Scaffolding Type-2 Classifier for Incremental Learning under Concept Drifts

Mahardhika Pratama, Jie Lu, Edwin Lughofer, Guangquan Zhang, Sreenatha Anavatti

Abstract— The proposal of a meta-cognitive learning machine that embodies the three pillars of human learning: what-to-learn, how-to-learn, and when-to-learn, has enriched the landscape of evolving systems. The majority of meta-cognitive learning machines in the literature have not, however, characterised a plug-and-play working principle, and thus require supplementary learning modules to be pre-or post-processed. In addition, they still rely on the type-1 neuron, which has problems of uncertainty. This paper proposes the Scaffolding Type-2 Classifier (ST2Class). ST2Class is a novel meta-cognitive scaffolding classifier that operates completely in local and incremental learning modes. It is built upon a multivariable interval type-2 Fuzzy Neural Network (FNN) which is driven by multivariate Gaussian function in the hidden layer and the non-linear wavelet polynomial in the output layer. The what-to-learn module is created by virtue of a novel active learning scenario termed the uncertainty measure; the how-to-learn module is based on the renowned Schema and Scaffolding theories; and the when-to-learn module uses a standard sample reserved strategy. The viability of ST2Class is numerically benchmarked against state-of-the-art classifiers in 12 data streams, and is statistically validated by thorough statistical tests, in which it achieves high accuracy while retaining low complexity.

Keywords: Fuzzy Neural Network, Neural Network, Evolving System, Concept Drift, Incremental Learning

1. Introduction

The variety of concept drifts in data streams has led to an increasing demand for an autonomous learning machine that mimics the mental development of a human being – a machine that is capable of starting the learning process from scratch and evolving its knowledge base from streaming data without tedious manual intervention. This requires a process whereby information can be memorised through examples and experiences and obsolete or inconsequential knowledge can be discarded (Angelov & Filev, 2004; Lughofer, 2008; Liu el at, 2007). Such learning machines, known as 'evolving systems', have appeal for many industrial applications because of their ability to cope with concept drift in data streams regardless of how data distributions change – whether they are slow, rapid, abrupt, gradual, local, global, cyclical or otherwise. Additionally, the evolving system is capable of overcoming the 'digital obesity issue', as raised in Martin (2005) as a result of its sample-wise learning of entire scenarios. Nonetheless, classical evolving systems are cognitive in nature and have not taken into account the meta-cognitive facet of human beings.

This issue has motivated the proposal of a meta-cognitive learning machine (Suresh et al, 2010), which essentially transforms the meta-memory model of Nelson and Narens (1990) in the realm of machine learning: what-to-learn; how-to-learn; when-to-learn. However, most meta-cognitive learning models in the literature have two underlying deficiencies: 1) these works have not taken into account scaffolding theory, which has the potential to generate a plug-and-play working principle. As such, they still require pre-or-post training processes to produce reliable learning performance; 2) the what-to-learn constituent still requires impractical labelling effort and therefore does not exemplify a semi-supervised learning principle. To correct these shortcomings, a meta-cognitive scaffolding algorithm was proposed in Pratama et al (2014b; 2015a), in which the key principle is the use of scaffolding theory to govern the how-to-learn part. Scaffolding theory itself is a renowned tutoring theory in the area of cognitive psychology which assists learners to complete complex learning tasks. Nevertheless, meta-cognitive scaffolding theory as developed by Pratama et al (2014b; 2015a) still deserves more in-depth investigation for five reasons: 1) They have not adequately addressed the uncertainty challenge

caused by the disagreement of expert knowledge, noisy measurement, inexact fuzzy rule identification, and noisy data (Karnik, Mendel & Liang, 1999); 2) Although they are suitable for a semi-supervised learning environment, they always assume the availability of expert knowledge, which is inevitably laborious to elicit, in annotating queried samples; 3) The online feature weighting strategy is only done by the relevance measure, performing a fully supervised working principle. Hence, the estimate of the input contribution is inaccurate for the semi-supervised learning environment and does not take into account mutual information among input variables; 4) Existing meta-cognitive classifiers make use of less reliable cognitive components. Specifically, the meta-cognitive-scaffolding classifiers in Pratama et al (2014b; 2015a) impose considerable memory burden owing to the use of the functional link-based Chebyshev function, whereas others (Angelov, 2004) do not possess full mapping ability due to the use of the zero or first order Takagi-Sugeno-Kang (TSK) output weights; 5) Their fundamental working principles are only valid for the type-1 fuzzy system, because they are particularly devised for the type-1 fuzzy system.

To remedy the bottlenecks of its predecessors, a novel, evolving, plug-and-play, and computationally efficient type-2 meta-cognitive scaffolding classifier – Scaffolding Type-2 Classifier (ST2Class) – is proposed in this paper. ST2Class automates the knowledge acquisition and organization phases as practiced by conventional evolving systems. It demonstrates the plug-and-play working principle, in which all learning modules are fully incorporated in a single training process with no pre-or-post-processed training steps, and is carried out in single-pass learning mode to retain scalability against big data. Moreover, ST2Class adopts the local learning scenario in which learning in a local sub-system does not harm the stability and convergence of other local sub-models. It can be assumed as a loosely coupled fuzzy model.

ST2Class's output is inferred by a multivariable interval type-2 Takagi-Sugeno-Kang (TSK) FNN, where the hidden node triggers a non-axis parallel ellipsoidal cluster by means of the interval type-2 multivariate Gaussian function. Specifically, an interval-valued fuzzy system, which is a special case of the interval type-2 fuzzy system featuring an interval primary variable, is utilised (Liang and Mendel, 2000). It is worth noting that most interval type-2 FNNs in the existing literature actualise an interval valued fuzzy system, which is a special case of the interval type-2 fuzzy system that uses an interval primary membership (Bustince et al, 2015). The wavelet function, which possesses dilation and translation parameters, constructs the output node. This kind of output node was inspired by Fuzzy Wavelet Neural Networks (FWNNs) (Abiyev & Kaynak, 2008; Abiyev, Kaynak & Kayaran, 2013). The ST2Class learning strategy starts with what-to-learn, steered by a brand new active learning procedure with a self-labelling property. The what-to-learn module extracts relevant training samples to be fed into the how-to-learn phase for model updates.

In the *how-to-learn* component, the hidden nodes are automatically generated and adjusted with respect to the conflict level induced by a datum admitted by the what-to-learn component to identify the regime-shifting properties of the system being learned. The conflict degree induced by the datum is probed by two rule-growing cursors termed a type-2 version of the Data Quality method and the Datum Significance concept (Pratama et al, 2015b), to determine a suitable learning scenario for scrutinizing the datum (accretion, tuning, and restructuring). Once the datum qualifies as a new fuzzy rule, its parameters are initialised by activating the so-called Bayesian class overlapping criterion to circumvent an overlapping region, induced by a new rule. Furthermore, the problematizing facet of the active scaffolding theory is manifested by consolidating the drift handling, rule pruning, rule recall and rule splitting techniques to decipher concept drift in the data stream. To

this end, the Type-2 Potential+ (T2P+) method is assembled, which in essence dismantles out-dated fuzzy rules by monitoring their density evolution with respect to up-to-date data distributions. The T2P+ method is also effective for executing the rule recall mechanism to overcome the circumstance of cyclic drift (Bartlett,1932), where the fuzzy rules deactivated by the T2P+ method can be reactivated when their attribute is compatible with the current data distribution. To cope with the gradual concept drift, the local error method to vet the error level in respective local regions is made use of orchestrating the local forgetting mechanism and adapting clusters more firmly to counterbalance the gradual concept drift. The cluster-splitting mechanism is used to apportion an over-sized cluster to preclude the so-called cluster delamination (Lughofer, 2012) and to surmount the problem of incremental concept drift. Moreover, the fading facet of the active scaffolding theory is imparted by two rule-based simplification procedures, where inconsequential fuzzy rules which contribute marginally during their lifespan are obviated by the so-called Type-2 Rule Significance concept (Pratama et al, 2015a). By extension, the redundant fuzzy rules, which strongly overlap other rules, are coalesced by implementing the vector similarity measure (Wu & Mendel, 2008).

ST2Class is endowed with the online input weighting module by virtue of the unsupervised mutual information estimation (Mitra, Murthy & Pal, 2002), which plays a complexity reduction role in active scaffolding theory. Lastly, the passive supervision part of scaffolding theory is driven by the parameter learning scenario. The parameter learning scheme uses the Fuzzily Weighted Generalized Recursive Least Square (FWGRLS) method to adapt the output weight. In addition, the parameter learning method using the zero-order density maximization principle (Silva, Alexandrem & Mardues de Sa, 2005) occurs in the adaptation of the design factors *ql* and *qr*, and the dilation and translation parameters of the wavelet functions. It is worth stressing that we do not utilise the renowned Karnik and Mendel (KM) type reduction procedure due to its computationally demanding characteristics (Karnik, Mendel & Liang, 1999). We instead adopt the construct of the *ql* and *qr* design coefficients of Abiyev and Kaynak (2010). In contrast, ST2Class is equipped by the standard sample reserved technique for the when-to-learn scenario.

