International Journal of Technology (2017) 4: 601-610
ISSN 2086-9614 © IJTech 2017

SIN ACTIVATION STRUCTURAL TOLERANCE OF ONLINE SEQUENTIAL CIRCULAR EXTREME LEARNING MACHINE

Sarutte Atsawaraungsuk[1*], Tatpong Katanyukul[1]

[1] *Department of Computer Engineering, Faculty of Engineering, Khon Kaen University,*
123 Moo 16 Mittapap Rd., Muang, Khon Kaen 40002, Thailand

(Received: July 2016 / Revised: April 2017 / Accepted: July 2017)

ABSTRACT

This article discusses the development of the online sequential circular extreme learning machine (OS-CELM) and structural tolerance OS-CELM (STOS-CELM). OS-CELM is developed based on the circular extreme learning machine (CELM) to enable sequential learning. It can update a new chunk of data by spending less training time to update the chunk than the batch CELM. STOS-CELM is developed based on an idea similar to that of OS-CELM, but with a Householder block exact inverse QR decomposition (QRD) recursive least squares (QRD-RLS) algorithm to allow sequential learning and mitigate the criticality of deciding the number of hidden nodes. In addition, our experiments have shown that given the same hidden node setting, STOS-CELM can deliver accuracy comparable to a batch CELM approach and also has higher accuracy than the original online sequential extreme learning machine (OS-ELM) and structural tolerance OS-ELM (STOS-ELM) in classification problems, especially those involving high dimension datasets.

Keywords: Circular extreme learning machine; Extreme learning machine; Householder block exact QRD recursive least squares algorithm; Online sequential extreme learning machine

1. INTRODUCTION

The extreme learning machine (ELM) is a supervised algorithm proposed by Huang et al. (2004; 2006; 2011). It has been shown that ELM enables a very fast learning process and leads to satisfactory accuracy. The conventional ELM relies on a batch-mode learning mechanism. When there is new available data, the batch-mode ELM has to retrain the model with an entire dataset, including previously used data and the newly available data. This requires a long time, even when the new data is very small.

Liang et al. (2006) developed the online sequential extreme learning machine (OS-ELM) to address this issue and claimed that its results had higher accuracy than the batch ELM accuracy. However, OS-ELM is very sensitive (Horata et al., 2015). That is, the performance of OS-ELM along a sequence of online updates leads to a lower accuracy. This can be explained by the fact that OS-ELM repeats ELM for its very first batch. The number of hidden nodes, which is a number of model parameters, is selected to best fit the data. However, although the selected number of nodes is suitable for the very first batch in the OS-ELM context, this number is fixed and may be less suitable later, after more data is accrued.

To solve this problem, Horata et al. (2015) introduced structural tolerance OS-ELM (STOS-

*Corresponding author's email: sarutte_a@kkumail.com, Tel. +66-4336-2145-6, Fax. +66-4336-2142
Permalink/DOI: https://doi.org/10.14716/ijtech.v8i4.4966

ELM), which uses the Householder block exact QR decomposition recursive least squares (HBQRD-RLS) (Rontogiannis & Theodoridis, 2009). This ensures that the online update is more robust, by enabling OS-ELM to have a large number of nodes from the beginning.

In another line of research to extend ELM capability, Decherchi et al. (2013) proposed a circular extreme learning machine (CELM) that can handle high dimension problems better than the conventional ELM. CELM is a single-hidden layer feedforward network (SLFN) that uses circular back propagation (CBP) (Ridella et al., 1997; Gastaldo et al., 2002) architecture. CBP adds one dimension, which is the norm of the input. That extra dimension provides for the better handling of high dimension problems than by basic input formulation (Ridella et al., 1997). The additional dimension improves the overall performance of the CELM without affecting the generalization of the ELM structure. CELM has been successfully applied to assessment of perceived image quality (Decherchi et al., 2013; Atsawaraungsuk & Horata, 2015).

This paper proposes two online sequential versions of CELM. The first version, an online sequential circular extreme learning machine (OS-CELM), aims to add an online update capability to CELM based on the OS-ELM approach. The second version, a structural tolerance OS-CELM (STOS-CELM) aims to extend CELM based on the HBQRD-RLS algorithm (Rontogiannis & Theodoridis, 2009; Horata et al., 2015). Our experimental results show that STOS-CELM can improve the accuracy of both online versions of ELM, especially when applied to high dimensional datasets. STOS-CELM also shows the robustness of its predecessor STOS-ELM.

