Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 3092 – 3106

doi:10.3906/elk-2103-122

Improved online sequential extreme learning machine: OS-CELM

Olcay TOSUN*, **Recep ERYİĞİT**
Department of Computer Engineering, Ankara University, Ankara, Turkey

Received: 25.03.2021 • **Accepted/Published Online:** 16.06.2021 • **Final Version:** 30.11.2021

Abstract: Online learning methods (OLM) have been gaining traction as a solution to classification problems because of rapid renewal and fast growth in volume of available data. ELM-based sequential learning (OS-ELM) is one of the most frequently used online learning methodologies partly due to fast training algorithm but suffers from inefficient use of its hidden layers due to the random assignment of the parameters of those layers. In this study, we propose an improved online learning model called online sequential constrained extreme learning machine (OS-CELM), which replaces the random assignment of those parameters with better generalization performance using the CELM method based on the distance between classes. We compare the performance and training times of OS-ELM, ELM, and the proposed models for four different data sets. The results indicate that the proposed model has better generalization and accuracy performance.

Key words: Machine learning, extreme learning machine (ELM), constrained extreme learning machine (CELM), online sequential CELM (OS-CELM)

1. Introduction

With the phenomenal increase in production and transmission of data necessary for continuation of human endeavour in the current era, autonomous systems and artificial intelligence applications are becoming the essential tools to make sense, process, and manage the data related tasks that are personal to global in scope. Artificial neural networks (ANN) have been the focus of interest by researchers since Rosenblatt published the first training algorithm of an artificial neural network (perceptron) in 1958[1]. Researchers have developed ANNs and training algorithms in different study structures. Back propagation (BP) and feedforward neural networks (FFNN) are the most well-known ANN types. In both types, training speeds are slow due to slow gradient-based training algorithms and all parameters are iteratively adjusted using these algorithms[2]. The training speed of FFNN and BP is slow in cases such as online learning, where model training needs to be carried out close to the data flow rate or big data applications where the data cannot fit into memory [3, 4]. In addition to the slowness of the training speeds of ANNs, the number of neurons and the number of hidden layers in ANNs need to be increased in order to have high accuracy rates, which is one of the main purposes of machine learning. Depending on the neuron number, this situation causes an increase in training time [5].

In 2004, Huang et al. developed a new scheme for FFNN called extreme learning machine (ELM) in order to shorten the training time in ANNs. ELM is a single layer FFNN. Since there is no need to update neuron parameters in ELM, the training speed is very fast. Weights and biases of neurons are assigned randomly. In

*Correspondence: olcaytosun@gmail.com

ELM, nonlinear systems are treated like linear systems and the solution of the system is reached by using the least squares method. The popularity of ELM has been increased in recent years due to enabling fast ANN training and it has a comparable performance to other machine learning algorithms [2, 6–8].

Many ELM based algorithms have been developed. RKELM[9] developed by Deng et al. to shorten the training time in embedded systems, IDS-ELM[10] developed by Xu and Wang for the classification of video images, and DW-ELM [11] developed by Wang et al. to use big data are just some of them. Fast training speed also creates an advantage in online learning. Liang et al. developed an ELM-based online sequential extreme learning machine (OS-ELM) structure for single-hidden layer feedforward neural networks[4]. OS-ELM can perform fast training from one by one or chunk by chunk data sequentially. In cases where the entire data set is available, it conducts model training as ELM. Despite the fast training speed, random selection of hidden layer parameters in ELM and other ELM-based algorithms prevents the effective use of hidden layer neurons [8]. This negatively affects the performance of ELM and ELM based algorithms. Zhang, Li, and Xiao proposed a model for enhancing the generalization performance of OS-ELM for modeling problems in nonstationary environments (AOS-ELM) [28]. Zhang et al. used the genetic algorithm (GA) to overcome the issue of random assignments of hidden layer input parameters [29].

In this study, an improved model called online sequential constrained extreme learning (OS-CELM) has been proposed for OS-ELM, which can perform online and sequential learning, to perform better classification and generalization performance. Performance tests are conducted on popular MNIST, skin segmentation, wireless indoor localization, and Cleveland heart disease datasets to measure the effectiveness of the proposed model.

2. Related works

2.1. Extreme learning machine (ELM)

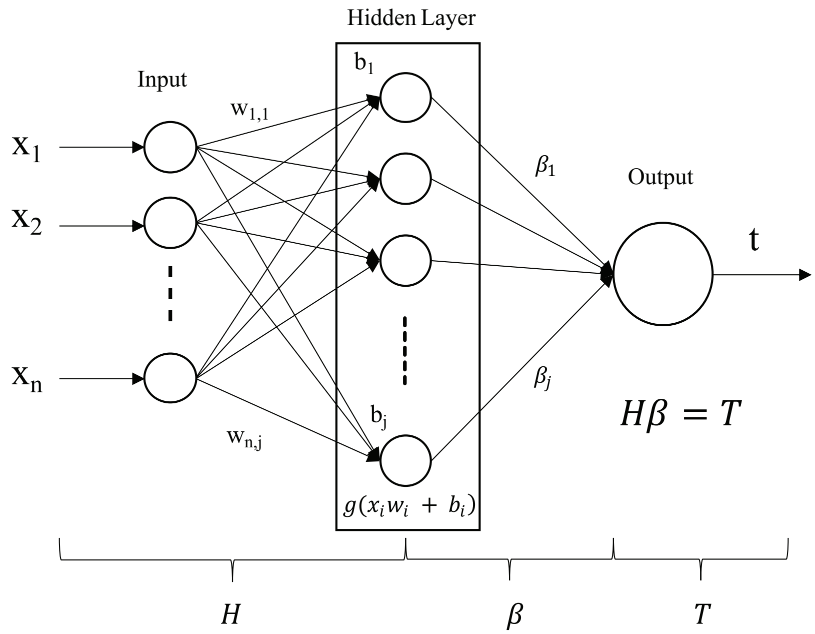
Huang et al. [2] proposed an algorithm designed for single-layer feed forward neural networks called ELM. Contrary to the traditional feed-forward or back-propagation learning algorithms which use gradient-based techniques, ELM calculates the output weights analytically from the randomly assigned hidden neuron weights and bias values. All of its parameters are set once which eliminates the need for repetitive training. The performance and speed of the model depends on randomly assigned input weights.

