EASR

Engineering and Applied Science Research

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

Identity activation structural tolerance online sequential circular extreme learning machine for highly dimensional data

Sarutte Atsawaraungsuk, Tatpong Katanyukul* and Pattarawit Polpinit

Department of Computer Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand

Received 1 November 2018
Revised 1 March 2019
Accepted 14 March 2019

Abstract

The Structural Tolerance Online Sequential Circular Extreme Learning Machine (STOS-CELM) was developed based on the Circular Extreme Learning Machine (CELM) to allow sequential learning and to mitigate the criticality of deciding the number of hidden nodes with the Householder Block exact inverse QRD Recursive Least Squares (HBQRD-RLS) algorithm. A previous study showed significant efficiency improvement using STOS-CELM with sine activation. However, sine activation is periodic. Its periodicity repeatedly maps multiple values of its input to the same output values. Within the context of STOS-CELM, input of the activation is a non-negative real value corresponding to the closeness of the data to CELM kernels. Mapping this non-negative real value to a limited range with a periodic nature causes loss of inherent information. That could restrain the STOS-CELM from reaching its full potential. This article proposes an Identity Activation Structural Tolerance Online Sequential Circular Extreme Learning Machine (ISTOS-CELM) to improve the STOS-CELM by removing the sine function to relieve this issue. Our experimental results show that ISTOS-CELM provides significantly higher accuracy than STOS-ELM and the original STOS-CELM, while retaining a comparable processing time and robustness to STOS-CELM.

Keywords: Extreme learning machine, Circular extreme machine, Online sequential · Quadratic function

1. Introduction

The Extreme Learning Machine (ELM) is a supervised algorithm that relies on a batch-mode learning mechanism. Huang et al. [1] proposed ELM to handle regression and classification problems and also reported many advantages of ELM. The results show that ELM has a faster learning process than Support Vector Machine (SVM) [2]. ELM also has been shown to have a higher accuracy than traditional learning using the Back Propagation algorithm (BP) [3].

Decherchi et al. [4] proposed a Circular Extreme Learning Machine (CELM) to improve the conventional ELM to handle higher dimensional problems. CELM is a Single-Hidden Layer Feedforward Network (SLFNs), but using a Circular Back Propagation (CBP) [5-6] architecture. CBP adds one extra dimension, which is the norm of the input. That extra dimension allows the capability to handle higher dimension problems better than the basic input formulation in ELM [6]. The additional dimension improves the overall performance of the CELM without affecting the generalization of the ELM structure.

When there is new data, batch-mode ELM and CELM have to be retrained with an entire dataset, including previously used and new data. Therefore, this convention requires significant time for retraining. To solve this problem, Liang et al. [7] proposed an Online Sequential Extreme Learning Machine (OS-ELM) that allows online update capability. However, OS-ELM may be affected by improper settings of a number of hidden nodes, which has to be determined at the beginning, such that it reduces the generalization of OS-ELM.

To address this issue, Atsawaraungsuk et al. [8] introduced a Structural Tolerance Online Sequential CELM (STOS-CELM) with a sine activation function. STOS-CELM uses the Householder block exact QR Decomposition Recursive least squares (HBQRD-RLS) [9] to allow sequential learning and to improve the robustness in online update situations. The STOS-CELM and CELM have the same input formulation, which is based on the circularity concept. The input or data point is mapped to a distance. Each distance is a Euclidean distance between its location on the input space and the corresponding center point. The center points are parameters of the model. Representing data points in distance space allows for better classification, especially in higher dimension problems [6]. STOS-CELM has been reported to be highly accurate on various datasets [8]. STOS-CELM has been used with sine activation [8].

Regarding distance representation, a data point in a higher dimensional space is likely to get mapped to longer distances than ones in low dimensional scenarios. When activated with a sine function, larger distances have a higher chance that multiple values are mapped to the same activated output. This puts sine activation under the question of whether large distances, common in higher dimensional data,

*Corresponding author. Tel.: +66 5151 0402
Email address: tatpong@kku.ac.th
doi: 10.14456/easr.2019.15

could lose significant information through the activation process due to its periodicity. That leads to a more concrete question: how does a sine activation function affect classification performance of an online CELM in a higher dimensional scenario.

This article addresses these issues and proposes an Identity activation Structural Tolerance OS-CELM or ISTOS-CELM. ISTOS-CELM is STOS-CELM without the activation function to unlock the distances of the STOS-CELM formulation. Notably, our study is different from Atsawaraungsuk et al. [8] in many aspects, including (1) our evaluation of higher dimensional datasets (c.f. low dimension data in [8]), (2) entropy analysis of activation output, (3) having a training time in perspective, and (4) extensive investigation on various activation functions. Our experiment aims to evaluate accuracy, processing time and the robustness of the ISTOSCELM and STOS-CELM when using with higher dimensional data.

The rest of this article is organized as follows. Section 2 describes the details of CELM, OS-CELM, and the proposed algorithm, ISTOS-CELM. Section 3 explains the experimental design and its results. The final section provides the conclusions of the study.

2. Materials and methods

2.1 Circular Extreme Learning Machine (CELM)

The CELM [4] was developed to extend the ELM (Extreme Learning Machine), with fast learning using the same structure as ELM. Additionally, CELM can handle higher dimensional data because of its Circular Back Propagation (CBP) architecture, which can map both linear and circular decision boundaries.

Letting CELM have K hidden nodes, the samples can be written as (x_i, t_i), i=1, 2, ..., N where $x_i \in \mathbb{R}^N$ is the training sample members of $\mathbf{X}$. Let $\mathbf{T}$ be the target sample set, which has the member $t_i \in \mathbb{R}^C$ by C, the number of classes. In the input layer of CELM, the input weights and biases are initially random numbers in the range [-1,1] and [0,1], respectively. CELM can be written in a least square form as follows:

$$\beta = \mathbf{H}^{\dagger}\mathbf{T} \tag{1}$$

where, $\beta = [\beta_{j1}, \beta_{j2}, \ldots, \beta_{jC}]^T$ j=1,2, ..., K is a matrix of output weights.

$\mathbf{H}^{\dagger}$ is the inverse of $\mathbf{H}$ from the Moore-Penrose pseudo inverse.

The CELM has the same structure as the ELM. However, CELM uses a quadratic input formulation that is the same as CBP and can be described by the following:

$$h_{ij} = br_j + \sum_{i=1}^{N} x_i w_{j,K} + w_{j,K+1} \sum_{i=1}^{N} x_i^2 \tag{2}$$

This formulation can be rewritten in a form similar to the Radial Basic Function as:

$$\mathbf{H} = [h_{ij}] = g\left(z_j \cdot \left\| x_i - c_j \right\|^2 - b_j\right) \tag{3}$$

where $g(\bullet)$ is an activation function, and:

$$z_j = w_{j,K+1,} \tag{4}$$

$$c_j = \left[-\frac{w_{j,1}}{2w_{j,K+1}}, \ldots, -\frac{w_{j,K}}{2w_{j,K+1}} \right], \tag{5}$$

$$b_j = -\frac{1}{w_{j,K+1}} \left(\sum_{k=1}^{K} \frac{w_{j,K+1}^2}{4w_{j,K+1}} - br_j \right), \tag{6}$$

where the input weights $\mathcal{W}_{j,K}$, $\mathcal{W}_{j,K+1}$, and biases br_j are randomly generated numbers.