This article is organized as follows: Section 2 describes OS-ELM, HBQRD-RLS, CELM, and the proposed algorithm OS-CELM and STOS-CELM; Section 3 explains the experimental design and experimental results; Section 4 briefly discusses the findings, and Section 5 presents the conclusion.

2. METHODOLOGY

2.1. Online Sequential Extreme Learning Machine

In a real situation, training samples may arrive sequentially, in a chunk-by-chunk form. To solve this problem, Liang et al. (2006) proposed an online sequential extreme leaning machine (OS-ELM) that can update the data immediately. Let $\mathbf{X}_{[k-1]} = \left\{(\mathbf{x}_i, \mathbf{t}_i)\right\}_{\mathrm{i}=[N_0+\ldots+N_{k-1}]}^{[N_0+\ldots+N_{k-1}]}$ be an initial the training sample set, $\mathbf{H}_{[k-1]}$ the hidden layer output matrix, and $\mathbf{T}_{[k-1]}$ the matrix that contains the target values. The objective of OS-ELM can then be written as

$$\min_{\beta} \left\| \begin{bmatrix} \mathbf{H}_{[0:k-1]} \\ \mathbf{H}_k \end{bmatrix} \beta - \begin{bmatrix} \mathbf{T}_{[0:k-1\backslash} \\ \mathbf{T}_k \end{bmatrix} \right\| \lim_{x\to\infty} \tag{1}$$

where $\mathbf{H}_k$ is the hidden layer output matrix that can be calculated from new arrival data $\mathbf{X}_k$, and $\mathbf{T}_k$ is a target values matrix of $\mathbf{X}_k$.

OS-ELM can summarize the steps to compute as follows. At the initial step, if the number of hidden nodes K is less than or equal to the number of samples N then the output weights from training $\mathbf{X}_{[0;k-1]}$ and $\mathbf{X}_k$ are defined as

$$\beta_k = \mathbf{K}_k^{-1} \begin{bmatrix} \mathbf{H}_{k-1} \\ \mathbf{H}_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{T}_{k-1} \\ \mathbf{T}_k \end{bmatrix} \tag{2}$$

where

$$\mathbf{K}_k = \begin{bmatrix} \mathbf{H}_{k-1} \\ \mathbf{H}_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{H}_{k-1} \\ \mathbf{H}_k \end{bmatrix}. \tag{3}$$

In OS-ELM, Equation 3 $\mathbf{K}_k$ can be rewritten to the terms of $\mathbf{K}_{k-1}$ for sequential learning as

$$\mathbf{K}_k = \mathbf{K}_{k-1} + \left(\mathbf{H}_k^{\mathrm{T}}\mathbf{H}_k\right) \tag{4}$$

with Equation 2 divided e as

$$\beta_k = \beta_{k-1} + \mathbf{K}_k^{-1}\mathbf{H}_k^{\mathrm{T}}\left(\mathbf{T}_k - \mathbf{H}_k\beta_{k-1}\right) \tag{5}$$

An important way to calculate β_k is $\mathbf{K}_k^{-1}$, $\mathbf{K}_{k-1}^{-1}$ from Equation 4, which can be reformed based on the Woodbury formula (Golub & Loan, 1996) as

$$\mathbf{P}_k = \mathbf{P}_{k-1} - \mathbf{P}_{k-1}\mathbf{H}_k^{\mathrm{T}}\left(\mathbf{I} - \mathbf{H}_k\mathbf{P}_{k-1}^{-1}\right)^{-1}\mathbf{H}_k\mathbf{P}_{k-1} \tag{6}$$

where $\mathbf{P}_k = \mathbf{K}_k^{-1}, \mathbf{P}_{k-1} = \mathbf{K}_{k-1}^{-1}$ and $\mathbf{I}$ is the identity matrix. The output weights of OS-ELM β_{k+1} in Equation 5 and Equation 6 are rewritten as

$$\beta_k = \beta_{k-1} + \mathbf{P}_k\mathbf{H}_k^{\mathrm{T}}\left(\mathbf{T}_k - \mathbf{H}_k\beta_{k-1}\right) \tag{7}$$

2.2. Householder Block Exact Inverse QRD-RLS Algorithm to Handle a New Added Chunk of Data to the System

The Householder-based QR-decomposition recursive least squares algorithms (HBQRD-RLS) (Rontogiannis & Theodoridis, 2009) can handle the least squares system and make the sequential learning robust. The HBQRD-RLS has to store and update the square root factor of the input data covariance matrix ($\tilde{\mathbf{R}}_{k-1}^{-1}$) that is used to update β_k in sequential learning. However, its computing steps are summarized as follows.