In relation to state-of-the art classifiers, the novelty of this study lies in four factors:

(1) *The multivariable interval type-2 neuro-fuzzy architecture:* ST2Class is supported by multivariable interval type-2 neuro-fuzzy topology, which is potent in dealing with erratic and noisy learning environments. Although some interval type-2 FNNs (Juang & Tsao, 2008) can be observed in the literature, they mostly rely on the classical interval type-2 neuro-fuzzy models, triggering an axis-parallel Gaussian function in the input space and a linear hyper-plane in the output space. Additionally, the fuzzy wavelet neural network put forward in Abiyev and Kaynak (2008) is still built upon the classical type-1 fuzzy system.

(2) *Type-2 Meta-cognitive scaffolding Algorithm:* A novel type-2 meta-cognitive scaffolding theory is introduced in this paper which can be seen as a generalized version of the type-1 meta-cognitive scaffolding theory of Pratama et al (2014b; 2015a). Although several interval type-2 FNNs exist in the literature, they arguably suffer from three disadvantages. First, most of these works do not apply meta-cognitive scaffolding theory. Second, a number of important learning components are excluded from their learning engines, which consequently may require pre-or-post-training processes. Third, the rule-growing module is still constructed using a distance-based input partitioning scenario. This is deemed to be vulnerable to outliers.

(3) *The online active learning-based uncertainty method:* The uncertainty method presents a novel active learning concept in which it is in charge of organising the what-to-learn module. In contrast to the what-to-learn

scenarios in Suresh et al (2010) and Savitha et al (2012), our method is capable of reducing operator labelling effort. Our proposed method advances the online active learning methods in Pratama (2014b; 2015a) because it does not necessarily request the ground truth or the expert domain knowledge in labelling the data stream as a result of the autonomous annotation process. Other uncertainty-based active learning techniques exist in the literature, but they are offline in nature due to a pool-based approach (Lewis & Carlett,1994).

(4) *The input weighting-based unsupervised mutual information estimation method:* The drawback of the online feature weighting scenario in the meta-cognitive scaffolding classifier of Pratama (2014b; 2014c) lies in its fully-supervised trait. The approximation of the input significance is inaccurate in the semi-supervised context, and therefore produces less reliable input weights. By extension, most online feature weighting scenarios in the realm of evolving classifiers are often composed of a relevance measure (Lughofer, 2011a) via Fisher Separability Criterion (FSC) without considering redundancy among the input features (Yu & Liu, 2004).

The viability of ST2Class was numerically validated by means of thorough empirical studies in 13 evolving and adaptive data streams, benchmarks with cutting-edge classifiers, and various statistical tests. In summary, ST2Class outperformed its counterparts in 12 study cases in attaining a trade-off between accuracy and complexity. The rest of this paper is structured as follows: Section 2 outlines the literature survey; Section 3 deliberates on the cognitive aspects of ST2Class; Section 4 details the meta-cognitive scaffolding learning policy of ST2Class, including computational complexity analysis; Section 5 elaborates on the numerical validation of each learning component; Section 6 describes numerical and theoretical benchmarks with prominent classifiers, statistical tests, open problems; and Section 8 concludes this paper.

2. Literature review

This section reviews two related areas: part 2.1 discusses the basic concept of human learning involving schema, scaffolding and meta-cognitive theories; part 2.2 outlines the prominent evolving classifiers in the literature.

2.1 Human Learning

The major stumbling block in learning from data streams when there is no access to previous training patterns is the stability-plasticity dilemma (Elwell and Polikar, 2011). Controlling the stability-plasticity dilemma can help to achieve a trade-off between an old belief system and a new one, as elaborated in Schema theory, which presents a psychological model for the process of human knowledge acquisition and memory organization for future decision making (Flavell, 1996;Vygotsky, 1978).

Schema theory consists of two parts – schemata construction and schemata activation. Schemata construction is a mechanism for building schemata that is adaptable to new information that leads to three possible learning scenarios – accretion, tuning and restructuring. Each phase is characterised by the degree of conflict caused by a data stream. Accretion characterises learning when an example causes a minuscule degree of conflict or none at all, and the example is simply associated with an existing schema. Tuning occurs when an example causes a minor degree of conflict and fine-tuning of the schema is required. Restructuring describes a noticeable degree of conflict and the schema needs to be reorganised or entirely replaced. Schemata activation denotes a self-regulatory process to assess the performance of an existing schema or extrapolate a suitable learning scenario for a new example.

Scaffolding theory outlines supervision theory for students completing a complex learning task (Vygotsky, 1978; Reiser, 2004). In essence, Scaffolding theory aims to reduce learning complexity by either actively or passively supervising the training process. Passive supervision of a learning scenario consolidates the

experience-consequence process and relies on the predictive quality of fresh data. In other words, passive supervision is inherent in the parameter learning scenario of the FNN such as the Recursive Least Square (RLS) method and gradient-descent method. Active supervision comprises three phases: complexity reduction, problematizing, and fading (Wood, 2001). The main goal of the complexity reduction phase is to alleviate the learning burden. It comprises data pre-processing and feature selection. The problematizing phase tackles concept drift, which can be addressed by the drift detection mechanism, and the rule splitting strategy. The fading phase inhibits redundancy in the knowledge base and is usually accomplished using the rule merging scenario or rule pruning technology.

The introduction of scaffolding theory to the realm of machine learning boosts learning fidelity and gives rise to a plug-and-play algorithm in which all learning modules are contained in a single training process. Both schema and scaffolding learning theories focus only on human aptitude for information processing and encompass perception, learning, remembering, judging and problem-solving – all of which are, arguably, cognitive in nature. However, both concepts disregard common sense and emotional reasoning – the salient traits of human learning. In the realm of machine learning, such algorithms cannot normally select remarkable training stimuli and discard inconsequential samples, nor can they point out expedient time instants in which to explore the training samples.

The ability of human beings to appraise new knowledge in the context of their previous knowledge and their environment has been recognised as meta-cognition (Joysula et al, 2009; Flavell, 1979). In accordance with Nelson and Narens (1990), the meta-memory model is composed of termination of study (when to learn), selection of processing method (how to learn), and item selection (what to learn) (Isacson & Fujita, 2006). Typically, meta-cognitive learning adds two complementary learning scenarios – one to extract relevant data (what-to-learn) and another to pinpoint suitable training episodes (when-to-learn) – to the how-to-learn process, which is the only focus of traditional machine learning.

2.2 State-of-the art evolving classifiers

Angelov and Zhou (2008) pioneered the construct of Evolving Classifier (eClass) for solving online classification problems. In Angelov, Lughofer, and Zhou (2008), various evolving classifier architectures were tendered which utilised eClass and FLEXFIS-class as base classifiers. The Simplified Evolving Classifier+ (simp_eClass+) was proposed in Baruah and Angelov (2011), with key ideas stemming from simp_eTS+ in Angelov (2011). Another prominent contribution was made by Lemos et al (2013), which put forward Evolving Multivariable Gaussian (eMG) to resolve the online fault diagnosis and detection problem. In Lughofer and Buchtala (2013), an all-pair classifier topology was designed to infer the decision boundary of FLEXFIS-Class, and an online conflict-and-ignorance active-learning approach was put into perspective. Bouchachia and Vanaret (2014) recently devised a zero-order evolving classifier under the framework of the interval type-2 fuzzy set, called Growing Interval Type-2 Fuzzy Classifier (GT2FC). In Pratama et al (2014d), the so-called Parsimonious Classifier (pClass) was initiated, where pClass was extended from the Generic Evolving Neuro-Fuzzy Inference System in Pratama et al (2014a). Notwithstanding the work of numerous researchers, most evolving classifiers are cognitive in nature and merely address the how-to-learn issue of meta-cognitive learning.

To remedy the bottleneck of cognitive evolving classifiers, Suresh et al (2010) introduced the Self-Adaptive Resource Allocation Network (SRAN) based on the meta-memory model in (Nelson and Narens, 1990). SRAN was later developed into fully-complex network architecture in (Savitha et al, 2012). The projection learning

algorithm was offered to polish up the output parameters of a meta-cognitive-based radial basis function neural network in (Babu & Suresh, 2013). Meta-cognitive learning theory was actualised in the context of fuzzy systems in (Subramanian et al, 2013), and was later strengthened by amalgamating the interval type-2 fuzzy model in (Subramanian et al, 2014). Although the meta-cognitive learning has been realised in various network architectures, these meta-cognitive learning machines have two key shortfalls: 1) they still discount the scaffolding theory and thus do not characterise the plug-and-play principle; (2) They rely on impractical assumption of fully labelled data, which requires considerable operator intervention.

Scaffolding theory was initiated to steer the how-to-learn module of meta-cognitive learning in (Pratama et al, 2014(b)) and was crafted in the generalized type-1 fuzzy neural network structure as the cognitive component. By extension, this work was enhanced in (Pratama et al, 2015(a)), using a local recurrent network architecture. However, meta-cognitive scaffolding learning machines deserve more in-depth investigation, since these earlier works bear five shortcomings: 1) these works are still crafted in the type-1 fuzzy system, which is not robust enough to encounter uncertainties in data streams; 2) They always require expert knowledge in labelling queried samples; 3) they adopt the supervised feature weighting algorithm, which is inaccurate to appraise feature contribution in the semi-supervised learning scenario; 4) the cognitive components of meta-cognitive classifiers suffer from several major bottlenecks.