ELM structure is illustrated in Figure 1. Output weights linking the hidden layer to the output layer are obtained by solving the linear system $H\beta = T$. Existence of the solution to the linear system depends on the singularity as well as the shape of the matrix H. If $\{(x_i, t_i) \,|x_i \in R^n, t_i \in R^m\}$ for $n \neq m$, H matrix is not a square matrix and the system can be solved by using the pseudo-inverse method of Moore–Penrose as $\beta = H^{\dagger}T$ where:

$$H^{\dagger} = \begin{cases} H^T(HH^T)^{-1} & \text{if } m < n \\ (H^TH)^{-1}H^T & \text{if } n < m \end{cases} \tag{1}$$

ELM algorithm can be summarized as follows:

- Step-1: Initialization
  Assign w_i and b_i values randomly.
- Step-2: Calculate hidden layer input, H.
- Step-3: Calculate output, $\beta' = H^{\dagger}T$ where $T = [t_1, t_2, \ldots, t_N]^T$.



$$H = \begin{bmatrix} g(w_1.x_1 + b_1) & \cdots & g(w_j.x_1 + b_j) \\ \vdots & \cdots & \vdots \\ g(w_1.x_N + b_1) & \cdots & g(w_j.x_N + b_j) \end{bmatrix}_{NxJ} \quad \beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_j^T \end{bmatrix}_{JxM} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{NxM}$$

Figure 1. ELM Network. Input is $x_i = (x_1, x_2, \ldots, x_n)^T\ R^n$, activation function is $g()$, i-th hidden neuron weight is w_i, i-th hidden neuron bias is b_i, β are hidden layer neuron output weights, and T is the output.

2.2. Constrained extreme learning machine (CELM)

Constrained extreme learning machine (CELM) algorithm was proposed by Zhu et al. [5] to calculate the hidden layer parameters. The main feature differentiating the CELM from the original ELM is the way it handles the hidden layer parameters. Instead of randomly assigning those parameters, Zhu et al. found that calculating the hidden layer weights and bias values by using the distances between classes improved the performance of the system. According to this algorithm, random vectors between different classes are drawn and the distance between classes is calculated. Weights and bias values of the hidden layer neurons are assigned depending on the calculated distances. To calculate the distance between classes in CELM, one of the classes (c_1) is mapped to 1 and the other class (c_2) is mapped to –1. For samples from two classes x_{c_1} and x_{c_2}, the assumption can be written as:

$$\begin{aligned} ax_{c_1}(x_{c_2} - x_{c_1}) + b &= -1 \\ ax_{c_2}(x_{c_2} - x_{c_1}) + b &= 1. \end{aligned}$$

Normalization factor a and the bias b can be calculated as:

$$
\begin{aligned}
a &= \frac{2}{\|x_{c_2} - x_{c_1}\|^2} \\
b &= \frac{(x_{c_1} + x_{c_2})(x_{c_1} - x_{c_2})}{\|x_{c_2} - x_{c_1}\|^2},
\end{aligned} \tag{2}
$$

where $(x_{c_1} - x_{c_2})$ is the difference vector. As a result of the normalization of the difference vector, weight vector can be calculated as:

$$
w = \frac{2\,(x_{c_1} - x_{c_2})}{\|x_{c_2} - x_{c_1}\|^2}. \tag{3}
$$

Instead of random assignment of weight and bias values, it can be calculated by using Equation (2) and Equation (3) based on distances between classes.

2.3. Online sequential extreme learning machine (OS-ELM)

Training speed is important in online learning. In some cases, the model update needs to keep up with the data flow rate. In such cases, ELM-based online learning models providing fast model training are used. Liang et al. developed ELM-based OS-ELM algorithm for online learning in 2006[4]. OS-ELM consists of two parts. First training part and then second part where the sequential training is done. β given in Equation (1) is the least squares solution of linear system. $(H^T H)$ in $H^{\dagger} = \left(H^T H\right)^{-1} H^T$ solution tends to be singular. Since the singular matrix cannot be inverted, the solution may not be found. To avoid this situation, the number of hidden layer neurons can be reduced, or larger initial training data can be selected. This situation is important in the first training phase of OS-ELM. OS-ELM algorithm can be written as:

- Step-1: Assign the values w, β and b randomly.
- Step-2: Calculate the following values.

$$
\begin{aligned}
H_0 &= G\,(w \cdot x_0 + b) \\
P_0 &= \left(H_0^T H\right)^{-1} \\
\beta_0 &= P_0 H_0^T t_0.
\end{aligned}
$$

- Step-3: Calculate following values as long as new data arrives.

$$
\begin{aligned}
P_{k+1} &= P_k - P_k H_{k+1}^T \left(I + H_{k+1} P_k H_{k+1}^T\right)^{-1} H_{k+1} P_k \\
\beta_{k+1} &= \beta_k + P_{k+1} H_{k+1}^T \left(T_{k+1} - H_{k+1} \beta_k\right).
\end{aligned}
$$

3. Proposed method

OS-ELM randomly assigns input values of the hidden layer. This situation prevents the effective use of hidden layers. To solve this issue, we use distances between classes for effective use of input parameters of hidden layers in this study. Assigning parameters of hidden layer neurons based on the difference vectors between class samples help the output parameters of hidden layer neurons to reflect differences between classes. In this way, more successful models in classification problems can be created. In the proposed method, we use the distinguishing features between the classes in the assignment of the initial parameters in order to provide a better classification performance of the OS-ELM model, which is capable of online and sequential learning. To achieve this, we assign hidden layer parameters using the method stated in CELM [5]. This proposed model is named OS-CELM. We calculate the distance between class members to determine differences between classes. To achieve this, random vectors are drawn between class members. Randomly drawn vectors between classes used in the proposed model are shown in Figure 2.

