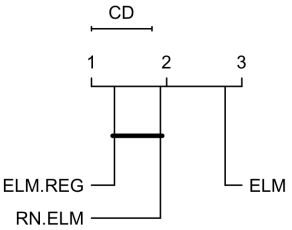


FIGURE 9. Critical difference diagram of the model with 100 neurons in the hidden layer.

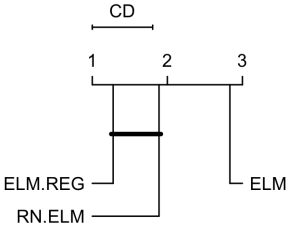


FIGURE 10. Critical difference diagram of the model with 500 neurons in the hidden layer.

In order to compare the three classifiers in the multiple datasets evaluated, the Friedman test and the Nemenyi post-hoc test were used [42], [43]. The results of the Friedman test can be seen on Table 2. For a significance level $\alpha = 0.05$, the null hypothesis of equality between ELM training approaches can be rejected for $L = \{10, 30, 100, 500, 1000\}$. The Nemenyi post-hoc test was applied next, yielding the results summarized on Fig. 7 to 11. It can be seen that, for mean accuracy values, RN-ELM is not statistically different from ELM-REG, as, for all cases, even though ELM-REG is ranked higher, both methods are less than one Critical Difference (CD) apart.

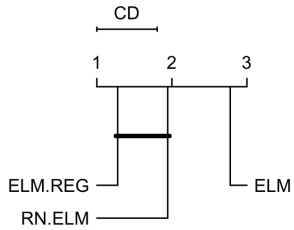


FIGURE 11. Critical difference diagram of the model with 1000 neurons in the hidden layer.

Regarding time complexity, according to [44], the hidden layer matrix has a time complexity of $O(NLm)$ whilst the output matrix has $O(L^3 + L^2N + LNC)$, where $C = 2$ for binary classification problems, L is the number of hidden layer neurons, m is the number of variables and N the number of samples. The time complexity of Gabriel Graph [31] algorithm is $O(mN^3)$ for the worst case scenario and $O(mN^2)$ for the best case.

Due to the lack of standardization to run experiments and to compare model performances on different datasets, 10-fold cross-validation, adopted in this work, seems to provide the most general methodology for benchmarking and comparing different models in the literature. Although there is no guarantee that the considered folds were exactly the same on different publications, statistical properties of 10-fold cross-validation yields a reliable approximation to the global performance [45]. The results presented in the benchmarking paper by Gestel *et al.* [46], which adopted 10-fold cross-validation and grid search, are very similar to the ones obtained in this work. For instance, performances obtained by the authors in the following datasets were: aca (87,00%), bpa (70,20%), ion (96,00%), pid (76,80%), snr (73,10%), sth (84,70%) and wbs (96,40%). As can be observed in Table 3, the results are quite close, what may suggest that, since grid-search was also adopted by Gestel *et al.* [46], the outcomes obtained with RN-ELM may provide a reliable general approximation without the need to make an exhaustive search on parameter's space. Comparisons with results from papers that did not adopt 10-fold cross-validation also suggest that the performance obtained with RN-ELM on the remaining datasets is within the expected range reported in the literature [47]–[50].

VI. CONCLUSION

It has been shown formally in this paper that ELM training with local sampling leads to Tikhonov regularization. This outcome follows Bishop's developments presented in the mid 1990's [21], however, the graph-based local resampling approach presented in this paper leads the separation function to the margin region, without the need to exhaustively cover the whole input space.

The results presented in this paper also show that such an approach reduces the norm of the weights, indicating that the methods yields smoother solutions, reducing the

effects of overfitting. Performance metrics also indicate that outcomes are statistically equivalent to the ones obtained by ELM-REG with regularization parameters obtained with cross-validation.

Directed resampling with the graph-based approach may reduce costly input space exploration in higher dimensional problems involving data augmentation. Although the graph needs to be generated in order to locate resampling, it is based on pairwise information which can be fully parallelized.

