

An Overview of Sequential Learning Algorithms for Single Hidden Layer Networks: Current Issues & Future Trends

M. H. Arshad¹, *Student Member, IEEE*, and M. A. Abido², *Senior Member, IEEE*

Abstract—In this paper, a brief survey of the commonly used sequential-learning algorithms used with single hidden layer feed-forward neural networks is presented. A glimpse at the different kinds that are available in the literature up until now, how they have developed throughout the years, and their relative execution is summarized. Most important things to take note of during the designing phase of neural networks are its complexity, computational efficiency, maximum training time, and ability to generalize the under-study problem. The comparison of different sequential learning algorithms in regard to these merits for single hidden layer neural networks is drawn.

Index Terms—RAN, EKF, MRAN, Generalized Growing and Pruning NNs OSELM, Direct Link RBF, EOSELM, ESOS ELM, MOSELM

IMPACT STATEMENT

Online sequential learning plays a pivotal role for real-time neural network applications. This review article summarizes the most commonly used online sequential learning algorithms with single hidden layer feedforward neural nets which will motive the interest of machine learning researchers with the span of the latest ongoing research work. We identify the major pros and cons of the studied algorithms and share some of the new promising directives for future work. This review also draws comparison between the studied algorithms for three different datasets to show the performance of each individual algorithm based on generalization ability, accuracy and training time. This comparison will benefit the new data science strategists in selecting any of the above techniques for their own problem.

I. INTRODUCTION

THE internal layout of Multi-layer feedforward neural networks (MLFNs) allows to generate an approximate representation which describes how well an input data is correlated and it has made these networks the most widely used neural networks (NNs) when it comes to pattern classifications [1], [2], [3]. Multiple hidden layer NNs are considered as general approximates but because of their complex nonlinear behavior,

they are not preferred in industrial applications where most applications require much faster and more generalized models [4]. Due to its simple network structure and excellent approximation performance, single hidden layer feed-forward neural networks (SHLFNs) have been widely used in pattern classification and function approximation applications [5]. Under normal circumstances with high dimensional large training data, the existing SHLFN algorithms has a slow learning speed, which limits the practical application of these learning algorithm [6]. Another disadvantage of the SHLFNs learning algorithm is its poor generalization ability [7]. At present the learning algorithms for SHLFNs can be widely classified as:

- Batch learning algorithms
- Online sequential learning algorithms.

For the NN batch-mode learning algorithm, the offline learning method is generally adopted [8]. That is, the network training process is divided into several independent stages in time. One typical feature of the batch learning algorithm based neural networks is that all sample information must be known beforehand. This feature makes the batch learning algorithm unable to process objects with time-varying characteristics i.e. real time data. In addition, before the network training, if the number of samples are not enough, it will seriously affect the performance of the network. Similarly, when adding new samples, all the samples must be used to retrain the whole neural network [4]. To compensate the shortcomings of SHLFNs batch learning algorithm, researchers are committed to explore new online sequential (OS) learning algorithm that provide compatible performance index with the existing batch learning based SHLFNs [9], [10]. With OS learning algorithm, all sample data enters the neural network one by one, and at any time, only one training sample is visible and used for learning. The training samples entering the network are discarded after the training is over. The most promising feature of this type of learning algorithm is that before the start of the learning process, there is no need of prior knowledge of the training dataset size. Therefore, these algorithms are more suitable for real time problems in industrial environment applications.

In the history of SHLFNs OS learning algorithms, the earliest algorithm was the Stochastic Gradient Decent Back Propagation (SGBP) algorithm which was formulated in [11]. It was proposed as an improvement to standard back propagation algorithm (BP) with input data samples are processed in

****This work has been submitted to the IEEE Transaction for Artificial Intelligence for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible****

M. H. Arshad & M. A. Abido are with Department of Electrical Engineering, King Fahd University of Petroleum & Minerals, Dhahran, 31261, Saudi Arabia, e-mail: haseeb.arshad.ee@gmail.com,

Corresponding author: M. A. Abido, phone: +966508757838; fax: +966138603535; e-mail: mabido@kfupm.edu.sa

chunks. Resource Allocating Network (RAN) learning algorithm proposed by Platt in [12]. RAN fundamentally suffers from a problem of unlearning over time and this problem was later addressed in [13] where the author proposed RAN-LTM. The shortcomings of RAN learning algorithm were further studied by Kadirkamanathan et al. and he proposed the extended Kalman filter (EKF) iterative strategy and replaced the previously used LMS method for network parameters training which resulted in RANEKF learning algorithm [14]. Although this method improves the convergence speed of the algorithm, the network complexity is also increased. To minimize the computational cost of the RANEKF learning algorithm, the MRAN algorithm added the deletion mechanism for the neurons in the hidden layer [15]. An improved MRAN learning algorithm (Hyper-MRAN) was proposed in 2006 which uses a new weight adjustment strategy that reduces the disadvantages of high memory requirements of the MRAN algorithm with higher accuracy [16].

In 2004, a new SHLFNs OS learning algorithm called GAP-RBF algorithm was proposed, which further improved the performance of MRAN [17]. It was later generalized in [18] and was called as GGAP-RBF. In an attempt to improve the computational complexity of the GGAP-RBF, in [19] GAP-DRBF was proposed that replaced the extended Kalman filter with decomposed extended Kalman filter and the underlying activation function was changed to direct-link RBF (DRBF). This learning algorithm has many adjustable parameters that needs manual tuning. When these parameters are not properly selected, the performance of the network will be seriously degraded. In view of this shortcoming of the GGAP-RBF learning algorithm an improved GGAP-RBF neural network learning algorithm (GIRAN) was formulated [20]. Although the improved GGAP-RBF can solve many practical problems related to manual tuning of parameters, the activation function of the algorithm is in the form of RBF which still was computationally not very effective and limits its use in real-time applications. The most famous RAN variant (SRAN) introduced in [21] uses self-adaptive mechanism to update the weight and biases of hidden layer neurons thereby drastically increases the learning speed of the algorithm.

The concept of batch learning based extreme learning machine (ELM) was introduced in [22] which Liang extended to an OS learning based ELM that significantly improved the computational efficiency of the algorithm and can select multiple activation functions [23]. To further improve the performance of Liang algorithm, recursive least squares (RLS) method was replaced with orthogonal least squares iterative strategy to further improve the generalization ability of the network [24]. Additional improvement in setting the initial random weights for the hidden neurons in OS-ELM was made in [25] by using set of regularized linear equations. With OS-ELM, the condition of having a greater number of hidden neurons than the dataset size and also the singularity problem encounter during the noisy data was studied in [26] and a new algorithm called Regularized online sequential extreme learning machine (ReOS-ELM) was proposed. The ReOS-ELM makes use of the Tikhonov regularization [27] with bi-optimization function to counter the problem of singularity and

ill-posed matrix inversion with standard OS-ELM. Ensemble technique [28] was studied with OS-ELM in [29], [30] to increase the generalization speed. To classify timeliness based data, a forgetting factor mechanism was introduced with OS-ELM in [31]. Class imbalance learning (CIL) is one of the most researched areas in NNs and in [32], the first ELM based framework was proposed for CIL problems with explicit feature mapping. In case of implicit feature mapping in the hidden layer neurons, a kernel based solution KOS-ELM was introduced [33]. Various other improved OS-ELM for multi-class data streams were studied in [34], [35], [36].

The SHLFNs OS learning algorithm can be divided into two categories:

- SGBP, RAN, RAN-LTM, RANEKF, MRAN, EMRAN, HMRAN, GGAP-RBF, GAP-DRBF, GIRAN and SRAN

The network parameters for these OS learning algorithms are done iteratively as new sample information is received. Also, before the algorithm training, there are many initialization parameters that need to be manually set for reliable performance

- OS-ELM, EOS-ELM, ReOS-ELM, FOS-ELM, VOS-ELM, WOS-ELM, KOS-ELM, I-ELM, ROS-ELM, ESOS-ELM, VWOS-ELM, MOS-ELM, M-OS-ELM, OS-RELM, OS-RKELM, WOS-ELMK, AFGR-OS-ELM and CW-OS-ELM

These OS learning algorithms eliminate the use of initialization parameters to improve the learning speed of the NNs and reduce the artificially adjusted initialization network parameters. Activation function for hidden layer neurons are also no longer limited to the RBF form which makes these algorithms computationally efficient.

To compare the performance of various OS learning algorithms for SHLFNs, most of the researchers used training time, number of hidden layer neurons, approximation accuracy, generalization ability and algorithm stability as performance indices.

II. ONLINE SEQUENTIAL LEARNING ALGORITHM BASED SHLFNS

The structure of the SHLFNs consists of three layers, as shown in Fig. 1, including the input layer, a single hidden layer, and the output layer. Passing of signals to the hidden layer is the major function of the input layer. The hidden layer neuron can be a radial basis function or other satisfying infinitely differentiable or arbitrary nonlinear bounded piecewise continuous qualitative function form, denoted here by $\varphi(x)$. The output layer node has generally a linear activation function, and the hidden layer and the output connection weight are represented by 'w'. In the RAN, RAN-LTM, RANEKF, MRAN, GGAP-RBF, GAP-DRBF and GIRAN, the hidden layer and the output connection weight is a constant value of '1', while in SGBP and sequential learning-based ELM algorithms, this parameter is randomly set.