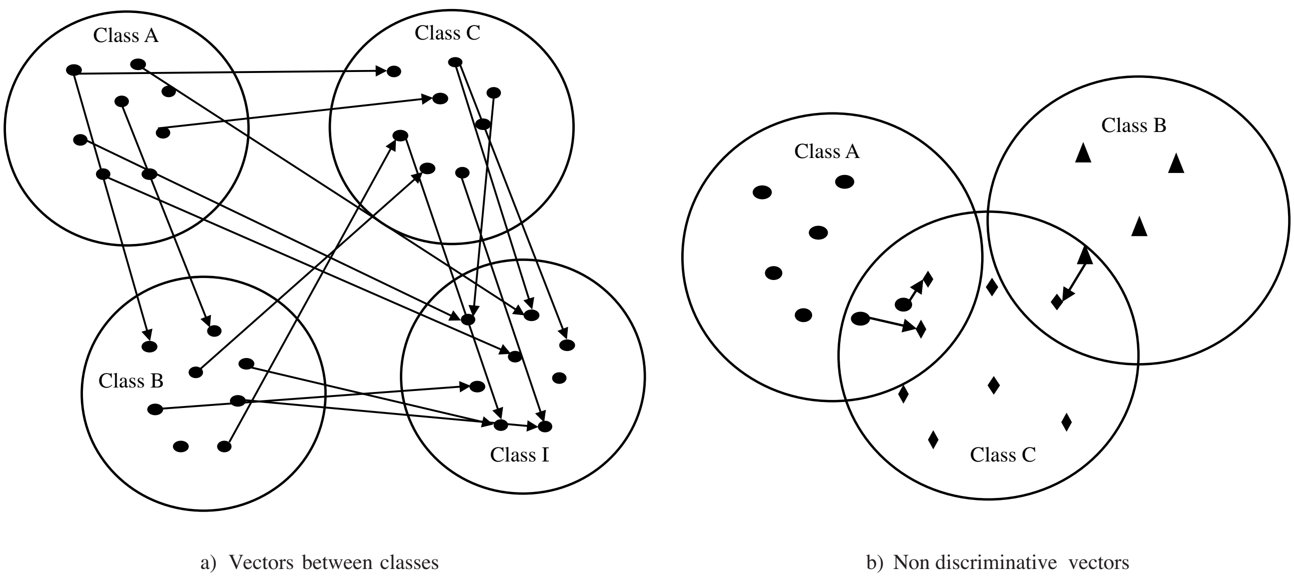


Figure 2. Representation of inter-class and noise difference vectors.

Every class in data set are numbered with a decimal in incremental order before the vectors are drawn. Since they are chosen randomly, vectors are drawn from the lower numbered classes to the higher numbered classes. How many vectors to draw from one class to another are again chosen randomly. Differences between classes in real data set will not be as distinct as in Figure 2(a). An example real situation is shown in Figure 2(b). The data in Figure 2(b) can also be noise data. These data make it difficult to distinguish the difference between classes. There are various methods and filters to remove noise in data sets [12]. In the proposed method, the lengths of the vectors drawn between classes are used to minimize the effect of the noise and the data that could not reveal the differences between classes. Vectors are deleted if length of drawn vector are below a specified threshold and new random vectors are drawn again between classes [5]. The following formula is used to find out whether the distance between classes is sufficiently distinctive.

$$\|x_{c_1} - x_{c_2}\|^2 < \frac{1}{fT}. \tag{4}$$

In Equation (4), the initial value of f is 1 and it increases during the weight calculation according to the density of the points close to each other. T is the number of output neurons which equals to class number. It should be determined the beginning of the training phase. If the length of the vector between different classes is less than the specified threshold, the vector is deleted and a new random vector is created. After the vectors between the classes are drawn, the weight matrix is calculated using Equation (3). The hidden layer bias values are calculated by using Equation (2). Data can be received one by one or chunk by chunk. If there is a previously created model, first training phase can be skipped and switch to the sequential training phase. If there is no previously created model, the sequential training phase must be started after first training phase. The inverse of $(H^T H)$ must be found to calculate P matrix. It is checked whether $(H^T H)$ is a singular matrix in the first training phase. If it is not a singular, the training process continues. If it is singular matrix, the input parameters of the hidden layer must be changed. In this case, hidden layer parameters need to be recalculated. In order to recalculate the parameters, random vectors must be drawn between classes and the weight and bias values are calculated again by using Equation (2) and Equation (3). Training phases are continued by calculating H and H^T with the new weight and bias values. Framework of proposed OS-CELM algorithm is given in Figure 3. We leverage the performance of OS-ELM by combining the method used in CELM into OS-ELM.

4. Experimental results

4.1. Dataset

We have analyzed four different data sets to compare the performance of the proposed method with the OS-ELM methodology of Ref. [4]. In the study, (wireless indoor localization (WIL) [13], skin segmentation (SS) [14], Cleveland heart disease (CHD) [15], and MNIST [16] data sets were used.

We present the diagrams of the datasets used in the present study in Figures 4(a)-(d) in two dimensions. Because of the difficulties in visualizing multidimensional data, the datasets used in the study were reduced to two dimensions with the t-distributed stochastic neighbor rmbedding (t-SNE) method [17]. This technique maps higher dimensional datasets to a lower dimension, ideally keeping near points close and far points far away. Perplexity, the most important parameter of the t-SNE algorithm, controls the width of the Gaussian kernel used to calculate the similarities between points and manages how many of the nearest neighbors of each point are drawn. Perplexity was taken as 30 in this study.

4.1.1. Wireless indoor localization dataset

WIL dataset is one of basic and most widely used datasets to test different classification techniques. It contains 2000 samples, each with 8 different features, which are produced by measuring signal strength from 7 different Wi-fi devices in four different rooms with a smartphone. Therefore, there are 4 different classes of the set and each data point contains 7 signal strength measurements in dB and one nominal data indicating the class value. The aim in solution of this problem is to determine the location for a given signal strength. The basic statistical information and t-SNE reduced images of the WIL dataset are listed in Table 1 and shown in Figure 4(a), respectively.

Table 1. Wifi indoor location.

Classes	Mean(dB)	Min	Max	Number of samples
Room-1	–68.668	–97	–47	500
Room-2	–57.251	–92	–10	500
Room-3	–62.156	–93	–40	500
Room-4	–64.403	–98	–36	500