REFERENCES

[1] N. J. Guliyev and V. E. Ismailov, "On the approximation by single hidden layer feedforward neural networks with fixed weights," *Neural Netw.*, vol. 98, pp. 296–304, Feb. 2018.

[2] H. T. Huynh and Y. Won, "Regularized online sequential learning algorithm for single-hidden layer feedforward neural networks," *Pattern Recognit. Lett.*, vol. 32, no. 14, pp. 1930–1935, Oct. 2011.

[3] T. Matias, F. Souza, R. Araújo, and C. H. Antunes, "Learning of a single-hidden layer feedforward neural network using an optimized extreme learning machine," *Neurocomputing*, vol. 129, pp. 428–436, Apr. 2014.

[4] W. F. Schmidt, M. A. Kraaijveld, and R. P. Duin, "Feed forward neural networks with random weights," in *Proc. Int. Conf. Pattern Recognit.*, 1992, pp. 1–11.

[5] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Trans. Electron. Comput.*, vol. EC-14, no. 3, pp. 326–334, Jun. 1965.

[6] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[7] E. Sevinc, "A novel evolutionary algorithm for data classification problem with extreme learning machines," *IEEE Access*, vol. 7, pp. 122419–122427, 2019.

[8] Y. Lei, L. Cen, X. Chen, and Y. Xie, "A hybrid regularization semi-supervised extreme learning machine method and its application," *IEEE Access*, vol. 7, pp. 30102–30111, 2019.

[9] C. Song, L. Pan, Q. Liu, Z. Jiang, and J. Jia, "Extreme learning machine under minimum information divergence criterion," *IEEE Access*, vol. 8, pp. 122026–122035, 2020.

[10] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.

[11] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford Univ. Press, 1995.

[12] R. D. Albuquerque Teixeira, A. P. Braga, R. H. C. Takahashi, and R. R. Saldanha, "Improving generalization of MLPs with multi-objective optimization," *Neurocomputing*, vol. 35, nos. 1–4, pp. 189–194, Nov. 2000.

[13] H. P. Rocha, M. A. Costa, and A. P. Braga, "Neural networks multiobjective learning with spherical representation of weights," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4761–4775, Nov. 2020.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[15] W. Deng, Q. Zheng, and L. Chen, "Regularized extreme learning machine," in *Proc. IEEE Symp. Comput. Intell. Data Mining*, Mar. 2009, pp. 389–395.

[16] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "OP-ELM: Optimally pruned extreme learning machine," *IEEE Trans. Neural Netw.*, vol. 21, no. 1, pp. 158–162, Jan. 2010.

[17] Y. Miche, M. van Heeswijk, P. Bas, O. Simula, and A. Lendasse, "TROP-ELM: A double-regularized ELM using LARS and tikhonov regularization," *Neurocomputing*, vol. 74, no. 16, pp. 2413–2421, Sep. 2011.

[18] Preeti, R. Bala, A. Dagar, and R. P. Singh, "A novel online sequential extreme learning machine with L 2, 1-norm regularization for prediction problems," *Int. J. Speech Technol.*, vol. 51, no. 3, pp. 1669–1689, Oct. 2020.

[19] L. J. Silvestre, A. P. Lemos, J. P. Braga, and A. P. Braga, "Dataset structure as prior information for parameter-free regularization of extreme learning machines," *Neurocomputing*, vol. 169, pp. 288–294, Dec. 2015.

[20] L. R. G. Araujo, L. C. B. Torres, L. J. Silvestre, C. Takahashi, and A. P. Braga, "Regularization of extreme learning machines with information of spatial relations of the projected data," in *Proc. 6th Int. Conf. Control, Decis. Inf. Technol. (CoDIT)*, Apr. 2019, pp. 593–597.

[21] C. M. Bishop, "Training with noise is equivalent to tikhonov regularization," *Neural Comput.*, vol. 7, no. 1, pp. 108–116, Jan. 1995.

[22] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 279–283, Mar. 2017.

