A Novel Meta-cognitive Extreme Learning Machine to Learning from Data Streams

Mahardhika Pratama, Jie Lu, Guangquan Zhang
Centre of Quantum Computation and Intelligent System, Faculty of Engineering and Information Technology,
University of Technology Sydney, Australia,
email: pratama@ieee.org,jie.lu@uts.edu.au,guangquan.zhang@uts.edu.au

***Abstract*—Extreme Learning Machine (ELM) is an answer to an increasing demand for a low-cost learning algorithm to handle big data applications. Nevertheless, existing ELMs leave four uncharted problems: complexity, uncertainty, concept drifts, curse of dimensionality. To correct these issues, a novel incremental meta-cognitive ELM, namely Evolving Type-2 Extreme Learning Machine (eT2ELM), is proposed. eT2ELM is built upon the three pillars of meta-cognitive learning, namely what-to-learn, how-to-learn, when-to-learn, where the notion of ELM is implemented in the how-to-learn component. On the other hand, eT2ELM is driven by a generalized interval type-2 Fuzzy Neural Network (FNN) as the cognitive constituent, where the interval type-2 multivariate Gaussian function is used in the hidden layer, whereas the nonlinear Chebyshev function is embedded in the output layer. The efficacy of eT2ELM is proven with four data streams possessing various concept drifts, comparisons with prominent classifiers, and statistical tests, where eT2ELM demonstrates the most encouraging learning performances in terms of accuracy and complexity.**

Keywords*—*extreme learning machine, fuzzy neural network, meta-cognitive learning, evolving neuro fuzzy system, sequential learning.

I. INTRODUCTION

Nowadays, we witness the advent of big data applications, which possess 4Vs characteristics [1],[2]: Volume, Velocity, Variety, Veracity. This phenomenon has led to substantial demand of computationally efficient algorithm, while retaining accurate predictive accuracy. This issue motivated Huang et al to initiate the so-called Extreme Learning Machine (ELM) [3] with a basic notion of learning without iterative tuning to expedite the processing time. Huang et al argue [3] that there exists L hidden nodes randomly generated from any continuous probability function to predict N distinct samples with probability one $L << N$. They provide a proof that ELM satisfies universal approximation criteria in [4]. Recently, a meta-cognitive ELM was proposed in [5], where it realizes the meta-memory model of [5]. The meta-cognitive learning scenario consists of three learning phases: termination of study (when-to-learn); selection of kind of processing (how-to-learn); item selection (what-to-learn), where they are respectively actualized by the sample deletion strategy, the sample learning strategy, the sample reserved strategy. It uses the Radial Basis Function (RBF) neural network as the cognitive constituent.

Although vast efforts have been devoted, more in-depth studies of ELM are required for four reasons: 1) the issue of structural complexity remains an open issue to be solved especially in the context of incremental learning. A hidden node pruning strategy has been designed in [6]. Nevertheless, this approach undermines the logic of online learning, because they rely on a multi-staged training mechanism; 2) currently applied ELMs have not adequately addressed the issue of uncertainties, because most ELMs are built upon the type-1 hidden nodes. Although the interval type-2 ELM has been proposed in [7], this algorithm constitutes a batched learning machine, which is offline in nature; 3) most ELMs adopt a static network topology, which cannot adapt to changing learning environments. The notion of a dynamic ELM was devised in [8]. Nevertheless, the structural learning scenario does not reflect the real data distribution, thereby being unable to discover the focalpoints of data streams. Another seminal work was proposed in [9], where it makes use of the ensemble approach endued by the drift detection method. However, it possesses a demanding complexity, since it utilizes the multi-classifiers principle; 4) to the best of our knowledge, ELMs cannot cope with the curse of dimensionality because of the absence of online feature selection.

To remedy these drawbacks, a novel interval type-2 ELM, namely Evolving Type-2 Extreme Learning Machine (eT2ELM), is proposed in this paper. eT2ELM is constructed by a generalized interval type-2 FNN as the cognitive component, where the interval type-2 multivariate Gaussian function with uncertain means is mounted in hidden layer, while benefiting from the nonlinear Chebyshev function in the output layer. The training process of eT2ELM is based on the three pillars of meta-cognitive learning: what-to-learn; how-to-learn; when-to-learn, where all of which run in the single-pass learning process. First, all data streams are selected by the what-to-learn part, using the certainty-based active learning method, to extract relevant training samples for model updates. The training samples are then injected to the how-to-learn phase, adjusting the cognitive component. The how-to-learn part adopts the underlying spirit of ELM, where no-tuning of hidden nodes is performed to curtail the training process. Specifically, the hidden nodes are evolved by the Type-2 Data Quality (T2DQ) method, which monitors the contribution of data streams. The hidden nodes, which play a minor impact of the classifier's output, are pruned with the use of Type-2 Relative Mutual Information (T2RMI) method to render a scalable network burden. The output weights are analytically tuned by the so-called Fuzzily Weighted Generalized Least Square (FWGRLS) method derived in [10]. This method constitutes a local learning version of the GRLS method in [11]. In addition, the how-to-learn process is equipped by the online feature selection method by virtue of the analysis of relevance and redundancy. Conversely, the when-to-learn process exploits the standard sample reserved strategy. The efficacy of eT2ELM has been numerically validated by using four data streams, featuring dynamic and evolving characteristics and is compared by the state-of-the art classifiers. Obviously, eT2ELM is capable of producing more reliable predictions, while retaining more parsimonious

structures. The remainder of this paper is structured as follows: section 2 outlines the cognitive architecture of eT2ELM and section 3 elaborates the meta-cognitive learning policy of eT2ELM. Section 4 details numerical studies and Section 5 concludes the paper.

