IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 27, 2019, accepted August 5, 2019, date of publication August 15, 2019, date of current version September 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2935454

Hybrid Data Regression Model Based on the Generalized Adaptive Resonance Theory Neural Network

SHEN YUONG WONG[1], **KEEM SIAH YAP**[2], **CHEE PENG LIM**[3], **AND ERIC WAI MING LEE**[4]

[1]Department of Electrical and Electronics Engineering, Xiamen University Malaysia, Sepang 43900, Malaysia
[2]Department of Electrical and Electronics Engineering, Universiti Tenaga Nasional, Kajang 43000, Malaysia
[3]Institute for Intelligent Systems Research and Innovation, Deakin University, Geelong, VIC 3220, Australia
[4]Department of Architecture and Civil Engineering, City University of Hong Kong, Hong Kong

Corresponding author: Keem Siah Yap (yapkeem@uniten.edu.my)

This work was supported by Universiti Tenaga Nasional (Grant No. RJO10436496 & RJO10436494/B/2019099) and Xiamen University Malaysia Research Fund (Grant No. XMUMRF/2018-C2/IECE/0001).

ABSTRACT The Generalized Adaptive Resonance Theory (GART) model is a supervised online learning neural network based on an integration of Adaptive Resonance Theory (ART) and the Generalized Regression Neural Network (GRNN). It is capable of online learning, and is suitable for undertaking both classification and regression problems. In this paper, we further enhance GART (EGART) with four improvements to formulate a new EGART model. Three operating strategies for the EGART model to undertake regression problems are suggested. The first operating strategy is a fully online learning EGART model. The second operating strategy involves Ordering Algorithm for determining the presentation sequence of training samples during the initial training of EGART model. This strategy is considered as offline learning because a set of data samples must be available for the Ordering Algorithm to compute the best presentation sequence (hereinafter denoted as Ordered-EGART). The third operating strategy aims to demonstrate online learning capability of EART model (the first operating strategy) can still be resumed after training on the Ordered-EGART. It is most suitable for applications with a set of ready data samples and their sequences are predetermined by Ordering Algorithm prior to training of EGART model in offline mode, and triggers online learning when more new data samples become available (hereinafter denoted as IO-EGART). A series of experiments with five benchmark data sets from various application domains is conducted to assess and compare the effectiveness of the EGART model and three operating strategies with those of other methods published in literature as well as two fire safety engineering problems, i.e., predicting the thermal interface height in a single compartment fire and evacuation times in the event of fire. The results and comparisons with other approaches positively demonstrate the efficacy and applicability of EGART model as a useful data regression model for tackling fire safety engineering problems.

INDEX TERMS Adaptive resonance theory, fire evacuation time, general regression neural network, ordering algorithm, regression, thermal interface height.

I. INTRODUCTION

Regression is often used to analyze data in statistics or other scientific research. It is used to examine whether there are some relations between two or more objects. It is desirable to fit a series of influencing factors and results into an equation that can be applied to other similar events for prediction purposes. Several neural network models have been proposed to solve data regression and estimation problems, i.e., the Recurrent Neural Network [1], Time Delay Neural Network [1], Echo State Networks [1], Dual Coordinate Descent Method [2], Successive Overrelaxation Algorithm [2], Modified Newton Method [2], Support Vector Regression [2], Genetic Adaptive Neural Network [3], and Backpropagation Neural Network [3].

Over the past decades, there are two networks which have consistently shown good accuracy in a variety of regression and estimation tasks, they are the Multilayer

The associate editor coordinating the review of this article and approving it for publication was Vicente Alarcon-Aquino.

Perceptron (MLP) [4], [5] and Radial Basis Function (RBF) [4], [6] models. The MLP network handles regression problems by exploiting its hidden neurons to build a nonlinear transformation of combined sigmoid functions based on the hidden neurons. On the other hand, the RBF network handles regression problems using the method of combining nonlinear semi-parametric functions. An example of such functions is the Gaussian kernel function.

However, in a standard MLP or RBF network, learning is based on a method that is known as the offline or batch learning mode [7]. During the process of batch learning, after receiving every single training sample, the network runs an iteration and only after that the network weights are updated [8]. Because batch learning typically involves multiple iterations through a number of training samples, it is usually a time consuming process of learning. A re-training procedure is also necessary when a new sample is presented to the network. During this re-training process, the new sample and all past samples are required for learning. [7]. In addition, choosing the proper learning and network parameters such as the number of nodes, number of learning epochs, learning rate, and stopping criteria require a lot of fine-tuning attempts [7]. Choosing of the network parameters is usually conducted after a series of trial-and-error experiments, which are time-consuming.

The batch learning approach is useful only when the data environment is stationary, and provided that the training samples are sufficiently representative. This is because during learning, information provided by the training samples collected from the environment is encoded by the adjustment (learning) of the network weights. After validating the network performance, the network is put into operation, and no further weight adaptation (or learning) takes place. When the network is presented with an unseen new sample, a built-in mechanism for the network to recognize the novelty is not available. In order to learn new information, the network needs to be re-trained using the new sample, together with all previous samples. This is a major drawback in most neural network models, and it arises from the so-called stability-plasticity dilemma [8]. The dilemma underlies a series of questions, i.e., how a learning system is able to remain plastic or adaptive in response to significant events, and yet remain stable in response to irrelevant ones; how a learning system is able to adapt to new information without corrupting or forgetting previously learned information [8].

For solving stability-plastic dilemma, an Adaptive Resonance Theory (ART)-based neural network namely Fuzzy ARTMAP (FAM) [9] has been proposed. The learning strategy involve minimum operator of Fuzzy and weight of a neuron (or commonly named as "category" in FAM) is a hyper-rectangular. It is capable of online learning of new data samples without disturbing the knowledge learned from previous data samples. When a set of data samples are available before online learning, an Ordering Algorithm [10] is proposed to arrange the presentation of data samples. This ordering algorithm required a parameter, i.e., number cluster centers that is number of class of the application plus one. The use of Ordering Algorithm improves the performance (better accuracy rate and smaller network size), however it is limited to pattern classification problem and it is considered as batch mode learning as all data samples must be available before a fix order of presentation can be decided. From the learning strategy of FAM or FAM with Ordering Algorithm, we can conclude that those approaches are only applicable for pattern classification problem. It is not feasible for data regression task.

Hence, it serves as our motivation to propose an extended ART-based model for online learning and capable of solving data regression task, namely Enhanced Generalized Adaptive Resonance Theory (i.e., EGART model). Unlike the standard ART-based neural networks, EGART model did not use Fuzzy minimum operator (hence, weights of a neuron will not be of a hyper rectangular shape), instead it uses Gaussian distribution for learning and representation of its weights. In addition, this paper presents three different operating strategies of the proposed EGART model, i.e., (i) a fully online learning EGART; (ii) When there is a set of data samples available, the proposed EGART model can be combined with Ordering Algorithm (namely Ordered-EGART model) for offline learning with better performance (smaller error for regression, smaller network size which means lesser number of category created) than the online EGART model; (iii) Initially EGART learned from a set data samples in offline mode, and then capable to continue learning new knowledge from newly available data (namely IO-EGART model).

Therefore, the objectives of this paper are four-fold that derive solutions for the following scenario:

(i) To propose an extended ART-based model for solving regression task.
(ii) To apply the Ordering Algorithm that originally designed for FAM to an Extended ART-based model.
(iii) To propose a strategy that determine number of cluster center of Ordering Algorithm for solving regression task.
(iv) To demonstrate the extended ART-based model that previously learned in batch mode (i.e., learning of a set of data samples, and the data presentation is determined Ordering Algorithm), can continuing performing online learning for newly available data samples.

The proposed EGART model with three operating strategies are tested with five UCI benchmark datasets and two other applications of fire safety engineering and the results are benchmarked against other approaches reported in the literature. The results suggested that all three operating strategies of EGART models achieve similar (if not superior) performance as of other approaches.

The organization of this paper is as follows. Section 2 presents the background of this research. Section 3 presents the detailed algorithms of the EGART model as well as the three different operating strategies. The experimental studies and results comparison are given in Section 4. A summary of this paper and suggestions for further work are presented in Section 5.

II. BACKGROUND

Over the last two decades, neural networks and learning systems have been developed for solving pattern classification problems in various domains. In power engineering, some successful applications using neural network and learning systems have been reported. These include in power load or price forecasting [11], condition monitoring of circulating water systems in power generation plant [12], [13], and prediction of harmful gas emission in power generation plant [14]. In image processing and pattern recognition, neural network and learning systems have been successfully implemented in real world applications such as deep learning for face verification [15], multi-task learning for blind source separation [16], patch-based low rank tensor decomposition algorithm for hyperspectral image compression and reconstruction [17], unsupervised deep learning of stacked convolutional denoising auto-encoder for feature representation [18]. In high speed railway systems, successful application of neural network and learning systems have been reported. These include modelling and parallel control systems [19], descriptor estimator-based incipient fault estimation [20], and deep neural network for mechanical fault diagnosis [21].