[23] H. Noh, T. You, J. Mun, and B. Han, "Regularizing deep neural networks by noise: Its interpretation and optimization," in *Advances in Neural Information Processing Systems*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 5109–5118.

[24] W. Zhang, X. Li, X.-D. Jia, H. Ma, Z. Luo, and X. Li, "Machinery fault diagnosis with imbalanced data using deep generative adversarial networks," *Measurement*, vol. 152, Feb. 2020, Art. no. 107377.

[25] T. DeVries and G. W. Taylor, "Dataset augmentation in feature space," 2017, *arXiv:1702.05538*. [Online]. Available: http://arxiv.org/abs/1702.05538

[26] Q. Quan, F. He, and H. Li, "A multi-phase blending method with incremental intensity for training detection networks," *Vis. Comput.*, vol. 37, pp. 1–15, Jan. 2020.

[27] J. Lemley, S. Bazrafkan, and P. Corcoran, "Smart augmentation learning an optimal data augmentation strategy," *IEEE Access*, vol. 5, pp. 5858–5869, 2017.

[28] R. E. Schapire, "The boosting approach to machine learning: An overview," in *Nonlinear Estimation Classification*. New York, NY, USA: Springer, 2003, pp. 149–171.

[29] K. Li, X. Kong, Z. Lu, L. Wenyin, and J. Yin, "Boosting weighted ELM for imbalanced learning," *Neurocomputing*, vol. 128, pp. 15–21, Mar. 2014.

[30] K. R. Gabriel and R. R. Sokal, "A new statistical approach to geographic variation analysis," *Syst. Biol.*, vol. 18, no. 3, pp. 259–278, 1969.

[31] L. C. B. Torres, C. L. Castro, F. Coelho, and A. P. Braga, "Large margin Gaussian mixture classifier with a gabriel graph geometric representation of data set structure," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 1400–1406, Mar. 2020.

[32] J. Arias-Garcia, A. Mafra, L. Gade, F. Coelho, C. Castro, L. Torres, and A. Braga, "Enhancing performance of gabriel graph-based classifiers by a hardware co-processor for embedded system applications," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1186–1196, Feb. 2021.

[33] S. Geman, E. Bienenstock, and R. Doursat. (1992). *Neural Networks and the Bias/Variance Dilemma*. [Online]. Available: http://www.mitpressjournals.org/doi/10.1162/neco.1992.4.1.1

[34] J. C. A. Barata and M. S. Hussein, "The Moore–Penrose pseudoinverse: A tutorial review of the theory," *Brazilian J. Phys.*, vol. 42, nos. 1–2, pp. 146–165, Apr. 2012.

[35] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[36] L. C. B. Torres, C. L. Castro, F. Coelho, F. Sill Torres, and A. P. Braga, "Distance-based large margin classifier suitable for integrated circuit implementation," *Electron. Lett.*, vol. 51, no. 24, pp. 1967–1969, Nov. 2015.

[37] M. DeGroot and M. Schervish, *Probability and Statistics*. Reading, MA, USA: Addison-Wesley, 2012.

[38] D. Dheeru and E. Karra Taniskidou. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[39] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, and S. García, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Multiple-Valued Logic Soft Comput.*, vol. 17, nos. 2–3, pp. 255–287, 2011.

[40] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, Oct. 1999.

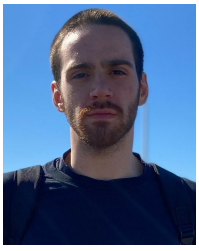
[41] K. R. Hess, K. Anderson, W. F. Symmans, V. Valero, N. Ibrahim, J. A. Mejia, D. Booser, R. L. Theriault, A. U. Buzdar, P. J. Dempsey, and R. Rouzier, "Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer," *J. Clin. Oncol.*, vol. 24, no. 26, pp. 4236–4244, 2006.