3 Network Architecture of ST2Class

This section articulates the inference scheme of ST2Class by virtue of the multivariable interval type-2 fuzzy neural network. The multivariable interval type-2 fuzzy rule empowers the multivariable Gaussian function as the rule antecedent and the wavelet function as the rule consequent. The ST2Class fuzzy rule is defined as follows:

$$R_i: \textbf{IF}\ \ X \ is\ \tilde{R}_i\ \textbf{Then}\ \tilde{y}_i^o = \phi_i(X_N)\tilde{\Omega}_{i,o}, \tilde{\Omega}_{i,o} = [\underline{\Omega}_{i,o}, \overline{\Omega}_{i,o}] \quad (1)$$

where $\tilde{R}_i = [\underline{R_i}, \overline{R_i}]$ denotes the multidimensional kernel, driven by the multivariate Gaussian function with uncertain non-diagonal covariance matrixes as follows:

$$\tilde{R}_i = \exp(-(X_n{}^{weight} - C_i)\tilde{\Sigma}_i^{-1}(X_n{}^{weight} - C_i)^T), \tilde{\Sigma}_i^{-1} = [\underline{\Sigma}_i{}^{-1}, \overline{\Sigma}_i{}^{-1}], X_n{}^{weight} = \lambda X_n \quad (2)$$

where $C_i \in \Re^{1\times u}$ is the centroid of *i-th* Gaussian fuzzy rule, and u is the number of input dimensions. $X_n{}^{weight} = \lambda X_n$ is a weighted input vector, where λ is the feature weight vector, produced by the online input weighting algorithm, outlined in Section 4.2.1 of this paper. $\tilde{\Sigma}_i{}^{-1} = [\underline{\Sigma}_i{}^{-1}, \overline{\Sigma}_i{}^{-1}] \in \Re^{u\times u}$ describes uncertain inverse covariance matrixes that trigger upper and lower membership degrees. Their elements indicate interrelationships between input features and control the orientation of the non-axis-parallel ellipsoidal clusters in the product space. The merit of the non-axis-parallel ellipsoidal cluster over the axis-parallel ellipsoidal and spherical clusters is that it can form a proper zone of influence – most notably when training samples are not scattered in the main input axes. It can be argued that this sort of fuzzy rule can evoke extra parameters stored in the memory owing to the use of the non-diagonal covariance matrix. However, we disagree with this claim, because the generalized fuzzy rule generates more exact input space partition, presumably dampening the fuzzy rule requirement to capture data distributions. Apart from that, this fuzzy rule variant can prevent information loss in input variable interactions, while featuring the advantage of scale-invariant and thus avoiding the need for the data pre-normalization process. The Gaussian neuron is used as a basis function in ST2Class because it

possesses an infinite support property and a steady differentiable merit. In addition, it inhibits undefined input states and provides a smooth approximation of a local data space.

The principal drawback of the multivariate Gaussian function in connection with the interval type-2 fuzzy inference scheme is that it is unable to provide the representation of the interval type-2 fuzzy set. To remedy this flaw, the fuzzy set representation of the multivariate Gaussian function should be extracted, which can be done by using Pratama et al (2013). Note that two methodologies in (Pratama et al, 2013) exist to illustrate the membership function of the multivariable Gaussian function. We employ the second method, since it confers instantaneous mathematical operation. Nevertheless, the fuzzy set form it imposes when encountering the ellipsoidal rule revolved around 45 degrees is too tiny. For brevity, the uncertain radii of interval type-2 Gaussian membership function with uncertain SDs are given as follows:

$$\tilde{\sigma}_i = \frac{r_i}{\sqrt{\tilde{\Sigma}_{ii}^{-1}}}, \tilde{\sigma}_i = [\underline{\sigma}_i, \overline{\sigma}_i], \tilde{\Sigma}_i^{-1} = [\underline{\Sigma}_i^{-1}, \overline{\Sigma}_i^{-1}] \tag{3}$$

where $\tilde{\Sigma}_{ii}^{-1}$ is the diagonal element of the inverse covariance matrix and r_i is the Mahalanobis distance between the datum and the *i-th* cluster. The goal of this transformation is to estimate the uncertain radii of the interval type-2 Gaussian neuron and ultimately shape the Footprint of Uncertainty (FOU), which is a peculiar feature of the interval type-2 fuzzy set in dealing with uncertainty (Bouchachia & Vanaret, 2014; Lin, Chang & Lin, 2014a). The centroid of the fuzzy set is the same as the non-axis parallel ellipsoidal cluster, so it can be injected directly to the fuzzy set level.

$\tilde{y}_i^{\,o} = [\underline{y}_i^o, \overline{y}_i^o] \in \Re^{1\times m}$ is the rule consequent of *i-th* rule and *o-th* class and is defined as $\underline{y}_i^o = \phi_i(X_N)\underline{\Omega}_i^o, \overline{y}_i^o = \phi_i(X_N)\overline{\Omega}_i^o$, $\tilde{\Omega}_i = [\underline{\Omega}_i, \overline{\Omega}_i] \in \Re^{u\times m}$ is an interval weight vector, in which *m* is the dimension of output space. $\tilde{q} = [q_l, q_r] \in \Re^{1\times m}$ stands for the design coefficient to reduce the type-2 fuzzy variable to the type-1 fuzzy variable. As a result of the *ql* and *qr* design factors in undertaking the type reduction mechanism, $\tilde{\Omega}_i = [\underline{\Omega}_i, \overline{\Omega}_i]$ puts forward the interval weight vectors, where $\overline{\Omega}_i = [\overline{w}_1, \ldots, \overline{w}_u], \underline{\Omega}_i = [\underline{w}_1, \ldots, \underline{w}_u]$. By extension, $\phi(X_N{}^{weight}) \in \Re^{(P\times u)}$ designates an extended input vector which is built upon a nonlinear mapping via the wavelet function to reinforce the local approximation property. We make use of a dilated and translated version of the Mexican Hat wavelet function $\phi_i(X_N{}^{weight}) \in \Re^{(1\times u)}$ as defined in Abiyev, Kaynak and Kayaran (2013) as follows:

$$\phi_{i,j}\left(\frac{x_j{}^{weight} - a_{i,j}}{b_{i,j}}\right) = \left(1 - \left(\frac{x_j{}^{weight} - a_{i,j}}{b_{i,j}}\right)^2\right)\exp\left(\frac{1}{2}\left(\frac{x_j{}^{weight} - a_{i,j}}{b_{i,j}}\right)^2\right) \tag{4}$$

In the vector form,

$$\phi_i\left(\frac{X_N{}^{weight} - A}{B}\right) = \left(1 - \left(\frac{X_N{}^{weight} - A_i}{B_i}\right)^2\right)\exp\left(\frac{1}{2}\left(\frac{X_N{}^{weight} - A_i}{B_i}\right)^2\right), A \in \Re^{P\times u}, B \in \Re^{P\times u}$$

where $a_{i,j}, b_{i,j}$ respectively represent the dilation and translation parameters. We deploy variable dilation and translation values in lieu of fixed values to easily figure out the different behaviours of each class and a nonlinear trait of the local part of the decision surface. It is worth emphasizing that the standard first order TSK rule output does not exhibit the full local mapping aptitude (Juang & Tsao, 2008), thus degenerating the classifier's generalization On the other hand, a functional link output benefiting from the Chebyshev function deploys massive parameters (Pratama et al, 2015a), which are prohibitive for limited computational resources.

The wavelet function is a plausible remedy for this shortcoming because it has a multi-resolution property which rigorously captures the global and local trend of any function with instantaneous convergence and reasonable memory demand.

The fuzzy inference scheme of the interval type-2 fuzzy system can be conveniently organised when (2) is executed, because we can conjecture that the upper and lower clusters are distinct in the higher dimensional space and are separated by a particular FOU. For brevity, the output of the interval type-2 hidden node can be produced in accordance with Mendel and John (2002) as follows:

$$\tilde{\mu}_{i,j} = \exp(-(\frac{x_j^{weight} - c_{i,j}}{\tilde{\sigma}_{i.j}})^2),\ \tilde{\sigma}_{i,j} = [\underline{\sigma}_{i,j}, \overline{\sigma}_{i,j}] \tag{5}$$

$$\overline{\mu}_{i,j} = N(c_{i,j}, \overline{\sigma}_{i,j}; x_j^{weight}), \underline{\mu}_{i,j} = N(c_{i,j}, \underline{\sigma}_{i,j}; x_j^{weight}) \tag{6}$$

where the upper fuzzy region is supposed to be larger than the lower region $\overline{\sigma}_{i,j} > \underline{\sigma}_{i,j}$. The upper and lower rule matching factors $\tilde{R}_i = [\underline{R_i}, \overline{R_i}]$ can be obtained by the *t-norm* operator in the interval type-2 fuzzy sense as follows:

$$\underline{R_i} = \prod_{j=1}^{u} \underline{\mu_{i,j}}, \overline{R_i} = \prod_{j=1}^{u} \overline{\mu_{i,j}} \tag{7}$$