In the literature, online or sequential learning is an appealing method to undertake the limitations associated with batch learning. In a dynamic environment, the ability of a learning system to operate in real time while being able to act and react autonomously is important. It has to adjust its own controlling parameters, and even its structure depending on the needs of the dynamic environment. A number of networks with such properties include the Resource Allocation Network (RAN) [22] and its extensions, e.g. Resource Allocation Network with Extended Kalman Filter (RANEKF) [23], Minimum Resource Allocation Network (MRAN) [10], Growing and Pruning RBF (GAPRBF) [24], and Generalized Growing, Pruning RBF (GGAPRBF) [25] and single and two hidden layer(s) Feedforward network [26]. In addition, the Online Sequential Extreme Learning Machine (OSELM) [27] uses the sequential least-squares method to minimize the error function. OSELM works with both additive-sigmoid and RBF hidden neurons. For the additive-sigmoid hidden neurons, the weights and biases for the input are generated randomly and they do not change while in the training phase; the weights for the output are determined analytically. Similarly, for OSELM with the RBF hidden neurons, the input weights, which are centers and widths of the RBF functions, are randomly generated, and the output weights are analytically determined.

The incremental non-iterative learning method in [28] constitutes an online and hyperparameter-free learning for the one-layer feedforward neural network without the concept of hidden layers. The principle is to use a square loss function to measure the error, and it has good competency in handling large scale real time learning problems [29]. Oscar *et al.* [30] presented an online learning algorithm for a two-layer feedforward neural network. The algorithm consists of a factor that puts the errors committed into numbers for each data sample. Such a method is effective for use both static and dynamic environments. A more recent work in Wang *et al.* [31] suggested an online reliability time series prediction method by combining a Convolutional Neural Network (CNN) and the Long Short Term Memory (LSTM). The proposed method was used to analyze the historical response time series and throughput time series of the service-oriented architecture services, and predict the reliability of the service system operation in the near future. Another interesting work on a new spiking neural network architecture also has the online learning capability [32]. It uses multiple learning strategies to learn by itself each input spike pattern. The learning algorithm is able to add a neuron, update the related network parameters, or delete a spike pattern. The algorithm information in the network from both local and global sources, and it performs better than batch-learning methods.

On the other hand, a family of online learning neural networks based on the Adaptive Resonance Theory (ART) has been developed for solving both clustering and classification problems [9], [33], [34]. While ART models are generally useful for solving classification tasks, there are some ART variants that are capable of solving both classification and regression problems [10], [35], [36], along with online learning capability. The following are the properties of *online learning* as derived from [34]. Similar online learning properties are also given in [7], [34].

Property–1: At any time of the training cycle, only the latest sample is needed for learning instead of all previous training samples.

Property–2: During the training cycle, a training sample is required to be presented only once for learning purposes. No re-iteration through the training set is necessary.

Property–3: Ability to perform incremental learning of new knowledge without corrupting the existing knowledge base.

Property–4: Ability to make a prediction if a new, unlabeled sample is arrived at any time during the training cycle.

Fuzzy ARTMAP (FAM) [35] is one of the most popular ART-based models. It has been pointed out that the performance of FAM is affected by the presentation sequence of training samples [35]. One of the solutions is to train several networks, each of which is given a different presentation sequence of training data, and then to combine the prediction results. However, such an approach demands an overly high computational load. In [10], an ordering algorithm that is based on the Max-Min clustering method to determine a fixed presentation sequence of training data to FAM has been proposed.

Later, GART [36] is developed that incorporates a modified Gaussian Adaptive Resonance Theory (GA) [34] which is a learning model based on a hybrid Gaussian classifier and

ART [33], [9], with the Generalized Regression Neural Network (GRNN) [37] in which it is a memory-based supervised learning neural network. GA learning aims to compress the new training samples into one of the existing categories that are based on PDFs and the Bayesian theorem. In the event where every existing category is not able to accommodate a new sample, a new category is created to encode the new sample. On the other hand, the GRNN learning process is conducted online. It only needs to perform a one-pass learning of the training data, and its learning process is instantaneous. Every hidden neuron in the GRNN function operates as a kernel. Upon receiving a new input sample, the kernels are used to calculate the probability density function (PDF). In the event that a new input sample is one that needs to be learned, the GRNN creates a new neuron to represent the new sample. This hybrid model is an improved version of GRNN, and at the same time preserves the online learning properties of ART. It is capable of conducting unsupervised clustering for the compression of the training data samples into a few hidden categories (or hidden node). And then decision function is constituted based on an enhanced Generalized Regression Neural Network (GRNN) for prediction. GART not only handles classification task, it manages to perform regression problem with the capability of online learning.

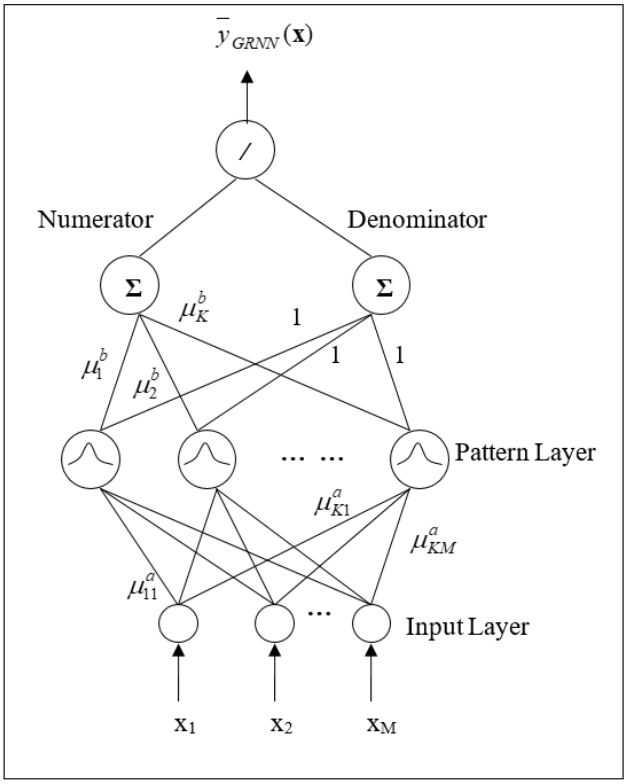

FIGURE 1. The GRNN structure.

Fig. 1 shows the GRNN structure. It consists of an input layer, a pattern layer, a summation layer, and an output layer. All the training samples are stored in the pattern layer. Fig. 2 shows the general GA structure. Category-j of GA uses M-dimensional vectors $\boldsymbol{\mu}_j$ and $\boldsymbol{\sigma}_j$ to represent its center (mean) and its standard deviation, respectively. A scalar n_j is used as its count of the number of training samples categorized into category-j. During the training phase, an M-dimension vector $\mathbf{A}_k$ (the k^{th}-training sample) is presented to GA for unsupervised learning.

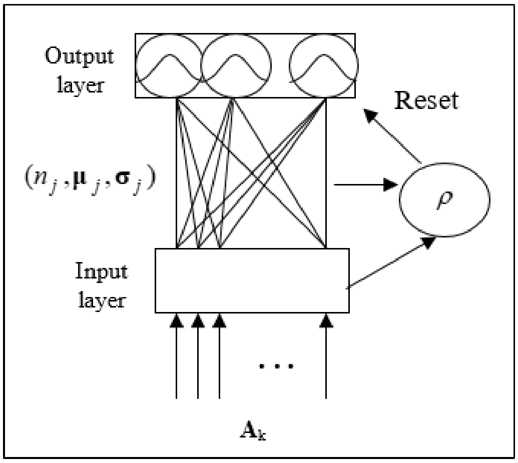

FIGURE 2. The GA structure.

In this paper, instead of using the ordering algorithm with ART-based models for solving *classification* problems, we propose a new Enhanced Generalized Adaptive Resonance Theory (EGART)-based model, based on a hybrid ART and the General Regression Neural Network (GRNN) [37], coupled a simplified ordering algorithm for tackling regression problems.

III. THE PROPOSED ENHANCED GENERALIZED ADAPTIVE RESONANCE THEORY (EGART) AND THE THREE OPERATING STRATEGIES

This section presents the detailed algorithms of the proposed EGART model. Based on the foundation of GART in [36], we further enhance it with four main improvements to become EGART, in order to increase the robustness of the resulting model in handling regression problems. Firstly, the Laplacian loss function is used in place of the ε-insensitive loss function. Secondly, instead of a general exponential kernel function, a Laplacian likelihood function (Equation 4) is instead adopted. Thirdly, a new definition of the vigilance function during the competition of ART is provided. Fourthly, after the ART competition, a match tracking mechanism is added.