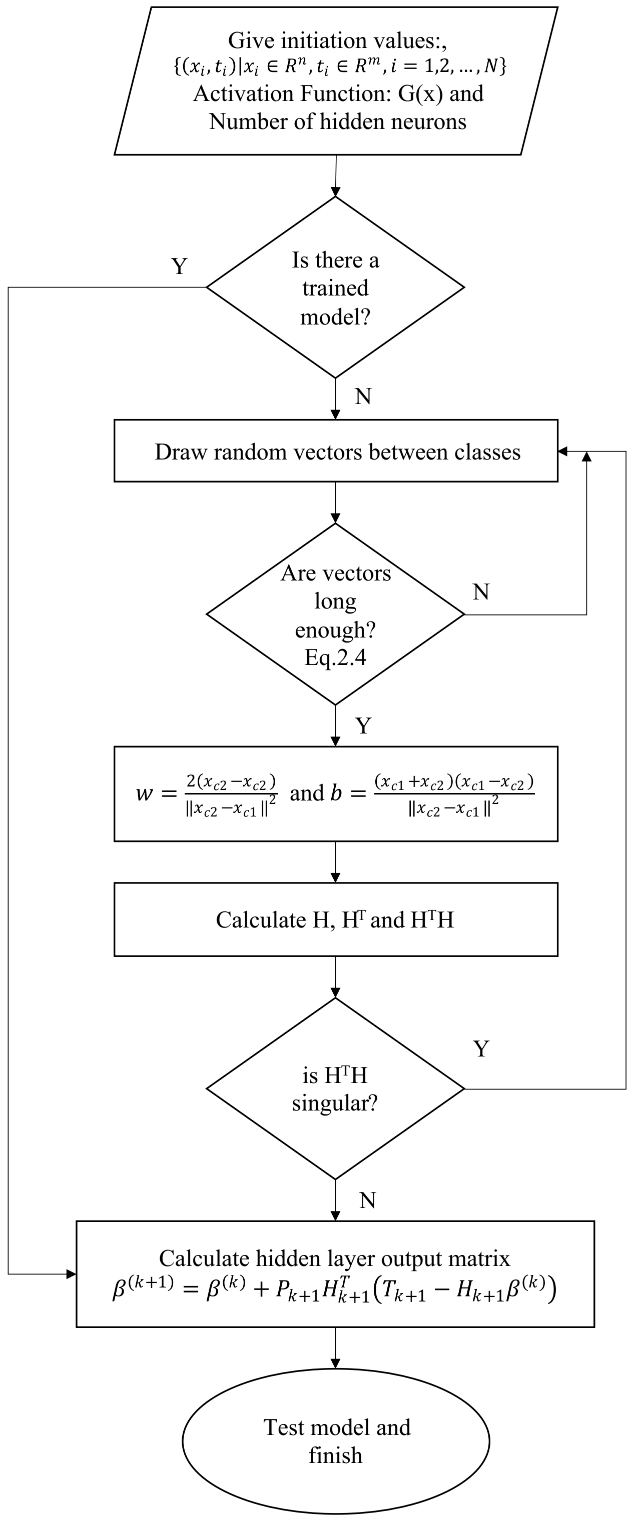
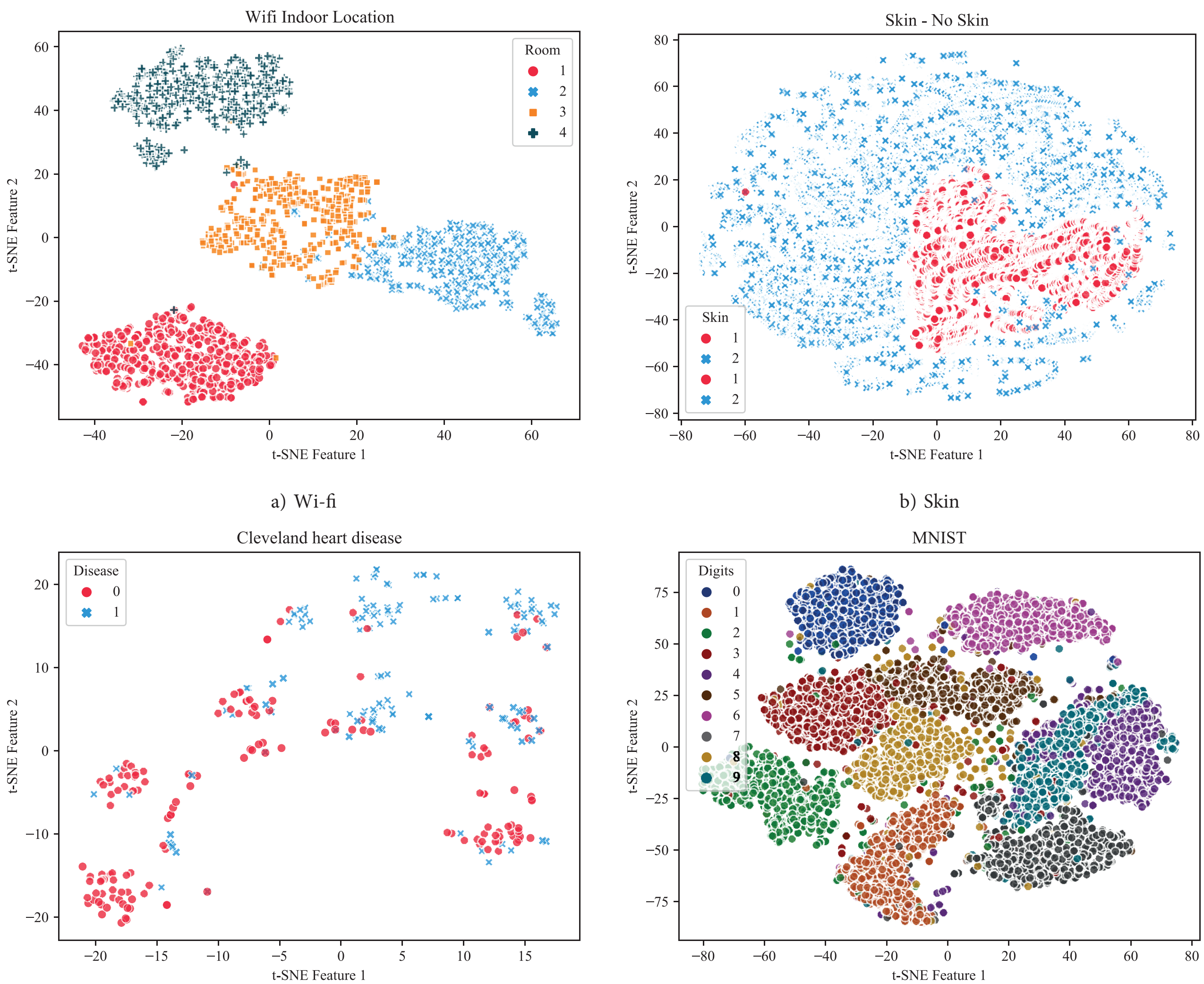


Figure 3. Framework of OS-CELM.



a) Wi-fi

b) Skin

c) Heart Disease

d) MNIST

Figure 4. Visualizing high dimensional data.