2.2 Identity activation Structural Tolerance Sequential Circular Extreme Learning Machine (ISTOS-CELM)

The Identity activation Structural Tolerance OS-CELM (ISTOS-CELM) [8] is an alternative approach to add sequential learning to CELM. ISTOS-CELM uses the HBQRD-RLS algorithm to store and update the square root factor covariance matrix ($\tilde{\mathbf{R}}_{k-1}^{-1}$). This is done to update the output weights (β_k) when there is new data. ISTOS-CELM computational steps are summarized as follows:

In the initial step, $\tilde{\mathbf{R}}_{k-1}^{-1}$ is calculated at time k = 1, where $\tilde{\mathbf{R}}_0^{-1} \approx \mathbf{H}^{\dagger}$ by:

$$\tilde{\mathbf{R}}_0^{-1} = \mathbf{R}_0^{-1}\mathbf{Q}^T \tag{7}$$

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

$\mathbf{R}$ is an $L \times L$ upper triangular matrix that has the same values as the square root of $\mathbf{H}^T\mathbf{H}$ or $\mathbf{R}^T\mathbf{R} = \mathbf{H}^T\mathbf{H}$. Therefore, Eq. (7) can be solved in a triangular system [10-11].

The first step of HBQRD-RLS is to compute the matrix product as follows:

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

The second step of HBQRD-RLS is storing and updating $\tilde{\mathbf{R}}_k^{-T}$ followed by determining $\mathbf{F}_k$ and $\mathbf{E}_k^T$. By applying Lemma 1 in [9], HBQRD-RLS uses the Householder transformation [12-13] to produce an orthogonal matrix $\mathbf{U}(k)$ such that:

$$\mathbf{U}(k)\begin{bmatrix} \mathbf{I}_C & 0_{C\times L} \\ \mathbf{G}_k & \tilde{\mathbf{R}}_{k-1}^{-T} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_k & \mathbf{E}_k^T \\ 0_{C\times L} & \tilde{\mathbf{R}}_k^{-T} \end{bmatrix} \tag{9}$$

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

$$\beta_k = \beta_{k-1} + \mathbf{E}_k\mathbf{F}_k^{-T}(\mathbf{T}_k - \mathbf{H}_k\beta_{k-1}) \tag{10}$$

where $(\mathbf{T}_k - \mathbf{H}_k\beta_{k-1})$, $\mathbf{F}_k^{-T}$ and $\mathbf{E}_k$ are called Kalman gain.

ISTOS-CELM is summarized in **Algorithm1.**

Algorithm1 ISTOS-CELM Algorithm
Steps:
1. Initialize parameters in K hidden neurons.
   - Centers $\mathcal{W}_{j,K}$ are assigned random patterns of X_{k-1}.
   - Weights $\mathcal{W}_{j,K+1}$ are the random numbers in the range [-1,1].
   - Widths br_j are randomly defined numbers in the range [0,1].

Table 1 The dataset

Dataset	Date size	Number of attribute	Number of Classes	Data type
Arrhythmia	452	279	13	Multivariate
Audio	226	93	24	Multivariate
Autos	205	72	6	Multivariate
CNAE-9	1,080	857	9	Sequence
Gene	3,175	120	3	Multivariate
Hilln	606	101	2	Sequence
Hillwn	606	101	2	Sequence
Musk	476	168	2	Multivariate
Promoters	106	58	2	Multivariate
Soybean	683	82	19	Multivariate
Urban	168	148	9	Multivariate

Table 2 The accuracy

Dataset	ELM	OS-ELM	STOS-ELM	CELM	OS-CELM	STOS-CELM	ISTOS-CELM
Arrhythmia	0.6390	0.6430	0.6431	0.6866	0.6713	0.6892	**0.6973**
	(0.0648)	(0.0551)	(0.0544)	(0.0569)	(0.0457)	(0.0607)	**(0.0521)**
Audio	0.7682	0.7909	0.7914	0.7977	0.6863	0.8059	**0.8154**
	(0.0777)	(0.0775)	(0.0744)	**(0.0732)**	(0.0736)	(0.0842)	(0.0855)
Autos	0.6629	0.6789	0.6855	**0.6944**	0.5716	0.6832	0.6791
	(0.1017)	**(0.0963)**	(0.1041)	(0.1152)	(0.1102)	(0.1048)	(0.1084)
CNAE-9	0.8862	0.8865	0.8868	0.9174	0.9293	0.9232	**0.9399**
	(0.0267)	(0.0274)	(0.0295)	(0.0227)	**(0.0191)**	(0.0215)	(0.0220)
Gene	0.8207	0.8207	0.8217	0.7523	0.8894	0.8859	**0.8954**
	(0.0200)	(0.0202)	(0.0183)	(0.0277)	(0.0199)	(0.0161)	**(0.0148)**
Hilln	0.6868	0.6864	0.6877	0.8669	0.8539	0.8473	**0.8928**
	(0.0417)	(0.0367)	(0.0472)	(0.0328)	(0.0322)	(0.0379)	**(0.0283)**
Hillwn	0.7930	0.7855	0.7908	0.9522	0.9675	0.9422	**0.9856**
	(0.0746)	(0.0778)	(0.0723)	(0.0249)	(0.0165)	(0.0242)	**(0.0143)**
Musk	0.8520	0.8572	0.8540	0.8821	0.8120	**0.9152**	0.8994
	(0.0575)	(0.0580)	(0.0481)	(0.0415)	(0.0431)	**(0.0360)**	(0.0404)
Promoters	0.8110	0.8298	0.8352	0.8415	0.8436	0.8242	**0.8587**
	(0.1147)	(0.1068)	(0.1213)	(0.1037)	(0.1178)	(0.1218)	**(0.1006)**
Soybean	0.9422	0.9429	0.9429	0.9400	0.9227	0.9433	**0.9448**
	(0.0249)	**(0.0243)**	(0.0249)	(0.0263)	(0.0261)	(0.0245)	**(0.0260)**
Urban	0.8216	0.8221	0.8232	0.8329	0.8242	0.8459	**0.8498**
	(0.0476)	(0.0524)	(0.0465)	(0.0521)	(0.0454)	(0.0455)	**(0.0440)**
AVG	0.7894	0.7949	0.7966	0.8331	0.8156	0.8460	**0.8598**
	(0.0593)	(0.0575)	(0.0583)	(0.0525)	(0.0499)	(0.0525)	**(0.0488)**

*Note: The bold letters show the highest accuracy of each dataset.