II. COGNITIVE ARCHITECTURE OF eT2ELM

The generalized interval type-2 FNN puts forward the interval type-2 multivariate Gaussian hidden node with uncertain means, which enhances the standard form usually crafted with the uni-dimensional Gaussian function. This hidden node is capable of triggering a non-axis parallel ellipsoidal cluster arbitrarily rotated in any direction. The underlying advantage of this cluster over the axis-parallel ellipsoidal or spherical cluster as generated in the traditional interval type-2 FNN leads to a more exact input space partition notably when data streams are not distributed in the main axis. This aspect compensates a possible increase of network parameters, because less fuzzy rules are crafted to properly cover the training data. Beside, this hidden node has a scale invariant trait, which can omit the need of data normalization. The activation function of hidden node is expressed as follows:

$$\widetilde{R}_i = \exp(-(X_n - \widetilde{C}_i)\Sigma_i^{-1}(X_n - \widetilde{C}_i)^T) \;,\; \widetilde{C}_i = [C_{i,1}, C_{i,2}] \quad (1)$$

where $\widetilde{C}_i \in \Re^{1\times u}$ stands for the centroid of the Gaussian function, which presents an interval valued set $C_{i,1} > C_{i,2}$ to form the Footprint of Uncertainty (FoU) and u denotes the number of hidden nodes. $\Sigma_i^{-1} \in \Re^{u\times u}$ labels a non-diagonal covariance matrix , whose elements steer the orientation of ellipsoidal cluster. Nevertheless, the multivariate Gaussian function operates in the high dimensional space, thus being unable to be fed directly in the fuzzy inference process. It also obscures the transparency of fuzzy rules because of the absence of atomic clauses. To correct this shortcoming, the multivariate Gaussian function requires to be projected onto one dimensional space (neuron level), where this can be done using the transformation strategy of [10]. We exploit the second method of [10], because it offers an instantaneous mechanism but it is less accurate given that the ellipsoidal cluster is revolved around 45 degrees. Specifically, we aim to elicit the radius of the multivariate Gaussian function as follows:

$$\sigma_i = \frac{r}{\sqrt{\Sigma_{ii}}} \quad (2)$$

where Σ_{ii} is a diagonal element of the multivariate Gaussian function in the *i-th* coordinate and r is the Mahalanobis distance between the datum and the *i-th* cluster. On the other hand, the centroid of the hidden node can be applied without any modification in the neuron level. Once obtaining the representation of neuron, the interval type-2 Gaussian function with uncertain means $\widetilde{c}_i = [{c_{j,1}}^i, {c_{j,2}}^i]$ can be defined as follows:

$$\widetilde{\mu}_{i,j} = \exp(-(\frac{x_j - \widetilde{c}_{i,j}}{\sigma_{i.j}})^2) \,, \widetilde{c}_i = [{c_{j,1}}^i, {c_{j,2}}^i] \quad (3)$$

This induces a bounded interval set $\widetilde{\mu}_{i,j} \in [\underline{\mu}_{i,j}, \overline{\mu}_{i,j}]$, which is built upon upper and lower memberships as follows:

$$\overline{\mu}_{i,j} = \begin{cases} N({c_{j,1}}^i, \sigma_{i,j}; x_j) & x_j < {c_{j,1}}^i \\ 1 & , {c_{j,1}}^i \le x_j {c_{j,2}}^i \\ N({c_{j,2}}^i, \sigma_{i,j}; x_j) & x_j > {c_{j,2}}^i \end{cases} \quad (4)$$

$$\underline{\mu}_{i,j} = \begin{cases} N({c_{j,2}}^i, \sigma_{i,j}; x_j) & x_j \le \frac{({c_{j,1}}^i + {c_{j,2}}^i)}{2} \\ N({c_{j,1}}^i, \sigma_{i,j}; x_j) & x_j > \frac{({c_{j,1}}^i + {c_{j,2}}^i)}{2} \end{cases} \quad (5)$$

We can thus produce an interval firing strength $\widetilde{R}_i = [\underline{R_i}, \overline{R_i}]$ by using the product *t-norm* operator for the upper and lower memberships as follows:

$$\underline{R_i} = \prod_{j=1}^{u} \underline{\mu_{i,j}}, \overline{R_i} = \prod_{j=1}^{u} \overline{\mu_{i,j}} \quad (6)$$