Initially, the step $\tilde{\mathbf{R}}_{k-1}^{-1} \approx \mathbf{H}^{\dagger}$ is calculated at time k-1 by

$$\tilde{\mathbf{R}}_{k-1}^{-1} = \mathbf{R}^{-1}\mathbf{Q}^T \tag{8}$$

from $\mathbf{Q}^T\mathbf{H}\beta = \mathbf{Q}^T\mathbf{T}$, where $\mathbf{QR} = \mathbf{H}$, and $\mathbf{Q} \in □^{N\times N}$ is an orthogonal matrix with the property $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_{N\times N}$. Then,

$$\begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}\beta = \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} \tag{9}$$

where β_{k-1} is the least squares solution at the previous time, $\mathbf{H}_k$ is a newly added data matrix, and $\mathbf{T}_k$ is a newly added target values matrix.

For $\mathbf{Q}^T\mathbf{H} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$ and $\mathbf{Q}^T\mathbf{T} = \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix}$, $\mathbf{R}$ is the $L \times L$ upper triangular that has values the same as the square root of $\mathbf{H}^T\mathbf{H}$ or $\mathbf{R}^T\mathbf{R} = \mathbf{H}^T\mathbf{H}$. Equation 9 can be solved in the triangular system (Pan & Plemmons, 1989; Moonen & Vandewalle, 1991).

The first step of HBQRD-RLS is to compute the matrix production, thus:

$$\mathbf{G}_k = -\mathbf{R}_{k-1}^{-T}\mathbf{H}_k^T. \tag{10}$$

The second step is to store and update the square root factor matrix $\mathbf{R}_k^{-1}$ and acquire $\mathbf{F}_k$ and $\mathbf{E}_k^T$. By applying Lemma 1 (Rontogiannis & Theodoridis, 2009), HBQRD-RLS uses the Householder transformation—see details of the Householder transforms algorithm in Golub and Loan (1996) and Trefethen and Bau (1997)—to produce an orthogonal matrix $\mathbf{U}(k)$, such that

$$\mathbf{U}(k)\begin{bmatrix} \mathbf{I}_C & \mathbf{0}_{C\times L} \\ \mathbf{G}_k & \mathbf{R}_{k-1}^{-T} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_k & \mathbf{E}_k^T \\ \mathbf{0}_{L\times C} & \mathbf{R}_k^{-T} \end{bmatrix}. \tag{11}$$

The last step of HBQRD-RLS is to update the new least square solution as follows:

$$\beta_k = \beta_{k-1} + \mathbf{E}_k\mathbf{F}_k^{-T}\left(\mathbf{T}_k - \mathbf{H}_k\beta_{k-1}\right) \tag{12}$$

where $\left(\mathbf{T}_k - \mathbf{H}_k\beta_{k-1}\right)$ is the error matrix predicted by β_{k-1}, $\mathbf{E}_k\mathbf{F}_k^{-T}$ and known as the Kalman Gain.

2.3. Circular Extreme Learning Machine

The CELM (Decherchi et al., 2013) is the one extended ELM. It has the same structure as ELM and circular back propagation (CBP) architecture to enable it to map both linear and circular decision boundaries. CELM is like ELM without the hidden layer output; the difference is that it can be calculated from

$$\mathbf{H} = \mathbf{h}_{ij} = \mathrm{g}(\mathbf{z}_j \bullet ||\mathbf{x}_i - \mathbf{c}_j||^2 - \mathbf{b}_j) \tag{13}$$

where $\mathbf{H}$ is the hidden layer output matrix (*K* is the number of hidden nodes),

$$\mathbf{z}_j = w_{j,K+1},$$

$$\mathbf{c}_j = \left[-\frac{w_{j,1}}{2w_{j,K+1}}, \ldots, -\frac{w_{j,K}}{2w_{j,K+1}} \right]$$

$$\mathbf{b}_j = \frac{1}{w_{j,K+1}}\left(\sum_{k=1}^{K} \frac{w_{j,K}^2}{4w_{j,K+1}} - \mathrm{br}_j \right)$$

w_j and $w_{j,K+1}$ are the weight vectors connecting the hidden layer and input layer, and b_j in br_j denotes the bias of the hidden nodes.