A. Stochastic Gradient Decent Back Propagation (SGBP) Algorithm

SGBP was proposed by LeChun et al., is an improvement to standard BP learning algorithm. In SGBP, the training

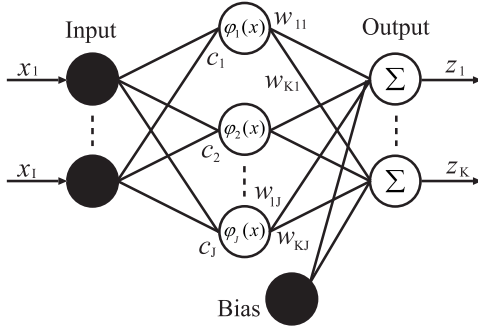


Fig. 1. SHLFNs Architecture

dataset sample is introduced one by one in the hidden layer neurons, and then based on the estimate of the error gradient, parameters are updated by the standard BP algorithm [11]. The algorithm was named as stochastic gradient back propagation because of the use of the estimate of error gradient. The purpose of using the estimate was to reduce the speed of convergence which is advantageous in case of noisy data. Because the input data is processed one by one, the SGBP results in faster learning with repeating data sequences and can track changing more effectively because of slow convergence with noisy data which is usually the case in most of practical applications.

B. Resource Allocation Network (RAN) Algorithm

The RAN was proposed by Platt in [12] is a dynamic single hidden layer radial basis function (RBF) neural network. The corresponding training algorithm is called RAN learning algorithm. In RAN, the "novelty" of training dataset sample is exploited for introducing the hidden layer neurons, and then the parameters are updated by the LMS algorithm [37]. The RAN learning algorithm is started with a no hidden layer neuron, the hidden layer neuron activation function type is RBF, and the first two input sample data are used for network initialization. Subsequently, if for an input sample there is unnecessarily large error between the output and the desired one, the sample is considered novel, and a neuron is added to the hidden layer of the network. If the input sample does not meet the novelty requirements, the hidden layer neurons are not added, but the LMS algorithm is initiated to update the current network parameters e.g. centers for neurons, the hidden and output layer connection weights etc. The novelty criteria of the RAN online-sequential learning algorithm is:

$$|e_i| = \|y_i - \hat{y}_i\| > \varepsilon \quad (1)$$

$$d_i = \min_{1 \leq k \leq K} \|x_i - c_k\| > \delta_i \quad (2)$$

where, e_i is the error in the output of the network, ε is the required approximation accuracy (manual adjustment parameter), d_i is the L_2 norm between the current input data sample x , and the center of the hidden layer neuron closest to it, $\delta_i = \max(\gamma^i \delta_{max}, \delta_{min})$ where, δ_{max} and δ_{min} are the max. and min. distance between all the input data samples, and

$0 < \gamma < 1$ is an attenuation coefficient. As the input data sample x_i increases, δ_i decreases at an exponential rate until δ_{min} . When the input data sample x_i does not meet the novelty requirements, the center point and the weights are updated using the LMS algorithm.

C. Resource Allocating Network with Long-Term Memory (RAN-LTM) Algorithm

The major drawback with RAN is the inherited unexpected forgetting which causes the network to unlearn over time [38]. This problem was addressed in 2003 by K. Okamoto who proposed the idea of external memory with RAN [13]. The new algorithm was called resource allocating network with long term memory. The learning phase of this algorithm was divided into 2 parts.

- Firstly, a hidden neuron is allocated and its weight connection with the output are determined.
- Then in the later stage output error is calculated and if the error shoots over a predetermined value, another neuron is added like in the case of RAN but together with it a new memory unit is also allocated in which the previous input output pair is stored

The architecture of the RAN-LTM is given in Fig. 2 [38].

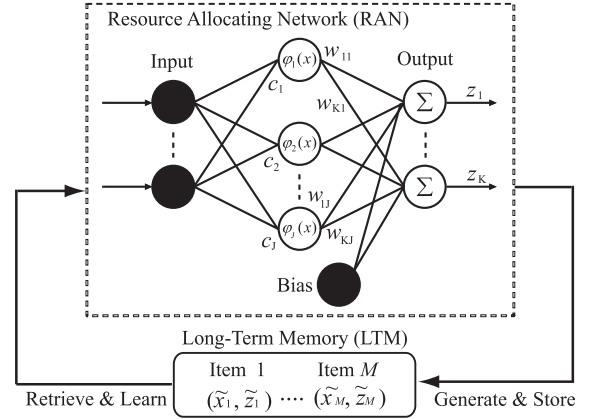


Fig. 2. RAN-LTM Architecture.

D. Resource Allocation Network using Extended Kalman Filter (RANEKF) Algorithm

The RANEKF algorithm is an improvement of RAN. The addition of neurons in the hidden layer is the same as for the RAN learning method. The only difference is that the adjustment of the network parameters (centers and weights) are done by the extended Kalman filter (EKF) method as a substitute for the more traditional LMS method [14]. The extended Kalman filter method has a faster convergence speed than LMS method, but it requires more computer resources. With the development of computer hardware technology, the EKF method is more advantageous than the LMS iterative method when the problem size is small.

E. Minimal Resource Allocation Network (MRAN) Algorithm

This algorithm combines the hidden layer growth criterion of RAN learning algorithm with an ideal deletion strategy for the hidden layer neurons. In 1994, Cheng proposed the method of deleting hidden layer nodes in batch learning algorithm [5]. In this method, each time the sample data enters the network processing, the weights of each hidden layer node must be checked and deleted in case if the weight of hidden layer node is not equal or greater than a certain wanted value. Inspired by the Cheng method, the MRAN algorithm with hidden layer node deletion strategy was proposed [15]. The proposed MRAN algorithm was compared both with RAN learning and the RANEKF algorithm. Since the learning process of MRAN algorithm entails the introduction of new hidden layer neurons along with iterative tuning of network parameters with the deletion strategy of hidden layer neurons, so it can be more streamlined under the premise that SHLFNs has better performance. Its common applications are listed in [39]. An extended MRAN (EMRAN) was proposed to reduce the computational cost associated with MRAN in case of larger input dimensional size [40]. A winner neuron approach was used to select a neuron and then update the parameters only related to this neuron using EKF method rather than updating the parameters of all the neurons in the hidden layer.

F. Hyper Minimal Resource Allocation Network (HMRAN) Algorithm

The computational complexity of MRAN increases drastically as the input data sample dimension increases as it requires to learn unwanted information from the input data sample. The underlying activation function in MRAN is RBF and in case of high dimensional input data sample, the computational cost increases because the increase in neurons number proportionally increases the size of the covariance matrix for extended Kalman filter. Therefore, the time complexity of the algorithm limits its use in real-time industrial applications. To overcome this problem with MRAN, extended minimal resource network (EMRAN) was proposed in [40] but this algorithm lags reasonable accuracy. In [16], Hyper MRAN (HMRAN) was formulated, with suitable input data sample dimension selection with hyper radial basis function (Hyper RBF) as an activation function for SHLFNs, to reduce the time complexity problem of MRAN by using localized extended Kalman filter approach also the hyper RBF activation function ensure high accuracy.

G. Generalized Growing and Pruning Radial Basis Function (GGAP-RBF) Algorithm.

Nanyang University of Technology in Singapore, in 2005, proposed a new OS learning algorithm for the existing shortcomings of various SHLFNs OS learning algorithms, called the GGAP-RBF algorithm [18], which makes the use of the traditional RBF. The performance of this SHLFNs online-sequential learning algorithm was improved over the MRAN algorithm. The parameters in MRAN are adjusted at each iteration using the extended Kalman filter method, which

leads to the increment in the hidden layer neurons during the parameter update process [16]. Also, the size of the covariance matrix used in EKF is usually very large, which increases the computational complexity of the network structure. This results in an excessive computational burden of the algorithm and a large amount of computer resources, which limits the real-time application of the MRAN algorithm. Although the GGAP-RBF learning algorithm uses the extended Kalman filter method in the parameter update process, but only updates the parameters (center and width) of the hidden layer neurons closest to the current input data and the weight of the corresponding neurons connecting the hidden layer neurons, which greatly reduces the computational cost of this algorithm. In addition, for the shortcomings of the initial parameter selection of the various heuristic algorithms mentioned above, for the uniformly distributed input data, the GGAP-RBF learning algorithm also estimates the importance of the hidden layer neurons, thereby reducing the initial parameters. Even though, the GGAP-RBF OS learning algorithm reduces the number of initialization parameters of MRAN, the disadvantage of having uniformly distributed input data drastically reduced the performance of the algorithm.