The conventional interval type-2 FNNs rely on the zero or first order Takagi Sugeno Kang (TSK) output node, which does not fully exemplify a local output mapping aptitude. To remedy this bottleneck, the input representation in the expanded input vector $x_e \in \Re^{1\times(2u+1)}$ is enhanced by using the notion of functional link neural network. Specifically, we extend the degree of freedom of the output layer by implementing the non-linear mapping based on the Chebyshev polynomial as follows:

$$T_{n+1}(x) = 2x_j T_n(x_j) - T_{n-1}(x_j) \quad (7)$$

where $T_0(x_j) = 1, T_1(x_j) = x_j, T_2(x_j) = 2{x_j}^2 - 1$. Because we employ up to the second order of the Chebyshev polynomial, the extended input vector is formed as follows $x_e = [1, x_1, T_2(x_1), x_2, T_2(x_2)]$. Note that other functional link neural networks, namely trigonometric and polynomial, also exist in the literature. Nevertheless, we do not apply them in the eT2ELM, because they incur more parameters to be stored in the memory. Henceforth, the type reduction mechanism is committed to transform the type-2 fuzzy output to the type-1 fuzzy output, called the type reduced set. In most interval type-2 FNNs, this mechanism is undertaken by the so-called Karnik-Mendel (KM) iterative procedure, which imposes the costly computational cost. Alternatively, the $q \in \Re^{1\times m}$ design factor is used and essentially controls the proportion of upper and lower outputs in the type-reduced set as follows:

$$y_o = \frac{(1-q_o)\sum_{i=1}^{P} \underline{R_i} y_{i,o} + q_o \sum_{i=1}^{P} \overline{R_i} y_{i,o}}{\sum_{i=1}^{P} \underline{R_i} + \overline{R_i}} = \frac{(1-q_o)\sum_{i=1}^{P} \underline{R_i} x_e \Omega_{i,o} + q_o \sum_{i=1}^{P} \overline{R_i} x_e \Omega_{i,o}}{\sum_{i=1}^{P} \underline{R_i} + \overline{R_i}} \quad (8)$$

where P is the number of hidden nodes. Because the eT2ELM poses the spirit of ELM, the design factor $q \in \Re^{1\times m}$ is randomly generated without being adapted in the training process. The classification decision is inferred with the use of MIMO architecture.

$$y = \max_{o=1,..,m}(\hat{y}_o) \quad (9)$$

where m is the number of class labels. It is worth noting that other classifier architectures like one against one and one against all are compatible with eT2ELM. However, the MIMO

architecture is explored in this paper, because it is widely studied in the literature.

III. META-COGNITIVE LEARNING POLICY OF ET2CLASS

A. How-to-Learn

1) Hidden Node Generation Mechanism: eT2Class adopts an open structure principle, where the knowledge building process can be automated by virtue of the Type-2 Data Quality (T2DQ) method. The crux of this method is to quantify the global density of data clouds recursively, thereby being able to discover focalpoints of data distributions to serve as the hidden nodes. In a nutshell, the T2DQ method is defined as follows:

$$FS_{P+1} = \sqrt{\frac{(1-q)^2 U_N}{U_N(1+b_N)-2h_N+g_N} + \frac{q^2 U_N}{U_N(1+b_N^{\,1})-2h_N^{\,1}+g_N}} \quad (10)$$

$$U_N = U_{N-1} + FS_{N-1} \quad , \quad b_N = \sum_{j=1}^{u+m} (\bar{c}_{j,P+1})^2 \quad , \quad b_N^{\,1} = \sum_{j=1}^{u+m} (\underline{c}_{j,P+1})^2 \quad ,$$

$$h_N = \sum_{j=1}^{u+m} c_{j,P+1} z_{N-1,j} \ , \ h_n^{\,1} = \sum_{j=1}^{u+m} c_{j,P+1} z_{N,j}^{\,1} \ , \ z_{N,j} = z_{N-1,j} + x_{j,N-1} U_{N-1,j}$$

$$z_{N,j}^{\,1} = z_{N-1,j}^{\,1} + x_{j,N-1} U_{N-1,j} \ \ g_{N,j} = g_{N-1,j} + \sum_{j=1}^{u+m} (x_{j,N-1})^2 U_{N-1,j}$$

where $x_{j,N-1}$ is a data stream at *N-1* in the *j-th* coordinate and $x_j^{\,N}$ is the latest incoming data stream. $\bar{c}_{j,P+1}, \underline{c}_{j,P+1}$ are the upper and lower centroids of the hypothetical hidden node (P+1st), which are crafted as $\bar{c}_{j,P+1} = x_j^{\,N} + \Delta x$, $\underline{c}_{j,P+1} = x_j^{\,N} - \Delta x$. Δx is an uncertainty factor , which is fixed as 0.1 in all our simulations in this paper. The design factor is involved to govern the dominance of upper and lower nodes, because it plays an important role to deal with the uncertainty of data streams leading to the inexact parameter identification. If *q=0.5* is chosen, it is akin to the average cardinality principle [12]. On the other hand, the weighting factor U_N is involved to alleviate the impact of outliers to the subsequent data streams. The new hidden node is grown, given that one of the two criteria is satisfied:

$$FS_{P+1} > \max_{i=1,\ldots,P}(FS_i) \, or \, FS_{P+1} < \min_{i=1,\ldots,P}(FS_i) \quad (11)$$

where FS_i can be elicited by replacing $\bar{c}_{j,P+1}, \underline{c}_{j,P+1}$ with $\bar{c}_{j,i}, \underline{c}_{j,i}$. $FS_{P+1} > \max_{i,=1,\ldots,P}(FS_i)$ implies that the hypothetical hidden node occupies a dense input region, thus being a promising candidate to be a new hidden node. On the other side, $FS_{P+1} < \min_{i,=1,\ldots,P}(FS_i)$ portrays an input region, uncharted by any existing hidden nodes. This data stream indicates a precursor of rapidly changing learning environments, thus being desired to be incorporated as a new hidden node. As a result, the knowledge exploratory module presented in this section is deemed equivalent with the variants of drift detection strategy in the conventional machine learning.

If the new hidden node is demanded, its parameters are initialized as follows:

$$C_{P+1} = X_N, \Sigma_{P+1}^{\,-1} = rand(A), A \in \Re^{u \times u} \quad (12)$$

where the data stream is assigned as the new datum to track the concept drifts, while randomly generating the inverse covariance matrix. Conversely, the new local sub-model is set as that of the winning node to expedite the training process, because the winning local sub-model is supposed to have an adjacent relationship with the new node. The output covariance matrix is crafted as a positive big definite matrix, which allows to rapidly emulate the real solution provided by the batched learning scenario [13].

$$\Omega_{P+1} = \Omega_{win}, \Psi_{P+1} = \Theta I \quad (13)$$

where ω stands for a large positive constant, fixed as $\Theta = 10^5$. Unlike most ELMs, eT2ELM employs a local learning principle, where each local sub-system is loosely coupled and possesses a unique covariance matrix. This concept is appealing, because the learning of a particular node only affects little to the stability and convergence of other local sub-models. Hence, no special setting of output covariance matrix is needed, when committing the structural learning mechanism.

The winning node is chosen with the Bayesian concept, where the hidden node with the highest posterior probability $win = \arg\max_{i=1,\ldots,P} \hat{P}(R_i|X)$ is selected. The key characteristic of this strategy is its probabilistic fashion. It is capable of capturing the true winning node, when there are some candidates lying on the roughly same distances to the sample of interest. The posterior probability $\hat{P}(R_i|X)$, the prior probability $\hat{P}(R_i)$, the likelihood function $\hat{P}(X|R_i)^{1,2}$ are respectively formulated as follows:

$$\hat{P}(R_i|X) = \frac{1}{2}\left(\frac{\hat{p}(X|R_i)^1 \hat{P}(R_i)}{\sum_{i=1}^{P} \hat{p}(X|R_i)^1 \hat{P}(R_i)} + \frac{\hat{p}(X|R_i)^2 \hat{P}(R_i)}{\sum_{i=1}^{P} \hat{p}(X|R_i)^2 \hat{P}(R_i)}\right) \quad (14)$$

$$\hat{P}(R_i) = \frac{\log(N_i + 1)}{\sum_{i=1}^{P} \log(N_i + 1)} \quad (15)$$

$$\hat{P}(X|R_i)^{1,2} = \frac{1}{(2\pi)^{1/2} V_i^{1/2}} \exp(-(X - C_i^{1,2}) \Sigma_i^{-1} (X - C_i^{1,2})^T) \quad (16)$$

where N_i denotes the number of populations of *i-th* cluster. The definition of the prior probability $\hat{P}(R_i)$ is softened from its original form to enable newly created nodes to win the competition more frequently.

2) Hidden Node Pruning Strategy: The rule pruning scenario, namely Type-2 Relative Mutual Information (T2RMI) method, is incorporated in eT2ELM. The RMI pruning method is pioneered in [14], but, the type-2 version of RMI method is unexplored. In this paper, the RMI method is enhanced to deal with the interval type-2 fuzzy system.

The T2RMI method aims to measure the mutual information between the hidden node and the class label, which is inherent with the relevance of hidden node. In essence, the correlation between the neuron and the target variable can be estimated using non-linear and linear

measures. Nevertheless, we utilize the nonlinear measure in this paper, since the interaction between two variables are often nonlinear. The symetrical uncertainty method is exploited to quantify the correlation, because it possessess three merits: simplicity, low bias for multi-valued features, insensitive to the order of two variables [17]. The T2RMI method is defined as follows:

$$RMI(\widetilde{R}_i, Y) = \left(\frac{(1-q)2I(\underline{R_i}, Y)}{H(\underline{R_i}) + H(Y)} + \frac{q2I(\overline{R_i}, Y)}{H(\overline{R_i}) + H(Y)}\right) \quad (17)$$