There are three operating strategies of the proposed model, i.e., (A) Enhanced GART with online learning (hereafter denoted as EGART); (B) EGART with data preprocessing by ordering algorithm for the best order (hereafter denoted as Ordered-EGART); (C) Initial training by chuck of data samples using Ordered-EGART followed by handling of new data

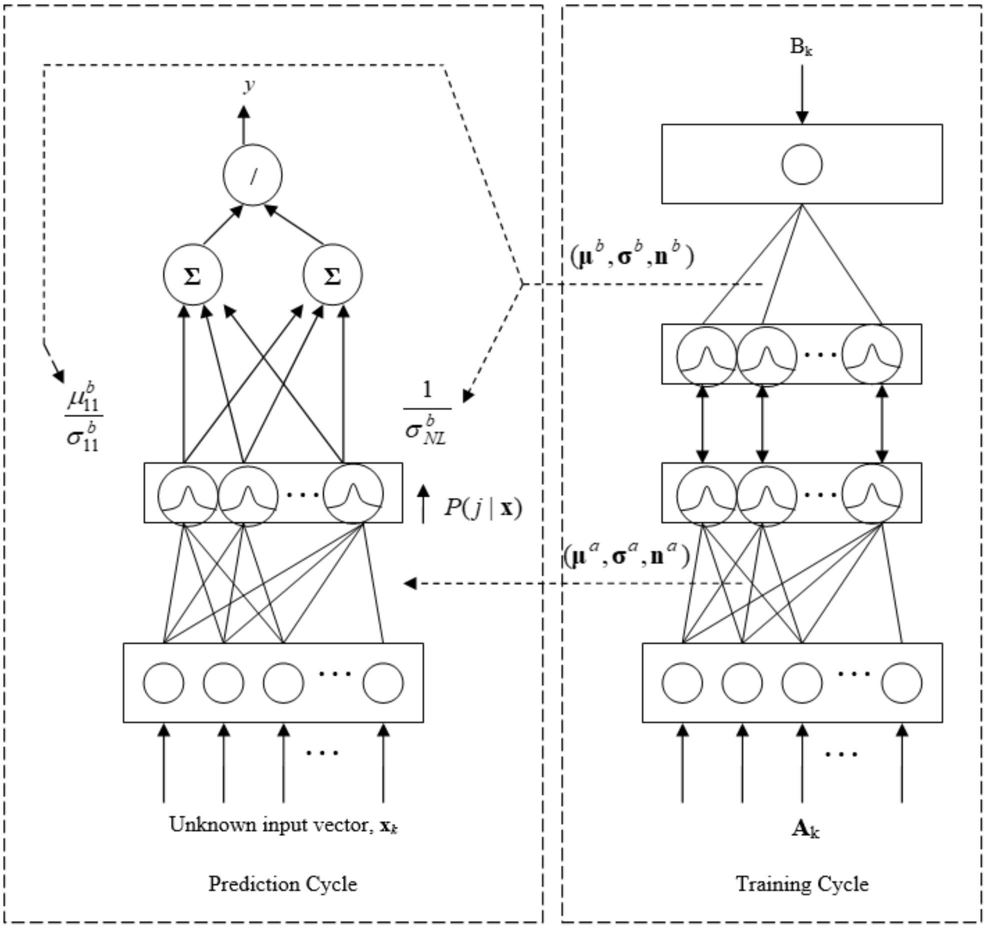

FIGURE 3. The Architecture of EGART neural network for training and prediction.

samples by online learning with EGART (hereafter denoted as IO-EGART).

A. EGART

Fig. 3 shows the EGART structure for training and prediction, while the core steps of the EGART learning algorithm are described, as follows.

Training samples – Let the training samples presented to ART-a and ART-b be $\{(\mathbf{A}_1, \mathrm{B}_1), (\mathbf{A}_2, \mathrm{B}_2), \ldots, (\mathbf{A}_k, \mathrm{B}_k)\}$, where $\mathbf{A}_k \in \mathbf{R}^{\mathbf{M}}$ and $\mathrm{B}_k \in \mathbf{R}^1$ are the input vector and kernel label of the k^{th} training sample, respectively. In the following discussion, the equations and variables are based on the ART-a module with input sample $\mathbf{A}_k$. The equations and variables of ART-b with input kernel label B_k are the same, but with subscript/superscript of "b" instead of "a".

Competition – The input sample is presented to ART-a, with its kernel label to ART-b, for computation of the *choice and match* functions. The choice and match functions are defined based on the Bayesian theorem. The Bayesian posterior probability for category-*j* of ART-a to input sample $\mathbf{A}_k$ is

$$P(j|\mathbf{A}_k) = \frac{P(\mathbf{A}_k|j)P(j)}{P(\mathbf{A}_k)} \tag{1}$$

The prior probability is

$$P(j) = \frac{n_j^a}{\sum_{i=1}^{N} n_i^a} \tag{2}$$

and

$$P(\mathbf{A}_k) = \sum_{i=1}^{N} P(\mathbf{A}_k|i)P(i) \tag{3}$$

where $P(\mathbf{A}_k|j)$ is a Laplacian likelihood function that is used to measure the degree of similarity between $\mathbf{A}_k$ and category-*j*, and is defined as

$$P(\mathbf{A}_k|j) = \frac{1}{2^M \prod_{i=1}^{M} \sigma_{ji}^a} \exp\left(-\sum_{i=1}^{M} \frac{1}{\sigma_{ji}^a} \left|\mu_{ji}^a - A_{ki}\right|\right) \tag{4}$$

where $\boldsymbol{\mu}_j^a$, $\boldsymbol{\sigma}_j^a$, and n_j^a are the center, standard deviation, count of category -*j* of ART-a, respectively. While GA [34] uses a standard quadratic loss function, EGART uses the Laplacian loss function. In the presence of noisy data, outliers have a high chance of presence in the quadratic loss function, which causes inaccurate recognition [38].

To find the "first round winner" of the competition, two measures are needed, i.e., the choice function of ART-a as defined in (1) and the match function, as follows

$$V(\mathbf{A}_k, j) = \left(2^M \prod\nolimits_{i=1}^{M} \sigma_{ji}^a\right) P(\mathbf{A}_k|j) \tag{5}$$

The first-round winner of ART-a, which is noted as J, is selected based on the choice function with the highest value. Its match function must be larger than or equal to the vigilance parameter, i.e.,

$$\begin{aligned} J = \ & \arg\max_j (P(j|\mathbf{A}_k)), \\ & \text{subject to } V(\mathbf{A}_k, J) \geq \rho_a \end{aligned} \tag{6}$$

where ρ_a is a pre-defined vigilance parameter between 0 and 1.

Match Tracking – The winner needs to be verified before it can be declared as the "final winner", which is carried out with the map-field vigilance test, i.e.,

$$V(B_k, J) \geq \rho_b \tag{7}$$

where ρ_b is a pre-defined vigilance parameter between 0 and 1. If the conditions set in (7) is fulfilled, category-J is declared as the final winner, and is ready for weight adaptation, i.e., learning. If not, a search based on (6) from the existing categories for a better candidate for the first-round winner is conducted, i.e., the match-tracking mechanism is triggered. During match tracking, the initial first-round winner, which has failed to fulfill (7), is deactivated temporarily by setting its choice function to be a negative value, i.e.,$P(j|\mathbf{A}_k) = -1$, and ρ_a is temporarily increased to $\rho_a = V(\mathbf{A}_k, J)$.

Learning – Learning involves the adjustment of the center, standard deviation, and counts of the final winner by using the following equations.

$$n_J^a \leftarrow n_J^a + 1 \tag{8a}$$

$$\boldsymbol{\mu}_J^a \leftarrow \left(1 - \frac{1}{n_J^a}\right) \boldsymbol{\mu}_J^a + \frac{\mathbf{A}_k}{n_J^a} \tag{8b}$$

$$\boldsymbol{\sigma}_J^a \leftarrow \left(1 - \frac{1}{n_J^a}\right) \boldsymbol{\sigma}_J^a + \frac{1}{n_J^a} \left|\boldsymbol{\mu}_J^a - \mathbf{A}_k\right| \tag{8c}$$

Addition of a new category– For the case where none of the existing categories that fulfills (6) is able to pass the map-field vigilance test (7), a new category is created to represent the new sample, i.e.,

$$N \leftarrow N + 1 \tag{9a}$$

$$\boldsymbol{\mu}_N^a = \mathbf{A}_k \tag{9b}$$

$$\boldsymbol{\sigma}_N^a = \gamma_a \tag{9c}$$

$$n_N^a = 1 \tag{9d}$$

where γ_a is a pre-defined initial standard deviation.