H. Growing and Pruning Direct Link Radial Basis Function (GAP-DRBF) Algorithm.

In [19], the authors improved the performance of GGAP-RBF by replacing the RBF with Direct-link RBF which is basically the augmented version of conventional RBF with a linear mapping of input and output to improve the accuracy of training and also the EKF was replaced with decomposed EKF proposed in [41] that reduces the number of initialization parameters thereby ensuring that the computational complexity of the GGAP-RBF is further reduced as the training data size increases. On the flip side, this algorithm has a significant disadvantage: it requires data to be uniformly distributed, which usually is not the case in practical applications hence requiring more hidden neurons, reducing the performance of the algorithm.

I. Improved GGAP-RBF (GIRAN) Algorithm.

The GGAP-RBF utilized the idea of dynamically estimating the importance of the hidden neurons but the formula given in [18] was further corrected in [20] where the author improved the GGAP-RBF algorithm in a bid to reduce the initialization parameters of the algorithm and introduced the new definition and estimation formula for measuring the importance of the hidden layer neurons. The corresponding algorithm is called GIRAN (Improved GGAP-RBF Learning Algorithm) learning algorithm. The dynamic adjustment formula for 'K' was given by

$$K = \frac{e_i}{(e_i + \|w_{ir}\|) \sqrt{-\ln \frac{\varepsilon}{e_i}}} \quad (3)$$

where, K equals the number of hidden neurons.

Based on the GGAP-RBF algorithm, combined with the update formula of parameter, and the adaptive adjustment

method of radial basis function width, the improved algorithm is as follows.

Given the estimated error, the single point output is expected to be accurate. For the input sample.

- Calculate the network output
- Calculate the novelty criterion
- Apply the novelty criterion to determine whether to increase the hidden layer neurons
- Else: Use the EKF method to update the hidden layer section of the network closest to the current input.
- Check if the deletion criteria of the hidden layer neurons are met
- If yes, delete the hidden layer neurons, correspondingly reduce the dimension of EKF. Else: GOTO Step 1.

J. Self Adaptive Resource Allocation Network (SRAN) Algorithm.

Usually, in RBF based sequential learning algorithms, the control parameters are manually set for reliable performance which results in poor generalization efficiency with redundant data. In [21], a solution to this problem was proposed by using self-regulated initial control parameters. These control parameters change automatically based on the difference between the new information in the coming sample and the already learnt network's information. The higher the difference, the greater is the probability of the incoming sample to contribute towards sequential learning process. If the difference is negligible, then the sample is considered as redundant and is simply discarded. It is to be noted that SRAN does not include pruning strategy and relies on EKF for training which makes it computationally more expensive.

III. EXTREME LEARNING MACHINE BASED ONLINE SEQUENTIAL LEARNING ALGORITHM

To counter the learning speed problems of the commonly used SHLFNs sequential-learning algorithms, Huang proposed an extreme learning machine (ELM) batch learning algorithm, which sets the weight connections between input and hidden layer and the bias of the hidden layer neurons randomly, and then analytically determines the weight connections of hidden layer neuron with that of the output layer of the SHLFNs by calculating a pseudo-inverse operation on the matrix generated by the output layer to obtain the weights [25]. In theory, the input weights and neuron deviations of SHLFNs do not have to be adjusted during training, and they can be arbitrarily assigned [22], which greatly improves the learning speed of the SHLFNs.

To solve the problem of large hidden layer neuron in ELM with batch learning, the Regularized Least-Squares Extreme Learning Machines (RLS-ELM) was proposed in [25]. In RLS-ELM, instead of random initialization of the hidden layer weights and biases, the decision is taken by using a system of regularized linear equations. The output weights are obtained in the same manner as the original ELM. This single act of changing the way in which the weights of the input layers are obtained makes the system obtain a very good performance in relation to the original ELM and ensures that the overall network is relatively smaller with a higher training speed.

A. Online Sequential Extreme Learning Machines (OS-ELM) Algorithm

Using the core idea of ELM algorithm, in [23], an online-sequential extreme learning machine based on recursive least squares algorithm (RLS), called OS-ELM (RLS) learning algorithm was proposed. This algorithm was further improved in [24] where the author used orthogonal least squares method and replaced the computationally bulky RLS method to update the network parameters. The improved OS-ELM was termed as OS-ELM (OLS). The major drawback related to OS-ELM was ill-posed and singular matrix obtained in case of noisy data. Also, the dependence of hidden layer neurons on the size of dataset greatly effects the efficiency of this learning algorithm

B. Ensemble based Online Sequential Extreme Learning Machine (EOS-ELM) Algorithm

To further enhance the generalization ability and speed of OS-ELM, in 2009, Lu proposed the ensemble-based OS-ELM termed as EOS-ELM [29]. It uses various OS-ELMs with different hidden neurons and the same data sample is passed to the hidden layer. Because of the different parametric settings of each OS-ELM, the output of OS-ELMs are different and the final output is calculated by the average of the outputs of each OS-ELM. This ensemble of different OS-ELMs effectively produces better results because of each ELM possessing distinct capacity of adaptation to the new streams of data and the fact that mean of a population is always close to the expected value rather than any individual value in the population.

C. Regularized Online Sequential Extreme Learning Machines (ReOS-ELM) Algorithm

The OS-ELM algorithm produced exemplary results for non-noisy training data. It can work with data arriving one by one or in small data streams. The issue of singular matrix and ill-posed optimization problem in case of noisy data was studied by H.T. Huynh associated with OS-ELM [26]. The author replaced the standard optimization problem associated with OS-ELM with bi-optimization problem composed of actual l_2 norm of current weight vector and training error that improved the generalization ability of the network for real-time noisy data as studied in [42]. Also, the singularity problem was solved by using Tikhonov regularization [27] instead of least squares method used in [25]. The ReOS-ELM learning algorithm not only produces faster learning but also significantly helps in overcoming the usual problem of setting the initial domain of impact factor and bias in case of noisy data.

D. Online Sequential Extreme Learning Machine with Forgetting Mechanism (FOS-ELM) Algorithm

The generalization ability of EOS-ELM hinders with the timeliness property associated with certain types of data e.g. weather and stock forecasting data. Because the input data has a limited period of validity, training the EOS-ELM with

this type of data results in wrong prediction and instability. To overcome this issue and to reflect the timeliness property of the incoming data, J. Zhao et al. proposed the forgetting mechanism with EOS-ELM to oust the out-of-date datum for the sequential learning in [31]. Zhao concluded that rather than re-training the network, as new data enters the hidden layer, training is only done by using newly available datum and with the already known network parameters only if the new data arriving is valid. A similar work was reported in [43] where, a timeliness online sequential extreme learning machine (TOS-ELM) was proposed based on incoming data's distribution and central tendency characteristics. In [44], a variable forgetting factor based on directional forgetting factor [45] with EOS-ELM, called DFF-OS-ELM, was used to better capture the timeliness property of the incoming data.

E. Voting base Online Sequential Extreme Learning Machine (VOS-ELM) Algorithm

VOS-ELM was proposed in [46] that used several independent OS-ELM in parallel with same number of hidden nodes and similar input data chunk but with different randomly generated initialization parameter for each extreme learning machine. A weight vector for the several OS-ELMs were calculated in parallel with RLS algorithm. The final measurement was calculated based on voting method between the OS-ELMs. The classification accuracy of VOS-ELM was found to be higher than the original OS-ELM also the speed of learning was greatly improved in contrast to V-ELM [30].

F. Weighting Online Sequential Extreme Learning Machine (WOS-ELM) Algorithm

The most widely encountered problem during the last decade in machine learning is class imbalance learning (CIL) problem. Majority of researchers have proposed various solutions based on batch learning to tackle CIL problems especially in the field of medicine and online fraud detection. It was not until 2013 that a first online sequential extreme learning machine-based algorithm was proposed to counter the bi-class CIL problem [32]. Mirza B. & Lin Z. proposed their weighted online sequential extreme learning machine algorithm that efficiently assign higher weights to minority class and lesser weight to majority class in order to alleviate the CIL problem. This weighting tuning was based on G_{mean} optimization which is a popular statistical evaluation criteria for the CIL problems [47]. Because of no need of external storage of past learned data and simple weight tuning method, the convergence rate of WOS-ELM is very fast, but it requires the setting of optimal number of neurons in the hidden layer. Also, the proposed algorithm works only in case of bi-class problems with known feature mappings.

G. Online Sequential Extreme Learning Machine with Kernels (KOS-ELM)

To overcome the limiting condition of known feature mappings for WOS-ELM, kernel based OS-ELM was presented in [33] called KOS-ELM. This was the first attempt to combine

OS-ELM with nonlinear adaptive filtering technique. Although the resultant algorithm has better classification capability for bi-class CIL problem, but it works only when the data is entered in the hidden layer one by one as for each time a new center is calculated for each input data sample which hinders its utilization for large datasets. Two variants of KOS-ELM; approximate linear dependency kernel-based OS-ELM (ALD-KOS-ELM) and fixed budget kernel based OS-ELM (FB-KOS-ELM) was also proposed in [33] in order to apply KOS-ELM algorithm to large datasets that are sparse in nature.