The positive facet of the interval type-2 fuzzy system lies on its fuzzy membership degree, which is more effective at dealing with uncertainties in data streams than the crisp membership degree portrayed by the type-1 fuzzy system. Nevertheless, we do not exploit the general type-2 fuzzy system because the type-1 fuzzy mathematics is not applicable for the type-2 fuzzy system and therefore induces over-complex working procedures. In addition, the best secondary grade shape in the general type-2 fuzzy system is to date unknown. The interval type-2 fuzzy system can be seen as a special case of its general type-2 counterpart, where the secondary grade is assumed to be unity (Mendel & John, 2002). Since the output of the rule layer $\tilde{R}_i = [\underline{R_i}, \overline{R_i}]$ is an interval set, a type reduction mechanism should be carried out to provide a crisp output. The K-M iterative procedure is usually performed to derive the L and R end points, but this method incurs prohibitive computational load and is impractical in a sequential learning framework. The *ql* and *qr* design factors $\tilde{q} = [q_l, q_r] \in \Re^{1 \times m}$ are used to correct this shortcoming, in lieu of the K-M type reduction concept (Abiyev & Kaynak, 2010; Lin et al, 2014a) which dictates the proportion of the output intervals. This in turn lands on the upper and lower outputs $[y_l, y_r]$ (Lin et al, 2014b) as follows:

$$y_{l,o} = \frac{q_l{}^o \sum_{i=1}^{P} \underline{R_i} y_{\underline{i},o} + (1 - q_l{}^o) \sum_{i=1}^{P} \overline{R_i} y_{\underline{i},o}}{\sum_{i=1}^{P} \underline{R_i} + \overline{R_i}},\ y_{r,o} = \frac{(1 - q_r{}^o) \sum_{i=1}^{P} \overline{R_i} \overline{y}_{i,o} + q_r{}^o \sum_{i=1}^{P} \underline{R_i} \overline{y}_{i,o}}{\sum_{i=1}^{P} \underline{R_i} + \overline{R_i}} \tag{8}$$

where P is the number of fuzzy rules and $[y_l, y_r]$ are the type reduced sets, which results in the final crisp output. Note that (8) is akin to the output expression of standard interval type-2 fuzzy system. The only difference lies on the use of q design factors in lieu of the KM method. Furthermore, should the MIMO architecture be used to output the classification decision, the maximum operator is benefited to generate the classifier's output as follows

$$y_o = \frac{1}{2}(y_{l,o} + y_{r,o}), y = \max_{o=1,..,m}(\hat{y}_o) \quad (9)$$

The MIMO architecture is chosen in this paper to craft the decision boundary, since it is widely used in the literature. This will also enable fair comparison with state-of-the art classifiers. The fundamental working principle of ST2Class is shown in Fig.1, while Fig.2 depicts the Gaussian interval type-2 membership function with interval SDs.

4 Meta-Cognitive Component of ST2Class

This section outlines the how to learn component of ST2Class, which is built upon a synergy between the Schema and Scaffolding theories, stemming from the cognitive psychology.

4.1 What-to-Learn

The meta-cognitive learning scenario is equipped by the what-to-learn mechanism, which is instrumental in reducing the streaming data to be learned in the training process, thus curtailing the training process. In addition, it is capable of boosting the classifier's generalization, because the redundant samples can be detected and exempted from the training process, thus limiting the over-fitting situation. In respect of the existing meta-cognitive learning, the sample deletion strategy was proposed to cultivate the what-to-learn scenario in the pioneering work on meta-cognitive classifiers (Babu & Suresh, 2013; Subramanian et al, 2013; Subramanian et al, 2014). This work is, nevertheless, inefficient, because it is implemented by virtue of the hinge error function, which necessitates fully labelled training stimuli. To correct this shortcoming, the online active learning mechanism is embedded in Pratama et al (2014b; 2015a) to govern the what-to-learn module, which is capable of lowering the operator annotation work, thus implementing the semi-supervised learning scenario. Even so, Pratama et al (2014b; 2015a) invoke the ground truth or the expert domain knowledge in annotating the unlabelled training patterns, whereas in online situations such information attracts vast manual intervention.

This research develops a novel active-learning mechanism, namely the uncertainty measure. This method not only underpins ST2Class as a semi-supervised classifier, but also is equipped by an autonomous labeling weapon. Although the uncertainty measure is the dominant method of appraising the sample sensitivity in the literature, most of the uncertainty methods are offline in nature because of the pool-based approach (Lewis & Carlett,1994). First, the significance of the streaming data is monitored by the neighborhood probability method, proposed in Xiong, Azimi, & Fern (2014), in which the key notion is to approximate the probability of a datum belonging to the existing data clouds. It is worth noting that the original version of the neighborhood probability relies on a batched learning scheme, and we tailor this method for the sequential learning framework. For brevity, the probability of the training instance X_n populating existing data clouds is predicted as follows:

$$P(X_n \in N_i) = \frac{\frac{1}{N_i}\sum_{n=1}^{N_i} M(X_n{}^{weight}, X_i{}^{weight})}{\sum_{i=1}^{P}\sum_{n=1}^{N_i}\frac{M(X_n{}^{weight}, X_i{}^{weight})}{N_i}} \quad (10)$$

where $M(X_n, X_i)$ denotes the similarity measure between the latest datum X_n and the population of *i-th* data cloud X_i and in turn implies the local cluster density. This is expressed in a recursive form as follows:

$$\frac{\sum_{n=1}^{N_i} M(X_n, X_i)}{N_i} = \frac{\sum_{n=1}^{(N_i-1)}\sum_{j=1}^{u}(x_{n,j}{}^{weight} - x_{N,j}{}^{weight})^2}{(N_i-1)u} = \frac{(\sum_{j=1}^{u}(N_i-1)x^{weight}{}_{N,j}{}^2 - 2\sum_{j=1}^{u} x^{weight}{}_{N,j}\kappa_{N_i,j} + \nu_{N_i})}{(N_i-1)u} \quad (11)$$

$$\kappa_{N_i,j} = \kappa_{N_i-1,j} + x^{weight}{}_{N_i-1,j}, \nu_{N_i} = \nu_{N_i-1} + \sum_{j=1}^{u} x^{weight}{}_{N_i-1,j}{}^2$$

Alternatively, (10) can be gauged by $P(R_i|X_N{}^{weight})$ as defined in (22), if (11) is deemed too demanding to be carried out. Nevertheless, (22) is less accurate, because it does not compute the proximity between the datum and all local supports. The uncertainty of the datum is elicited by finding the entropy of its membership, expressed as follows:

$$H(\mathrm{N}|X_n{}^{weight}) = -\sum_{i=1}^{P} P(X_n{}^{weight} \in N_i) \log P(X_n{}^{weight} \in N_i) \tag{12}$$

Once the uncertainty of the data points has been observed, we need next to determine the sensitivity of the datum, whether or not it can be accepted for the training process. The streaming data are admitted for the training process on condition that the uncertainty level of the data is lower than the uncertainty threshold $H(\mathrm{N}|X_n) < \delta$, where δ is initialised as $\delta = 0.5$. Intrinsically, the stray or spurious samples, which should be removed without being engaged in the training process, influence the increase in uncertainty. We also introduce the budget *B*, which is the maximum labeling cost in the training process and is fixed at *B=0.2* in this paper. *B=0.2* means 20% of the total training samples to be labeled in the training process. On the other side, the labeling expense is gained as $\hat{b} = Z/N$, where *Z* is the number of queried streaming data, and *N* denotes the number of training observations seen so far. Nonetheless, this labeling cost formula may be insensitive for dealing with the lifelong learning scenario, so it is amended as follows:

$$\hat{Z}(n+1) = \hat{Z}(n)o + labelling,\ o = (\xi - 1)/\xi,\ \hat{b} = \hat{Z}/\xi \tag{13}$$

where ξ no longer describes the number of training examples, but the size of the sliding window. ξ is assigned as $\xi = 100$, while *labelling* signifies whether or not a datum is annotated (1 or 0). The actual labelling cost should not violate the allocated budget $B \geq \hat{b}$ otherwise the training process will be terminated. The final condition in respect of these two criteria for accepting the training instances is set as follows:

$$H(\mathrm{N}|X_n{}^{weight}) < \delta \text{ and } B \geq \hat{b} \tag{14}$$

The uncertainty threshold δ is adapted in conjunction with the learning context to deal with the changing learning context. It is evident that a non-stationary circumstance entails a considerably more detailed annotation process than a static case. Note that such situations can jeopardise the model update, because the training process may be capped off too early when the classifier is not yet mature enough to sort out the learning task. We therefore adopt the variable uncertainty sampling strategy of Zliobaite et al (2014), which aims to counterbalance the actual labelling expense and in turn, to consume the budget uniformly irrespective of the types of concept drift in the data streams. The adaptation mechanism is devised as $\delta_{N+1} = \delta_N(1 \pm s)$, where the threshold augments $\delta_{N+1} = \delta_N(1+s)$ when the training sample is trivial for the learning process $H(\mathrm{N}|X_n{}^{weight}) \geq \delta$. Vice versa, it diminishes $\delta_{N+1} = \delta_N(1-s)$ when the training sample makes a substantial contribution to the learning process $H(\mathrm{N}|X_n{}^{weight}) < \delta$. Moreover, the adjustment factor *s* plays a paramount role in specifying the step size, so it is fixed as *s=0.01* according to Zliobaite et al (2014).