2. Calculate the hidden layer output matrix $\mathbf{H}_0$ using Eq. (3) without an activation function, equivalent to g(a) = a.
3. Initialize $\widetilde{\mathbf{R}}_{k-1}^{-1} = \mathbf{H}^{\dagger}$ and the corresponding solution $\beta_{k-1} = \mathbf{R}^{-1}\mathbf{H}_0^T\mathbf{T}_0$. When a new sample set $\mathbf{X}_k$ comes into the system, a new hidden layer output matrix $\mathbf{H}_k$ is computed.
4. Calculate $\mathbf{G}_k$ as in Eq. (8).
5. Store and update $\widetilde{\mathbf{R}}_k^{-1}$, $\mathbf{F}_k$ and $\mathbf{E}_k$ using Eq. (9).
6. Update the new output weights β_k at the time of the new data as in Eq. (10).
7. Calculate k=k+1, when the new sample enters the training process and then go to step (4).
End

3. Results and discussion

3.1 Experimental setup

The 11 datasets of the AYRNA research group [14] and UCI (University of Irvine, California) repository [15] are used to test the performance of the investigated methods (Table 1). All methods are implemented in MATLAB Version R2014a on a computer with the Core i3 environment, 3.40 GHz Ram 4.00 GB. A 10-fold cross-validation method is used to validate the accuracy, which depends on the random input weights and biases with 10 rounds. It should be noted that results presented here may look different from [8]. The difference is from the variety of evaluation methods used. Atsawaraungsuk & Kantanyukul's [8] work evaluated only one repetition. Our work includes 10 replicates. A representative model of each method is selected from the best performing complexity, selecting a number of hidden nodes from 1 to 200.

3.2 Accuracy of the ISTOS-CELM

ISTOS-CELM is compared with five algorithms: ELM, OS-ELM [7], STOS-ELM [16], the original CELM and STOS-CELM [8] to analyze its performance. Then experimental results comparing ISTOS-CELM and the five other methods is evaluated according to their accuracy (the meta-metrics evaluation) [17-18] as:

$$\text{Accuracy}=\frac{\text{Number of correct predictions}}{\text{Number of samples}} \tag{11}$$

From the performance results, accuracy and SD (shown in brackets) of the 10-round test are shown in Table 2. ISTOS-CELM has the highest accuracy in 9 out of 11 datasets: CNAE-9, Arrhythmia, Audio, Gene, Hilln, Hillwn, Promoters, Soybean and Urban. These results show that ISTOS-CELM outperforms the other methods in most cases, which can be clearly seen in the Hilln and Hillwn datasets. It should be noted that accuracy reported in Table 2 is based on a full scale of 1. ISTOS-CELM on Arrhythmia has

Table 3 The t and p values of the t-test

Dataset	ELM	OS-ELM	STOS-ELM	CELM	OS-CELM	STOS-CELM
Arrhythmia	**-7.0082**	**-7.1683**	**-7.1988**	-1.3954	**-3.8622**	-1.0140
	(0.0000)	**(0.0000)**	**(0.0000)**	(0.1645)	**(0.0002)**	(0.3118)
Audio	**-4.0851**	**-2.1244**	**-2.1247**	-1.5717	**-10.4753**	-0.7921
	(0.0001)	**(0.0349)**	**(0.0348)**	(0.1176)	**(0.0000)**	(0.4292)
Autos	-1.0909	-0.0148	0.4262	0.9711	**-6.6769**	0.2716
	(0.2766)	(0.9882)	(0.6705)	(0.3327)	**(0.0000)**	(0.7862)
CNAE-9	**-15.5147**	**-15.1807**	**-14.4198**	**-7.1040**	**-3.5211**	**-5.4109**
	(0.0000)	**(0.0000)**	**(0.0000)**	**(0.0000)**	**(0.0005)**	**(0.0000)**
Gene	**-30.0490**	**-29.8869**	**-31.4015**	**-45.5558**	**-2.8481**	**-4.3566**
	(0.0000)	**(0.0000)**	**(0.0000)**	**(0.0000)**	**(0.0049)**	**(0.0000)**
Hilln	**-40.8318**	**-44.5372**	**-37.2941**	**-5.9813**	**-9.0307**	**-9.6297**
	(0.0000)	**(0.0000)**	**(0.0000)**	**(0.0000)**	**(0.0000)**	**(0.0000)**
Hillwn	**-25.3367**	**-25.2862**	**-26.4095**	**-11.6243**	**-6.8218**	**-15.4227**
	(0.0000)	**(0.0000)**	**(0.0000)**	**(0.0000)**	**(0.0000)**	**(0.0000)**
Musk	**-6.7519**	**-5.9672**	**-7.2340**	-2.9779	**-11.7826**	2.9208
	(0.0000)	**(0.0000)**	**(0.0000)**	**(0.0033)**	**(0.0000)**	(0.0039)
Promoters	**-3.1280**	**-1.9706**	-1.4940	-1.1958	-0.9912	**-2.1876**
	(0.0020)	(0.0502)	(0.1368)	(0.2332)	(0.3228)	**(0.0299)**
Soybean	-0.7392	-0.5358	-0.5342	-1.3117	**-5.5602**	-0.4198
	(0.4607)	(0.5927)	(0.5938)	(0.1911)	**(0.0000)**	(0.6751)
Urban	**-4.3449**	**-4.0348**	**-4.1474**	**-2.4785**	**-4.1787**	-0.6158
	(0.0000)	**(0.0001)**	**(0.0000)**	**(0.0140)**	**(0.0000)**	(0.5388)

*Note: The bold letters show *p* value that is smaller than 0.05.

Table 4 Precision, recall and F1 score

Dataset	Performance	ELM	OS-ELM	STOS-ELM	CELM	OS-CELM	STOS-CELM	ISTOS-CELM
Arrhythmia	**precision**	**0.5924**	0.5556	0.5477	0.6308	0.4748	0.5838	0.5752
	recall	**0.5680**	0.5580	0.5540	0.5665	0.4751	0.5995	0.5575
	F1	0.5799	0.5568	0.5508	0.5969	0.4750	**0.5915**	0.5662
Audio	**precision**	0.2513	0.2663	0.2716	0.3152	0.3157	**0.3387**	0.3295
	recall	0.4119	0.4211	0.3576	0.3600	**0.4471**	0.3837	0.4302
	F1	0.3122	0.3263	0.3088	0.3361	0.3701	0.3598	**0.3732**
Autos	**precision**	0.6735	0.7574	0.7893	0.8457	0.5739	0.7470	**0.8721**
	recall	0.3747	0.4752	0.5223	0.4249	0.1959	0.4038	**0.4814**
	F1	0.4815	0.5840	**0.6286**	0.5656	0.2921	0.5242	0.6203
CNAE-9	**precision**	0.8907	0.9000	0.8874	0.9195	0.9272	0.9387	**0.9442**
	recall	0.8892	0.8962	0.8816	0.9180	0.9249	0.9371	**0.9416**
	F1	0.8900	0.8981	0.8845	0.9187	0.9261	0.9379	**0.9429**
Gene	**precision**	0.8068	0.8064	0.8004	0.7007	**0.8927**	0.8675	0.8925
	recall	0.8021	0.8012	0.7996	0.7505	0.8699	0.8567	**0.8748**
	F1	0.8044	0.8038	0.8000	0.7248	0.8811	0.8621	**0.8836**
Hilln	**precision**	0.6912	0.6887	0.6911	0.8641	0.8500	0.8284	**0.8921**
	recall	0.7411	0.7385	0.7402	0.8642	0.8500	0.8284	**0.8921**
	F1	0.7153	0.7127	0.7148	0.8641	0.8500	0.8284	**0.8921**
Hillwn	**precision**	0.7956	0.7870	0.7852	0.9567	0.9697	0.9146	**0.9876**
	recall	0.8331	0.8376	0.8273	0.9573	0.9692	0.9151	**0.9876**
	F1	0.8139	0.8115	0.8057	0.9570	0.9695	0.9148	**0.9876**
Musk	**precision**	0.8498	0.8598	0.8544	0.8813	0.7993	**0.9152**	0.9022
	recall	0.8474	0.8578	0.8518	0.8813	0.7971	**0.9152**	0.8999
	F1	0.8486	0.8588	0.8531	0.8813	0.7982	**0.9152**	0.9010
Promoters	**precision**	0.8335	0.8394	0.8394	0.8278	**0.8833**	0.8204	0.8511
	recall	0.8321	0.8386	0.8386	0.8301	**0.8900**	0.8196	0.8491
	F1	0.8328	0.8390	0.8390	0.8290	**0.8866**	0.8200	0.8501
Soybean	**precision**	**0.9726**	0.9788	0.9664	0.9659	0.9551	0.9698	0.9689
	recall	**0.9740**	0.9734	0.9692	0.9653	0.9494	0.9683	0.9650
	F1	**0.9733**	0.9761	0.9678	0.9656	0.9522	0.9691	0.9669
Urban	**precision**	0.8126	0.8160	0.8098	0.8157	0.8172	0.8397	**0.8415**
	recall	0.8386	0.8555	0.8446	0.8673	0.8604	**0.8729**	0.8687
	F1	0.8254	0.8353	0.8268	0.8407	0.8383	**0.8560**	0.8548
AVG	**precision**	0.8170	0.8255	0.8243	0.8723	0.8459	0.8764	**0.9057**
	recall	0.8112	0.8253	0.8187	0.8386	0.8229	0.8500	**0.8748**
	F1	0.8077	0.8202	0.8180	0.8480	0.8239	0.8579	**0.8839**