H. An Incremental Extreme Learning Machine for Online Sequential Learning (I-ELM) Algorithm

Based on the idea of ELM algorithm, in [48], three variants of incremental extreme learning machine (IELM) was proposed to solve online-sequential learning problems namely:

- Minimal Norm Incremental Extreme Learning Machine (MN-IELM)
- Least Square Incremental Extreme Learning Machine (LS-IELM)
- Kernel Based Incremental Extreme Learning Machine (KB-IELM)

MN-IELM is used with smaller dataset but for larger dataset LS-IELM is preferred with known feature mappings. For unknown feature mapping KB-IELM was formulated.

I. Robust Online Sequential Extreme Learning Machine (ROS-ELM) Algorithm

Even though the generalization ability of OS-ELM improves by using ensemble learning technique as studied in [29], Zhou et al. in 2002 deduced that the selective ensemble learning technique is even better than the standard ensemble learning technique for NNs [28]. Based on this idea of selective ensemble learning a ROS-ELM algorithm was presented in [49]. An adaptive selective ensemble method based on particle swarm optimization was used with OS-ELM. The output error in root mean square sense is compared to a threshold value to decide whether to do selective ensemble using PSO or to just proceed with standard EOS-ELM algorithm. This adaptive selective mechanism results in the improvement of the inherent instability of the EOS-ELM.

J. Ensemble of Subset Online Sequential Extreme Learning Machine (ESOS-ELM) Algorithm

To solve the CIL problem of data having timeliness property with ensemble-based OS-ELM, Mirza B. & Lin et al. proposed ensemble of subsets of OS-ELM algorithm termed as ESOS-ELM in the literature [50]. ESOS-ELM consists of ordinary EOS-ELM with external memory to store previously learned information. A control is implemented to detect the validity of the incoming data which enters the main ensemble of OS-ELMs in balanced subsets. The main architecture of ESOS is shown in Fig. 3. Though this framework helps getting better classification efficiency with imbalance data having timeliness property, the major disadvantage was its restriction to just bi-class classification problems.

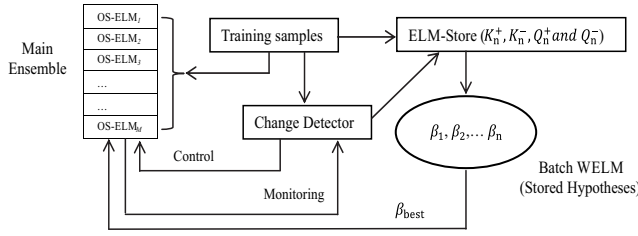


Fig. 3. ESOS-ELM Architecture.

K. Volting based Weighted Online Sequential Extreme Learning Machine (VWOS-ELM) Algorithm

The bi-class classification shortcoming of original WOS-ELM for CIL problems and even in ESOS-ELM with timeliness data was further addressed in [34] by the same authors and they proposed an improved framework called VWOS-ELM. Mirza B. et al. extended the idea presented in [50] and replaced OS-ELM with WOS-ELM proposed initially for stationary bi-class CIL problems. The architecture thus proposed is shown in Fig. 4. The control for the detection of implicit time information in the incoming datum was replaced with the majority voting system that selects the trained WOS-ELM. Each WOS-ELM in VWOS-ELM has same number of hidden neurons with different initial parametric settings. The enhanced sequential learning algorithm thus produced has better classification property in comparison to WOS-ELM for bi-class CIL problems and can also handle multi-class CIL problems. This technique however cannot work with timeliness data as there is no detection mechanism for time validity of the input data stream. Also, another drawback of this framework was to specify the optimal number of neurons in the hidden layer.

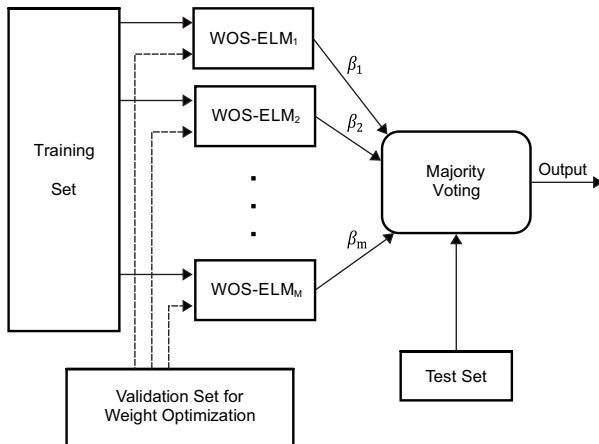


Fig. 4. VWOS-ELM Architecture.

L. Meta-cognitive Online Sequential Extreme Learning Machine (MOS-ELM) Algorithm

Newly proposed MOS-ELM [35] for CIL problems resolves the major issues in WOS-ELM & VWOS-ELM of classifying

the timeliness data and the problem of only working with the bi-class dataset. MOS-ELM can solve both timeliness and class imbalance classification problems. Effective cost weighting with data sampling was used to extend bi-class CIL to multi-class CIL and a windowing method to detect the validity of change required for timeliness data classification problems. There was no need to specify the optimal number of neurons in the hidden layer during the initialization process that resulted in faster convergence with better generalization efficiency [32].

M. M-estimator based Online Sequential Extreme Learning Machine (M-OS-ELM) Algorithm

Recently a variation of OS-ELM was proposed to cater the classification of noisy data especially in case of chaotic time series [51]. By replacing the least square with M-estimator in the cost function of the optimization function of OS-ELM, an iterative solution, that solves the M-estimator based model, is devised. To further find the optimal threshold value for M-estimator, a sequential parametric estimation was proposed. The proposed M-OS-ELM learning algorithm was found to be very robust in contrast to OS-ELM and ReOS-ELM for noisy data.

N. Online Sequential Regularized Extreme Learning Machine (OS-RELM) Algorithm

Owing to the ill-posed and stability issues with OS-ELM in case of smaller number of neurons in the hidden layer than the datum size, in [52], an improved version of OS-ELM similar to the ReOS-ELM [26] was presented. OS-RELM makes use of the regularization to deal with the inherent ill-posed issue. Moreover, it uses Leave-One-Out cross-validation method [53] to learn from the new incoming data. A novel update scheme was formulated to eliminate the required initialization phase for stability and optimal performance during the sequential learning. OS-RELM outperforms the basic OS-ELM in terms of computational cost with superior generalization ability.

O. Online Sequential Reduced Kernel Extreme Learning Machine (OS-RKELM) Algorithm

Although OS-RELM somewhat addressed the issue of computationally ill-posed and stability with OS-ELM, it still would get trapped in singularity problem when the incoming datum size is larger as compared to neurons in the hidden layer. Deng et al. proposed a solution to this problem by incorporating different kernels for the hidden neurons. The presented framework of OS-RKELM takes a subset of training samples in the initialization phase to train the kernel based hidden neurons and then afterwards the algorithm can handle incoming data streams arriving either one by one or in chunks [54]. OS-RKELM overcome the drawback of OS-ELM of specifying the optimal random initial weights for better generalization ability in case of implicit feature mapping.

P. Weighted Online Sequential Extreme Learning Machine with Kernels (WOS-ELMK) Algorithm

ESOS-ELM [50], and WOS-ELM [32] works only for bi-class CIL problems. Also, both these algorithms require to specify the number of neurons in the hidden layer for optimal performance. VWOS-ELM [34], presented for multi-class CIL problems, resolves the major stability problem inherited with single WOS-ELM when hidden layer neurons are less in number than the incoming data size via majority voting based ensemble of WOS-ELM. Lately, another online sequential learning algorithm (MOS-ELM) [35] proposed for multi-class CIL classification. All these variations of online sequential ELM frameworks for CIL problems require explicit feature mapping in the hidden layer. The idea of using kernel based hidden layer neurons for training the data with implicit feature mapping was exploited in [36] and the author presented a new framework called WOS-ELMK. The proposed algorithm works extremely well with multi-class CIL problems by utilizing implicit kernel mapping rather than explicit random feature mapping. To cater larger class-imbalance data streams, a window approach for transferring the data to hidden layer was used with an external memory control to improve the convergence.

Q. Adaptive Forgetting Factor with Generalized Regularization Online Sequential Extreme Learning Machine (AFGR-OS-ELM) Algorithm

DFF-OS-ELM, proposed in [44], uses a variable exponential-forgetting factor with regularization which causes the regularization effect to slowly fade away. Thus, eventually the DFF-OS-ELM will get trapped in the same problematic operation of ill-posed matrix inversion. In order to overcome the instability and limited use of DFF-OS-ELM, W. Guo et al. presented a more sophisticated framework called AFGR-OS-ELM in [55]. The author employed a novel adaptable forgetting mechanism with a more generalized regularization term [56], [57] in the cost function instead of using the exponential forgetting regularization term in [44]. Doing so, the improved AFGR-OS-ELM framework ensured a constant regularization effect without fading throughout the entire learning process which ultimately removed the ill-posed problem with most of OS-ELM algorithms.