4.1.2. Skin segmentation dataset

Data set is constructed for predicting whether the information given as the (R, G, B) color triple comes from skin or not. It consists of 245,057 data collected from FERET and PAL databases from face images of people of different colors, sexes, and ages. There are 50,859 skin samples and 194,198 no-skin samples in this dataset. Each sample consists of 3 data points which gives the R, G, B values of the texture in the 0–255 range, and 1 nominal data indicating class value. The two-dimensional view of skin segmentation dataset constructed by using t-SEN method is given in Figure 4(b) with red and blue colors indicating the skin and the no-skin samples, respectively. There is no missing value in this dataset. The details of dataset are listed in Table 2.

Since the skin segmentation data set contains duplicate samples, the duplicate samples are removed from the data set. After the duplicate samples are removed 51,444 samples containing 14,654 skin and 36,790 no-skin remained in the data set.

Table 2. Skin segmentation.

Classes	Mean	Min	Max	Number of samples
Skin	154.821	26	255	50,859
No-skin	119.609	0	255	194,198

4.1.3. Cleveland heart disease dataset

Data set is taken from UCI repository and constructed for predicting whether individual has a heart disease or not from the given information. It consists of 303 individual's data and has 14 different features including class value. There are 7 nominal data, 6 ordinal data, and 1 class value in this dataset. The details of dataset are listed in Table 3.

Table 3. Heart disease.

Feature	Mean	Min	Max
Age	54.5420	29	77
Resting blood pressure	131.693	94	200
Cholesterol	247.350	126	564
Max heart rate	149.599	71	202
ST depression	1.055	0	6.2
Number of major vessels	0.676	0	3

There are 6 missing values in the dataset. Missing values are removed from the dataset and the remaining 297 data consisting of 160 "absence" values and 137 "present" values are used in this study. The two-dimensional view of the Cleveland heart disease dataset constructed using the t-SEN method is given in Figure 4(c). Each class is shown in separate colors and markings.

4.1.4. MNIST dataset

MNIST, a database of images of handwritten numerals consisting of 70,000 samples containing 60,000 training and 10,000 test samples, is a subset of the NIST data set containing both handwritten numerals and letters [18]. Every image in the dataset belongs to one of 10 categories (0–9). Each grey level digit image was centered in 28x28 dimensions according to pixel densities. It is one of the widely used data sets for training and testing in the field of machine learning. We have used the 1000 training data for t-SNE dimensional reduction and the result is presented in Figure 4(d) which shows 10 well-separated blobs for each one of the digits. Number of digits in the dataset listed in Table 4.

Table 4. Number of digits in the MNIST data set.

	0	1	2	3	4	5	6	7	8	9
Train	5923	6742	5958	6131	5842	5421	5918	6265	5851	5949
Test	980	1135	1032	1010	982	892	958	1028	974	1009

4.2. Performance assessment

In order to compare the performance of the proposed OS-CELM model with those of OS-ELM and ELM, all models were trained with the same training data sets and their performances were measured on test data sets for various number of hidden neurons. Sigmoid and hyperbolic tangent were used as the activation function in training all the models. Activation function with hyperbolic tangent performed better scores so the results created with hyperbolic tangent were given in the study. Five-fold cross-validation is employed to evaluate the model. For datasets from UCI, the number of training sample is 80% of total samples and the rest of samples were used to test models. Training and testing data were normalized within the range of [0,1]. Accuracy rate, F1 score, and the training time were used as the performance metrics to compare the models. For each one of the four datasets examined here, fifty training and test cycles were carried out for each considered neuron number. The statistics of the accuracy results obtained from those 50 experiments are presented as function of the number of neurons for OS-ELM, ELM, and the proposed OS-CELM networks in Figures 5(a)-(d) showing the distribution of the accuracy scores in 50 runs. For each hidden number of neurons, the accuracy, F1 score, and training time for all the datasets were calculated by averaging over 50 rounds and the results are shown in Figure 6(a)-(d).

As can be observed from the boxplots in Figures 5(a)-(d), the proposed learning algorithm has better performance in accuracy metric compared to the OS-ELM and ELM for all four datasets and hidden neuron numbers. It is interesting to note that increasing number of neurons beyond 80 leads to a large deterioration in the accuracy of the OS-ELM algorithm for the skin dataset.

For the Wi-Fi dataset, 200 samples were randomly selected for the first training phase and 100 samples for sequential training. Since the data set is balanced, we display the statistics of the fifty rounds of model accuracy rate and the average of those values as function of number of hidden neurons in Figures 5 and 6, respectively. It is seen from the figures that the OS-ELM has stabilized at 97.19%, ELM at 97.98%, and OS-CELM at 98.3% accuracy. The accuracy of classification of WIL dataset with ELM is also reported as 96.27%[19]. The training time of the algorithm we proposed increased after 100 neurons compared to OS-ELM.

After removing the duplicate samples, the remaining 51,444 samples of the SS dataset were used to test the performance of ELM, OS-ELM, and the proposed OS-CELM methods. For each of the first and the sequential training phases of the study of this dataset, 400 randomly chosen samples of the set were used for OS-ELM and OS-CELM. Since the dataset is not balanced, we present the weighted F1 scores instead of classification accuracy rates. Statistics of the weighted F1 scores for every round for the OS-ELM, ELM, and OS-CELM approaches are displayed in Figure 5(b) while the mean of F1 scores and training times of the model in the end of 50 rounds are plotted as function of number of neurons in the network in Figure 6(b). As can be seen from Figure 6(b), OS-ELM suffers from the over-fitting problem and becomes insensitive to data after the number of neurons in the hidden layer is increased beyond 80. It is obvious that proposed method and ELM do not suffer from similar problem and its classification performance increases with increasing neuron numbers. F1 score (training time) of the proposed model constructed with 100 neurons is 0.9863 (0.9134 s), at the end of 50 rounds (Figure 6(b)). ELM shows the same performance (0.9863) as OS-CELM does in SS dataset. MapReduce-based distributed AdaBoosting of extreme learning machine (ELM) model was proposed in the study [20]. OS-ELM/ELM algorithm was used to build weak learners (classifier functions) and then, strong learner was built from a set of weak learners. Performance of the model was evaluated on SS dataset. The best F1 score that was achieved in the study on this dataset was 98.42%.