2.4. The Proposed OS-CELM

The proposed algorithm, the OS-CELM, is the online sequential version of CELM. It can train the samples that arrive sequentially chuck by chunk. We can summarize the process of OS-CELM as shown in Figure 1.

2.5. The Proposed STOS-CELM

The structural tolerance OS-CELM is one way to enable sequential learning for CELM. STOS-CELM uses the HBQRD-RLS algorithm to store and update $\mathbf{R}_{k-1}^{-1}$ for updating β_k in new data. STOS-CELM is summarized in Figure 2.

```
Steps:
1. Initialize parameter in $K$ hidden neurons
    ▪ Centers $w_j$ are assigned by random patterns of X
    ▪ Weights $w_{j,K+1}$ are the random numbers in the range [-1,1]
    ▪ Widths br are randomly defined numbers in the range [0,1]
2. Calculate the hidden layer output matrix $\mathbf{H}_0$ using Equation 13.
3. Initially, $\mathbf{K}_0 = \left(\mathbf{H}_0^{\mathrm{T}}\mathbf{H}_0\right)$ and $\beta_0 = \mathbf{K}_0^{-1}\mathbf{H}_0^{\mathrm{T}}\mathbf{T}_0$. When a new sample set $\mathbf{X}_k$ arrives, a new hidden layer output matrix $\mathbf{H}_k$ is computed
4. Store and update $\mathbf{K}_k^{-1}$ using Equation 6.
5. Update new output weights $\beta_k$ at the time a chunk is coming as in Equation 7
6. Go to step (4), $k=k+1$, when a new sample comes to the training process
End
```

Figure 1 Algorithm of OS-CELM

```
Steps:
1. Initialize parameter in $K$ hidden neurons.
    ▪ Centers $w_j$ are assigned random patterns from X.
    ▪ Weights $w_{j,K+1}$ are the random numbers in the range [-1,1].
    ▪ Widths br are randomly defined numbers in the range [0,1].
2. Calculate the hidden layer output matrix $\mathbf{H}_0$ using Equation 13.
3. Initial $\tilde{\mathbf{R}}^{-1} = \mathbf{H}^{\dagger}$ and corresponding solution $\beta_{k-1} = \mathbf{R}^{-1}\mathbf{H}_0^{T}\mathbf{T}_0$. When a new sample set $\mathbf{X}_k$ comes to the system, a new hidden layer output matrix $\mathbf{H}_k$ is computed.
4. Calculate $\mathbf{G}_k$ as in Equation 10.
5. Store and update $\mathbf{R}_k^{-1}$ , $\mathbf{F}_k$ and $\mathbf{E}_k$ using Equation 11.
6. Update new output weights $\beta_k$ while a chunk is coming as in Equation 12.
7. Go to step (4), $k=k+1$ when a new sample comes to the training process.
End
```

Figure 2 Algorithm of STOS-CELM

3. RESULTS

3.1. Experimental Setup

This section describes the experiments in detail. The 46 datasets from the AYRNA research group—most of which are housed in the University of Irvine, California (UCI) repository—are used to test the performance of methods that run on MATLAB version R2014a in environment Core i3 3.40 GHz Ram 4.00 GB. The 10-fold cross-validation method was used to validate the random input weight and biases and find the optimal number of hidden nodes selected in the range of 1–200.

3.2. Performance Comparison of Activation Function in STOS-CELM

To analyze the STOS-CELM of six activation functions—sigmoid, sin, hardlim, tribas, radbas, and Q-Gaussian activation—the experimental results were quantified by several statistical methods, thus: the percentage of correct classification T, the percentage of the mean ($\overline{\mathrm{T}}$),percentage of the median ($\tilde{\mathrm{T}}$), average ranking ($\overline{\mathrm{R}}$), and standard deviation (SD).