R. Improved Online Sequential Extreme Learning Machine Algorithm

To avoid the ill-posed matrix inversion problem associated with OS-ELM, the size of hidden neurons is usually larger than the input datum size. Genetic algorithm (GA), [58] which is a renowned global optimization tool, was used recently to optimize the randomly initialized weight and biases of simple OS-ELM [59]. The improved OS-ELM was able to give better generalization efficiency and opened the door to a new phase in the research of finding a new sequential learning algorithm which combines OS-ELM with various evolutionary algorithms.

S. Combination Weight based Ensemble of Online Sequential Extreme Learning Machine (CW-EOS-ELM) Algorithm

In January 2019, CW-EOS-ELM algorithm was proposed which selects the OS-ELM from the ensemble, based on individual ELMs correlation and running error [60]. Adaboost algorithm was used to set the initial weights and biases of every OS-ELM in the ensemble and then later are updated analytically during the update phase in accordance with the aggregate game theory. The proposed framework can only work with bi-class classification problems and does not require any beforehand settings of weights and biases of individual OS-ELMs establishing the fact that re-learn process is dynamic.

IV. SUMMARY & DISCUSSION

Usually the performance of various SHLFNs online-sequential learning algorithms are compared in terms of the type of problem and input data stream type i.e. imbalance or timeliness-based data. There are very few algorithms that can be used to solve bi-class as well as multi-class classification problem. In terms of handling imbalance data streams, the proposed framework makes use of computationally efficient weighting method for different classes of data or intelligent sampling techniques to extract balanced subsets of data. Windowing method is widely used to capture the timeliness property of the incoming data during the retraining of the network. Even though major improvements in the sequential learning algorithms have been suggested over the years for the SHLFNs, all have some limitation of their own:

- Very few perform well with timeliness-based data e.g. RAN-LTM, SRAN, FOS-ELM, DFF-OS-ELM, MOS-ELM and AFGR-OS-ELM.
- Some are better suited for the imbalanced data streams e.g. ESOS-ELM.
- Few of them can handle both bi-class and multi-class classification problem with timeliness-based imbalanced input data streams e.g. MOS-ELM and VWOS-ELM.
- Even though some can solve problems with input data having gradual change in its validity, they are unable to give satisfactory results with rapid timeliness drifts, such as SRAN, OS-ELM.
- SRAN, GIRAN, OS-ELM, EOS-ELM & WOS-ELM are commonly studied algorithms by most scholars for online sequential learning and are further compared with the improved algorithms

Table I summarizes all the above methods just to provide a theoretical guidance for the practical engineering applications when using any one of these learning algorithms.

The most common performance indices for any training algorithms are the approximation accuracy (training error) and the generalization ability (test error) which are calculated using the average value of the root mean square error produced by the algorithm. The stability is measured using the average result's standard deviation. The complexity of the network trained by the algorithm can also be obtained using the total number of the hidden layer neurons and learning speed is approximated by the amount of training time required. The activation functions of the RAN and its derivatives are fixed

TABLE I
SEQUENTIAL LEARNING ALGORITHMS WITH THEIR APPROPRIATE SCENARIOS

Algorithms	Bi-Class Classification	Multi-Class Classification	Imbalanced Data Stream	Timeliness Based Data
RAN	✓	×	×	×
RAN-LTM	✓	×	×	✓
RAN-EKE	✓	×	×	×
MRAN	✓	×	×	×
EMRAN	✓	×	×	×
HMRAN	✓	×	×	×
GGAP-RBF	✓	×	×	×
GAP-DRBF	✓	×	×	×
GIRAN	✓	×	×	×
SRAN	✓	×	×	✓
RLS-ELM	✓	✓	×	✓
OS-ELM	✓	✓	×	✓
EOS-ELM	✓	✓	×	✓
ReOS-ELM	✓	✓	×	✓
FOS-ELM	✓	✓	×	✓
VOS-ELM	✓	✓	×	✓
WOS-ELM	✓	×	✓	×
KOS-ELM	✓	×	✓	×
I-ELM	✓	✓	×	✓
ROS-ELM	✓	✓	×	✓
ESOS-ELM	✓	×	×	✓
VWOS-ELM	✓	✓	✓	✓
MOS-ELM	✓	✓	✓	✓
M-OS-ELM	✓	✓	×	✓
OS-RELM	✓	✓	×	✓
OS-RKELM	✓	✓	×	×
WOS-ELMK	✓	×	✓	×
AFGR-OS-ELM	✓	✓	×	✓
Improved OS-ELM	✓	✓	×	✓
CW-OS-ELM	✓	×	×	✓

and is in the form of RBF except in GAP-DRBF where DRBF was used whereas in case of OS-ELM and its different variants of, two different activation functions of RBF and Sigmoid (SIG) forms are mostly used.

In RAN, RAN-LTM, RAN-EKE, MRAN, EMRAN, HMRAN, GGAP-RBF, GAP-DRBF and GIRAN algorithms, the parameter of the expected approximation accuracy is required, and the performance of the network is controlled by the adjustment of this parameter which is usually achieved through the trial and error method. Similarly, in ELM variants, the setting of the number of hidden layer neurons is used to control the performance of the network.

The sequential learning algorithms including RAN, RAN-LTM, RAN-EKE, MRAN, EMRAN, HMRAN, GGAP-RBF, GIRAN and SRAN shows good stability when the activation function is RBF. Also, the standard deviation in this case is zero which confirms robustness. In case of learning algorithm based on OS-ELM, the standard deviation usually is large which makes the algorithm less robust. The main reason for the poor stability of the algorithm is that the weight connections between the inputs and hidden layer neurons are set randomly before training which are later updated by the iterative algorithm. Since the initial weights are different for each iteration setting thereby, the performance of the network is slightly different at each iterative step which reduces the stability of the algorithm.

In addition, for most of the variants of OS-ELM learning algorithm, it is necessary to use the training samples to initialize

the network parameters. When the size of hidden layer is large, more training samples are needed to initialize the parameters and since these input data sample have not been learned by the network, they will reduce the approximation ability of the network. Conversely if the total number of sample data is small, it will result in large training error of the algorithm or even instability. To get good approximation accuracy and generalization ability different activation function have been used for the hidden layer neurons [61].

For any OS learning algorithm, the process of learning from the previous data sample must end before the new sample enters the network allowing the algorithm to be fast enough to deal with real-time problems. The training time associated with RAN learning algorithm and its variants is longer as compared with sequential learning ELM variants.

Table II shows the performance of some of the studied sequential learning algorithms in terms of learning time, generalization ability (testing error), accuracy (training error) and no. of hidden layer neurons used for Mackey chaotic time series dataset. A better visualization for learning time for these algorithms is given in Fig. 5. The Mackey chaotic time series dataset used include 4000 data samples with 300 testing and validation samples.

Table III-IV shows the results of three major algorithms in case of bi-class and multi-class imbalance problems. For bi-class CIL problem, phoneme dataset is used with a class imbalance factor of 2.4, 4500 training samples, 300 test samples and 300 validation samples. Similarly, for multi-class

TABLE II
COMPARISON OF DIFFERENT SEQUENTIAL LEARNING ALGORITHMS
BASED ON DIFFERENT PERFORMANCE INDICES FOR MACKEY CHAOTIC
TIME SERIES DATASET

Algorithm	No. of Hidden Neurons	Learning Time (sec.)	RMSE (Training)	RMSE (Testing)
RAN	39	58.12	0.1006	0.0431
RAN-EKF	23	62.24	0.0725	0.0236
MRAN	16	57.5	0.1102	0.0337
EMRAN	18	53.6	0.0941	0.0245
GGAB-RBF	19	24.32	0.0667	0.0312
GIRAN	13	26.42	0.1289	0.0609
OS-ELM (SIG)	120	7.11	0.0177	0.0183
ReOS-ELM (SIG)	120	6.89	0.0181	0.0175
ROS-ELM	120	6.94	0.0146	0.0149
EOS-ELM (SIG)	120	6.86	0.0179	0.0183
VOS-ELM	120	5.91	0.0172	0.0175
FOS-ELM	120	7.34	0.0165	0.0192
KOS-ELM	120	2.54	0.0153	0.0191
OS-ELMK	120	9.42	0.0142	0.0242
OS-RKELM	120	7.05	0.0042	0.0045
M-OS-ELM	120	8.52	0.0078	0.0074
CW-OS-ELM	120	6.94	0.0152	0.0165

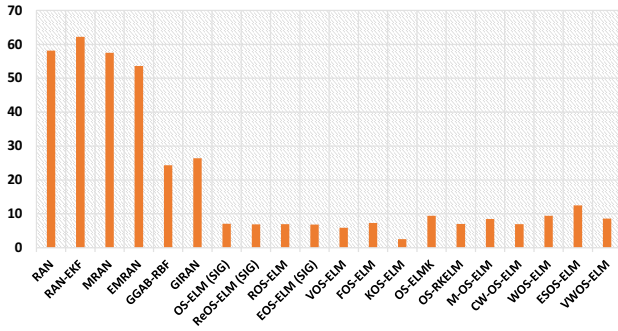


Fig. 5. Learning Time for Different Sequential Learning Algorithms for Mackey Chaotic Time Series Dataset.