The target class label is henceforth predicted in the Bayesian concept as defined in Subramanian et al (2014) and Suresh et al (2014), where the posterior probability is utilised as a cursor of the true class label, which can be written as follows:

$$P(\hat{y}_o|X^{weight})^{output} = \min(\max(conf_{final},0),1)\,,\; conf_{final} = \frac{\hat{y}_1}{\hat{y}_1+\hat{y}_2} \tag{15}$$

where $\hat{y}_1, \hat{y}_2$ denote the most dominant, and second most dominant output classes. Clearly, the prediction of the true class label via (15) hinges on the spatial proximity between the decision boundary built upon the local sub-models and the input pattern. Hence, the reliability of the self-labelling process is dependent on the quality of the rule consequent in classifying the streaming data. The autonomous labelling scenario is consequently undertaken when the classifier features a trustworthy decision boundary in subsuming the streaming data or the classifier is certain of its prediction. Expert knowledge is otherwise solicited in annotating the streaming data to stave off misguided information about the true class label, which exacerbates classification complexity due to an erroneous adjustment of the decision surface. The classifier is confident of its own prediction if the class posterior probability fathomed by (15) outstrips the labelling threshold as follows:

$$P(\hat{y}_o|X^{weight})^{output} \geq \delta_1 \tag{16}$$

where δ_1 designates the labelling threshold, which can be set as $\delta_1 = 0.55$. Seemingly, (16) indicates the dominance of the most eminent class label against the second most superior category. In conjunction with the classifier's confidence, $P(\hat{y}_o|X^{weight})^{output} \approx 0.5$ points out the classifier confusion in categorizing the training patterns into one of these two categories. The classifier confusion can be perceived as the training stimuli lying on or adjacent to the decision surface in the output space. Once (14) is satisfied, the self-labelling process is switched on as follows:

$$T_n = \max_{o=1,..,m}(P(\bar{y}_o|X^{weight}))^{output} = true\_class\_label^{output} \tag{17}$$

In spite of partially utilizing expert domain knowledge in annotating the data stream, most unlabelled samples will be self-annotated. This is later confirmed by our in-depth numerical studies in Section V.

4.2 How-to-Learn

This section outlines the how to learn component of ST2Class, which is built upon a synergy between the Schema and Scaffolding theories, stemming from the cognitive psychology.

4.2.1 Scaffolding Concept (Complexity Reduction Item) : Input Weighting Mechanism

The complexity reduction constituent of scaffolding theory aims to unravel a complex learning problem. This learning module is realised with data normalization or feature selection to cope with the curse of dimensionality. Since the multivariate Gaussian function that possesses the scale-invariant property is employed in the hidden node, the normalization of the training data can be ignored. In the evolving system, the notion of an online feature selection method has been pioneered in several algorithms, where the online feature selection scenario was either compiled by the online input pruning mechanism (Angelov, 2011; Pratama et al 2014b) or by the sequential feature weighting procedure (Pratama et al, 2014c, Lughofer, 2011). Generally speaking, the underlying drawback of the online input pruning module is that once the input feature has been discarded from the training process, it cannot be resurrected in the future. This learning manner prompts a discontinuity of the training process, which invokes a retraining phase from scratch to avoid instability. To the best of our knowledge, the variants of the online input weighting algorithm in the literature are solely targeted at the fully supervised learning machine. This learning perspective is incompatible with the ST2Class learning context, which is designed for the semi-supervised learning scenario.

ST2Class is underpinned by the online feature weighting mechanism, utilizing an unsupervised feature similarity measure termed the Maximal Information Compression Index (MICI) method (Mitra, Murthy & Pal,

2002). The key construct of the MICI method is to extract the mutual information conveyed by two input variables, where an input feature that has strong similarity to other input features can be ruled out from the training process by the allocation of marginal input weights, thus allowing its leverage to be smoothly suppressed to a low level. The MICI method is expressed as follows:

$$\xi(x_1, x_2) = \frac{1}{2}(\text{var}(x_1) + \text{var}(x_2) - \sqrt{(\text{var}(x_1) + \text{var}(x_2))^2 - 4\,\text{var}(x_1)\,\text{var}(x_2)(1-\rho(x_1, x_2)^2)}) \quad (18)$$

$$\rho(x_1, x_2) = \frac{\text{cov}(x_1, x_2)}{\sqrt{\text{var}(x_1)\,\text{var}(x_2)}} \quad (19)$$

where $\text{cov}(x_1, x_2), \text{var}(x_1), \text{var}(x_2)$ can be recursively quantified by the same strategy as (22). In essence, the underlying construct of this measure reflects the eigenvalue for normal direction to the principal orientation of two input variables (x_1, x_2), in which the maximum information compression is attained when the streaming data are projected in the direction of their principal component. For brevity, (18) conceives the amount of information loss when one of these two variables is discarded from the learning process. It is worth noting that the MICI method is more appealing than the correlation coefficient method in (19), because (19) is sensitive to rotation. In summary, the MICI method holds the following appealing properties: 1) $0 \leq \xi(x_1, x_2) \leq 0.5(\text{var}(x_1) + \text{var}(x_2))$. 2) $\xi(x_1, x_2) = 0$ if and only if x_1 and x_2 are linearly dependent. 3) $\xi(x_1, x_2) = \xi(x_2, x_1)$ (symmetric). 4) It is invariant to translation of the dataset, because its expression in (19) overlooks mean expression and only incorporates the variance and covariance terms. 4) It is insensitive to rotation, because the geometric interpretation of λ_2 incorporates a property whereby the perpendicular distance of a point to a line is not dependent on the rotation of the input features.

The contribution of an input feature is thereafter pinpointed by its average similarity to other input variables. Accordingly, an input variable is deemed inconsequential provided that it has a strong similarity to other input attributes. Formally speaking, the significance of one input attribute can be defined as follows:

$$J_l = \underset{j=1,..\mu}{mean}(\xi(x_l, x_j)), l \neq j \quad (20)$$

(20) is cultivated *u-times* per input variable to monitor the mutual information contained in the input feature $[\xi(x_l, x_1),\ldots,\xi(x_l, x_u)]$. Since the sensitivity of an input feature is inversely proportional to (19), the setback of the input attribute contribution is commensurate with the decline of (19). In the high dimensional space, it is expected that the contribution of the input attributes will be low. Therefore, the feature weight $\lambda_j \in [0,1]$ is normalised in relation to the most pivotal input variable as follows:

$$\lambda_j = \frac{J_j}{\max\limits_{j=1,..\mu}(J_j)} \quad (21)$$

This input weight is exploited to craft the weighted input variable $X_n^{weight} = \lambda X_n$ to overcome the curse of dimensionality bottleneck.

4.2.2 Scaffolding Concept (Problematizing Item) : Local Forgetting Mechanism

The problematizing aspect of the scaffolding concept is intended to handle concept drift that is unsolved by the schema theory. The concept drifts resolved in the problematizing module encompass gradual, incremental and recurring types (Bose, van der Aalst, Zliobaite & Pechenizkiy, 2014), which are more complicated to manage from the adaptation viewpoint than abrupt concept drift. In the gradual drift scenario, the data regime gradually moves from one input space zone to another region with a particular intensity and speed. As discussed

in Bose, van der Aalst, Zliobaite and Pechenizkiy (2014), and Lu, Zhang and Lu (2014), new and old data concepts initially take place simultaneously, but the old data distribution slowly attenuates and in turn no longer exists in the data stream. A conventional adaptation scheme without a specific forgetting scheme is incapable of pursuing changing data distributions, thus resulting in a downward trend in the classifier's accuracy. In light of this issue, the forgetting scheme assists in fading the old belief of the system, while emphasizing new knowledge. Some attempts have been made to troubleshoot the gradual concept drift by means of the global forgetting scheme, levelling the forgetting power to the whole feature space (Lughofer & Angelov, 2011c; Angelov, 2011). Nevertheless, the major shortfall of the global forgetting procedure is the existence of an unnecessary cluster evolution, because the concept drift often prevails in some local regions but with different degrees of severity. To this end, the so-called local drift management technology was proposed in Shaker and Lughofer (2014), in which each local region is assigned with a unique forgetting degree by exploiting the extended version of the Page-Hinkley test. In Pratama et al (2014c), local data quality estimation was put forward. However, the central deficiency of this scheme is that changing data distribution in the feature space does not always mean the changing concept of the target concept. In this paper, the drift intensity is computed by the rate of the local system error, which is defined as follows:

$$\bar{e}_{N_i,i} = \frac{\sum_{n=1}^{N_i} e_{n,i}}{N_i} = \frac{N_i - 1}{N_i}\bar{e}_{N_i-1} + \frac{1}{N_i}e_{N_i},\ \bar{\sigma}_{N_i,i}{}^2 = \frac{N_i - 1}{N_i}\bar{\sigma}_{N_i,-1}{}^2 + \frac{1}{N_i}(e_{N_i} - \bar{e}_{N_i-1}) \quad (22)$$