Table 1 Performance comparison of activation function in STOS-CELM

Dataset	Sigmoid	Sin	Hardlim	Tribas	Radbas	Q-Gaussian
Best accuracy	10	**27**	1	3	4	15
$\bar{T}$	85.23	**86.37**	61.89	73.88	80.38	83.94
$\tilde{T}$	86.74	**87.76**	64.64	76.99	84.14	85.24
$\bar{R}$	3.87	**2.89**	7.80	6.35	4.93	3.87
#SD	**2.3424**	2.7182	2.8331	2.7203	2.7569	2.7328

Table 1 shows the results of the performance of STOS-CELM with the sin activation function as the highest rank in 27 datasets. This yields the highest mean ($\bar{T}$ = 86.37), median ($\tilde{T}$ = 87.76), and average ranking ($\bar{R}$ = 2.89). The sigmoid function has the lowest SD with sin function the second lowest. These results show that the sin activation function has a winning score of 4-1, which is suggested as the activation function of STOS-CELM.

3.3. Performance Comparisons of STOS-CELM

To analyze the performance of STOS-CELM, it was compared with five algorithms: ELM, OS-ELM, STOS-ELM (Horata et al., 2015), the original CELM, and OS-CELM. Statistical measures were used to evaluate performance as above (Section 3.2) and add win ("+"), tie ("="), and lose ("–") rates among STOS-CELM and the compared methods (the meta-metrics evaluation) (Stefani & Xenos, 2009; Horata et al., 2013).

Table 2 Performance comparison of STOS-CELM with the compared methods

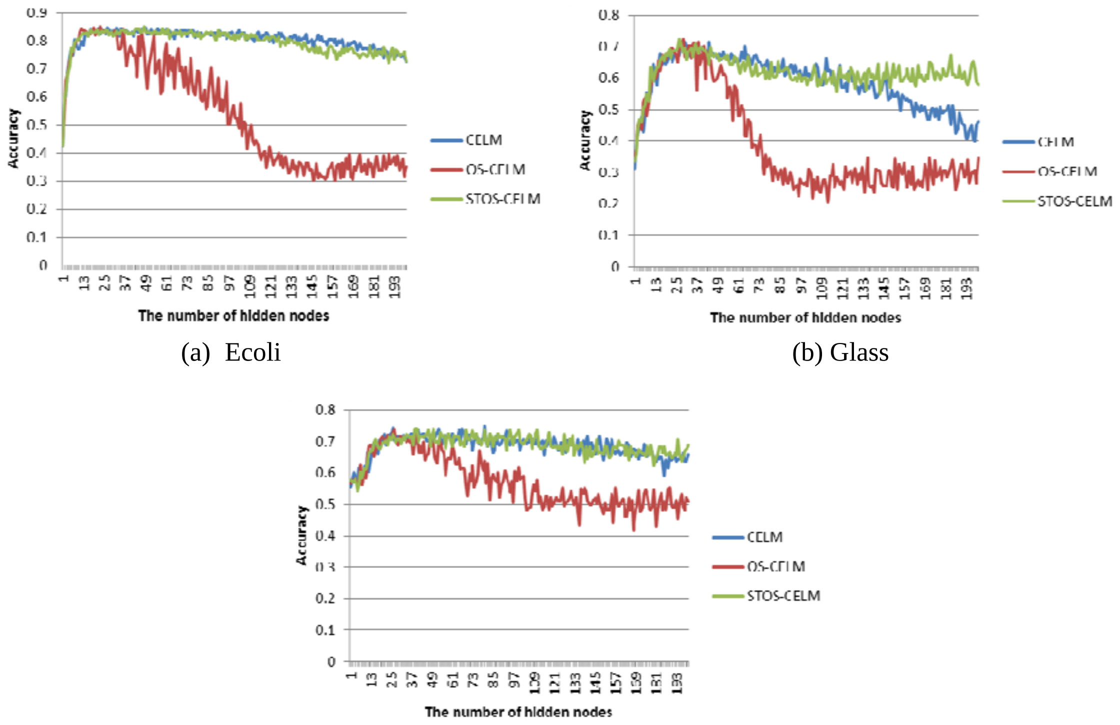
Dataset	ELM	OS-ELM	STOS-ELM	CELM	OS-CELM	STOS-CELM
Anneal	0.9889-	**0.9900-**	**0.9900-**	0.9811+	0.9866-	0.9844
Audio	0.7791+	0.8134+	0.8051+	0.8101+	0.8142+	**0.8180**
Autos	0.6538+	0.6781-	0.6638+	0.6683=	**0.6826-**	0.6683
Balance	0.9249-	**0.9281-**	0.9249-	0.9201-	0.9200+	0.9201
Breast	0.7448+	**0.7589-**	0.7378+	0.7486+	0.7520+	0.7555
Breast-w	0.9671-	0.9699-	**0.9700-**	0.9685-	0.9685-	0.9671
Card	**0.8725-**	0.8710-	0.8710-	**0.8725-**	0.8681+	0.8696
Dermatology	0.9836-	**0.9837-**	**0.9837-**	0.9809-	0.9755-	0.9755
Ecoli	**0.8665-**	0.8574-	0.8603-	0.8488+	0.8514+	0.8516
Gene	0.8202+	0.8148+	0.8211+	0.7386+	**0.8850-**	0.8828
German	0.7580+	0.7580+	0.7610+	0.7640+	0.7640+	**0.7650**
Glass	0.7052+	0.7093+	0.7050+	0.7147+	**0.7238-**	0.7236
glassG2	0.7235+	0.7239+	**0.7430-**	0.7408-	0.6989+	0.7294
Heart	0.8370-	**0.8407-**	0.8296-	0.8259=	0.8333-	0.8259
Heart-c	0.8152-	0.8214-	**0.8282-**	0.8215-	0.8146+	0.8148
heartY	0.8593-	0.8481=	0.8519-	**0.8630-**	0.8444+	0.8481
Hepatitis	0.8958+	0.8967+	0.8963+	**0.9029-**	**0.9029-**	0.9025
Horse	0.6610+	0.6753+	0.6613+	0.6804+	0.6726+	**0.6861**
Hypo	0.9287+	0.9290+	0.9292+	**0.9311-**	0.9305-	0.9300
Ionos	0.9034+	0.8975+	0.9033+	**0.9460-**	0.9459-	0.9458