CIL problem, yeast dataset having 9 classes with a ratio of 463:429:244:163:51:44:35:30:20 and 1450 samples was used.

TABLE III
COMPARISON OF DIFFERENT SEQUENTIAL LEARNING ALGORITHMS FOR
BI-CLASS PHONEME DATASET

Algorithm	No. of Classifiers	Learning Time (sec.)	G _{means}
WOS-ELM	7	9.42	84.2
ESOS-ELM	15	12.51	84.4
VWOS-ELM	7	8.59	84.7

TABLE IV
COMPARISON OF DIFFERENT SEQUENTIAL LEARNING ALGORITHMS FOR
MULTI-CLASS YEAST DATASET

Algorithm	No. of Classifiers	Learning Time (sec.)	G _{means}
WOS-ELM	7	12.57	90.6
ESOS-ELM	15	21.51	90.4
VWOS-ELM	7	10.64	91.6

Finally, Table V summarizes the all these sequential learning techniques in terms of their merits & demerits. In summary, when you need to choose a higher stability algorithm, you can choose anyone from RAN or its derivatives while if you

need a higher processing speed algorithm, you should choose OS-ELM alternatives.

V. FUTURE WORK

The main focus of this review article is to summarize the sequential learning algorithms used with single hidden layer feed-forward neural networks. The authors tried to list the advantages as well as the various limitations associated with the existing algorithms to provide a prior guideline for anyone using one of the above mentioned algorithms. The following problems were identified as open research queries and are worth exploring for potential researchers.

- 1) Even after more than a decade of research in ELM, the problem of settings the optimal number of neurons in the hidden layer is still there and needs to be randomly selected beforehand. Thus, implying that the improved algorithms have not solved the fundamental problem associated with ELM stability.
- 2) At present, the practical applications of sequential learning variants are very limited and there is plenty of work needs to be done especially to use SHLNs sequential learning algorithms for video and text streams classification.
- 3) Although the sequential learning algorithms for multi-layer neural network have attracted many researchers, their applications to practical problems are very rare. Therefore to explore application of sequential learning with MLFNs is much needed.
- 4) Another contribution, for the improvement in the generalization ability of sequential learning algorithms, can be made by combining conventional learning algorithms e.g. support vector machine, decision trees, random forest, nearest neighbour with them.
- 5) Timeliness based data streams, over the period of time, completely reshape the correlation between the data stream features thus opening a new research objective of using correlation to predict the data stream type.

VI. CONCLUSION

Compared with the batch learning algorithm of SHLFNs, the OS learning algorithm can be applied to the processing of real-time problems, which is more suitable for industrial environment. Built on the brief summary of existing SHLFNs online-sequential learning algorithm, this paper compares the performance of various algorithms, based on training time, approximation accuracy, generalization ability, and algorithm stability. The merits and demerits of various online learning algorithms and the scope of their applications are pointed out, which provides a theoretical guidance for the practical application of the SHLFNs online learning algorithm.

VII. ACKNOWLEDGMENT

The authors acknowledge the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia for this research work.

REFERENCES

- [1] R. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, vol. 4, no. 2, pp. 4–22, 1987.
- [2] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and intelligent laboratory systems*, vol. 39, pp. 43–62, 1997.
- [3] B. Widrow, R. G. Winter, and R. A. Baxter, "Layered neural nets for pattern recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1109–1118, 1988.
- [4] E. Blanzieri, "Learning algorithms for radial basis function networks: synthesis, experiments and cognitive modelling," *PHD Dissertation*, 1998.
- [5] B. Cheng and D. M. Titterton, "Neural networks: A review from a statistical perspective," *Statistical science*, pp. 2–30, 1994.
- [6] G. B. Huang, Y. Q. Chen, and H. A. Babri, "Classification ability of single hidden layer feedforward neural networks," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 799–801, 2000.
- [7] G. B. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 274–281, 2003.
- [8] J. Wang, A. Belatreche, L. Maguire, and M. McGinnity, "Online versus offline learning for spiking neural networks: A review and new strategies," *2010 IEEE 9th International Conference on Cybernetic Intelligent Systems*, pp. 1–6, 2010.
- [9] S. Smale and Y. Yao, "Online learning algorithms," *Foundations of computational mathematics*, vol. 6, no. 2, pp. 145–170, 2006.
- [10] S. C. Hoi, J. Wang, and P. Zhao, "Libol: A library for online learning algorithms," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 495–499, 2014.
- [11] Y. A. Lecun, L. Bottou, G. B. Orr, and K. R. Müller, "Efficient backprop," *Neural networks: Tricks of the trade*, pp. 9–48, 2012.
- [12] J. Platt, "A Resource-Allocating Network for Function Interpolation," *Neural Computation*, vol. 3, pp. 213–225, 1991.
- [13] K. Okamoto, S. Ozawa, and S. Abe, "A fast incremental learning algorithm of RBF networks with long-term memory," *Proceedings of the International Joint Conference on Neural Networks*, vol. 1, pp. 102–107, 2003.
- [14] V. Kadirkamanathan and M. Niranjan, "A function estimation approach to sequential learning with neural networks," *Neural Computation*, vol. 5, no. 6, pp. 954–975, 1993.
- [15] L. Yingwei, N. Sundararajan, and P. Saratchandran, "A sequential learning scheme for function approximation using minimal radial basis function neural networks," *Neural Computation*, vol. 9, no. 2, pp. 461–478, 1997.
- [16] K. Nishida, K. Yamauchi, and T. Omori, "An online learning algorithm with dimension selection using minimal hyper basis function networks," *Systems and Computers in Japan*, vol. 37, no. 11, pp. 11–21, 2006.
- [17] G. B. Huang, P. Saratchandran, and N. Sundararajan, "An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks," *IEEE Transactions on Systems, Man, Cybernetics, Part B*, vol. 34, no. 6, pp. 2284–2292, 2004.
- [18] —, "A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 57–67, 2005.
- [19] D. Xun and W. Chang-Shan, "An Efficient Sequential Learning Algorithm for Growing and Pruning Direct-Link RBF (DRBF) Networks," *Neural Networks and Brain, 2005. ICNN&B'05. International Conference on*, vol. 1, pp. 494–498, 2005.
- [20] B. Li and X. Lai, "An improved GGAP-RBF algorithm and its application to function approximation," *Pattern Recognition and Artificial Intelligence*, vol. 20, no. 2, pp. 230–253, 2006.
- [21] S. Suresh, K. Dong, and H. J. Kim, "A sequential learning algorithm for self-adaptive resource allocation network classifier," *Neurocomputing*, vol. 73, pp. 3012–3019, 2010.
- [22] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
- [23] N. Y. Liang, G. B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, 2006.
- [24] B. Li, J. Wang, Y. Li, and Y. Song, "An improved on-line sequential learning algorithm for extreme learning machine," *International Symposium on Neural Networks*, pp. 1087–1093, 2007.
- [25] H. T. Huynh and Y. Won, "Online training for single hidden-layer feed-forward neural networks using RLS-ELM," *Computational Intelligence in Robotics and Automation (CIRA)*, pp. 469–473, 2009.
- [26] —, "Regularized online sequential learning algorithm for single-hidden layer feedforward neural networks," *Pattern Recognition Letters*, vol. 32, no. 14, pp. 1930–1935, 2011.
- [27] A. N. Tikhonov, A. Goncharsky, V. Stepanov, and A. G. Yagola, 2013.
- [28] Z. H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: many could be better than all," *Artificial intelligence*, vol. 137, no. 1–2, pp. 239–263, 2002.
- [29] Y. Lan, Y. C. Soh, and G. B. Huang, "Ensemble of online sequential extreme learning machine," *Neurocomputing*, vol. 72, pp. 3391–3395, 2009.
- [30] J. Cao, Z. Lin, G. B. Huang, and N. Liu, "Voting based extreme learning machine," *Information Sciences*, vol. 185, no. 1, pp. 66–77, 2012.
- [31] J. Zhao, Z. Wang, and D. S. Park, "Online sequential extreme learning machine with forgetting mechanism," *Neurocomputing*, vol. 87, pp. 79–89, 2012.
- [32] B. Mirza, Z. Lin, and K. A. Toh, "Weighted online sequential extreme learning machine for class imbalance learning," *Neural processing letters*, vol. 38, no. 3, pp. 465–486, 2013.
- [33] S. Scardapane, D. Comminiello, M. Scarpiniti, and A. Uncini, "Online sequential extreme learning machine with kernels," *IEEE transactions on neural networks and learning systems*, vol. 26, pp. 2214–2220, 2014.
- [34] B. Mirza, Z. Lin, J. Cao, and X. Lai, "Voting based weighted online sequential extreme learning machine for imbalance multi-class classification," *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 565–568, 2015.
- [35] B. Mirza and Z. Lin, "Meta-cognitive online sequential extreme learning machine for imbalanced and concept-drifting data classification," *Neural Networks*, vol. 80, pp. 79–94, 2016.
- [36] S. Ding, "Kernel based online learning for imbalance multiclass classification," *Neurocomputing*, vol. 277, pp. 139–148, 2018.
- [37] P. C. Ko and P. C. Lin, "Resource allocation neural network in portfolio selection," *Expert Systems with Applications*, vol. 35, pp. 330–337, 2008.
- [38] S. Ozawa, S. Abe, S. Pang, and N. Kasabov, "Online Incremental Face Recognition System Using Eigenface Feature and Neural Classifier," *State of the Art in Face Recognition*, 2009.
- [39] N. Sundararajan, P. Saratchandran, and Y. W. Lu, 1999.
- [40] Y. Li, N. Sundararajan, and P. Saratchandran, "Analysis of minimal radial basis function network algorithm for real-time identification of nonlinear dynamic systems," *IEEE Proceedings-Control Theory and Applications*, vol. 147, no. 4, pp. 476–484, 2000.
- [41] V. S. Asirvadam, S. F. McLoone, and G. W. Irwin, "Fast and efficient sequential learning algorithms using direct-link RBF networks," *Neural Networks for Signal Processing*, pp. 209–218, 2003, NNSP'03.
- [42] P. L. Bartlett, "The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network," *IEEE transactions on Information Theory*, vol. 44, no. 2, pp. 525–536, 1998.
- [43] Y. Gu, J. Liu, Y. Chen, X. Jiang, and H. Yu, "TOSELM: timeliness online sequential extreme learning machine," *Neurocomputing*, vol. 128, pp. 119–127, 2014.
- [44] S. G. Soares and R. Araújo, "An adaptive ensemble of on-line extreme learning machines with variable forgetting factor for dynamic system prediction," *Neurocomputing*, vol. 171, pp. 693–707, 2016.
- [45] R. Kulhav, "Restricted exponential forgetting in real-time identification," *Automatica*, vol. 23, no. 5, pp. 589–600, 1987.
- [46] J. Cao, Z. Lin, and G. B. Huang, "Voting base online sequential extreme learning machine for multi-class classification," *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, pp. 2327–2330, 2013.
- [47] R. Batuwita and V. Palade, "FSVM-CIL: fuzzy support vector machines for class imbalance learning," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 3, pp. 558–571, 2010.
- [48] L. Guo, J. H. Hao, and M. Liu, "An incremental extreme learning machine for online sequential learning problems," *Neurocomputing*, vol. 128, pp. 50–58, 2014.
- [49] Y. Liu, "ROS-ELM: A Robust Online Sequential Extreme Learning Machine for Big Data Analytics," *Proceedings of ELM-2014*, vol. 1, pp. 325–344, 2015.
- [50] B. Mirza, Z. Lin, and N. Liu, "Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift," *Neurocomputing*, vol. 149, pp. 316–329, 2015.
- [51] W. Guo, T. Xu, and K. Tang, "M-estimator-based online sequential extreme learning machine for predicting chaotic time series with outliers," *Neural Computing and Applications*, vol. 28, no. 12, pp. 4093–4110, 2017.