where $\bar{e}_{N_i,i}$ denotes the mean local error of the *i-th* rule and $\bar{\sigma}_{N_i,i}$ stands for the standard deviation of the *i-th* local error, whereas N_i expresses the number of supports of the *i-th* cluster. Note that the *i-th* local error can be elicited as the deviation of the true class label and the local output produced by the *i-th* category, because we adopt the local learning concept. The rate of change of the local system error can be formulated as $\Delta\tau = \bar{e}_{N_i,i} + \bar{\sigma}_{N_i,i}{}^2 - (\bar{e}_{N_i-1,i} + \bar{\sigma}_{N_i-1,i}{}^2)$, which can be regarded as the intensity of the local drift. To guarantee stability, the local forgetting degree should hover around $\zeta_i \in [0.9,1]$, where the substantial forgetting degree is given by $\zeta_i = 0.9$, whereas the minuscule forgetting level is given by $\zeta_i = 1$. To assure $\zeta_i \in [0.9,1]$, the following setting is explored:

$$\zeta_i = \min(\max(1 - 0.1\Delta\tau_i, 0.9), 1),\ N_i = N_i - N_i \min(\zeta_{trans}{}^i, 0.99),\ \zeta_{trans}{}^i = -9.9\Delta\tau_i + 9.9 \quad (23)$$

The drift handling mechanism is applied in both the input and output concepts to produce more intense parameter adaptation, which counterbalances concept drift. The concept drift in the feature space is compensated for by a reduction in its population, included in the rule premise adaptation in (38)-(40), while the concept drift in the target space is compensated for by the local forgetting factor ζ_i, explicitly contained in the FWGRLS method (passive scaffolding step) in Section 4.2.8. In addition, a cluster ought to hold at least 30 samples to activate the local forgetting scenario to deter the unlearning effect. That is, the unlearning effect refers to the complete or too dramatic forgetting of previously learned knowledge (Lughofer, 2006).

4.2.3 Schema Theory

- *Measuring Conflict*

The foremost step in the schema learning theory is to perform a knowledge exploratory mechanism, which is instrumental in enacting appropriate learning scenarios for data streams according to the amount of conflict caused by the datum. In the ST2Class learning engine, two conflict cursors, namely Type-2 Data Quality

(T2DQ) and Type-2 Data significance (T2DS), are coupled, and are extended from their type-1 variant in Pratama et al (2014c). In principle, the T2DS method is potent for estimating the statistical contribution of a datum, which also hints at the forthcoming contribution of a datum. Suppose that an incoming datum acts as a hypothetical cluster, its statistical contribution can be described as follows:

$$DS_n = \frac{1}{2S(X)}\left(\int_X \exp(-(X_n^{weight} - C_{P+1})\overline{\Sigma}_{p+1}^{-1}(X_n^{weight} - C_{p+1})^T + \exp(-(X_n^{weight} - C_{p+1})\underline{\Sigma}_{p+1}^{-1}(X_n^{weight} - C_{p+1})^T)\right)dx \quad (24)$$

where $S(X)$ denotes a size of the range *X*, which can be simply acquired from $S(X) = \int_X 1dx$. Note that the underlying assumption of this statistical contribution formula is that the streaming data are uniformly distributed with the sampling density function $p(x) = \frac{1}{S(x)}$. $C_{p+1}, \tilde{\Sigma}_{p+1}^{-1} = [\underline{\Sigma}_{P+1}^{-1}, \overline{\Sigma}_{P+1}^{-1}]$ labels the centroid and uncertain covariance matrix of the hypothetical new rule (P+1st) tailored from the new training pattern X_N. The hypothetical rule is created using the initialization strategy elaborated in the restructuring phase (30)-(35). This strategy aims to accommodate the likelihood of class overlapping in the rule-growing phase, leading to the accurate approximation of fuzzy rule significance. In practice, it is cumbersome set $S(X)$ before the process runs, since the true data distribution is unknown. To remedy this bottleneck, it can be substituted by the total contribution of all existing categories, because a contribution of a rule is shown with its contribution with respect to other rules. The *u*-fold numerical integration for any arbitrary density function $p(x)$ is subsequently executed, consequently arriving at the final formula of the type-2 DS method as follows:

$$DS_n = \frac{1}{2}\left(\frac{\overline{V}_{P+1}}{\sum_{i=1}^{P+1}\overline{V}_i} + \frac{\underline{V}_{P+1}}{\sum_{i=1}^{P+1}\underline{V}_i}\right), \tilde{V}_i = [\underline{V}_i, \overline{V}_i] \quad (25)$$

where $\tilde{V}_{p+1}$ stands for the volume of the hypothetical non-axis parallel ellipsoidal cluster, which can be dispatched by either the determinant measure or more reliable method in Pratama et al (2014c). In essence, a datum deserves to be a new rule when its volume is greater than the maximum volume of the existing fuzzy rules as follows:

$$(\underline{V}_{P+1} + \overline{V}_{P+1}) > \max_{i=1,..,P}(\underline{V}_i + \overline{V}_i) \quad (26)$$

One can visualise the uniform data distribution assumption of the T2DS method, which is not in line with the characteristic of real-world data streams, which may follow various and even non-smooth data distributions (Rong, Sundarajan, Huang & Zhao, 2011). Furthermore, the *u-fold* numerical integration is only reliable for a small input dimension *(u<5)* (Bortman & Aladjem, 2009). To rectify this shortcoming, the Type-2 Data Quality (T2DQ) rule-growing strategy is embedded in the ST2Class learning engine. The crux of this method is to gauge the spatial relationship between a new datum and all preceding data without maintaining all the data in the memory. This strategy is essentially equivalent a recursive density estimation. The T2DQ method is customised in such a way that the interval type-2 fuzzy rule is conceived as follows:

$$DQ_n = \frac{1}{2}\sqrt{\frac{1}{\left(1 + \frac{\sum_{n=1}^{N-1} DQ_n \sum_{j=1}^{u}(x_{n,j}^{weight} - c_j^{P+1})}{\sum_{n=1}^{N-1} DQ_n}\right)}} \quad (27)$$

This definition necessitates revisiting all previous training stimuli, thus imposing costly computational cost. Hence, we have to express its recursive form. To keep this paper compact, we do not show the complete mathematical derivation, but the interested reader is recommended to refer to Pratama et al (2014c) for clarity. We derive the recursive formula of (27) as follows:

$$DQ_N = \sqrt{\frac{U_n}{U_n(1+b_n)-2h_n+g_n}} \tag{28}$$

$$U_n = U_{n-1} + DQ_{N-1}, b_n = \sum_{j=1}^{u} (x_{N,j}{}^{weight})^2 , h_n = \sum_{j=1}^{u} x_{N,j}{}^{weight} p_n{}^{j} , p_n = p_{n-1} + DQ_{N-1} X_n{}^{weight},$$

$$g_n = g_{n-1} + DQ_{N-1} b_n$$

where these recursive parameters are initialised as zero in a priori. Note that the outlier disadvantage presents in the DQ method, as disclosed in Wang, Ji and Jin, (2013). This issue is resolved in this paper by a weighting factor DQ_{N-1}, lessening the outlier leverage in quantifying the data quality formula.

In short, the streaming data are deemed to be paramount for the evolution of the classifier, either when occupying populated regions that illustrate main data orientations, or when traversing an uncharted input space region. Formally speaking, these two circumstances are pinpointed by the following two criteria:

$$DQ_N \geq \max_{i=1,..,P}(DQ_i) \text{ or } DQ_N \leq \min_{i=1,...,P}(DQ_i) \tag{29}$$

On one hand, the condition $DQ_N \leq \min_{i=1,...,P}(DQ_i)$ paves the way for outliers to be recruited as new fuzzy rules, since the outliers are usually situated in a distant region far from the influence of existing clusters. Therefore, ST2Class should be equipped with a rule pruning mechanism, preventing outliers from being engaged in the rule base. On the other hand, the criterion $DQ_N \geq \max_{i=1,..,P}(DQ_i)$ should be circumspectly executed, since it reveals the contingency of an overlapping fuzzy region as a result of positing a new fuzzy rule in a congested feature zone. In light of this problem, a class overlapping criterion should be integrated in the initialization of the new fuzzy rule parameters to sidestep an overlapping region with inter-class clusters that might result in misclassification.

Measuring the conflict degree is consistent with the recent finding of the basic control theoretic notion in the neuroscience field, which discloses the presence of a hierarchical relationship. This module is capable of triggering other learning scenarios with respect to certain criteria (Roy, 2000).

- *Restructuring Phase*

If a training observation meets the criteria (26) and (29), the datum offers desirable novelties to the rule base. Nonetheless, this learning step should be rigorously designed to avoid the development of an overlapping region by a newly created fuzzy region. In Savitha et al (2012), the class overlapping strategy was proposed, in which the authors relied on the distance ratio principle. A majority class assumption of this method however does not coincide with the unclean cluster paradigm, where the cluster might contain supports from different classes. In Pratama et al (2014b), the so-called potential per class method is offered. However, this method has high computational and memory cost as a result of the update of recursive parameters in every training observation. The ST2Class is equipped by the Bayesian-inspired class overlapping criterion, in which the crux is to apprehend the spatial relationship between the datum and the class distribution.