During the prediction cycle, a, new, unlabeled sample, $\boldsymbol{x}$, is firstly presented to ART-a. The prediction of EGART is obtained by a distributed posterior probability estimation based on the GRNN algorithm, as follows,

$$f(\mathbf{x}) = \frac{\sum_{j=1}^{N} \frac{\mu_j^b}{\sigma_j^b} P(j|\mathbf{x})}{\sum_{j=1}^{N} \frac{1}{\sigma_j^b} P(j|\mathbf{x})} \tag{10}$$

B. ORDERED-EGART

The applicability and performance of ART-based neural network has been reported [9], [33], [34]. As an online learning algorithm, the performance maybe disturbed by the sequence or the order of training data presentation [10]. For managing this situation, an Ordering Algorithm for FAM has been proposed [10]. Instead of random order in online learning, ordering algorithm a preprocessing method to compute the best order of training data presentation to the collected data samples. However, such algorithm required a set of many data samples to be collected and involved preprocessing. The ordering algorithm is a type of min-max clustering algorithm that is used to find the presentation order of training data for FAM [10]. This algorithm requires a pre-defined number of cluster centers, which poses an obstacle to satisfying the online learning properties. One way to solve this problem is to define this number to be equal to one larger than the number of classes [10]. Therefore, when FAM employs ordering algorithm, it will become offline learning.

Instead of solving classification task using FAM with ordering algorithm, this paper proposes a new EGART model with implementation of ordering algorithm -to solve data regression problem. At the same time, we also propose a new ordering strategy for regression, i.e., the ordering algorithm is simplified to only one cluster center and applied in data regression task since there is no class information involved in regression. For training of EGART with ordering algorithm (Ordered-EGART), as there is only one cluster center hence the original three-stage procedure of the ordering algorithm [10] is now simplified to only two stages, as follows:

Stage 1–Identify the cluster center (the first training sample in the sequence): For each of the M-dimension input vectors,$\mathbf{A}_k$, find the respective complement-coded [33] vector, $\mathbf{I}_k \in \mathbf{R}^{2M}$ i.e.,

$$\begin{aligned} \mathbf{I}_k &= (\mathbf{A}_k, \mathbf{A}_k^c) = (\mathbf{A}_k, 1 - \mathbf{A}_k) \\ &\equiv (A_{k,1}, .., A_{k,M}, 1 - A_{k,1}, \ldots, 1 - A_{k,M}) \\ &\equiv (A_{k,1}, \ldots, A_{k,M}, A_{k,M+1}, \ldots, A_{k,2M}) \end{aligned} \tag{11}$$

The k^{th} input vector that has the largest value according to Equation (12) is selected as the first sample in the presentation sequence.

$$K = \arg\max_k \left(\sum\nolimits_{i=1}^{M} \left|A_{k,M+i} - A_{k,i}\right|\right) \tag{12}$$

Stage 2–The presentation sequence of the remaining input vectors is determined based on the minimum Euclidean distance from the input vectors to the cluster center.

Once the order of presentation of data is decided by ordering algorithm, the ordered data is send to EGART for training. Under this scheme, it is referred to as Ordered-EGART as shown in Fig. 4.

C. IO-EGART

In most of the real world applications with nonstationary data samples, a set of training input and respective targeted output

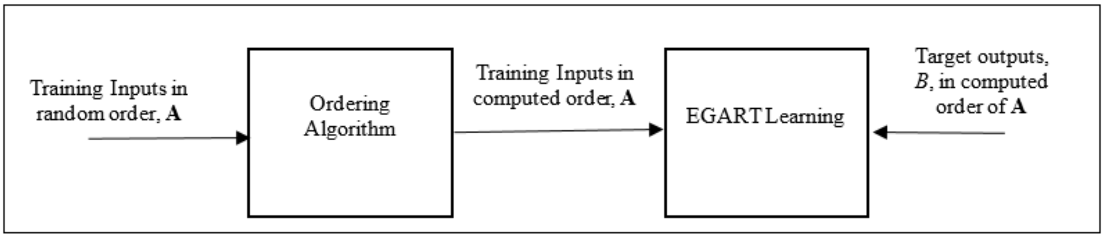
Training Inputs in random order, A
Ordering Algorithm
Training Inputs in computed order, A
EGART Learning
Target outputs, B, in computed order of A

FIGURE 4. Training of Ordered-EGART. The inputs of Ordered-GART are the training input and respective target output (A, *B*). The outputs are weights of trained Ordered-EGART.

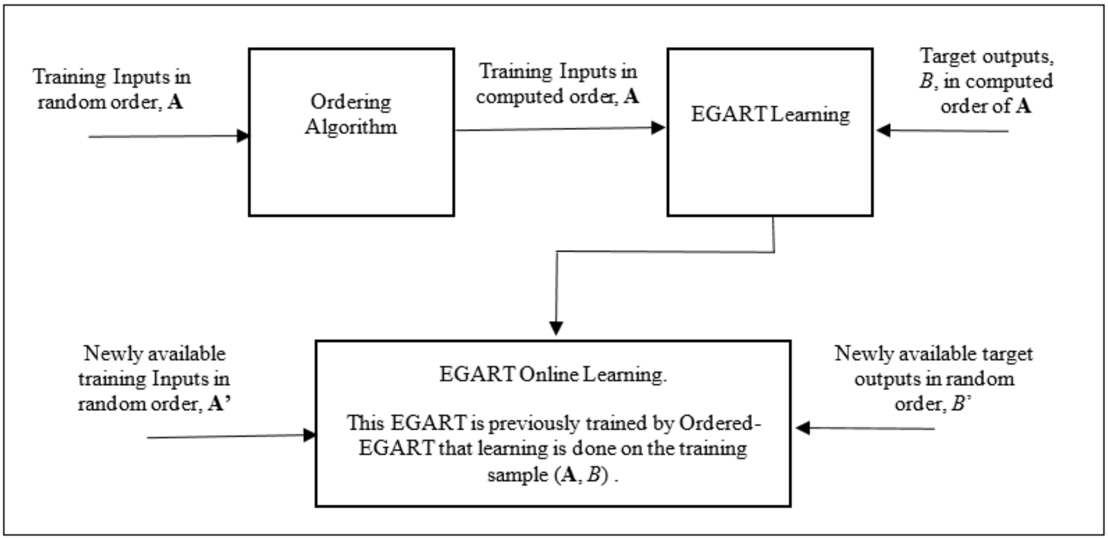
Training Inputs in random order, A
Ordering Algorithm
Training Inputs in computed order, A
EGART Learning
Target outputs, B, in computed order of A
Newly available training Inputs in random order, A'
EGART Online Learning.
This EGART is previously trained by Ordered-EGART that learning is done on the training sample (A, B) .
Newly available target outputs in random order, B'

FIGURE 5. Training of IO-EGART. The inputs are the collected training input and respective target output (A, *B*) for Ordered-EGART, and newly available inputs and respective target output (*A'*, *B'*) for EGART. The outputs are weights of trained IO-EGART.

(**A**, *B*) may be available before training of a EGART. In this scenario, ordering algorithm is applied to the set of available training data samples before the training of EGART (i.e., Ordered-EGART) for best performance. As this is a nonstationary application, after the training of Ordered-EGART is completed, more new training data (**A**', *B*') may be available since then. In order to ensure the trained EGART (by Ordered-EGART) can continue to learn new training data, the online learning algorithm can be applied to the trained EGART that has been trained by Ordered-EGART strategy. This operating strategy is thereby named as Initial-Ordered-EGART (IO-EGART) as shown in Fig. 5.

IV. EXPERIMENTS

Three versions of EGART are evaluated in this paper. Firstly, EGART is applied with no pre-collected training data. In other words, EGART is expected to learn from the very first training sample. Secondly, the simplified ordering algorithm is combined with EGART (Ordered-EGART). For this model, a collection of training samples is available, and the problem undertaken is a stationary one. Note that by using the simplified ordering algorithm to determine the presentation sequence of training data, Ordered-EGART violates *Property–1* of online learning. The advantage is that Ordered-EGART is expected to exhibit a lower error rate and a smaller network size in line with the theory advocated by ordering algorithm [26] as compared with those of EGART. Another advantage ensues by using the simplified ordering algorithm is that the training of the Ordered-EGART will become more straightforward, thus resulting in shorter training time. Thirdly, the concept of an "initial training" for EGART with the simplified ordering algorithm is proposed (IO-EGART). For this model, it is assumed that some data samples are available for training, and the simplified ordering algorithm is used to determine the presentation sequence of these data samples. After the initial training, the trained IO-EGART model is ready to perform online learning. This means that the simplified ordering algorithm is not used to process subsequent new data samples that are available after the initial training. Note that Ordered-EGART can be extended to be IO-EGART if the new data samples are presented one-by-one after the initial training without further ordered.

In this paper, all three EGART-based models (i.e., EGART, Ordered-EGART, IO-EGART) are evaluated using five regression data sets from various application domains. The Ordered-EGART is expected to produce the best prediction accuracy rates with the simplest network structures; but without the properties of online learning. On the other hand, EGART is a fully online learning model, but may perform with lower prediction accuracy rates with more complex network structures. Finally, IO-EGART is expected to be a compromise between EGART and Ordered-EGART, which preserves the online learning properties.