[42] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.
[43] N. Japkowicz and M. Shah, *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge, U.K.: Cambridge Univ. Press, 2011.
[44] A. Iosifidis, A. Tefas, and I. Pitas, "On the kernel extreme learning machine classifier," *Pattern Recognit. Lett.*, vol. 54, pp. 11–17, Mar. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865514003705
[45] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statist. Surveys*, vol. 4, pp. 40–79, Jan. 2010.
[46] T. van Gestel, J. A. K. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. de Moor, and J. Vandewalle, "Benchmarking least squares support vector machine classifiers," *Mach. Learn.*, vol. 54, no. 1, pp. 5–32, Jan. 2004.
[47] A. F. Agarap, "On breast cancer detection: An application of machine learning algorithms on the wisconsin diagnostic dataset," *CoRR*, vol. abs/1711.07831, pp. 5–9, Feb. 2017.
[48] R. Das, "A comparison of multiple classification methods for diagnosis of parkinson disease," *Expert Syst. Appl.*, vol. 37, p. 1568, Jan. 2010.
[49] B. Nie, J. Luo, J. Du, L. Peng, Z. Wang, and A. Chen, "Improved algorithm of C4.5 decision tree on the arithmetic average optimal selection classification attribute," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Nov. 2017, pp. 1376–1380.
[50] K. Polat and M. Nour, "Parkinson disease classification using one against all based data sampling with the acoustic features from the speech signals," *Med. Hypotheses*, vol. 140, Jul. 2020, Art. no. 109678.

ALEX D. ASSIS received the B.Sc. degree and the M.Sc. degree in computer science from the Federal University of Viçosa (UFV), Brazil, in 2006 and 2010, respectively. He is currently pursuing the Ph.D. degree with the Department Electrical Engineering, Universidade Federal de Minas Gerais (UFMG), Brazil. He is also an Assistant Professor with the Universidade Federal de Juiz de Fora (UFJF).

LUIZ C. B. TORRES received the B.Sc. degree in computer science from the University Center of Belo Horizonte in Graph Theory, in 2010, and the master's degree in electrical engineering and the Ph.D. degree in neural networks from the Universidade Federal de Minas Gerais (UFMG), Brazil, in 2012 and 2016, respectively. He received a Postdoctoral Researcher from UFMG through Brazilian agency CNPq (National Council for Scientific and Technological), in 2017. In 2018, he received a postdoctoral degree in international cooperation with the Optical Networks Laboratory (ONLAB) Laboratory, Royal Institute of Technology (KTH) of Sweden, in 2018. Since 2019, he has been with the Computer and Systems Department, Universidade Federal de Ouro Preto (UFOP), where he is currently an Adjunct Professor and the Head of the Machine Learning Research Group, UFOP. He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS.

LOURENÇO R. G. ARAÚJO received the B.Sc. degree in chemical engineering and the master's degree in electrical engineering from the Universidade Federal de Minas Gerais (UFMG), in 2016 and 2019, respectively, where he is currently pursuing the Ph.D. degree with the Graduate Program in Electrical Engineering.



VÍTOR M. HANRIOT is currently pursuing the bachelor's degree in control and automation engineering with the Universidade Federal de Minas Gerais (UFMG). In his final semester, he is also an Intern with the Computational Intelligence Laboratory (LITC), UFMG.

ANTONIO P. BRAGA received the B.Sc. degree in electrical engineering and the master's degree in computer science from the Universidade Federal de Minas Gerais (UFMG), Brazil, in 1987 and 1991, respectively, and the Ph.D. degree in electrical engineering from the University of London, Imperial College, in 1995, in recurrent neural networks. Since 1991, he has been with the Electronics Engineering Department, Universidade Federal de Minas Gerais (UFMG), where he is a Full Professor and the Head of the Computational Intelligence Laboratory. He is also an Associate Researcher of the Brazilian National Research Council. As a Professor and a Researcher, he has coauthored many books, book chapters, journal, and conference papers. He has served in program committees for many international conferences and was the Program Co-Chair for IJCNN 2018. He was also an Associate Editor of many international journal, including *Engineering Applications of Artificial Intelligence*, *Neural Processing Letters*, *International Journal of Computational Intelligence and Applications*, and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.