Dataset	ELM	OS-ELM	STOS-ELM	CELM	OS-CELM	STOS-CELM
Iris	**0.9867-**	0.9800=	0.9800=	**0.9867-**	0.9800=	0.9800
Krvskp	0.9643+	0.9625+	0.9603+	0.9662+	0.9756+	**0.9762**
Labor	0.9600+	0.9833+	**1.0000=**	**1.0000=**	0.9833+	**1.0000**
Lenses	0.7000+	**0.8333=**	**0.8333=**	**0.8333=**	**0.8333=**	**0.8333**
Liver	0.7453-	**0.7657+**	0.7540-	0.7511-	0.7397-	0.7394
Lymph	0.8171-	0.8310+	0.8376-	**0.8438-**	0.8238-	0.8167
Newthyroid	0.9069+	0.9162=	0.9117+	**0.9632-**	0.9444+	0.9489
Optdigits	0.9783-	**0.9786-**	0.9778-	0.9715+	0.9778-	0.9776
Page-blocks	0.9593+	0.9582-	0.9598+	**0.9607-**	0.9541+	0.9605
Pima	**0.7734-**	0.7670+	0.7683-	0.7721-	0.7631+	0.7643
Post-op	0.7222=	0.7222-	**0.7333-**	0.7222=	**0.7333-**	0.7222
Primary-tumor	0.4929-	0.4898+	**0.4987-**	0.4633+	0.4690-	0.4665
Prompters	0.8127+	0.8018-	0.8009+	0.8200+	0.8036+	**0.8400**
Satimage	0.8796+	0.8797=	0.8803+	0.8869+	0.8923+	**0.8942**
Segment	0.9541+	0.9450-	0.9515+	0.9087+	0.9398+	**0.9606**
Sick	0.9483+	0.9480+	**0.9499-**	0.9478+	0.9483+	0.9486
Sonar	0.7933+	0.8176+	0.8069+	0.8324+	**0.8467-**	0.8329
Soybean	0.9503+	0.9488+	0.9488+	0.9459+	0.9518+	**0.9532**
Tic-tac-toe	0.9937+	0.9916+	0.9927+	0.9906+	**0.9948=**	**0.9948**
Vehicle	**0.8735-**	0.8712+	0.8724-	0.8581+	0.8629+	0.8723
Vote	0.9747=	**0.9770-**	0.9770-	0.9770-	0.9770-	0.9747
Vowel	0.9434+	0.9394+	0.9354+	0.9253+	0.9192+	**0.9515**
Waveform	0.8534+	0.8530+	0.8522+	**0.8600-**	0.8586-	0.8582
Wine	0.9944+	0.9944+	0.9944+	**1.0000=**	**1.0000=**	**1.0000**
Yeast	0.6031+	0.6051+	0.6024+	0.6186+	0.6010+	**0.6206**
Zoo	**0.9900-**	**0.9900-**	**0.9900-**	0.9800=	0.9700+	0.9800
The best accuracy	6	10	11	13	9	**14**
Win/Tie/Loss	**27**/2/17	**25**/4/17	**23**/3/20	**21**/7/18	**24**/4/18	-
$\bar{T}$	85.35	85.90	85.88	85.90	86.04	**86.37**
$\tilde{T}$	87.30	87.11	87.17	86.77	87.66	**87.76**
$\bar{R}$	4.61	4.09	4.02	3.83	4.26	**3.65**
SD	**2.4458**	2.8914	2.7492	2.8278	2.8414	2.7182