where $I(\overline{R}_i, Y) = H(\overline{R}_i) + H(Y) - H(\overline{R}_i, Y)$ is the information gain. $H(\overline{R}_i)$ is the entropy of $\overline{R}_i$ and $H(\overline{R}_i, Y)$ is the joint entropy of $\overline{R}_i, Y$. The symmetrical uncertainty method hovers around [0,1], where zero indicates that the two variable are uncorrelated. $H(\overline{R}_i), I(\overline{R}_i, Y)$ are usually elicited with the use of the discretization method or the Parzen window method. Nevertheless, the two techniques are incompatible for the online learning scenario, since it induces computationally prohibitive cost. To this end, we adopt the notion of differential entropy [15], which assumes the training data are normally distributed as follows:

$$H(\overline{R}_i) = \frac{1}{2}(1 + \log(2\pi \operatorname{var}(\overline{R}_i))) \quad (18)$$

$$I(\overline{R}_i, Y) = -\frac{1}{2}\log(1 - \rho^2_{\overline{R}_i Y}) \quad (19)$$

where $I(\overline{R}_i, Y)$ stands for Pearson's correlation coefficient, which is defined as follows:

$$\rho_{\overline{R}_i Y} = \frac{\operatorname{cov}(\overline{R}_i, Y)}{\sqrt{\operatorname{var}(\overline{R}_i)\operatorname{var}(Y)}} \quad (20)$$

Note that the mean, variance, and covariance can be processed in the incremental mode as done in [16]. Performing the same operation for $H(\underline{R}_i), I(\underline{R}_i, Y)$ completes the T2RMI procedure (17). The hidden node is pruned given that this criterion is satisfied as follows:

$$RMI < mean(RMI) - 2std(RMI) \quad (21)$$

where *mean(RMI)* is the average of RMI of hidden nodes and *std(RMI)* is the standard deviation of RMI of hidden nodes.

3) Online Feature Selection Mechanism: the underlying shortcoming of existing ELMs lies on the absence of feature selection method, which leads this mechanism to be executed before the process runs. Obviously, it does not coincide with the concept of online learning, where the learning process is supposed to run with negligible human intervention. eT2ELM is equipped by the input pruning mechanism, which enables to get rid of the inconsequntial input attributes on the fly without any significant loss of classifier's generalization. The online feature selection scenario of eT2ELM is constructed by the analysis of relevance and redundancy based on the Markovblanket criterion. The main strength of the Markov blanket concept can be perceived in its redundancy analysis, which guarantess once observing the Markov blanket in the earlier episode a feature can be discarded, because it will be no longer required for the future training episode. Therefore, it refines the perspective of conventional input pruning method [17], which is hampered by the instability issue, because neither the redundancy analysis is performed nor the input feature pruned in earlier episode can be recalled.

In the Markov Blanket concept, the input attributes can be classified into five criteria: irrelevant, weakly relevant, weakly relevant but non-redundant, strongly relevant. The optimal feature subset encompasses the strongly relevant features and the weakly relevant but non-redundant features, whereas other input attributes can be obviated. To this end, the relevance of input attributes is scrutinized and the input features that happen to be irrelevant are discarded. The analysis then continues on the issue of redundancy to find slightly relevant but redundant input variables to be pruned as well. Two measures are formulated as follows.

Definition 1(C-correlation) [18]: *The correlation between any feature x_j and the class T is termed as C-correlation, denoted by $SU(x_j, T)$.*

Definition 2 (F-correlation) [18]: *The correlation between any pair of features x_j, x_i $(i \neq j)$ is termed F-correlation, labeled by $SU(x_j, x_i)$.*

Both the C-correlation test $SU(x_j, T)$ and the F-correlation test $SU(x_j, x_i)$ can be quantified as the T2RMI method by means of the symmetrical uncertainty method. First of all, the C-correlation is undertaken to extract the irrelevant features, where an input feature is deemed superfluous and is then pruned, if it meets $SU(x_j, T) < \gamma$. γ is a predefined constant, which is fixed at 0.8 in respect to the recommendation of [18]. Those features, surviving the C-correlation test, are subject to the F-correlation test to delve their correlations with other input variables. The input features satisfying $SU(x_j, T) < SU(x_j, x_i)$ are deemed redundant, thereby being removed to relive the curse of dimensionality. It is worth mentioning that a two-staged approach is committed in eT2ELM to mitigate the computation cost, since not all features are included in the C-correlation check, which is computationally demanding.

4) Extreme Parameter Learning Mechanism: the concept of ELM can be found in the parameter learning scenario of eT2ELM, where the hidden nodes are randomly generated, whereas the output weights are analytically adjusted using the FWGRLS method. The FWGRLS method offers a greater robustness against noise and more flexible adaptation method than the conventional RLS method in current ELMs, because it relies on the local adaptation principle, where the FWGRLS method is modified from its global learning formula in [10]. The salient trait of the FWGRLS method is its generalized decay function, which boosts the weight decay effect of the RLS method. The weight decay effect is capable of suppressing the weight vector to concentrate on a small bounded interval, thereby substantiating the generalization. In addition, it aids to render the parsimonious network structure,

because a superflous local sub-model can be detected easily with the rule pruning method. The FWGRLS method is expressed as follows:

$$\psi(n) = \Psi_i(n-1)F(n)\left(\frac{R(n)}{\Lambda_i(n)} + F(n)\Psi_i(n-1)F^T(n)\right)^{-1} \quad (22)$$

$$\Psi_i(n) = \Psi_i(n-1) - \psi(n)F(n)\Psi_i(n-1) \quad (23)$$

$$\Omega_i(n) = \Omega_i(n-1) - \varpi\Psi_i(n)\nabla\xi(\Omega_i(n-1)) + \Psi(n)(t(n) - y(n)) \quad (24)$$

$$y(n) = x_{en}\Omega_i(n) \text{ and } F(n) = \frac{\partial y(n)}{\partial \Omega(n)} = x_{en} \quad (25)$$

where $\tilde{\Lambda}_i(n) \in \Re^{(P+1)\times(P\times 1)}$ denotes a diagonal matrix, whose diagonal elements comprise the firing strength of the fuzzy rule $\tilde{R}_i$. $\Delta(n)$ stands for the output covariance matrix, which can be set as an identify matrix [11]. Moreover, $\nabla\xi(\Omega_i^{l,r}(n-1))$ shows the gradient of the weight decay function and $\varpi \approx 10^{-15}$ exhibits a case-insensitive predefined constant. Note that the weight decay function can be any nonlinear function, which may not be differentiable. Hence, it is expanded to *n-1* step, whenever the weight decay function has an inexact gradient expression. We, however, make use of the quadratic function, because it is capable of proportionally adapting the weight vector to its current values.

B. What-to-Learn

First of all, the significance of data streams is evaluated with the certainty-based active learning method, which is triggered by the Bayesian concept. The advantage of eT2ELM over the sample deletion strategy in [5] is its aptitude to underpin the semi-supervised learning scenario, because it can relieve the annotation efforts by operators. In realm of evolving fuzzy systems, some active learning methods were proposed in [19], but cannot effectively handle the concept drifts. This bottleneck imposes severe labeling, when observing concept drifts in data streams.

The conflict level is investigated in both input and output spaces with the use of Bayesian posterior probability. The conflict measure in the input space aims to examine whether or not a data streams lies on a class overlapping region, whereas the conflict case in the output case scrutinizes whether or not a data stream is adjacent to the decision surface. The conflict in the output space is probed using the Bayesian posterior probability as follows:

$$p(\hat{y}_o|X)^{output} = \min(\max(conf_{final},0),1), \; conf_{final} = \frac{\hat{y}_1}{\hat{y}_1 + \hat{y}_2} \quad (26)$$

where $p(\hat{y}_o|X)^{output}$ stands for the output posterior probability, and $\hat{y}_1, \hat{y}_2$ are the most and the second most dominant outputs. This measure hinges on the quality of decision boundary when classifying a data stream. On the other hand, the Bayesian posterior probability in (14) is extended using the estimate of joint-category and class probability $P(\hat{y}_o|R_i)$ to produce the Bayesian class posterior probability as follows:

$$P(\hat{y}_o|X) = \frac{1}{2}\left(\frac{\sum_{i=1}^{P} P(\hat{y}_o|R_i)P(X|R_i)^1 P(R_i)}{\sum_{o=1}^{m}\sum_{i=1}^{P} P(\hat{y}_o|X)P(X|R_i)^1 P(R_i)} + \frac{\sum_{i=1}^{P} P(\hat{y}_o|R_i)P(X|R_i)^2 P(R_i)}{\sum_{o=1}^{m}\sum_{i=1}^{P} P(\hat{y}_o|X)P(X|R_i)^2 P(R_i)}\right) \quad (27)$$

$$P(\hat{y}_o|R_i) = \frac{\log(N_i^o + 1)}{\sum_{o=1}^{m}\log(N_i^o + 1)} \quad (28)$$

where N_i^o is the number populations of *i-th* cluster falling into the *o-th* class and $p(\hat{y}_o|X)^{input}$ is the Bayesian class posterior probability in the input space. This measure is employed to check the conflict level in the input space, because it abstracts the class overlapping case due to $P(\hat{y}_o|R_i)$. A data stream is worth learning, if it incurs substantial conflict as follows:

$$p(\hat{y}_o|X)^{output} > \theta \text{ and } p(\hat{y}_o|X)^{input} > \theta \quad (29)$$

where θ is the conflict threshold and is initialized as $\theta = \frac{1}{m} + 0.2(1 - \frac{1}{m})$. The initialization strategy is derived with assumption that the training data are uniformly distributed. The conflict threshold has to be adjusted during the training process to compensate the concept drifts $\theta_{N+1} = \theta_N(1 \pm s)$ [23]. Specifically, the threshold increases $\theta_{N+1} = \theta_N(1+s)$, when $p(\hat{y}_o|X) \geq \theta$, whereas it decreases $\theta_{N+1} = \theta_N(1-s)$ when $p(\hat{y}_o|X) < \theta$. We set *s=0.05* as done in [20].