The point of departure of this method is to check the spatial proximity between the datum and the winning rule to grasp whether or not the datum possesses a high risk of redundancy. It can be solved by vetting the compatibility degree of the winning cluster, where a non-trivial case is designated by $(\underline{R_{win}} + \overline{R_{win}})/2 \geq \rho_a$,

because the training pattern lies an unsafe distance from the winning rule. ρ_a itself labels a predefined constant, which can be statistically stipulated by the critical value of a χ^2 distribution with Z degrees of freedom and a significance level of α (Tabata & Kudo, 2010), termed as ${\chi_p}^2(\alpha)$. A typical value of α is 5%, and the degree of freedom is represented by the dimensionality of the learning problem, thus resulting in $Z = u$. Hence, the predefined constant ρ_a is elicited as $\rho_a = \exp(-{\chi_p}^2(\alpha))$. The Bayesian-inspired class overlapping criterion is subsequently activated to fathom the class posterior probability that reveals the best matching category for a data stream. This concept can be obtained by approximating the joint category-and-class probability $P(\hat{y}_o, R_i)$ (Vigdor & Lerner, 2007) as follows:

$$P(\tilde{y}_e|X) = \frac{1}{2}\left(\frac{\sum_{k=1}^{P} P(\hat{y}_e|R_k)P(R_k)P(X^{weight}|\overline{R}_k)}{\sum_{o=1}^{m}\sum_{i=1}^{P} P(\hat{y}_o|R_i)P(R_i)P(X^{weight}|\overline{R}_i)} + \frac{\sum_{k=1}^{P} P(\hat{y}_e|R_k)P(R_k)P(X^{weight}|\underline{R}_k)}{\sum_{o=1}^{m}\sum_{i=1}^{P} P(\hat{y}_o|R_i)P(R_i)P(X^{weight}|\underline{R}_i)}\right) \quad (30)$$

where $P(\hat{y}_o|R_i), P(R_i), P(X|\widetilde{R}_i) = [P(X|\underline{R}_i), P(X|\overline{R}_i)]$ respectively denote the conditional probability, the prior probability and the likelihood function. It is worth mentioning that the joint probability $P(\hat{y}_o|R_i)$ is achieved by the estimate for the class joint probability $P(\tilde{y}_o, R_i)$ making use of the frequency count approach (Sasu & Andonie, 2013) and the type-1 version of (30) was derived in Vigdor and Lerner (2007). The conditional probability, the prior probability and the likelihood function are defined as follows:

$$P(\hat{y}_o|R_i) = \frac{\log({N_i}^o + 1)}{\sum_{o=1}^{m}\log({N_i}^o + 1)}, \hat{P}(R_i) = \frac{\log(N_i + 1)}{\sum_{i=1}^{P}\log(N_i + 1)}, \hat{P}(X|\widetilde{R}_i) = \frac{1}{(2\pi)^{1/2}\widetilde{V}_i^{1/2}}\exp(-(X^{weight} - C_i)\widetilde{\Sigma}_i^{-1}(X^{weight} - C_i)^T) \quad (31)$$

where ${N_i}^o$ labels the number of the *i*-th cluster's population falling into the *o-th* class. It is worth noting that the number of populations is incremented given that it is selected as the winning rule (40) and no new rule is grown, whereas their class labels can be elicited as a result of the automatic or manual labelling process from the what-to-learn scenario (Section 4.1) (40). Moreover, a cluster can be populated by different-class samples, leading to ${N_i}^o$, where $N_{win} = \sum_{o=1}^{m} {N_{win}}^o$. Note that the formula of the prior and conditional probabilities is softened using the log operation from their original versions to enable a newly-born cluster to compete with older clusters in the category choice phase. In the following, the class overlapping situation possibly takes place in $\max_{o=1,..,m}(P(\tilde{y}_o|X)) \neq true\_class\_label$, since the datum is more imminent to inter-class data clouds than those of the intra-class. In this circumstance, a model update exacerbates a nonlinearity degree. Therefore, we may end up with an over-complex decision boundary, which inevitably downgrades the generalization potential. To overcome this shortcoming, the new fuzzy rule should be placed in a location where it can move away from the inter-class data clouds to reduce the likelihood of class overlapping. To this end, suppose that *ir* represents the nearest intra-class cluster, whereas *ie* denotes the nearest inter-class cluster. The new fuzzy rule is initialised as follows:

$$c_{P+1,j} = x_j - \rho_2(c_{ie,j} - {x_j}^{weight}), \overline{dist}^j = \frac{\rho_1}{N_{win}}|c_{P+1,j} - c_{ie,j}|, \underline{dist}^j = \frac{2\rho_1}{N_{win}}|c_{P+1,j} - c_{ie,j}|, \widetilde{\Sigma}_{p+1}^{-1} = (\widetilde{dist}^T\widetilde{dist})^{-1}, N_{P+1} = 1, {N_{P+1}}^o = 1 \quad (32)$$

where ρ_1 labels an overlapping parameter, steering the shrinkage of the fuzzy set spread. In this context, it is set as a distance ratio between the inter- and intra-class clusters $\rho_1 = r_{ir}{}^j / r_{ie}{}^j$. This setting is plausible to create the new fuzzy region, because the influence zone of the new fuzzy rule should diminish, given that the inter-class cluster lies in a vicinity to the new cluster to attenuate the class overlapping contingency and vice versa. $\rho_2 \in [0.01-0.1]$ indicates a problem-independent shifting factor, fixed as 0.01 in this paper. This predefined parameter is useful for orchestrating a shift of the centroid of the new cluster away from the winning inter-class cluster. Because we employ the interval type-2 Gaussian fuzzy rule with uncertain SDs, the coverage span of the cluster is modulated by the winning cluster population. The upper fuzzy rule should have a greater zone of influence by multiplying with N_{win} , whereas the lower fuzzy rule is by $N_{win}/2$.

In contrast, the new fuzzy rule is more similar to the nearest intra-class cluster, if we come across $\max_{o=1,..,m}(P(\bar{y}_o|X)) = true\_class\_label$. This condition does not generally have a detrimental impact on the classifier's generalization because there is no major modification of the decision surface in the target concept. Nonetheless, the winning intra-class rule and the new fuzzy rule can eventuate in a significantly overlapping position in the future due to the rule resonance granted to these two rules. We can guarantee that this situation essentially does not undermine the classifier's accuracy, because it can be easily sorted out by the rule merging scenario, detailed elsewhere in this paper. As a minor likelihood of misclassification is observed, more confident parameters are allocated to the new rule as follows:

$$c_{P+1,j} = x_j{}^{weight} + \rho_2(c_{ir,j} - c_{ie,j}),\ \overline{dist}^j = \frac{\rho_1}{N_{win}}\left|c_{ie,j} - c_{ir,j}\right|,\ \underline{dist}^j = \frac{2\rho_1}{N_{win}}\left|c_{ie,j} - c_{ir,j}\right|\ \tilde{\Sigma}_{p+1}{}^{-1} = (\tilde{dist}^T \tilde{dist})^{-1},\ N_{P+1} = 1, N_{P+1}{}^o = 1 \quad (33)$$

Note that although the rule-growing process on this occasion does not harm the stability of the decision boundary in the target space, we still need to shift the centroid of the new cluster away the gap between the intra-class cluster and the inter-class cluster to circumvent the rule redundancy. If the new fuzzy rule occupies a remote area from the existing fuzzy rules indicated by $(\underline{R_{win}} + \overline{R_{win}})/2 < \rho_a$, the new fuzzy rule is crafted as follows:

$$c_{P+1,j} = x_j{}^{weight},\ \overline{dist}^j = \frac{\rho_1}{N_{win}}\left|x_j{}^{weight} - c_{ir,j}\right|,\ \underline{dist}^j = \frac{2\rho_1}{N_{win}}\left|x_j - c_{ir,j}\right|\ \tilde{\Sigma}_{p+1}{}^{-1} = (\tilde{dist}^T \tilde{dist})^{-1},\ N_{P+1} = 1, N_{P+1}{}^o = 1 \quad (34)$$

Clearly, this situation merely results in a subtle leverage of the evolution of other clusters, because the new cluster spotlights a remote fuzzy region which is uncorrelated with existing input partitions.

The new rule consequent and the new output covariance matrix are assigned as follows:

$$\tilde{\Omega}_{P+1} = \tilde{\Omega}_{win}\ \tilde{\gamma}_{P+1} = \omega \boldsymbol{I}\ ,\tilde{\Omega}_{P+1} = [\underline{\Omega}_{P+1}, \overline{\Omega}_{P+1}], \tilde{\gamma}_{P+1} = [\underline{\gamma}_{P+1}, \overline{\gamma}_{P+1}] \quad (35)$$

where ω indicates a large positive constant which is fixed as $\omega = 10^5$ and $\tilde{\gamma}_{P+1} = [\underline{\gamma}_{P+1}, \overline{\gamma}_{P+1}] \in \Re^{u \times u}$ stands for the new output covariance matrix. The new rule consequent is assigned in the same way as the winning rule to expedite the convergence time, because the rule consequent of the winning rule is expected to portray a pertinent data trend with the new rule. Note that when a data stream occupies a remote region from the coverage span of existing rules, such a sample is likely to have little influence on the convergence of existing rules. Furthermore, the adaptation of the rule output is endowed with the local forgetting factor ζ_i , which captures this type of situation. Meanwhile, the output covariance matrix is arranged as a large positive definite matrix which is capable of swiftly emulating a real solution as attained by the batched learning scheme (Lughofer, 2011b).