After removing the data with missing fields in Cleveland heart disease dataset, performance tests were

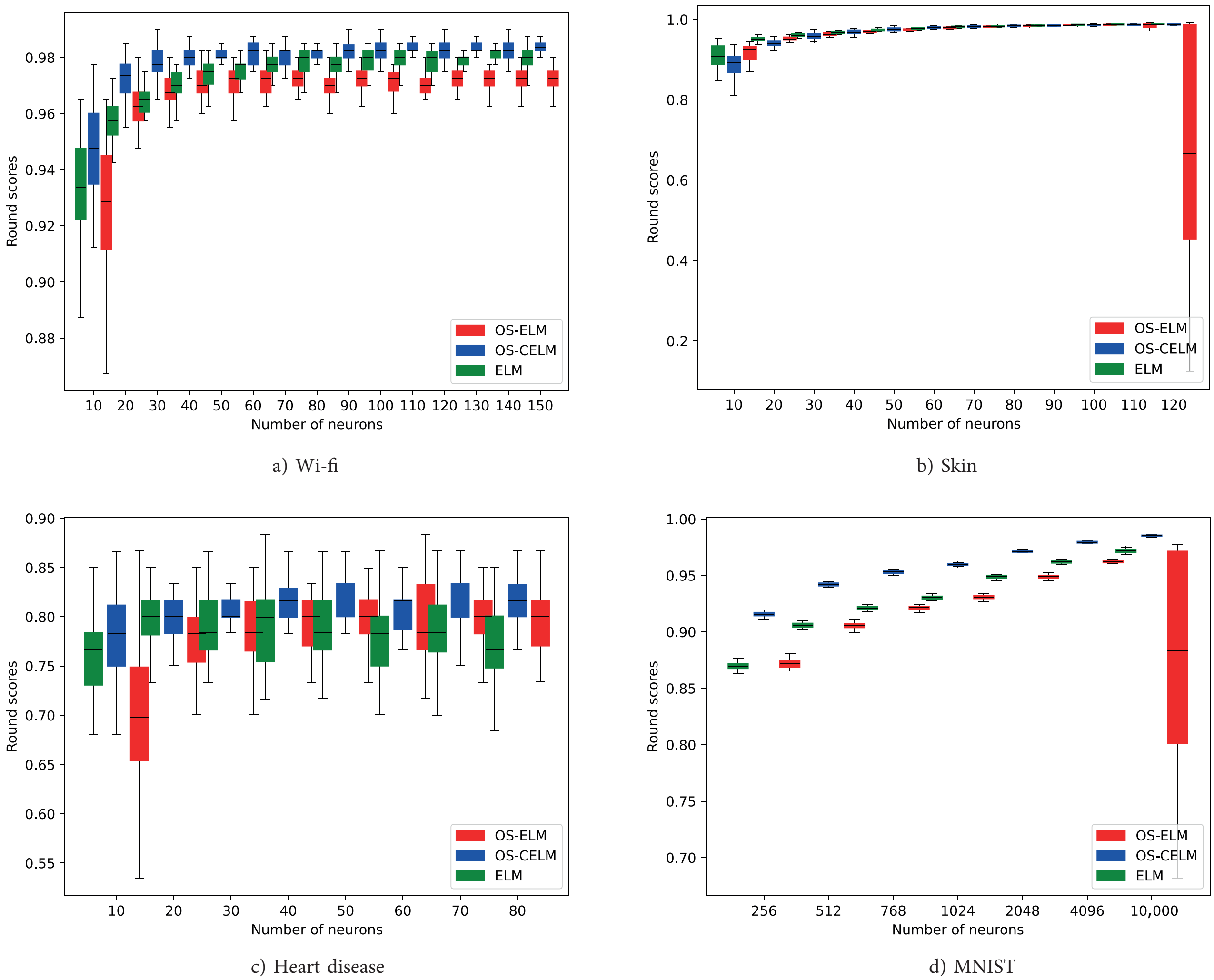


Figure 5. Accuracy and/or F1 scores of OS-ELM and OS-CELL methods over 50 turns.

conducted on the remaining 297 samples. The initial and the sequential training were carried out with 150 and 50 samples, respectively. Since the dataset is not balanced, we present the discuss the weighted F1 values instead of the classification accuracy rate. Statistics of the 50 rounds of F1 scores and the average of those along with the training times as function of number of hidden neurons for the Cleveland dataset are displayed in Figures 5(c) and 6(c), respectively. The F1 score is found to peak around 70 hidden neurons for both OS-ELM and the proposed method for this dataset while 20 hidden neuron for ELM. Highest F1 score with 70 neurons is found to be 0.8206 and 0.8 for the OS-CELM and OS-ELM, respectively. As the number of neurons increase, the success rate of ELM decreases after 20 neurons. Mean training time with 70 neurons is 6.62 ms in OS-CELM while it is 2.5 ms in OS-ELM. In a data mining related study [21] for Cleveland heart disease dataset, 8100 combination of features were selected and tested to find out which features gives highest accuracy rate. The average accuracy achieved with ANN is reported to be 75.18% (with the highest rate of 84.85% which is obtained by using 11 features). For the dataset, accuracy rates of 80.99% and 80% are also reported with ANN in other studies [22, 23].