Seven regression data sets are used for evaluation, i.e., SinE (artificial mapping data), Delta Ailerons (aircraft control), Boston Housing (economy), Sante-Fe Series-E (Astrophysical data), Abalone (Physical measurement), Thermal Interface Location (fire safety engineering) and Fire Evacuation Time (Building evacuation exercise). The same training data samples are used for training the three EGART-based models, the difference lies in the training operating strategies, in which EGART will handle all training samples online, Ordered-EGART will consider all training samples offline with ordering algorithm in place, and IO-EGART will take half of the training samples for preprocessing with ordering algorithm while the remaining half of the training samples will be presented and learned one by one online. Besides that, the test procedure is also based on the same set of test data samples. The Root Mean Square Error (RMSE) is used as the performance indicator. The training and test cycles are repeated 50 times. The average number of categories (nodes) created over 50 runs is reported. The final error rate is obtained by using the bootstrap method [30], [31], i.e., the bootstrap mean with 1,000 re-samplings.

Bootstrapping is a technique that roughly calculate the change of a certain criterion statistically in certain situations whereby the concealed sampling distribution of the criterion is not known and/or rather demanding to estimate. It is handy for computing the statistics of population parameters in situations with small or limited data samples [30]. The principles of bootstrapping for the calculation of the mean of a set of data samples is as following steps:

1. A data set is first acquired. Assuming that data set $\mathbf{X} = x_1, \ldots, x_n$ is acquired, and n is the sample size that is examined from a totally undefined distribution of probability F, ψ is the average of all the values in the data set $\mathbf{X}$ and is the total number of times bootstrapping is repeated.
2. Select a random sample of n data points with replacements from the data set $\mathbf{X}$. This new set of data, $\mathbf{X^*}$, is the bootstrap sample.
3. The mean of the bootstrap sample $\mathbf{X^*}$,ψ_1^* is calculated. Re-sample the data by repeating steps 2 to 3 N times to get the bootstrap estimates of $\psi_1^*, \psi_2^*, \psi_3^* \ldots, \psi_N^*$.

TABLE 1. RMSE and number of categories (in parentheses) for the sine problem.

ρ_b	EGART	IO-EGART	Ordered-EGART
0.95	0.0119 (760.40)	0.0079 (733.22)	0.0068 (656.46)
0.9	0.0176 (593.62)	0.0105 (563.76)	0.0083 (466.58)
0.8	0.0261 (458.06)	0.0156 (432.32)	0.0120 (319.98)
0.7	0.0347 (381.14)	0.0246 (371.54)	0.0158 (250.28)
0.6	0.0473 (345.40)	0.0248 (310.40)	0.0196 (204.22)
0.5	0.0637 (352.98)	0.0294 (264.80)	0.0242 (169.98)

TABLE 2. Performance comparison of the sine problem [24], [36].

Algorithms	Testing RMSE	
	Mean	Stdev
Ordered-EGART	0.0068	0.0005
IO-EGART	0.0079	0.0005
EGART	0.0119	0.0035
GAPRBF [24]	0.0269	0.0087
RANEKF [24]	0.0275	0.0124
MRAN [24]	0.0496	0.0212
RAN [24]	0.0673	0.0107
SVR [24]	0.2629	0.0056
ELM	0.2560	0.0045
OSELM [36]	0.2754	0.0051

For all experiments, parameters ρ_a and γ_b were set to their default values, i.e., 0 and 1, respectively. Parameter γ_a was obtained after several trials, and parameter ρ_b was varied from 0.5 to 0.95 (while other parameters fixed) in order to evaluate the effectiveness (network size and error rate) of the three EGART-based models.

A. SINE

In this experiment, the three EGART-based models were used to estimate a fast changing continuous function known as "SinE" [24], [36], i.e.,

$$y = 0.8\exp(-0.2x)\sin(10x) \tag{13}$$

A total of 50 runs were carried out in total, each of which contained 3,000 training samples. The 3,000 training samples were composed of randomly generated x values between 0 and 10, with their respective y values. The test samples were produced by utilizing similar procedures, and a total of 1,500 samples were created.

Table 1 shows the results of three EGART-based models for various values of ρ_b. Comparing with EGART, IO-EGART and Ordered-EGART created smaller network sizes (from 2.52% to 24.98%, and 13.67% to 51.84%, for various values of ρ_b) and smaller error rates (from 29.11% to 53.85%, and 42.86% to 62.01% for various values of ρ_b). Table 2 shows the best results and standard deviations of the three

TABLE 3. RMSE and number of categories (in parentheses) for the delta ailerons problem.

ρ_b	RMSE and number of categories for EGART	RMSE and number of categories for IO-EGART	RMSE and number of categories for Ordered-EGART
0.95	0.0417 (728.06)	0.0413 (714.36)	0.0411 (650.76)
0.9	0.0414 (533.14)	0.0410 (517.24)	0.0408 (460.08)
0.8	0.0416 (334.24)	0.0412 (319.02)	0.0408 (265.22)
0.7	0.0419 (252.84)	0.0414 (235.60)	0.0412 (182.18)
0.6	0.0423 (203.86)	0.0417 (185.14)	0.0414 (135.22)
0.5	0.0425 (168.18)	0.0421 (144.84)	0.0418 (103.16)

EGART-based models, as well as the results of other methods in [24], [36]. All three EGART-based models achieved smaller error rates as compared with those in [24], [36]. Their error rates reduced by 55.76 % (EGART), 70.63% (IO-EGART), and 74.72% (Ordered-EGART) than that of GAPRBF (the best in [24], [36]).

B. DELTA AILERONS

This set of data was concerned with the control of ailerons on an F16 aircraft [39]. A total of 7,192 samples were available. Each pair of data sample consisted of five continuous variables (roll rate, pitch rate, current pitch, current roll, and difference of roll rate) and its respective label (control action).

According to [26], [39]–[41], 3,000 of the 7,192 samples were used as the training set, and the remaining as the test set. The input and output variables were set between 0 and 1, while the RMSE was derived based on the same range.

Table 3 shows the results (RMSE and number of categories) of the three EGART-based models for various ρ_b. Again, IO-EGART and Ordered-EGART established smaller network sizes (from 1.88% to 12.88% and 10.62% to 38.66 % for various ρ_b) and produced slightly lower error rates than those of EGART. Table 4 shows the best results and standard deviations of three EGART-based models. All three EGART models achieved lower error rates as compared with those reported in [26], [39]–[41].

C. BOSTON HOUSING

This data set was concerned with predicting the housing values in Boston. There were 506 samples. Each sample consisted of 13 input attributes (12 of which are continuous and one binary attribute), and one continuous target output. According to [24], 481 of the samples were selected randomly as the training set, and the remaining as the test set. Again, all input and output variables were normalised between 0 and 1, while the RMSE was derived based on the same range.

TABLE 4. Performance comparison of the delta ailerons problem [26], [40]–[41].

Algorithms	Testing RMSE	
	Mean	Stdev
Ordered-EGART	0.0408	0.0009
IO-EGART	0.0410	0.0008
EGART	0.0414	0.0008
SVR [40]	0.0429	0.0010
BP [40]	0.0481	0.0015
SLFN [26]	0.0472	0.0049
EI-ELM (Sigmoid) [41]	0.0445	0.0062
EI-ELM (RBF) [41]	0.0448	0.0065
ELM [40]	0.0431	0.0035
OSELM	0.0440	0.0054

TABLE 5. RMSE and number of categories (in parentheses) for the boston housing problem.

ρ_b	RMSE and number of categories for EGART	RMSE and number of categories for IO-EGART	RMSE and number of categories for Ordered-EGART
0.95	0.1095 (269.54)	0.1090 (264.54)	0.0993 (250.52)
0.9	0.1127 (204.14)	0.1079 (199.04)	0.1083 (186.86)
0.8	0.1129 (148.48)	0.1111 (146.16)	0.1062 (136.20)
0.7	0.1183 (123.50)	0.1145 (118.26)	0.1122 (108.18)
0.6	0.1236 (107.94)	0.1177 (103.10)	0.1164 (96.60)
0.5	0.1291 (95.58)	0.1239 (92.02)	0.1208 (81.10)

Table 5 below depicts the results (RMSE and number of categories) of three EGART-based models for various ρ_b. Comparing with EGART, the best improvements of Ordered-EGART are 15.15% (network size) and 9.32% (prediction error rate). Table 6 shows the best results and standard deviations of three EGART-based models, and a comparison with those of other methods in [24]. All three EGART-based models achieved significantly better rates of errors, for example, the rate of errors for Ordered-EGART was 26.77% and 71.07% lower as compared with those of MRAN and RAN [24], respectively.

D. SANTA-FE SERIES-E

This problem was concerned with the Sante-Fe Time Series Competition, specifically the Series-E [43]. The astrophysical data (univariate time series data with only one variable) was noisy, discontinuous, and nonlinear. In accordance

TABLE 6. Performance comparison of the boston housing problem.