*Note: The bold letters show the highest precision, recall, and F1 score of each dataset.

an accuracy of 0.6973, equivalent to 69.73% (5% higher than ELM).

The t-test is used to verify the results comparing ISTOS-CELM and the other methods. Table 3 shows the corresponding pairs of *t*-values and *p*-values (in brackets). Bold letters indicate that the null-hypothesis is rejected when *p*-value is smaller than 0.05 and the *t* value is negative, indicating that the ISTOS-CELM performs significantly better than other methods. The results show that the ISTOS-CELM has significantly higher accuracy than the STOS-CELM with 5 out of 11 datasets: CNAE-9, Gene, Hilln, Hillwn and Promoters.

From the precision, recall and F1 results in Table 4, ISTOS-CELM has the highest accuracy in 6 out of 11 datasets: CNAE-9, Auto, Audio, Gene, Hilln, and Hillwn. These results show that ISTOS-CELM outperforms other methods in most cases. This can be clearly seen in the Hilln and Hillwn datasets.

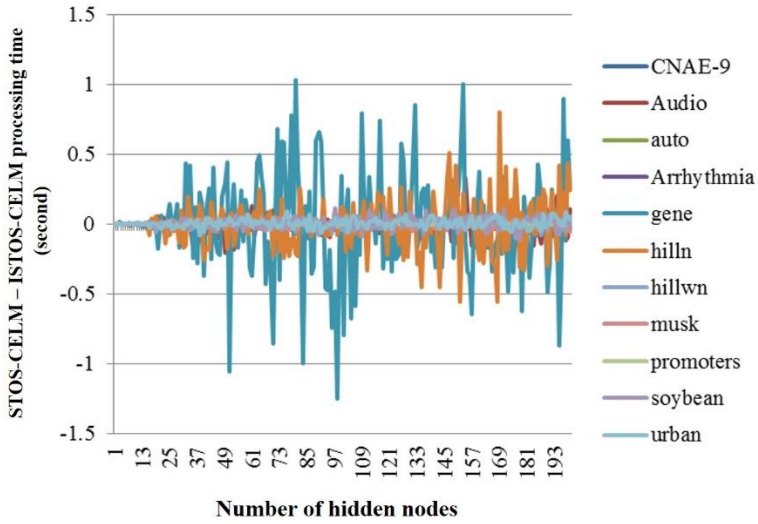


Figure 1 Time difference between STOS-CELM and ISTOS-CELM on every dataset

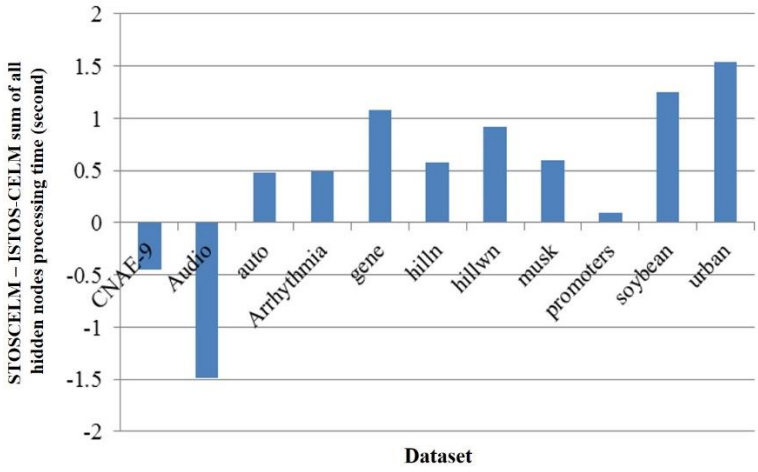


Figure 2 Sum of all hidden node time differences between STOS-CELM and ISTOS-CELM on every dataset

From the experimental result, STOS-CELM works well using a periodic activation function, while ISTOS-CELM performs using a linear activation function. This fact puzzles us since these activation functions are of totally different families. Since an activation function maps input information, **X,** to feature information, $g(\mathbf{H})$, our assumption is that information loss during mapping affects the final performance. Our investigation on this assumption is presented in the APPENDIX.

3.3 Performance comparison of ISTOS-CELM and other classification algorithms

This section presents comparison of ISTOS-CELM with three algorithms: Naïve Bayes (NB) [19], Back Propagation (BP) [20], and Support Vector Machine (SVM) [21]. The four competing methods are evaluated on the basis of accuracy, SD, precision, recall, F1 score, and training time (in seconds) as shown in Table 5.

BP, NB, and SVM results are implemented as MATLAB functions: patternnet, fitcecoc, and fitcnb functions for BP, NB, and SVM, respectively.

From the results in Table 5, ISTOS-CELM has the highest accuracy in 8 out of 11 datasets: Arrhythmia, Audio, CNAE-9, Hilln, Hillwn, Musk, Soybean, and Urban. These results imply the viability of ISTOS-CELM. CELM training time is comparable to SVM and higher than NV.