- [52] Z. Shao and M. J. Er, "An online sequential learning algorithm for regularized extreme learning machine," *Neurocomputing*, vol. 173, pp. 778–788, 2016.
- [53] G. C. Cawley and N. L. Talbot, "Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers," *Pattern Recognition*, vol. 36, no. 11, pp. 2585–2592, 2003.
- [54] W. Y. Deng, Y. S. Ong, P. S. Tan, and Q. H. Zheng, "Online sequential reduced kernel extreme learning machine," *Neurocomputing*, vol. 174, pp. 72–84, 2016.
- [55] W. Guo, T. Xu, K. Tang, J. Yu, and S. Chen, "Online sequential extreme learning machine with generalized regularization and adaptive forgetting factor for time-varying system prediction," *Mathematical Problems in Engineering*, vol. 2018, 2018.
- [56] B. Prez-Sanchez, O. Fontenla-Romero, and B. Guijarro-Berdias, "A review of adaptive online learning for artificial neural networks," *Artificial Intelligence Review*, vol. 49, no. 2, pp. 281–299, 2018.
- [57] C. Wilson, Y. Bu, and V. Veeravalli, 2019, arXiv preprint.
- [58] J. H. Holland, "Outline for a logical theory of adaptive systems," *Journal of the ACM (JACM)*, vol. 9, no. 3, pp. 297–314, 1962.
- [59] B. Dai, C. Gu, E. Zhao, K. Zhu, W. Cao, and X. Qin, "Improved online sequential extreme learning machine for identifying crack behavior in concrete dam," *Advances in Structural Engineering*, vol. 22, no. 2, pp. 402–412, 2019.
- [60] H. Yu, X. Sun, and J. Wang, "Ensemble OS-ELM based on combination weight for data stream classification," *Applied Intelligence*, vol. 49, no. 6, pp. 2382–2390, 2019.
- [61] B. Li, Y. Li, and X. Rong, "The extreme learning machine learning algorithm with tunable activation function," *Neural Computing Applications*, vol. 22, no. 3–4, pp. 531–539, 2013.
- [62] N. Shiraga, S. Ozawa, and S. Abe, "A reinforcement learning algorithm for neural networks with incremental learning ability," *Proceedings of the 9th International Conference on Neural Information Processing*, vol. 5, pp. 2566–2570, 2002.
- [63] T. Kidera, S. Ozawa, and S. Abe, "An incremental learning algorithm of ensemble classifier systems," *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pp. 3421–3427, 2006.
- [64] V. S. Asirvadam, S. F. McLoone, and G. W. Irwin, "Computationally efficient sequential learning algorithms for direct link resource-allocating networks," *Neurocomputing*, vol. 69, no. 1–3, pp. 142–157, 2005.
- [65] Y. Li, N. Sundararajan, and P. Saratchandran, "Neuro-flight controllers for aircraft using minimal resource allocating networks (MRAN)," *Neural Computing & Applications*, vol. 10, no. 2, pp. 172–183, 2001.
- [66] K. Salahshoor and A. S. Kamalabady, "On-line identification of multi-variable processes using EKF learning-based adaptive neural networks," *2008 IEEE Conference on Cybernetics and Intelligent Systems*, pp. 12–17, 2008.
- [67] S. Li, Q. Chen, and G. B. Huang, "Dynamic temperature modeling of continuous annealing furnace using GGAP-RBF neural network," *Neurocomputing*, vol. 69, no. 4–6, pp. 523–536, 2006.
- [68] H. G. Han, J. F. Qiao, and Q. L. Chen, "Model predictive control of dissolved oxygen concentration based on a self-organizing RBF neural network," *Control Engineering Practice*, vol. 20, no. 4, pp. 465–476, 2012.
- [69] S. Padma, S. S. Kumar, and R. Manavalan, "Performance analysis for classification in balanced and unbalanced data set," *2011 6th International Conference on Industrial and Information Systems*, pp. 300–304, 2011.
- [70] Y. Li, R. Qiu, and S. Jing, "Intrusion detection system using Online Sequence Extreme Learning Machine (OS-ELM) in advanced metering infrastructure of smart grid," *PloS one*, vol. 13, no. 2, pp. 192 216–192 216, 2018.
- [71] Y. Li and Z. Yang, "Application of EOS-ELM with binary Jaya-based feature selection to real-time transient stability assessment using PMU data," *IEEE Access*, vol. 5, pp. 23 092–23 101, 2017.
- [72] H. Trung Huynh and Y. Won, "Regularized online sequential learning algorithm for single-hidden layer feedforward neural networks," 2011.
- [73] J. Wang, R. Ran, and Y. Zhou, "A Short-term photovoltaic power prediction model based on an FOS-ELM algorithm," *Applied Sciences*, vol. 7, no. 4, pp. 423–423, 2017.
- [74] H. Zhang, S. Zhang, and Y. Yin, "Online sequential ELM algorithm with forgetting factor for real applications," *Neurocomputing*, vol. 261, pp. 144–152, 2017.
- [75] E. I. Georga, V. C. Protopappas, D. Polyzos, and D. I. Fotiadis, "Online prediction of glucose concentration in type 1 diabetes using extreme learning machines," *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3262–3265, 2015.
- [76] H. Zhang, S. Zhang, and Y. Yin, "Kernel online sequential ELM algorithm with sliding window subject to time-varying environments," *Memetic Computing, journal article*, vol. 10, no. 1, pp. 43–52, 2018.
- [77] Z. Tian, G. Wang, Y. Ren, S. Li, and Y. Wang, "An adaptive online sequential extreme learning machine for short-term wind speed prediction based on improved artificial bee colony algorithm," *Neural Network World*, vol. 28, no. 3, pp. 191–212, 2018.
- [78] L. Hu, Y. Chen, J. Wang, C. Hu, and X. Jiang, "OKRELM: online kernelized and regularized extreme learning machine for wearable-based activity recognition," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 9, pp. 1577–1590, 2018.
- [79] P. A. Alaba, "Towards a more efficient and cost-sensitive extreme learning machine: A state-of-the-art review of recent trend," *Neurocomputing*, vol. 350, pp. 70–90, 2019.
- [80] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast kernel classifiers with online and active learning," *Journal of Machine Learning Research*, vol. 6, pp. 1579–1619, 2005.