As Table 2 shows, STOS-CELM had the highest accuracy in 14 datasets, with a mean ($\bar{T}$ = 86.37), median ($\tilde{T}$ = 87.76), and average ranking ($\bar{R}$ = 3.65). The SD of the STOS-CELM was the second lowest (after ELM). These results show STOS-CELM as outperforming the compared methods. It is more accurate than STOS-ELM, especially when applied to high dimensional datasets such as Gene, Promoter, Audio, and Soybean that have 120, 114, 93, and 82 attributes, respectively.

3.4. The Generalization Performance of STOS-CELM with Varied Numbers of Hidden Nodes

Figure 3 shows the three methods' accuracy rates when the number of hidden nodes varied in the range of 1–200. The accuracy of OS-CELM peaked and then declined when the number of hidden nodes increased more than the initial data, while STOS-CELM and CELM maintained robustness.



(a) Ecoli

(b) Glass

(c) Liver

Figure 3 The generalization performance of STOS-CELM for the ecoli, glass, and liver datasets was more stable than that of OS-CELM when the number of hidden nodes varied in a wide range

The experimental results showed the accuracy rate of OS-CELM to be very much affected by an inappropriate number of hidden nodes. On the other hand, the accuracy rate of STOS-CELM and CELM were only slightly affected by this, which means that an appropriate number of hidden nodes is necessary for OS-CELM but STOS-CELM and CELM can be used with many hidden nodes, so the general performance of STOS-ELM for classification problems is comparable to that of CELM, while it also has a higher accuracy than ELM, OS-ELM, and STOS-ELM, especially when applied to high dimensional datasets.

4. DISCUSSION

CELM is a version of ELM that can improve it to handle high dimension datasets and complex problems. However, in many practical applications, such as the real-time radio-frequency identification (RFID) indoor positioning system for shop-floor management (Yang et al., 2015) and time series prediction (Lu et al., 2017), data are acquired sequentially, and CELM has to retrain all of the CELM processes to account for the new data. This causes a large amount of unnecessary training time. STOS-CELM is proposed here is to mitigate this issue by employing the advantage of CELM and improving its accuracy for long sequences of online updates. However, the accuracy of STOS-CELM still depends on the number of hidden nodes, which are fixed from the onset. Our research group envisions a mechanism to enable the addition of nodes during the process. This mechanism should be able to monitor prediction error and restructure the model to best suit the current state. Such a mechanism would allow for more flexibility in the STOS-CELM design and lead to wider applications of the technique.

5. CONCLUSION

This paper has proposed a structure tolerance online sequential circular extreme learning machine (STOS-CELM) based on the Householder block exact inverse QRD-RLS (HBQRD-RLS) algorithm to improve OS-CELM. The proposed method was evaluated according to 46 classification problems using several performance statistics. The results showed that OS-CELM is sensitive to the number of hidden nodes. If an inappropriate number of hidden nodes were selected, it was likely to produce a low accuracy. However, STOS-CELM and CELM were less affected by this. Furthermore, the quality of the solution from STOS-CELM was also shown to be better than that of ELM, OS-ELM, STOS-ELM CELM, and OS-CELM.