3.4 Processing time comparison between STOS-CELM and ISTOS-CELM

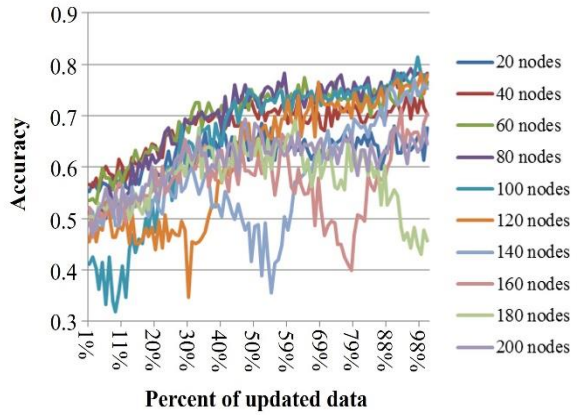
Processing time comparison is made to investigate the different results of the STOSCELM and ISTOS-CELM. The time difference is used as measurement index. The calculation results are shown in Figure 1.

Figure 1 shows the processing time difference between STOS-CELM and ISTOS-CELM. If the time difference values are higher than zero then the processing time of STOS-CELM is longer than those of ISTOS-CELM. Conversely, if the time difference values are less than zero then the processing time of ISTOS-CELM is the longer of the two. The magnitudes of time differences for most datasets are small. However, the time differences for the Gene and Hilln datasets are significantly higher. Figure 2 shows the time differences as the sum of processing time of STOS-CELM and those of ISTOS-CELM close to zero, and the sums of processing times over all hidden nodes for each dataset. The results show that ISTOS-CELM has a slightly lower processing time than the STOS-CELM in 9 out of 11 datasets.

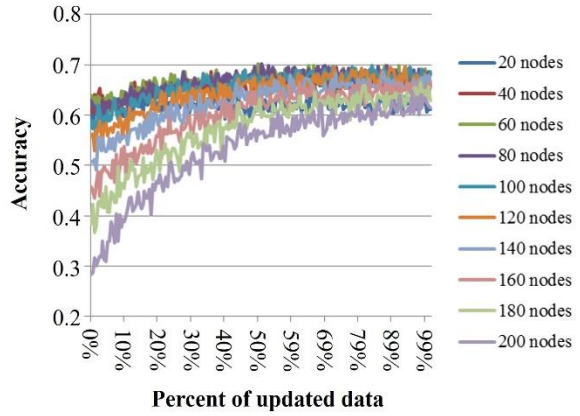
Table 5 Performance comparison of NB, BP, SVM, and ISTOS-CELM

Dataset	Performance	NB	BP	SVM	ISTOS-CELM
Arrhythmia	**accuracy**	0.3655	0.5714	0.6931	**0.6973**
	SD	0.0817	0.0981	0.1175	**0.0521**
	precision	0.1919	0.4550	0.2748	**0.5752**
	recall	0.4082	0.4722	0.3270	**0.5575**
	F1	0.2610	0.4634	0.2987	**0.5662**
	training time	**0.0297**	0.5094	0.0734	0.1000
Audio	**accuracy**	0.5176	0.6526	0.6748	**0.815**4
	SD	0.0746	**0.0579**	0.0619	0.0855
	precision	0.0625	0.2233	0.1386	**0.3295**
	recall	0.0324	0.3193	0.2296	**0.4302**
	F1	0.0426	0.2628	0.1729	**0.3732**
	training time	**0.1516**	1.8422	0.7344	0.5016
Autos	**accuracy**	0.6368	0.7075	**0.7567**	0.6791
	SD	0.1149	0.0877	**0.0778**	0.1084
	precision	0.2146	0.3118	0.3570	**0.8721**
	recall	0.2142	0.3077	0.3647	**0.4814**
	F1	0.2144	0.3097	0.3608	**0.6203**
	training time	**0.0719**	0.8625	0.9750	0.1156
CNAE-9	**accuracy**	0.7528	0.8713	**0.9444**	0.9399
	SD	0.0378	0.2391	0.0180	**0.022**
	precision	0.7566	0.8727	0.8900	**0.9442**
	recall	0.7759	0.8849	0.8874	**0.9416**
	F1	0.7661	0.8788	0.8887	**0.9429**
	training time	**0.3422**	12.6063	0.7484	3.7625
Gene	**accuracy**	0.8954	0.8784	**0.9137**	0.8954
	SD	0.0203	0.0174	0.0163	**0.0148**
	precision	0.8869	0.8742	0.8851	**0.8925**
	recall	0.8806	0.8590	**0.8789**	0.8748
	F1	**0.8837**	0.8665	0.8820	0.8836
	training time	**0.0719**	1.2250	3.9406	2.2266
Hilln	**accuracy**	0.4670	0.5033	0.1206	**0.8928**
	SD	**0.0183**	0.0539	0.2178	0.0283
	precision	0.4683	0.5032	0.4684	**0.8921**
	recall	0.4670	0.5032	0.4671	**0.8921**
	F1	0.4677	0.5032	0.4678	**0.8921**
	training time	**0.0422**	0.7797	0.1531	0.4625
Hillwn	**accuracy**	0.4884	0.5100	0.4092	**0.9856**
	SD	0.0481	0.1035	0.1473	**0.0143**
	precision	0.4745	0.5144	0.4746	**0.9876**
	recall	0.4602	0.5150	0.4606	**0.9876**
	F1	0.4673	0.5147	0.4675	**0.9876**
	training time	**0.0578**	1.3969	0.1656	0.5000
Musk	**accuracy**	0.5462	0.8804	0.0799	**0.8994**
	SD	0.0375	0.0512	**0.0330**	0.0404
	precision	0.5000	0.8790	0.5004	**0.9022**
	recall	0.2731	0.8793	0.5230	**0.8999**
	F1	0.3533	0.8792	0.5114	**0.901**
	training time	0.0406	1.0734	**0.0359**	0.3016
Promoters	**accuracy**	**0.9245**	0.6427	0.8691	0.8587
	SD	**0.0752**	0.1518	0.0906	0.1006
	precision	**0.9227**	0.6509	0.9073	0.8511
	recall	**0.9257**	0.6582	0.9100	0.8491
	F1	**0.9242**	0.6545	0.9086	0.8501
	training time	0.0219	0.3484	**0.0172**	0.0672
Soybean	**accuracy**	0.8784	0.9239	0.9399	**0.9448**
	SD	0.0270	**0.0214**	0.0290	0.026
	precision	0.9338	0.9556	0.9680	**0.9689**
	recall	0.9236	0.9590	0.9560	**0.965**
	F1	0.9287	0.9573	0.9620	**0.9669**
	training time	**0.0672**	1.7406	1.0469	0.2141
Urban	**accuracy**	0.1585	0.7998	0.8428	**0.8498**
	SD	**0.0234**	0.0501	0.0517	0.0440
	precision	0.1033	0.7832	0.1753	**0.8415**
	recall	0.1422	0.8069	0.3269	**0.8687**
	F1	0.1197	0.7949	0.2282	**0.8548**
	training time	**0.0703**	1.0813	0.3797	0.4109
AVG	**accuracy**	0.6028	0.7219	0.6586	**0.8598**
	SD	0.0508	0.0847	0.0783	**0.0488**
	precision	0.5014	0.6385	0.5490	**0.9057**
	recall	0.5003	0.6513	0.5756	**0.8748**
	F1	0.4935	0.6441	0.5590	**0.8839**
	training time	**0.0879**	2.1332	0.7518	0.7875

*Note: The bold letters show the best evaluating value of each dataset.