C. When-to-Learn

The when-to-learn scenario terminates the training process by exploiting the sample reserved strategy. The reserved samples (XS_n, TS_n) are those complying to(40).

$$\min_{i,=1,..,P}(FS_i) \leq FS_{P+1} \leq \max_{i=1,..,P}(FS_i) \text{ and } y_N = t_N \quad (30)$$

This condition designates that a data stream does not convey meaningful information in the input space and can be classified easily with the current structure. The reserved samples are pushed in the rear sample stack and are learned after fully completing the underlying training samples. Ideally, the training process is terminated when both reserved and training samples are fully completed. In practice, this strategy is impossible to be carried out due to the nature of data streams, which may be infinite. Therefore, the training process is ended, when the number of reserved samples is constant. The reserved samples are useful to enrich the concept of the main training samples.

IV. NUMERICAL STUDY

eT2ELM is numerically validated with four data streams, characterizing various concept drifts. We benefit from a semi artificial data stream, yeast, stemming from the UCI machine learning repository. The data stream is customized in [21] to include the sudden concept drift. The synthetic problem, namely10dplane, is exploited to test the viability of eT2ELM, where it was devised in DDD database [21]. Note that there are three versions of the 10dplane study case in the DDD database. We, however, make use the most complex version, where changing data distributions take place in most samples. As with the Yeast study case, this problem features the abrupt concept drift. In addition, we utilize the rotating checkerboard

and rotating spiral study cases, which were designed in [22]. These two problems feature the recurrent concept drift.

Table 1. consolidated results of benchmarked system in four datasets

algorithm		ROCB	Yeast	ROSP	10dplane
eT2Class	CR	0.65±0.07	0.43±0.2	**0.53±0.03**	0.56±0.4
	HN	**1.4±0.5**	4±1	**1.5±0.5**	3.2±1.4
	ET	1.9±0.5	5.03±1.2	2.4±0.5	0.2±0.09
	NS	2000	890	**2000**	100
	NI	**8.5±0.01**	**4.5±0.03**	**2**	**2**
rClass	CR	0.65±0.1	0.46±0.08	0.54±0.03	0.76±0.2
	HN	3.1±1.4	3.6±1.5	**2.6±0.05**	3.4±4.7
	ET	1.4±0.16	0.79±0.16	1.5±0.6	0.09±0.06
	NS	837.3	199	425	45.3
gClass	CR	0.68±0.08	0.51±0.06	0.51±0.02	0.69±0.3
	HN	2.3±0.6	**2.6±0.9**	2.96±0.91	3.1±1.5
	ET	1.7±0.3	1.6±0.1	0.76±0.08	0.06±0.01
	NS	1654.3	879.2	462.89	49.6
eT2ELM	CR	**0.69±0.03**	**0.66±0.16**	**0.6±0.02**	**0.78±0.17**
	HN	5.8±1.4	10.6±1.3	7.1±4.7	7.2±0.2
	ET	**0.6±0.05**	**0.1±0.02**	0.13±0.01	**0.05±0.03**
	NS	**115.**	**105.1**	**130.5**	**38.8**

eT2ELM is benchmarked with three prominent algorithms: eT2ELM [12], rClass [16], gClass [23] and the predefined parameters of consolidated classifiers are set according to the rules of thumb in their publication. Our experimental procedure relies on the periodic hold-out process by means of the data stream generator in (www.cs.bham.ac.uk/~flm/opensource/DriftGenerator.zip). Meanwhile, our computational resources are an Intel (R) core (TM) i7-2600 CPU @3.4 GHz processor and 8 GB memory. The consolidated classifiers are assessed in five viewpoints: Classification Rate (CR), Hidden Nodes (HN), Number of Samples (NS), Execution Time (ET), Number of Input Attributes (NI). The numerical results are reported in Table 1.

Referring to Table 1, eT2ELM outperforms other algorithms in three facets: classification rate, execution time, and number of samples. Although eT2ELM evolves a higher number of hidden nodes, it possesses the lowest computational complexity and undergoes the most rapid runtime. Apart from that, eT2ELM is the sole classifier, which is capable of reducing the number of input attributes, thereby being able to surmount the curse of dimensionality.

V. CONCLUSION

This paper presents a novel ELM, termed Evolving Type-2 Extreme Learning Machine (eT2ELM). The fundamental working principle of eT2ELM is in line with the notion of metacognitive principle, where the learning engine is driven by the three learning components: what-to-learn; how-to-learn; when-to-learn. The what-to-learn constituent is governed by the certainty-based active learning method, while exploiting the sample reserved strategy in the when-to-learn part. The how-to-learn facet is triggered by two learning phases: structural learning; parameter learning. eT2ELM actualizes an open structure concept, where the network structure can be evolved, pruned automatically from data streams, is equipped by the online feature selection scenario. Furthermore, the parameter learning phase uses the extreme learning principle under the concept of learning without iterative learning. The potency of eT2ELM is numerically validated using four data streams and is benchmarked with its counterparts, where it outperforms other classifiers in achieving a tradeoff between complexity and simplicity.