Algorithms	Testing RMSE	
	Mean	Stdev
Ordered-EGART	0.0993	0.0370
IO-EGART	0.1079	0.0385
EGART	0.1095	0.0371
MRAN [24]	0.1356	0.0411
GAPRBF [24]	0.1418	0.0466
RANEKF [24]	0.1437	0.0464
RAN [24]	0.3432	0.0770
ELM	0.1103	0.0114
OSELM	0.1132	0.0136

TABLE 7. RMSE and number of categories (in parentheses) for the Santa-Fe Series-E problem.

ρ_b	RMSE and number of categories for EGART	RMSE and number of categories for IO-EGART	RMSE and number of categories for Ordered-EGART
0.95	0.0567 (545.70)	0.0564 (525.52)	0.0562 (472.05)
0.9	0.0565 (264.62)	0.0564 (233.14)	0.0561 (199.25)
0.8	0.0564 (122.78)	0.0565 (117.80)	0.0566 (97.50)
0.7	0.0565 (80.08)	0.0567 (77.40)	0.0564 (61.15)
0.6	0.0566 (56.42)	0.05670 (55.60)	0.0574 (45.56)
0.5	0.0568 (39.76)	0.0572 (40.56)	0.0575 (33.89)

with [42], [44], a total of 2048 of the samples were taken into account, each with five inputs and one output, i.e., $x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5})$, with x_t being the intensity of the star at time, t. The first 90% of the data samples were used as the training set, and the remaining were used as the test set.

Table 7 shows the results (MSE and number of categories) of three EGART-based models for various ρ_b. Comparing with EGART, the best improvements of Ordered-EGART are 24.80 % (network size) and 1.86% (prediction error rate). Table 8 shows the best results of three EGART-based models, and a comparison with those of other methods in [42], [44]. The error rates of three EGART-based models are slightly better than that of GRNNFA [34], but better than those reported in [44].

E. ABALONE

The abalone data set [26] consists of 4177 samples for predicting the age of abalone based on physical measurements. Each samples comprises eight continuous input attributes (physical measurements) and one integer output (age of abalone). In each run, 2784 training samples and 1393 testing samples were randomly selected from the data set. Table 9 shows the results (RMSE and number of categories) of three EGART-based models for various values of ρ_b. For this data set, comparing with EGART, the best improvements of Ordered-EGART have been improved by 20.35 % (network size) and 2.60% (prediction error rate).

TABLE 8. Performance comparison of the Santa-Fe Series-E problem.

Algorithms	Testing RMSE
Ordered-EGART	0.0562
IO-EGART	0.0564
EGART	0.0564
GRNNFA [42]	0.0571
PMRS [42]	0.1225
NN [42]	0.0574
ES [42]	0.0883
ELM	0.0581
OSELM	0.0595

TABLE 9. RMSE and number of categories (in parentheses) for the abalone problem.

ρ_b	RMSE and number of categories for EGART	RMSE and number of categories for IO-EGART	RMSE and number of categories for Ordered-EGART
0.95	0.0816 (1170.54)	0.0796 (1134.96)	0.0782 (1071.10)
0.9	0.0788 (823.48)	0.0772 (797.14)	0.0769 (713.38)
0.8	0.0762 (505.22)	0.0755 (480.78)	0.0748 (402.40)
0.7	0.0775 (379.50)	0.0769 (351.46)	0.0758 (279.02)
0.6	0.0776 (297.08)	0.0773 (268.16)	0.0761 (200.70)
0.5	0.0779 (238.84)	0.0774 (210.64)	0.0767 (147.60)

Table 10 shows the best results of three EGART-based models, and a comparison with Support Vector Regression (SVR), Least Square Support Vector Regression (LSSVR), Extreme Learning Machine (ELM) and Online Sequential Extreme Learning Machine (OSELM). The best model with the smallest error is LSSVM, then all three EGART-based model has smaller errors than SVM, ELM and OSELM.

F. THERMAL INTERFACE HEIGHT

The dividing line between cold air and hot gas layers of a compartment fire is called the thermal interface. This height of the interface is affected mainly by the mass of the air that is entraining into the plume of fire. A set of full scale

TABLE 10. Performance comparison of abalone problem.

Algorithms	Testing RMSE
Ordered-EGART	0.0748
IO-EGART	0.0755
EGART	0.0762
SVM [45]	0.0764
LSSVM [45]	0.0746
ELM[45]	0.0761
OSELM[45]	0.0777

steady-state experiments of flow induced in a single compartment fire was reported in [46], and 55 samples were collected. Each sample consisted of six attributes, i.e., size (height and width) of the sill of the opening, location of the fire bed, strength of the fire source, and ambient temperature, as well as its respective label (i.e. thermal interface height).

In general, a thermal interface's height can be approximated based on the profile of the room temperature along the room height. The thermal interface's height is defined as the height of the profile with the largest change of room temperature. However, due to the effects of mixing and diffusion of fluids, the height of the thermal interface cannot be stably measured. It is possible for the thermal interface heights to be bound between an error envelope of ±8% to ±50% from the estimated height [46]. Therefore, for each sample, an error value was included (in addition to the label) to form the envelope of thermal interface height.

Since the set of data contained only 55 samples, the leave-one-out (LOO) method was utilized. In each LOO run, 54 of the 55 samples were used as the training set and the remaining one was used as the testing set. Note that other than RMSE and number of categories, the prediction accuracy (sum of predictions within the acceptable envelope range) was included.

Table 11 shows the results (RMSE, accuracy rate, and number of categories) of the three EGART-based models for various values of ρ_b. Fig. 4 shows the prediction of Ordered-EGART as compared with the envelope range of thermal interface heights.

Table 12 shows the best results of EGART and Ordered-EGART as compared with those of [47], Extreme Learning Machine (ELM) [40] and Online Sequential Extreme Learning Machine [27]. The ELM is an offline learning algorithm that similar to Ordered-EGART. On one hand, OSELM is an extension of ELM that requires some data samples for initial training before online learning which can be deemed having the similar learning strategy as of IO-EGART. The performances of the models (based on the prediction within the range envelope and prediction fallen out of range envelope) are recorded in terms of accuracy. The results appear consistent, i.e., 94.55% for offline algorithms that encompass Ordered-EGART, ELM and GRNNFA, while 89.09% when they are trained online that includes EGART, IO-EGART, and OSELM) The result is rather close because the size of data samples of this dataset is rather small and Leave One Out (LOO) strategy is used.

TABLE 11. The record of RMSE, percentage of accuracy, and number of categories for the thermal height problem.

ρ_b	**EGART**	**IO-EGART**	**Ordered-EGART**
	RMSE	RMSE	RMSE
	Accuracy(%)	Accuracy (%)	Accuracy (%)
	Categories	Categories	Categories
0.95	0.0696	0.0729	0.0646
	89.09	89.09	92.73
	38.06	37.39	35.55
0.9	0.0827	0.0764	0.0700
	87.27	89.09	94.55
	28.62	28.25	26.73
0.8	0.0916	0.0871	0.0904
	83.64	89.09	87.27
	23.55	22.61	18.89
0.7	0.1077	0.0969	0.1040
	80.00	87.27	83.64
	20.10	18.83	17.53
0.6	0.1254	0.1029	0.1046
	74.55	81.82	83.64
	17.20	15.63	13.73
0.5	0.1292	0.1176	0.1018
	72.73	80.00	78.18
	14.72	13.47	8.33

TABLE 12. The performance comparison of the thermal interface height problem.

Algorithms	Accuracy (%)
Ordered-EGART	94.55
IO-EGART	89.09
EGART	89.09
ELM	94.55
OSELM	89.09
GRNNFA [47]	94.55

Although Ordered-EGART and GRNNFA achieved the same accuracy rate (94.55%), Ordered-EGART used only one LOO training cycle, since the simplified ordering algorithm was used to ascertain the presentation sequence of training samples. However, GRNNFA applied 20 LOO cycles [37] with different presentation sequences of training samples before obtaining the final average results, as in Table 10. Therefore, Ordered-EGART has the advantage of simplicity and of a faster processing time (with only one LOO cycle).