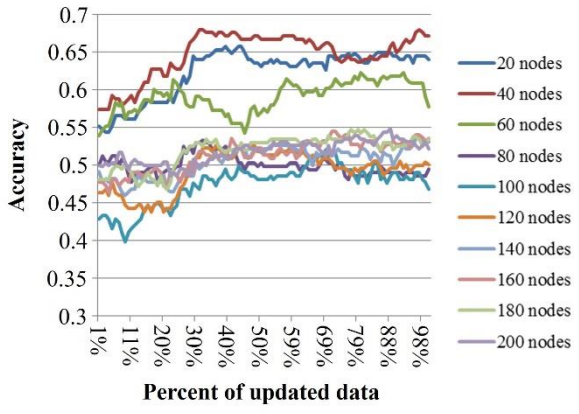


(a) Audio dataset

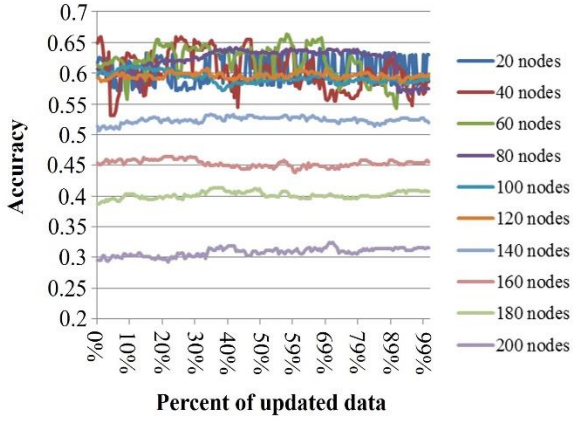


(b) Arrhythmia dataset

Figure 3 Accuracy of CELM with various numbers of hidden nodes



(a) Audio dataset



(b) Arrhythmia dataset

Figure 4 Accuracy of OS-CELM with various numbers of hidden nodes

3.5 Robustness of the ISTOS-CELM when updating the data with different hidden nodes

The aim of this subsection is to analyze the robustness of the ISTOS-CELM as a result of using HBQRD-RLS, in particular, when the hidden nodes are varied [16]. Furthermore, this experiment shows that the robustness of the ISTOS-CELM is stable even when the data is updated. Figures 3-6 show the accuracy of CELM, OS-CELM, STOS-CELM and ISTOS-CELM with various percentages of updated data in different numbers of hidden nodes. Each figure demonstrates the accuracy using two datasets, Audio and Arrhythmia.

Figure 3 shows that the accuracy trend when the data of CELM is updated. The result shows that the accuracy increases when data is updated in both datasets. However in the Audio dataset, some periods using 20, 120, 140, 160 and 180 hidden nodes for accuracy show a trend that drops sharply and then increases.

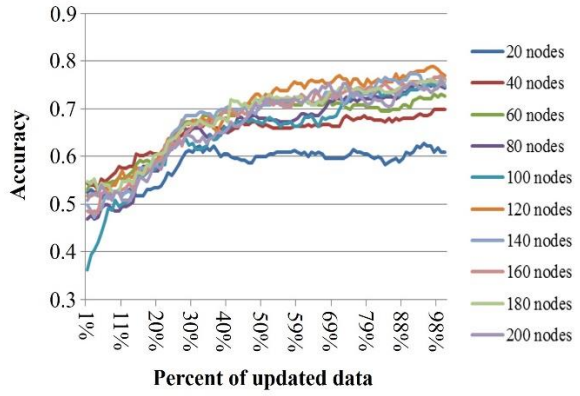
Figure 4 shows that most of the OS-CELM accuracy unexpectedly did not increase even when the data was updated, especially with the number of hidden nodes was varied between 100-200 in the Arrhythmia dataset. However, the accuracy using the Arrhythmia dataset fluctuated considerably without increasing when using 20-80 hidden nodes.

Figure 5 shows that the accuracy on both Audio and Arrhythmia using STOS-CELM tends to increase with the percentage of updated data at all settings with various numbers of hidden nodes. From Figure 5, STOS-CELM accuracy has trends to increase while updating the data to ISTOS-CELM in all hidden nodes of both datasets. Figure 6 shows a trend of increasing accuracy for ISTOS-CELM that is similar STOS-CELM.

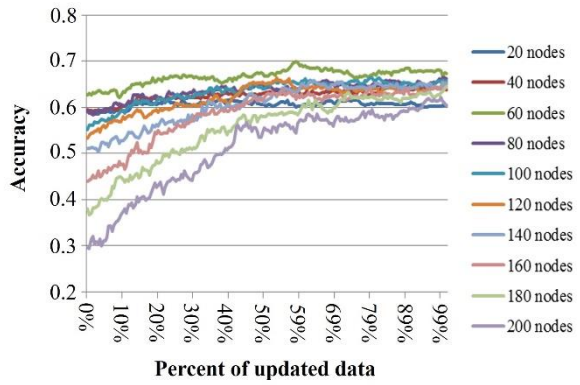
4. Discussion

Our proposed method, ISTOS-CELM, does not employ sine activation, but rather uses the CELM distances directly for classification. The CELM distance formulation is quadratic, which allows for a nonlinear decision boundary. Unlike ELM, whose input formulation is linear so that non-linear activation is essential, CELM distance formulation is natively non-linear. Therefore, non-linear activation is not needed as much. This argument is supported by ISTOS-CELM performance, shown in three ways in the experimental results, as follows.

(1) Accuracy. ISTOS-CELM has a significantly higher accuracy than STOS-CELM. Our initial explanation proposes a clustering resulting from periodic behavior. That is, the STOS-CELM uses the sine function as its activation function. A sine function is periodic. This behavior clusters multiple values into the same activated output. This clustering causes unnecessary loss of information and might

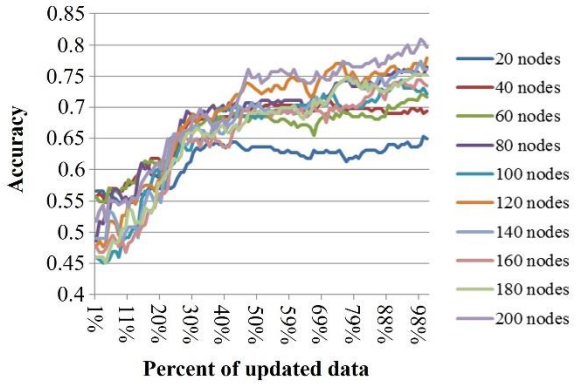


(a) Audio

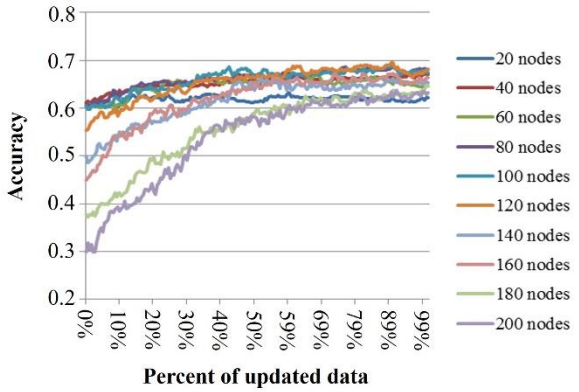


(b) Arrhythmia

Figure 5 Accuracy of STOS-CELM with different number of hidden nodes



(a) Audio



(b) Arrhythmia

Figure 6 Accuracy of ISTOS-CELM with different number of hidden nodes

be a source of STOS-CELM performance degradation. However, our investigation using entropy presented in the APPENDIX does not support this explanation, since the entropy of the sine activation functions is higher than the input and other activation functions. This contradiction of our initial explanation might not be entirely accurate since the entropy in our investigation may not be sufficiently representative.