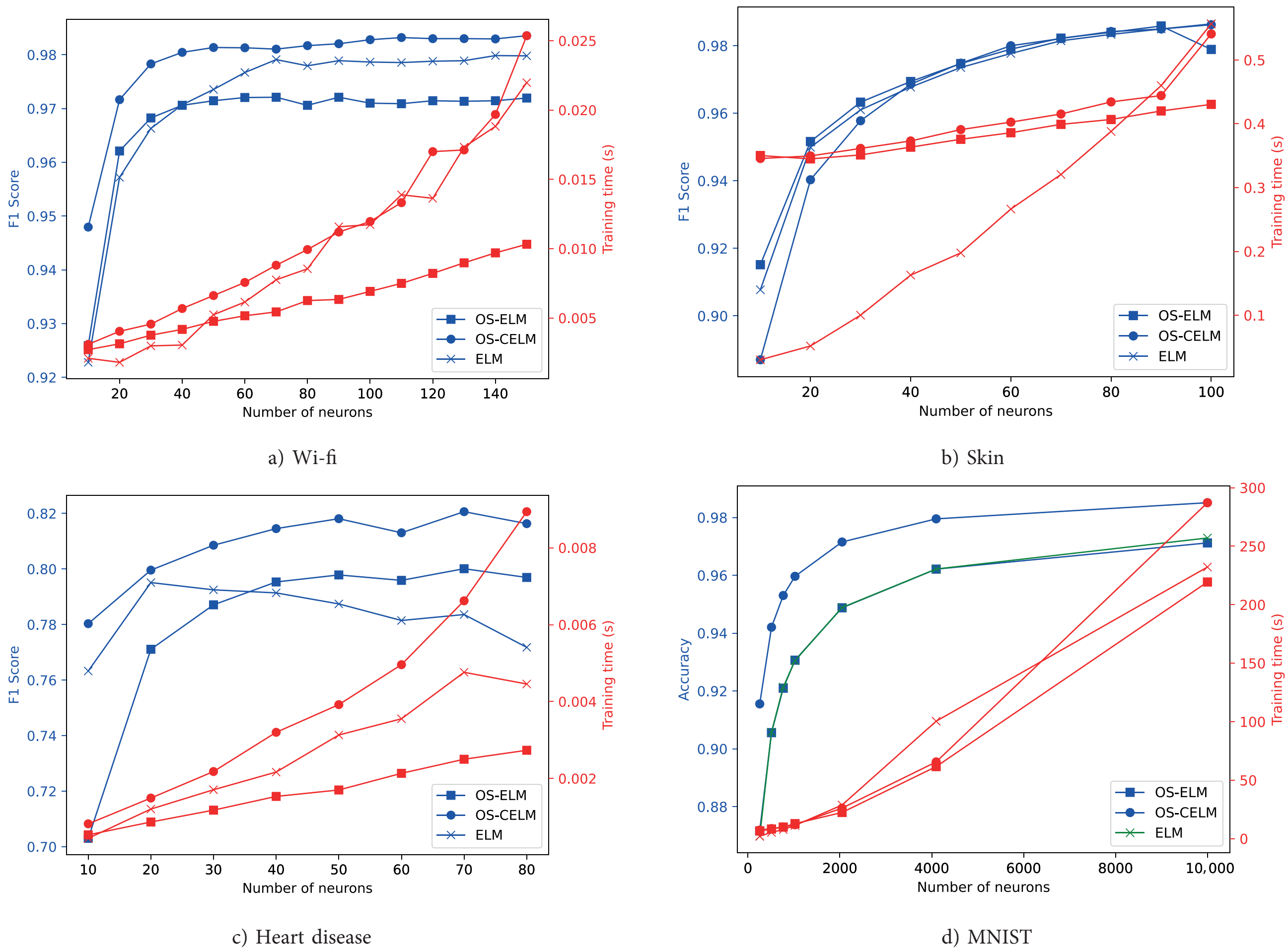


Figure 6. Accuracy and/or F1 score and the training times of OS-ELM and OS-CELM methods for the analyzed datasets.

For the MNSIT dataset, we have used 256, 512, 768, 1024, 2048, 4096, and 10,000 hidden neurons for the performance comparison of the three models. Since one needs more data points than the number of neurons in the hidden layer for the considered methods, we have used 2000, 3000, and 5000 samples for the initial training for the neuron numbers 256–1024, 2048, and 4096, respectively. Two thousand randomly chosen data were used in the sequential training irrespective of the number of neurons for OS-ELM and OS-CELM. Since the MNIST data is, also, balanced, we present the model accuracy rate instead of F1 score. Finally, all models are trained with 10,000 neurons for 50 rounds, and the accuracy rate and training time of the models are measured.

Statistics of the accuracy rates for the 50 rounds of training and test for MNIST data are shown in Figure 5(d). It can be deduced from the figure that the accuracy of the proposed OS-CELM is $\approx 5\%$ higher than that of OS-ELM and ELM for the 256 hidden neuron network and the rate uniformly increases with increasing neuron number before reaching 98% in the high neuron number limit. Also, the dispersion in accuracy rate decreases with the increasing neuron number. Training times of the proposed method and OS-ELM are found to be almost identical for the MNIST dataset (Figure 6)(d) till 4096 neurons. Accuracy rates of ELM and OS-ELM are almost identical for this dataset. Multilayer ELM model using the CELM method is proposed for MNIST dataset in the study [24]. With the model they named R^2CELM, they proposed ANN network consisting of 6 hidden layers with 2000 neurons in each layer. With R^2CELM, 98.47% accuracy was achieved. In this study, better accuracy rate (98.51%) is achieved with a single layer and 10,000 neurons with OS-CELM. Comparisons and scores for datasets used in this study are summarized in Table 5.

Table 5. Performance comparison of OS-CELM with other algorithms on UCI datasets.

Dataset	Method	Accuracy/F1 (%)
Wi-Fi	FPSOGSA[13]	95.16
	ELM[19]	96.27
	BP[25]	91.48
	ELM	97.98
	OS-ELM	97.19
	OS-CELM	98.3
SS	Fuzzy decision tree [14]	94.1
	AdaBoosting of ELM [20]	98.42
	ANFIS [26]	90.1
	CNN [27]	94.99
	ELM	98.63
	OS-ELM	97.89
	OS-CELM	98.63
C. Heart disease	PSO+MLP [21]	75.18
	MLP+BP [23]	80.99
	ANN [22]	80.0
	ELM	79.51
	OS-ELM	80.0
	OS-CELM	82.06
MNIST	CELM [5]	91.0
	R^2CELM [24]	98.47
	ELM	97.29
	OS-ELM	97.12
	OS-CELM	98.62

5. Conclusion

The use of data science in almost all business lines causes data to be produced very quickly and large increases in data volume. With the growth of data, the process of creating model over the data demands more calculation process and calculation time on user computers. Processing large amounts of data requires high-capacity memory. To issue this problem, the data must be fragmented enough to fit in the memory and the models created must be updated sequentially or partially. To address this, we have presented an improved online learning model called online sequential constrained extreme learning machine which assigns the hidden layers parameters according to the distance between the classes instead of random assignment. OS-CELM allows the data to be processed in sequentially or partially.

The proposed method was tested and its performance on accuracy and training time for classification problem is compared with those of OS-ELM and ELM on four widely used datasets. On the accuracy and the F1 score accounts, the proposed method is found to be better than the OS-ELM for each dataset at every hidden neuron number. From the results presented above, it is possible to say that the proposed OS-CELM model shows better classification performance for online sequential learning in four dataset without losing fast training speed compared to the OS-ELM method. In addition, its fast performance in creating models and the fact that it does not require time-consuming brute force search approaches to find hyper-parameters allows applications to be developed very quickly and efficiently. Since OS-CELM is based on the distance between classes, it works in the same way as OS-ELM in regression problems. In this context, the proposed model has the same performance in regression problems as OS-ELM does. It is the limitation of proposed model. Class independent methods can be applied in future works to increase regression performance of OS-ELM.