G. FIRE EVACUATION TIME

In this experiment, the EGART models were applied to building evacuation. The study focused on predicting the time taken for people to evacuate in the event of fire emergency in a typical karaoke center in Hong Kong. The evacuation time

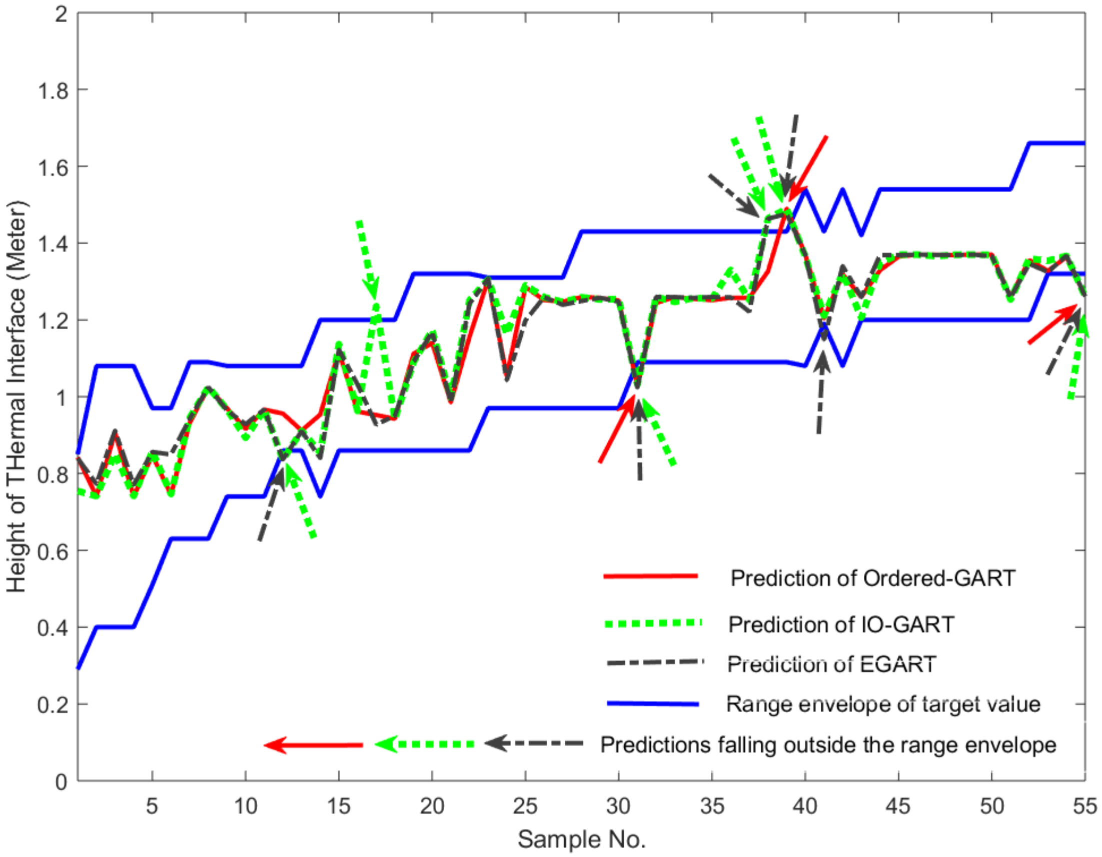

FIGURE 6. Prediction of Ordered-EGART, IO-EGART and EGART for the thermal interface height problem (sorted ascending). For results of Ordered-EGART, out of 55 samples, 52 (or 94.55%) have been successfully predicted to be within the envelope.

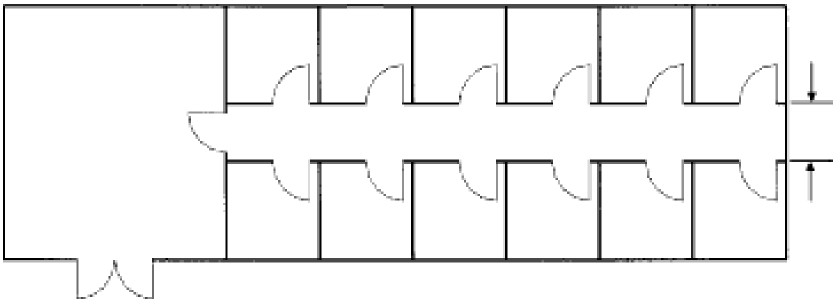

FIGURE 7. Typical layout of a Karaoke center in Hong Kong.

is distinctly linked to normal human behaviors like the route choice and human behavioral responses to the fire alarm. Fig.7 depicts a common floor plan or layout of a typical karaoke center in Hong Kong.

In a standard karaoke center in Hong Kong, a straight corridor to which the rooms are connected is directly connected to a lobby area with a clear exit. It is intriguing to perform investigations on the relationship between the variables of the layout of the karaoke center, and the evacuation time. The evacuation time is defined as the time taken for everyone to evacuate the center starting from the time the fire alarm sounds.

For this case study, there were three layout parameters or variables, i.e., the number of rooms (ranging from 4 to 22 rooms), width of the corridor (ranging from 1.05 meters to 1.4 meters), and the area of the lobby (ranging from 15 square-meters to 35 square-meters). The variables were varied in order to examine their effects on the evacuation time. This evacuation time was determined by using the Spatial Grid Evacuation Model [48]. There were a total of 750 data

TABLE 13. RMSE and number of categories (in parentheses) for the fire evacuation time problem.

ρ_b	RMSE and number of categories for EGART	RMSE and number of categories for IO-EGART	RMSE and number of categories for Ordered-EGART
0.95	5.3306 (218.78)	5.2666 (210.66)	5.1564 (178.74)
0.9	7.0942 (169.78)	6.5958 (155.30)	5.2935 (119.34)
0.8	9.6600 (139.32)	9.0405 (122.82)	6.4261 (77.54)
0.7	12.0313 (124.30)	11.0514 (114.16)	8.1896 (67.98)
0.6	14.9028 (112.78)	13.0484 (104.56)	9.5011 (61.14)
0.5	17.9250 (101.78)	16.2891 (93.36)	11.1001 (53.54)

TABLE 14. The performance comparison for the fire evacuation time problem.

Algorithms	Testing RMSE
Ordered-EGART	5.1564
IO-EGART	5.2666
EGART	5.3306
ELM	5.2173
OSELM	5.5028
GRNN	8.7173

samples for experimentation, with 550 training samples and 200 test samples.

Table 13 shows the results (RMSE and number of categories) of three EGART-based models for various values of ρ_b. Comparing with EGART, the best improvements of Ordered-EGART are 18.30 % (network size) and 3.27% (prediction error rate). Table 14 depicts the best results of three EGART-based models, and a comparison with ELM, OSELM, and GRNN. The Ordered-EGART achieved smallest error rates as compared with other models.

V. SUMMARY

In this paper, an enhanced GART neural network, known as EGART, with a simplified ordering algorithm for determining the presentation sequence of training data has been proposed. Three operating strategies of EGART, i.e. fully online (EGART), semi-online (IO-EGART), and off-line (Ordered EGART) operations, have been suggested. The effectiveness of the three EGART-based models in tackling regression problems have been demonstrated using four benchmark problems (SinE, Delta Ailerons, Boston Housing, and Sante-Fe Series-E). In additional, the three EGART-based models have been applied to fire safety engineering, i.e., to prediction of the thermal interface height in single compartment fire and fire evacuation times. All the experimental results have been quantified by the bootstrap technique. The results positively demonstrate that EGART, IO-EGART, and Ordered-EGART perform better than other machine learning methods in tackling regression problems.

Further research is focused on improving the EGART-based models. A pruning algorithm for the EGART-based models is needed to reduce the network complexity in the case when too many redundant categories are generated during the training cycle. On the other hand, Ordered-EGART can be used for designing an ensemble of neural networks for handling regression problems. The training samples can be divided into several blocks, and the simplified ordering algorithm is applied before presenting the samples to Ordered-EGART for learning. A cooperative multiple Ordered-EGART regression model can also be investigated to provide better solutions when a single estimator fails to solve the underlying problem.

REFERENCES

[1] Y. T. Wu, M. Yuan, S. Dong, L. Li, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, Jan. 2018.

[2] X. Pan, X. Pang, and H. Wang, "A safe screening based framework for support vector regression," *Neurocomputing*, vol. 287, pp. 163–172, Apr. 2018.

[3] K. Zhang, Y. Chen, J. Zheng, J. Huang, and X. Tang, "Adaptive filling modeling of butt joints using genetic algorithm and neural network for laser welding with filler wire," *J. Manuf. Process.*, vol. 30, pp. 553–561, Dec. 2017.

[4] C. M. Bishop, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford, 1995.

[5] D. E. Rumelhart, G. E. Hinton, R. J. Williams, D. E. Rumelhart, J. L. McClelland, and The PDP Research Group, Eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, U.K.: MIT Press, 1986, pp. 318–362.

[6] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Netw.*, vol. 2, no. 2, pp. 302–309, Mar. 1991.

[7] Y. Yingwei, N. Sundararajan, and P. Saratchandran, "Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm," *IEEE Trans. Neural Netw.*, vol. 9, no. 2, pp. 308–318, Mar. 1998.

[8] G.-M. Park, J.-W. Choi, and J.-H. Kim, "Developmental resonance network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1278–1284, Apr. 2019.

[9] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 698–713, Sep. 1992.

[10] I. Dagher, M. Georgiopoulos, G. L. Heileman, and G. Bebis, "An ordering algorithm for pattern presentation in fuzzy ARTMAP that tends to improve generalization performance," *IEEE Trans. Neural Netw.*, vol. 10, no. 4, pp. 768–778, Jul. 1999.