A strong correlation between mapping entropy and best performance, such as those using activation functions with high entropy, is likely to result in high performance. Our second observation is that mapping entropy of an identity function is closer to entropy of the input data than to the sine function. Our third observation is that the identity function has better efficacy than the sine function, such that when (i) the entropy of the identity function is smaller than that of the sine function, the accuracies of both functions are relatively comparable, and (ii) the entropies of identity and sine functions are roughly in the same level, the accuracy of the identity function is clearly higher than that of the sine function. A sine function may unnecessary disturb the mapping entropy, so that it has lower efficacy than the identity function.

(2) Processing time. The ISTOS-CELM has processing times comparable to the STOS-CELM. Although ISTOS-CELM is less computationally intensive as it has no activation function, the activation computation in STOS-CELM requires a very short time. This explains a little of the difference in the processing times of both methods. Figure 2 shows that ISTOS-CELM requires slightly shorter processing times than the STOS-CELM.

(3) Robustness. As the experimental results show, STOS-CELM and ISTOS-CELM are slightly affected by online updating, even when using pre-selected hidden nodes. That is, accuracies obtained from both STOS-CELM and ISTOS-CELM show a strongly increasing trend as more training data is added in an online fashion. This is credited to a key of STOS-CELM and ISTOS-CELM using the HBQRD-RLS (Householder Block exact inverse QR Decomposition Recursive Least Square). The increasing relationship between accuracy and data update also can be observed in CELM (Figure 3a), but the CELM increase is noticeably weaker than those of STOS-CELM (Figure 5a) and ISTOS-CELM (Figure 6a). It is notable that the OSCELM (Figures 4a-b) is very sensitive to this.

The ISTOS-CELM method allows training of a large dataset in a sequential manner without the need to repeat the entire training process and avoid the negative effects of online data updating, for which OSCELM is notoriously. Furthermore, ISTOS-CELM has been shown to handle highly dimensional data well. These characteristics of ISTOS-CELM may well address the challenges of big data. As the present is the era of big data, much data is processed automatically on the web to respond in real time. This is commonly seen in the applications and features of Facebook, YouTube, and Google. For example, identification of people currently used by Facebook learns a large amount of highly dimensional data in real time. This kind of application could be addressed by ISTOS-CELM. Specifically, images dynamically tagged can be used in online training. ISTOS-

CELM is designed for such situations and hopefully ISTOS-CELM will become a viable tool in this big data era.

The accuracy of the ISTOS-CELM still depends on the number of the hidden nodes that are held fixed from the onset. Our research group envisions a mechanism to allow additional nodes as the process proceeds. Such a mechanism should be able to monitor the prediction error and restructure the model to best suit its current state. This would allow more flexibility to ISTOS-CELM design and lead to wider applications of the technique.

5. Conclusions

This paper proposes an Identity activation Structure Tolerance Online Sequential Circular Extreme Learning Machine (ISTOS-CELM) based on the Householder Block exact inverse QRD-RLS (HBQRD-RLS) algorithm to improve STOS-CELM in highly dimension problems. The proposed method was evaluated on 11 classification problems using several performance statistics. The results showed that the quality of the solution from ISTOS-CELM was better than ELM, OS-ELM, CELM, OS-CELM, and STOS-CELM to a statistically significant level in most problems. Furthermore, ISTOSCELM has comparable processing time to STOS-CELM, while retaining the same level of robustness.

6. Acknowledgements

This research was partially supported by the Faculty of Computer Engineering, Khon Kaen University, Thailand.

7. References

[1] Huang GB, Zhu QY, Siew CK. Extreme learning machines: theory and applications. Neurocomputing. 2006;70(1):489-501.

[2] Huang GB, Wang DH, Lan Y. Extreme learning machines: a survey. Int J Mach Learn Cyb. 2011;2(2): 107-22.

[3] Huang GB, Zhou H, Ding X, Zhang R. Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern B Cybern. 2012;42(2):513-29.

[4] Decherchi S, Gastaldo P, Zunino R, Cambria E, Redi J. Circular-elm for the reduced reference assessment of perceived image quality. Neurocomputing. 2013;102: 78-89.

[5] Ridella S, Rovetta S, Zunino R. Circular backpropagation networks for classification. IEEE Trans Neural Network. 1997;8(1):84-97.

[6] Gastaldo P, Zunino R, Heynderickx I, Vicario E. Circular back-propagation networks for measuring displayed image quality. Lect Notes Comput Sci. 2002; 2415:1219-24.

[7] Liang NY, Huang GB, Saratchandran P, Sundararajan N. A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Trans Neural Network. 2006;17(6):1411-23.

[8] Atsawaraungsuk S, Katanyukul T. Sin activation structural tolerance of online sequential circular extreme learning machine. Int J Tech. 2017;8(4):601-10.

[9] Apolinario JA, editor. QRD-RLS adaptive filtering. USA: Springer; 2009.

[10] Pan CT, Plemmons R. Least squares modifications with inverse factorizations: parallel implications. J Comput Appl Math. 1989;27(1-2):109-27.

[11] Moonen M, Vandewalle J. A square root covariance algorithm for constrained recursive least squares estimation. J VLSI Sign Process Syst Sign Image Video Technol. 1991;3(3):163-72.

[12] Golub GH, Van Loan CF. Matrix computations. 4th ed. USA: Johns Hopkins University Press; 2013.

[13] Trefethen LN, Bau III D. Numerical linear algebra. USA: Society for Industrial and Applied Mathematics; 1997.

[14] AYRNA research group. AYRNA dataset [Internet]. 1994 [cited 2017 Sep]. Available from: http://www.uco.es/grupos/ayrna/index.php/en/investigacion-y-difusion/partitions-and-datasets

[15] UCI machine learning repository. UCI dataset [Internet]. 1987 [cited 2017 Sep] Available from: https://archive.ics.uci.edu/ml/datasets.html.

[16] Horata P, Chiewchanwattana S, Sunat K. Enhancement of online sequential extreme learning machine based on the householder block exact inverse qrd recursive least squares. Neurocomputing. 2015;149:239-52.

[17] Stefani A, Xenos M. Meta-metric evaluation of e-commerce-related metrics. Electron Notes Theor Comput Sci. 2009;233:59-72.

[18] Horata P, Chiewchanwattana S, Sunat K. Robust extreme learning machine. Neurocomputing. 2013; 102:31-44.

[19] McCallum A, Nigam K. A comparison of event models for naive Bayes text classification. AAAI-98 workshop on learning for text categorization; 1998 Jul 26-27; Wisconsin, USA. USA: AAAI Press; 1998. p. 41-8.