As a result, in this study, OS-CELM performance in different data sets is measured and an improved ELM-based model that can compete with other methods is proposed. OS-CELM is a single layer, easy to implement, and fast ANN that can be used in real world applications.

References

[1] Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological Review 1958; 65: 386-408.

[2] Huang GB, Zhu QY, Siew C. Extreme learning machine: a new learning scheme of feedforward neural networks. In: IEEE International Joint Conference on Neural Network; Budapest; 2004. pp. 985-990. doi: 10.1109/IJCNN.2004.1380068.

[3] Apaydin E. Introduction to Machine Learning. Cambridge: MIT Press, 2010.

[4] Liang N, Huang GB, Saratchandran P, Sundararajan N. A fast and accurate online sequential learning algorithm for feed forward networks. IEEE transactions on neural networks a publication of the IEEE Neural Networks Council 2006; 17: 1411-1123.

[5] Zhu W, Miao J, Qing L. Constrained extreme learning machine: a novel highly discriminative random feedforward neural network. In: International Joint Conference on Neural Networks (IJCNN); Beijing; 2014. pp. 800-807. doi: 10.1109/IJCNN.2014.6889761

[6] Chy TS, Rahaman MA. A comparative analysis by knn, svm & elm classification to detect sickle cell anemia. In: International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST); Bangladesh; 2019. pp. 455-459. doi: 10.1109/ICREST.2019.8644410

[7] Huang GB, Ding X, Zhou H. Optimization method based extreme learning machine for classification. Neurocomputing 2010; 74: 155-163.

[8] Qiu S, Gao L, Wang J. Classification and regression of elm, lvq and svm for e-nose data of strawberry juice. Journal of Food Engineering 2015; 144: 77-85.

[9] Deng WY, Zheng Q, Wang Z. Cross-person activity recognition using reduced kernel extreme learning machine. Neural networks: the official journal of the International Neural Network Society 2014; 53: 1-7.

[10] Xu S, Wang J. A fast-incremental extreme learning machine algorithm for data streams classification. Expert Systems with Applications 2016; 65: doi: 10.1016/j.eswa.2016.08.052

[11] Wang Z, Xin J, Yang H, Tian S, Yu G et al. Distributed and weighted extreme learning machine for imbalanced big data learning. Tsinghua Science and Technology 2017; 22: 160-173.

[12] Garcia L, Saez JA, Luengo J, Lorena A, Carvalho A et al. Using the one-vs-one decomposition to improve the performance of class noise filters via an aggregation strategy in multi-class classification problems. Knowledge-Based Systems 2015; 90: 153-164. doi: 10.1016/j.knosys.2015.09.023

[13] Rohra JG, Perumal B, Narayanan SJ, Thakur P, Bhatt RB. User localization in an indoor environment using fuzzy hybrid of particle swarm optimization & gravitational search algorithm with neural networks. In: Proceedings of Sixth International Conference on Soft Computing for Problem Solving. Advances in Intelligent Systems and Computing; Singapore; 2015. doi:/10.1007/978-981-10-3322-3_27

[14] Bhatt RB, Dhall A, Sharma G, Chaudhury S. Efficient skin region segmentation using low complexity fuzzy decision tree model. In: Proceedings of INDICON An IEEE India Council Conference; India 2009. doi: 10.1109/INDCON.2009.5409447

[15] Detrano R, Janosi A, Steinbrunn W, Pfisterer M, Schmid J et al. International application of a new probability algorithm for the diagnosis of coronary artery disease. The American Journal of Cardiology 1989; 64 (5): 304-310.

[16] Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. In: Proceedings of the 1998 IEEE International Frequency Control Symposium; Pasadena, CA, USA; 1998. pp. 2278-2324.

[17] Van der Maaten L, Hinton G. Visualizing high-dimensional data using t-SNE. Journal of Machine Learning Research 2008; 9: 2579-2605.

[18] Grother P. NIST special database 19 hand printed forms and characters database. 1995.

[19] Sabanci K, Yigit E, Ustun D, Toktas A, Aslan MF. Wifi based indoor localization: Application and comparison of machine learning algorithms. In: XXIIIrd International Seminar Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED) ;Tbilisi,Georgia; 2018. pp. 246-251.

[20] Çatak FO. Classification with boosting of extreme learning machine over arbitrarily partitioned data. Soft Computing 2017; 21: 2269-2281.

[21] Amin M, Chiam Y, Varathan K. Identification of significant features and data mining techniques in predicting heart disease. Telematics and Informatics 2018. doi: 10.1016/j.tele.2018.11.007

[22] Khemphila A, Boonjing V. Heart disease classification using neural network and feature selection. In: International Conference on Systems Engineering; Las Vegas, USA; 2011. pp. 406-409.

[23] Verma L, Srivastava S, Negi P. A hybrid data mining model to predict coronary artery disease cases using non-invasive clinical data. Journal of Medical Systems 2016; 40: 178.15.

[24] Song G, Dai Q, Han X, Guo L. Two novel elm-based stacking deep models focused on image recognition. Applied Intelligence 2020; 50: 1345-1366.

[25] Chen J, Dong C, He G, Zhang X. A method for indoor Wi-Fi location based on improved back propagation neural network. Turkish Journal of Electrical Engineering & Computer Sciences 2019; 27: 2511-2525.

[26] Idoko J, Arslan M, Abiyev R. Fuzzy neural system application to differential diagnosis of erythemato-squamous diseases. Cyprus Journal of Medical Sciences 2018; 90-97.

[27] Ma CH, Shih H. Human skin segmentation using fully convolutional neural networks. In: IEEE 7th Global Conference on Consumer Electronics; Nara, Japan; 2018. pp. 168-170.

[28] Zhang J, Li Y, Xiao W. Adaptive online sequential extreme learning machine for dynamic modeling. Soft Computing 2021; 25: 2177-2189. doi: 10.1007/s00500-020-05289-6

[29] Zhang P, Huang Y, Li M, Wang Y, Wen P et al. Fault diagnosis method of analog circuit based on GA-OS-ELM. In: 2020 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC); Beijing, China; 2020. pp. 273-278. doi: 10.1109/SDPC49476.2020.9353152