M. H. Arshad received his B.Sc degree in Electrical Engineering from University of the Punjab, Lahore - Pakistan. He is currently enrolled as a full-time graduate scholar in Electrical Engineering Department with major in 'Control Engineering' at King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia. His research focus is on power electronics, nonlinear dynamical systems, variable speed drives for multiphase induction motor, digital filter design and flatness based robust control of nonlinear systems.



M. A. Abido (SM'15) received the B.Sc. (Honors with first class) and M.Sc. degrees in EE from Menoufia University, Shebin El-Kom, Egypt, in 1985 and 1989, respectively, and the Ph.D. degree from King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 1997. He is currently a Distinguished University Professor at KFUPM, KSA and also a Senior Researcher at K.A.CARE Energy Research & Innovation Center, Dhahran, KSA. His research interests are power system stability, planning, operation, and optimization techniques applied to power systems. Dr. Abido is the recipient of KFUPM Excellence in Research Award, 2002, 2007 and 2012, KFUPM Best Project Award, 2007 and 2010, First Prize Paper Award of the Industrial Automation and Control Committee of the IEEE Industry Applications Society, 2003, Abdel-Hamid Shoman Prize for Young Arab Researchers in Engineering Sciences, 2005, Best Applied Research Award of 15th GCC-CIGRE Conference, Abu-Dhabi, UAE, 2006, and Best Poster Award, International Conference on Renewable Energies and Power Quality (ICREPQ'13), Bilbao, Spain, 2013. Dr. Abido has been awarded Almarai Prize for Scientific Innovation 2017/2018, Distinguished Scientist, Saudi Arabia, 2018 and Khalifa Award for Education 2017-2018, Higher Education, Distinguished University Professor in Scientific Research, Abu Dhabi, UAE, 2018. Dr. Abido has published two books and more than 350 papers in reputable journals and international conferences. He participated in 50+ funded projects and supervised 50+ MS and PhD students

TABLE V: Summary of Sequential Learning Algorithms in SHLFNs

No.	Algorithms	Advantages	Disadvantages	Compared with
1	SGBP	Sequential learning of data in form of one by one form, faster convergence with redundant data	Slow learning rate with noisy data	RAN [15]
2	RAN	1 st true sequential learning algorithm with input stream in form of one by one or in chunks	Convergence is very slow, un-reliable to use with real-time applications, un-learn over time with redundant input data	SGBP [15], RAN-LTM [62]
3	RAN-LTM	Overcome the issue of unlearning in RAN with external memory	Convergence is very slow, un-reliable to use with real-time applications	RAN [13], [63]
4	RAN-EKF	Replace LMS with EKF, faster convergence rate compared to RAN	Higher computational cost because of EKF based learning method	RAN [14], DRAN-EKF [64]
5	MRAN	Low computational cost because of the method of deleting hidden layer neurons, better generalization accuracy	Convergence is very slow, un-reliable to use with real-time applications	RAN-EKF [65], GAP-RBF [66], EMRAN [16]
6	EMRAN	First RAN variant for real-time applications, faster convergence as compared to MRAN, can work with high dimensional dataset	Un-learn over time with redundant input data, poor generalization accuracy	MRAN [40], HMRAN [16]
7	HMRAN	Improved accuracy, less training time, localized EKF method	Un-learn over time with redundant input data	MRAN [16], GGAP-RBF [67]
8	GGAP-RBF	Faster convergence, well suited real-time applications, less initial parameter settings as compared to RAN and its variants	Conventional EKF method, require more computational resources, Input data needs to be uniformly distributed	RAN-EKF [67], MRAN [68]
9	GAP-DRBF	Replaced RBF with DRBF in GGAP-RBF, decomposed EKF method, low computational cost	Input data needs to be uniformly distributed	RAN, RAN-EKF, MRAN [17]
10	GIRAN	Improved update formula for the addition of new hidden layer neuron, higher accuracy, less initial parameter settings	Same as GGAP-RBF	RAN, MRAN, IRAN, GGAP-RBF [20]
11	SRAN	No need of initial parameter settings	No pruning strategy, computationally expensive because of EKF learning method	MRAN, ELM [69]
12	OS-ELM	Online sequential-learning algorithm, input data can arrive as one by one or chunk by chunk	Not very robust, falls to singularity trap when the incoming datum size is larger than the hidden layer neurons	RAN [12], GGAP-RBF [18], GAP-RBF [17], ELM [70]
13	EOS-ELM	OS ensemble-based learning algorithm, stable with much improved accuracy	No weighting mechanism for individual OS-ELMs	OS-ELM [29], SGBP, MRAN [71]
14	ReOS-ELM	Replaced single factor cost function with bi-optimization cost function, overcome inherit singularity problem of OS-ELM with noisy data, better accuracy, robust	Require balance stationary input data	SGBP, MRAN, GGAP-RBF [26], OS-ELM [72], ELM, M-OS-ELM [51]

Continued on next page

TABLE V: continued

15	FOS-ELM	Forgetting factor mechanism with OS-ELM to predict timeliness information of input data streams, stable, less training time	Only valid for short-term prediction problems	OS-ELM [73], RAN, GGAP-RBF, WOS-ELM [74]
16	VOS-ELM	A majority voting method was introduced with a weighting vector for different OS-ELM in ensemble, less training time, improved accuracy	The neurons in the hidden needs to be randomly set beforehand	OS-ELM [46]
17	WOS-ELM	First OS-ELM ensemble-based algorithm to solve CIL problem	Explicit feature mapping required, number of hidden layer neurons needs to be randomly assigned	OS-ELM [32], TOS-ELM [43], VWOS-ELM [34]
18	KOS-ELM	Similar to WOS-ELM, small generalization error, faster convergence rate, no need of explicit feature mapping	Poor generalization accuracy with large data stream, input data can only be processed in one by one form	OS-ELM [75], FOS-ELM [76],
19	I-ELM	Overcome ill-posed matrix inversion and stability issue with OS-ELM, three variants based on different types of input data streams	Dataset size needs to be known beforehand, activation function depends on the feature vector	OS-ELM [23], ELM [77], OKRELM [78]
20	ROS-ELM	Selective ensemble-based OS-ELM, more stable	Similar to EOS-ELM	OS-ELM, EOS-ELM [49]
21	ESOS-ELM	Can work with bi-class CIL problem, use external memory to detect timeliness information in the input datum	Works only with bi-class problems, poor generalization ability with stationary balance dataset	WOS-ELM [50], FOS-ELM, VWOS-ELM [76]
22	VWOS-ELM	WOS-ELM ensemble-based algorithm to solve CIL problem, higher classification accuracy	Similar to WOS-ELM	ESOS-ELM [76], OS-ELM, MRAN [79]
23	MOS-ELM	Similar to ESOS-ELM, can handle multi-class classification problems	Explicit feature mappings are required, kernels cannot be used	ELM, OS-ELM [79], WOS-ELM, VWOS-ELM [35]
24	M-OS-ELM	Robust as compared to OS-ELM & ReOS-ELM	Similar to ReOS-ELM	ELM, ReOS-ELM [51]
25	OS-RELM	Improved accuracy with better generalization ability in comparison to OS-ELM	Same as OS-ELM	OS-ELM [26], [52]
26	OS-RKELM	Singularity issue of OS-ELM was resolved using Kernels	Unable to solve nonstationary CIL problems	SGBP, OS-ELM [36], BSGD [80]
27	WOS-ELMK	Similar to WOS-ELM, can work with multi-class data, faster convergence rate, no need of explicit feature mapping	No mechanism to detect the timeliness property of the incoming data	KOS-ELM, VWOS-ELM [36]
28	AFGR-OS-ELM	Completely overcome the ill-posed matrix inversion problem of OS-ELM by using directional forgetting factor mechanism with OS-ELM, can work with bi-class & multi-class problem	Unable to predict the validity period of the incoming data	OS-ELM, ReOS-ELM [77]

Continued on next page

TABLE V: continued

29	Improved OS-ELM	Better generalization accuracy as compared to OS-ELM	Similar to OS-ELM, slow convergence because of hybridization of OS-ELM with evolutionary algorithms	ELM, OS-ELM [59]
30	CW-OS-ELM	Very fast, no need for initial parameters settings	Works only with bi-class problems	EOS-ELM [60]