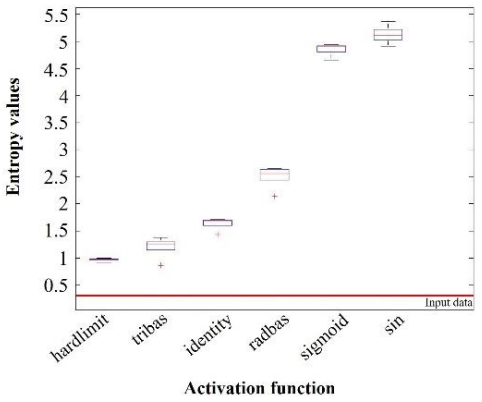
[20] LeCun Y, Boser BE, Denker JS, Henderson D, Howard RE, Hubbard WE, et al. Handwritten digit recognition with a back-propagation network. In: Touretzky DS, editor. Advances in neural information processing systems 2. San Francisco: Morgan Kaufmann Publishers; 1990. p. 396-404.

[21] Joachims T. Text categorization with support vector machines: learning with many relevant features. In: Nédellec C, Rouveirol C, editors. 10th European Conference on Machine Learning; 1998 Apr 21-23; Chemnitz, Germany. Berlin: Springer; 1998. p. 137-42.

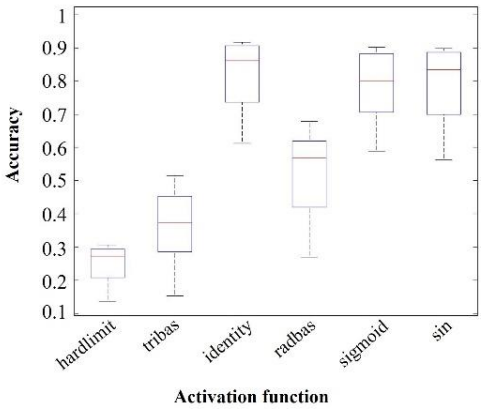
8. Appendix

8.1 The effect of the activation function that affect to the STOS-CELM

Each figure (Figures 7-10) consists of 2 sub-figures. Sub-figure (a) shows the entropy of **H** matrix of STOS-CELM that passes through each activation function with various numbers of hidden nodes (10, 20, 50, 100 and 200 with red horizontal lines depicting the entropy of input data. Sub-figure (b) shows the accuracies. The sequence of the activation function in sub-figure (a) and sub-figure (b) are in ascending order of their entropies. Figures 7-10 show the samples from all of the dataset. From the Figures 7-10, we can divide the results into two groups as follows, (1) when the entropy of the identity function is smaller than that of the sine function, and (2) when the entropies of identity and sine functions are roughly equal. Only the Hilln and Hillwn datasets are in the second group, while the others are in the first group.

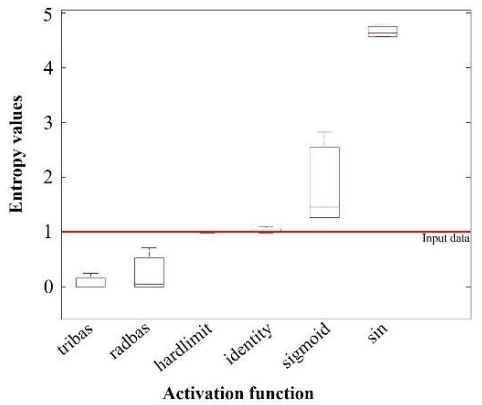


(a) Entropy

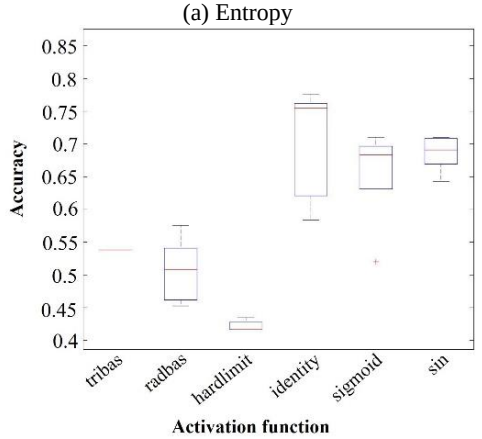


(b) Accuracy

Figure 7 Arrhythmia dataset

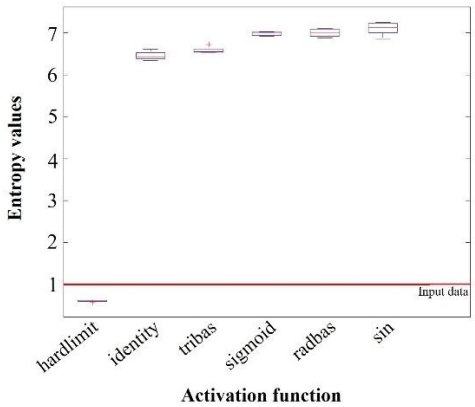


(a) Entropy

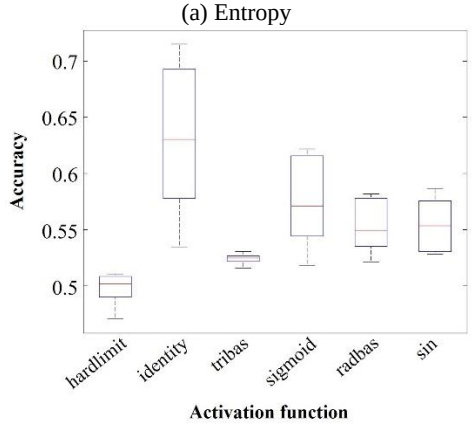


(b) Accuracy

Figure 8 Audio dataset

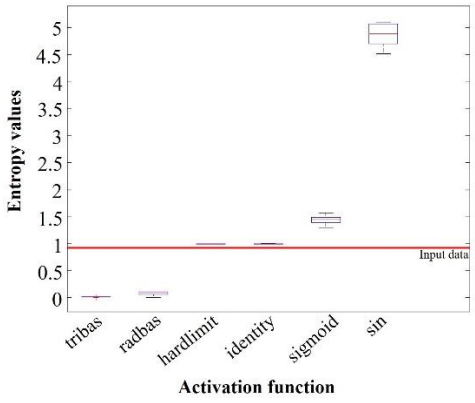


(a) Entropy

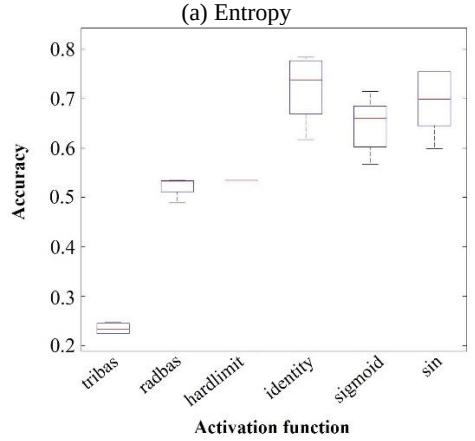


(b) Accuracy

Figure 9 Hilln dataset



(a) Entropy



(b) Accuracy

Figure 10 Hillwn dataset