6. REFERENCES

Atsawaraungsuk S., Horata, P., 2015. Evolutionary Circular-ELM for the Reduced-reference Assessment of Perceived Image Quality. *Lecture Notes in Electrical Engineering*, Volume 339, pp. 657–664

Decherchi, S., Gastaldo, P., Zunino, R., Cambria, E., Redi, J., 2013. Circular-ELM for the Reduced-reference Assessment of Perceived Image Quality. *Neurocomputing*, Volume 102, pp. 78–89

Gastaldo, P., Zounino, R., Heynderickx I., Vicario, E., 2002. Circular Back-propagation Networks for Measuring Displayed Image Quality. *Lecture Notes in Computer Science*, Volume 2415, pp. 1219–1224

Golub, G.H., Loan, C.F.V., 1996. *Matrix Computations (3rd ed)*. Johns Hopkins University, Maryland

Horata, P., Chiewchanwattana, S., Sunat, K., 2013. Robust Extreme Learning Machine. *Neurocomputing*, Volume 102, pp. 31–44

Horata, P., Chiewchanwattana, S., Sunat, K., 2015. Enhancement of Online Sequential Extreme Learning Machine based on the Householder Block Exact Inverse QRD Recursive Least Squares. *Neurocomputing*, Volume 149, pp. 239–252

Huang, G.-B., Wang, D.H., Lan, Y., 2011. Extreme Learning Machines: A Survey. *International Journal of Machine Learning and Cybernetics*, Volume 2(2), pp. 107–122

Huang, G.-B., Zhu, Q.-Y., Siew, C.-K., 2004. Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks. *In*: Proceedings of the IEEE International Joint Conference on Neural Networks 2004, Budapest, 25 July, Hungary, pp. 985–990

Huang, G.-B., Zhu, Q.-Y., Siew, C.-K., 2006. Extreme Learning Machine: Theory and Applications. *Neurocomputing*, Volume 70(1-3), pp. 489–501

Liang, N.-Y., Huang, G.-B., Saratchandran, P., Sundararajan, N., 2006. A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks. *IEEE Transactions on Neural Networks*, Volume 17(6), pp. 1411–1423

Lim, J., 2013. Partitioned Online Sequential Extreme Learning Machine for Large Ordered System Modeling. *Neurocomputing*, Volume 102, pp. 59–64

Lu, J., Huang, J., Lu, F., 2017. Time Series Prediction based on Adaptive Weight Online Sequential Extreme Learning Machine. *Applied Sciences*, Volume 7(3), pp. 217–231

Moonen, M., Vandewalle, J., 1991. A Square Root Covariance Algorithm for Constrained Recursive Least Squares Estimation. *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, Volume 3(3), pp 163–172

Pan, C.-T., Plemmons, R., 1989. Least Squares Modifications with Inverse Factorizations: Parallel Implications. *Journal of Computational and Applied Mathematics*, Volume 27(1–2), pp. 109–127

Ridella, S., Rovetta S., Zunino, R., 1997. Circular Backpropagation Networks for Classification. *IEEE Transactions on Neural Networks*, Volume 8, pp. 81–97

Rontogiannis, A.A., Theodoridis, S., 2009. QRD-RLS Adaptive Filtering. *In*: J.A. Apolinário Jr. (ed.). *Householder-based RLS Algorithms*, Springer, pp. 9–64

Stefani, A., Xenos, M., 2009. Meta-metric Evaluation of E-Commerce-related Metrics. *Electronic Notes in Theoretical Computer Science*, Volume 233, pp. 59–72

Trefethen, L.N., Bau, D., 1997. *Numerical Linear Algebra*. Philadelphia, Penn.: SIAM (Society for Industrial and Applied Mathematics)

Yang, Z., Chen, L., Zhang, P., 2015. OS-ELM Based Real-time RFID Indoor Positioning System for Shop-floor Management. *In*: Proceedings of ELM-2014, Volume 2, pp. 233–241