[11] J. Nagi, K. S. Yap, F. Nagi, S. K. Tiong, and S. K. Ahmed, "A computational intelligence scheme for the prediction of the daily peak load," *Appl. Soft Comput.*, vol. 11, no. 8, pp. 4773–4788, 2011.

[12] S. Y. Wong, K. S. Yap, H. J. Yap, S. C. Tan, and S. W. Chang, "On equivalence of FIS and ELM for interpretable rule-based knowledge representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1417–1430, Jul. 2015.

[13] S. Y. Wong, K. S. Yap, H. J. Yap, and S. C. Tan, "A truly online learning algorithm using hybrid fuzzy ARTMAP and online extreme learning machine for pattern classification," *Neural Process. Lett.*, vol. 42, no. 3, pp. 585–602, 2015.

[14] S. Y. Wong, K. S. Yap, and H. J. Yap, "A constrained optimization based extreme learning machine for noisy data regression," *Neurocomputing*, vol. 171, pp. 1431–1443, Jan. 2016.

[15] S. Y. Wong, K. S. Yap, Q. Zhai, and X. Li, "Realization of a hybrid locally connected extreme learning machine with DeepID for face verification," *IEEE Access*, vol. 7, pp. 70447–70460, 2019. doi: 10.1109/ACCESS.2019.2919806.

[16] B. Du, S. Wang, C. Xu, N. Wang, L. Zhang, and D. Tao, "Multi-task learning for blind source separation," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4219–4231, Sep. 2018.

[17] B. Du, M. Zhang, L. Zhang, R. Hu, and D. Tao, "PLTD: Patch-based low-rank tensor decomposition for hyperspectral images," *IEEE Trans. Multimedia*, vol. 19, no. 1, pp. 67–79, Jan. 2017.

[18] B. Du, W. Xiong, J. Wu, L. Zhang, L. Zhang, and D. Tao, "Stacked convolutional denoising auto-encoders for feature representation," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1017–1027, Apr. 2017.

[19] B. Ning, T. Tang, H. Dong, D. Wen, D. Liu, S. Gao, and J. Wang, "An introduction to parallel control and management for high-speed railway systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1473–1483, Dec. 2011.

[20] Y. Wu, B. Jiang, and N. Lu, "A descriptor system approach for estimation of incipient faults with application to high-speed railway traction devices," *IEEE Trans. Syst., Man, Cybern. Syst.*, to be published. doi: 10.1109/TSMC.2017.2757264.

[21] H. Hu, B. Tang, X. Gong, W. Wei, and H. Wang, "Intelligent fault diagnosis of the high-speed train with big data based on deep neural networks," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2106–2116, Aug. 2017.

[22] J. Platt, "A resource-allocating network for function interpolation," *Neural Comput.*, vol. 3, pp. 213–225, Jun. 1991.

[23] V. Kadirkamanathan and M. Niranjan, "A function estimation approach to sequential learning with neural networks," *Neural Comput.*, vol. 5, no. 6, pp. 954–975, Nov. 1993.

[24] G.-B. Huang, P. Saratchandran, and N. Sundararajan, "An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 6, pp. 2284–2292, Dec. 2004.

[25] G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 57–67, Jan. 2005.

[26] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2006.

[27] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1411–1423, Nov. 2006.

[28] G.-B. Huang and C.-K. Siew, "Real-time learning capability of neural networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 863–878, Jul. 2006.

[29] O. Fontenla-Romero, B. Pérez-Sánchez, and B. Guijarro-Berdiñas, "An incremental non-iterative learning method for one-layer feedforward neural networks," *Appl. Soft Comput.*, vol. 70, pp. 951–958, Sep. 2018.

[30] B. Pérez-Sánchez, O. Fontenla-Romero, B. Guijarro-Berdiñas, and D. Martínez-Rego, "An online learning algorithm for adaptable topologies of neural networks," *Expert Syst. Appl.*, vol. 40, no. 18, pp. 7294–7304, Dec. 2013.

[31] H. Wang, Z. Yang, Q. Yu, T. Hong, and X. Lin, "Online reliability time series prediction via convolutional neural network and long short term memory for service-oriented systems," *Knowl.-Based Syst.*, vol. 159, pp. 132–147, Nov. 2018.

[32] S. Dora, S. Suresh, H. Hongui, and N. Sundararajan, "Online meta-neuron based learning algorithm for a spiking neural classifier," *Inf. Sci.*, vol. 414, pp. 19–32, Nov. 2017.

[33] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Netw.*, vol. 4, no. 6, pp. 759–771, 1991.

[34] J. R. Williamson, "Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps," *Neural Netw.*, vol. 9, no. 5, pp. 881–897, Jul. 1996.

[35] R. Andonie and L. Sasu, "Fuzzy ARTMAP with input relevances," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 929–941, Jul. 2006.

[36] K. S. Yap, C. P. Lim, and I. Z. Abidin, "A hybrid ART-GRNN online learning neural network with Aε-insensitive loss function," *IEEE Trans. Neural Netw.*, vol. 19, no. 9, pp. 1641–1646, Sep. 2008.

[37] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Netw.*, vol. 2, no. 6, pp. 568–576, Nov. 1997.

[38] W. Chu, S. S. Keerthi, and C. J. Ong, "Bayesian support vector regression using a unified loss function," *IEEE Trans. Neural Netw.*, vol. 15, no. 1, pp. 29–44, Jan. 2004.

[39] D. Malerba, F. Esposito, M. Ceci, and A. Appice, "Top-down induction of model trees with regression and splitting nodes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 612–625, May 2004.

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

[41] G.-B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, nos. 16–18, pp. 3460–3468, Oct. 2008.

[42] E. W. M. Lee, C. P. Lim, R. K. K. Yuen, and S. M. Lo, "A hybrid neural network model for noisy data regression," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 2, pp. 951–960, Apr. 2004.

[43] A. S. Weigend and N. A. Gersehnfield, *Time Series Prediction: Forecasting The Future And Understanding The Past*. Reading, MA, USA: Addison-Wesley, 1994.

[44] S. Singh, "Noise impact on time-series forecasting using an intelligent pattern matching technique," *Pattern Recognit.*, vol. 32, pp. 1389–1398, Aug. 1999.

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

[46] K. D. Steckler, J. G. Quintiere, and W. J. Rinkinen, *NBSIR 82-2520: Flow Induced by Fire in A Compartment*. Washington, DC, USA: National Bureau of Standards, 1982.

[47] E. W. M. Lee, R. K. K. Yuen, S. M. Lo, K. C. Lam, and G. H. Yeoh, "A novel artificial neural network fire model for prediction of thermal interface location in single compartment fire," *Fire Saf. J.*, vol. 39, no. 1, pp. 67–87, Feb. 2004.

[48] S. M. Lo and Z. Fang, "A spatial-grid evacuation model for buildings," *J. Fire Sci.*, vol. 18, no. 5, pp. 376–395, 2000.

SHEN YUONG WONG received the bachelor's (hons.) and master's degrees (hons.) in electrical and electronic engineering, and the Ph.D. degree in engineering from Universiti Tenaga National (National Energy University), Malaysia, in 2010, 2012, and 2015, respectively. She is currently an Assistant Professor with the Department of Electrical and Electronics Engineering, Xiamen University Malaysia. She is also a Professional Engineer registered to the Board of Engineers Malaysia, accredited under the Washington Accord. Her research interests include computer vision and image processing, extreme learning machine, deep learning, classification, regression, and other applications of artificial intelligence. She serves as an Editorial Board Member for *Applied Soft Computing* Journal.

KEEM SIAH YAP received the B.Eng. degree (Hons.) in electrical and the M.Sc. degree in electrical engineering from Universiti Teknologi Malaysia, in 1998 and 2000, respectively, and the Ph.D. degree in electronics engineering from Universiti Sains Malaysia, in 2010. He is currently a Professor with the College of Engineering, Universiti Tenaga Nasional, Malaysia. He is also a Professional Engineer registered to the Board of Engineers Malaysia. His research interests include theory and applications of artificial intelligence.

CHEE PENG LIM received the Ph.D. degree in intelligent systems from The University of Sheffield, Sheffield, U.K., in 1997. He is currently an Associate Professor with the Institute for Intelligent Systems Research and Innovation, Deakin University, Geelong, VIC, Australia. He has authored more than 350 technical papers in journals, conference proceedings, and books, and edited three books and 12 special issues in journals. His research interests include computational intelligence-based systems for data analytics, condition monitoring, optimization, and decision support. He served on the Editorial Board for five international journals. He received seven best paper awards.

ERIC WAI MING LEE received the B.Eng. degree (Hons.) in building services engineering and the Ph.D. degree in fire engineering from the City University of Hong Kong. He is a Professional Engineer and currently a Panel member of the Fire Discipline of the Hong Kong Institution of Engineers, the Assessor of the Hong Kong Laboratory Accreditation Scheme of the HKSAR Government, and a member of various technical committees of the Buildings Department of the HKSAR Government.