ACKNOWLEDGEMENTS

The work presented in this paper is supported by the Australian Research Council (ARC) under Discovery Projects DP140101366.

REFERENCES

[1] T. Martin, "Fuzzy sets in the fight against digital obesity," *Fuzzy Sets Systems*, vol. 156, no. 3, pp. 411–417, (2005)

[2] V.Lopez, S.D.Rio, J.M.Benitez, F.Herrera," Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data", *Fuzzy Sets and Systems,* Vol.258, pp.5-38, (2015)

[3] G-B.Huang, Q-Y.Shu, C-K.Siew, " Extreme Learning Machine: Theory and Applications", Vol.70, pp.489-501, (2006)

[4] G-B.Huang, L.Chen, C-K.Siew, " Universal Approximation Using Incremental Constructive Feedforward Networks with Random Hidden Nodes", *IEEE Transactions on Neural Networks,* Vol.17(4), (2006)

[5] R. Savitha, S. Suresh, H. J. Kim, "A Meta-Cognitive Learning Algorithm for an Extreme Learning Machine Classifier", *Cognitive Computation*, Vol.6, pp.253-263, (2014)

[6] H-J. Rong, et al, " A fast pruned-extreme learning machine for classification problem", *Neurocomputing,* Vol.72, pp.359-366, (2008)

[7] Z. Deng, K-S. Choi, L. Cao, S. Wang, "T2FELA: Type-2 Fuzzy Extreme Learning Algorithm for Fast Training of Interval Type-2 TSK Fuzzy Logic System", *IEEE Transactions on Neural Networks and Learning Systems,* Vol. 25(4), pp.664-676, (2014)

[8] G. Feng, G-B. Huang, Q.Lin, R.Gay, "Error Minimized Extreme Learning Machine With Growth of Hidden Nodes and Incremental Learning", *IEEE Transactions on Neural Networks,* Vol.20(8), pp.1352-1357, (2009)

[9] B. Mirza, Z. Lin, N. Liu, "Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift", *Neurocomputing,* Vol.149, pp.315-329, (2015)

[10] M. Pratama, et al, "GENFIS: Towards an effective localist network", *IEEE Transactions on Fuzzy Systems*, on line and in press, (2013)

[11] Y. Xu, K.W. Wong, C.S. Leung, " Generalized Recursive Least Square to The Training of Neural Network", *IEEE Transactions on Neural Networks*, Vol.17(1), (2006)

[12] M. Pratama, J. Lu, G. Zhang, S. Anavatti," Evolving Type-2 Fuzzy Classifier", in press, *in proceeding of 2015 IEEE conference on fuzzy systems*, (2015)

[13] M. Pratama, S. Anavatti, E. Lughofer, " pClass: an effective classifier to streaming examples", *IEEE Transactions on Fuzzy Systems*, vol.23(2), pp.369-386, (2014)

[14] H. Gan, et al, "Nonlinear Systems Modeling Based on Self-Organizing Fuzzy-Neural-Network with Adaptive Computation Algorithm ", *IEEE Transactions on Cybernetics*, Vol. 44(4), pp.554-564, (2014)

[15] A. Lazo, et al," On the entropy of continuous probability distributions", *IEEE Transactions on Information Theory*, 24(1), 120–122, (1978)

[16] M. Pratama, S. Anavatti, J. Lu, " Recurrent classifier based on an incremental meta-cognitive scaffolding algorithm", *IEEE Transactions on Fuzzy Systems*, in press, (2015)

[17] P. Angelov, " Fuzzily Connected Multimodel Systems Evolving Autonomously From Data Streams" *IEEE Transaction on Systems, Man and Cybernetics-part b: Cybernetics*, vol. 40(4), pp.898-910,(2010)

[18] L. Yu, H. Liu, "Efficient Feature Selection via Analysis of Relevance and Redundancy", *Journal of Machine Learning Research*, Vol.5, pp.1205-1224, (2004)

[19] E. Lughofer, et al, " Reliable All-Pairs Evolving Fuzzy Classifiers", *IEEE Transactions on Fuzzy Systems*, Vol. 21(3), pp.625-541, (2013)

[20] I. Zliobaite, A. Bifet, B. Pfahringer, G. Holmes, " Active Learning with Drifting Streaming Data", *IEEE Transactions on Neural Networks and Learning Systems,* Vol.25, no.1, pp.27-39, (2014)

[21] L.L Minku, X. Yao, " DDD: A New Ensemble Approach for Dealing with Drifts", *IEEE Transactions on Knowledge and Data Engineering,* Vol.24(4), (2012)

[22] G. Ditzler, R. Polikar, "Incremental learning of concept drift from streaming imbalanced data", *IEEE Transactions on Knowledge and Data Engineering*, Vol.25(10), pp.2283-2301, (2012)

[23] M.Pratama, M-J.Er, S.Anavatti, E.Lughofer,N.Wang, I.Arifin, " A Novel Meta-Cognitive-based Scaffolding Classifier to Sequential Non-stationary Classification Problems", *in proceeding of IEEE International Conference on Fuzzy Systems*, Beijing , pp.369-376, 2014