The upper and lower output covariance matrix of a new rule is initially fixed using the same setting. An interval-valued output covariance matrix is produced after undergoing rule adaptation with the FWGRLS method in Section 4.2.8. As with the class overlapping criterion, the winning rule is chosen in accordance with the Bayesian concept, in which the rule possessing a maximum posterior probability is chosen as the winning rule $win = \arg\max_{i=1,\ldots,P} \widehat{P}(R_i \left| X^{weight} \right.)$. The key trait of the Bayesian concept in seeking the winning rule is its prior probability, which investigates the true winning rule from the probabilistic standpoint. This method is worthwhile when some clusters are situated in on par proximity to the focal point of interest. For brevity, the posterior probability $\hat{P}(R_i \left| X \right.)$ is expressed as follows:

$$\widehat{P}(R_k \left| X^{weight} \right.) = \frac{1}{2}\left(\frac{\hat{p}(X^{weight} \left| \overline{R}_k \right.)\hat{P}(R_k)}{\sum_{i=1}^{P} \hat{p}(X^{weight} \left| \overline{R}_i \right.)\hat{P}(R_i)} + \frac{\hat{p}(X^{weight} \left| \underline{R}_k \right.)\hat{P}(R_k)}{\sum_{i=1}^{P} \hat{p}(X^{weight} \left| \underline{R}_i \right.)\hat{P}(R_i)}\right) \tag{36}$$

where $P(X^{weight} \left| \widetilde{R}_i \right.) = [P(X^{weight} \left| \overline{R}_i \right.), P(X^{weight} \left| \underline{R}_i \right.)], P(R_i)$ can be found by (30) as well. In the realm of scaffolding theory, the restructuring phase of the schemata construction is also coincident with the problematizing facet of the active scaffolding theory, since the conflict measure, enumerated by the T2DS and T2DQ methods, can be regarded the abrupt concept drift detection. In the Bayesian sense, the concept drift ensues on condition that the class posterior probability changes over time $P_{N+1}(\hat{y}_o \left| X^{weight} \right.) \neq P_N(\hat{y}_o \left| X^{weight} \right.)$, induced by the non-stationary component of the prior probability $P(\hat{y}_o)$ or the class conditional probability $P(X^{weight} \left| \hat{y}_o \right.)$ (Zhu et al, 2010; Elwell & Polikar, 2011). The restructuring phase in schema theory literally portrays the conditional probability concept drift, where the data distribution suddenly shifts from previously seen location, or an abrupt variation of the class conditional probability occurs.

- *Tuning Scenario*

The tuning phase of schema theory delineates a circumstance in which the training sample imposes a marginal conflict with the current knowledge base, thus soliciting a slightly more fine-grained rule premise of the winning cluster. This goal can be achieved by simply fine-tuning the rule premise of the winning rule as follows:

$$\min_{i=1,..,P}(DQ_i) < DQ_{P+1} < \max_{i=1,..,P}(DQ_i) \text{ and } (\overline{V}_{P+1} + \underline{V}_{P+1}) > \max_{i=1,..,P}(\overline{V}_i + \underline{V}_i) \tag{37}$$

It can be seen that on this occasion, the datum does not bring substantial novelty to the existing rule base, since the datum is not located in the populated fuzzy region, nor does it convey novel knowledge. Even so, it does make a considerable statistical contribution, which is sufficient to trigger a rule antecedent learning process. Intrinsically, the winning rule can be adjusted as follows:

$$C_{win}^{N} = \frac{N_{win}^{N-1}}{N_{win}^{N-1}+1} C_{win}^{N-1} + \frac{(X_N^{weight} - C_{win}^{N-1})}{N_{win}^{N-1}+1} \tag{38}$$

$$\widetilde{\Sigma}_{win}(N)^{-1} = \frac{\widetilde{\Sigma}_{win}(N-1)^{-1}}{1-\alpha} + \frac{\alpha}{1-\alpha}\frac{(\widetilde{\Sigma}_{win}(N-1)^{-1}(X_N^{weight} - C_{win}^{N-1}))(\widetilde{\Sigma}_{win}(N-1)^{-1}(X_N^{weight} - C_{win}^{N-1}))^T}{1+\alpha(X_N^{weight} - C_{win}^{N-1})\widetilde{\Sigma}_{win}(old)^{-1}(X_N^{weight} - C_{win}^{N-1})^T}, \widetilde{\Sigma}_{win}^{-1} = [\underline{\Sigma}_{win}^{-1}, \overline{\Sigma}_{win}^{-1}] \tag{39}$$

$$N_{win}^{N} = N_{win}^{N-1} + 1, N_{win}^{o} = N_{win}^{o} + 1, N_{win}^{N} = \sum_{o=1}^{m} N_{win}^{o} \tag{40}$$

where $\alpha = 1/(N_{win}{}^{N-1}+1)$. In principle, this adaptation mechanism is motivated by the sequential maximum likelihood method for a spherical cluster, and is specifically customised to gear the requirement of the non-axis parallel ellipsoidal cluster adjustment. Moreover, (39) is modified to sustain a direct update of the uncertain non-diagonal inverse covariance matrixes without the re-inversion phase to avoid numerical instability when dealing with an ill-posed matrix. Aside from that, it is laborious to carry out the re-inversion scenario in the case of the high dimensionality problem.

The accretion phase of schema theory is not recounted in this section, because it virtually deciphers a negligible conflict degree or the datum is superfluous to the training process. This circumstance is reflected by the what-to-learn scenario, when the training stimuli are ruled out from the training process.

4.2.4 Scaffolding Concept (Fading Item) : Rule Pruning Mechanism

The underlying aim of the fading facet in the scaffolding context is to avoid the redundancy problem, which can compromise the compactness and parsimony of the rule base. The fuzzy rule is superfluous to the resultant system output when it either contributes marginally to the learning progress or shares strong mutual information with another rule in the rule base. This fuzzy rule can be inevitably pruned to mitigate the rule base complexity without major loss of the classifier's accuracy, while expediting the training process.

The fading aspect of the active scaffolding is embodied by putting the rule pruning strategy, termed the Type-2 Extended Rule Significance (T2ERS) concept, into perspective. This strategy is developed from its type-1 version in Pratama et al (2014b) to serve the interval type-2 fuzzy system. That is, the T2ERS method aims to oversee the statistical contribution of the fuzzy rule, which indicates a possible future contribution of the fuzzy rule. The fuzzy rules which are captured by the T2ERS method are deemed to be inactive during their lifespan, such that they can be dispossessed from the rule base without compromising learning stability. For brevity, the statistical contribution of the fuzzy rule can be seen in an identical way of the T2DQ method as follows:

$$ERS_i{}^n = \frac{1}{2}|\delta_i|E_i{}^n \tag{41}$$

$$\delta_i = \sum_{j=1}^{u}\sum_{o=1}^{m}\overline{\Omega}_i{}^{o,j} + \underline{\Omega}_i{}^{o,j},\ E_i{}^n = \frac{1}{S(X)}\left(\int_X \exp(-(X_n{}^{weight}-C_i)\overline{\Sigma}_i{}^{-1}(X_n{}^{weight}-C_i)^T + \exp(-(X_n{}^{weight}-C_i)\underline{\Sigma}_i{}^{-1}(X_n{}^{weight}-C_i)^T))dx\right.$$

where $C_i, \tilde{\Sigma}_i{}^{-1} = [\underline{\Sigma}_i{}^{-1}, \overline{\Sigma}_i{}^{-1}]$ indicate the centroid and covariance matrix of the *i-th* fuzzy rule. By applying *u*-fold numerical integration, the final expression of the T2ERS method can be derived as follows:

$$ERS_i{}^n = \frac{1}{2}\left|\sum_{j=1}^{u}\sum_{o=1}^{m}\overline{\Omega}_i{}^{o,j} + \underline{\Omega}_i{}^{o,j}\right|\left(\frac{\overline{V}_{i,n}}{\sum_{i=1}^{p}\overline{V}_{i,n}} + \frac{\underline{V}_{i,n}}{\sum_{i=1}^{p}\underline{V}_{i,n}}\right) \tag{42}$$

The fuzzy rule is pruned if the following condition is met as follows:

$$ERS_i{}^n \le mean(ERS_i{}^n) - std(ERS_i{}^n)\quad mean(ERS_i{}^n) = \frac{\sum_{i=1}^{P} ERS_i{}^n}{P},\ std(ERS_i^n) = \sqrt{\frac{\sum_{i=1}^{P}(ERS_i{}^n - mean(ERS_i{}^n))^2}{P-1}} \tag{43}$$

Apart from being endowed with a solid statistical foundation, the T2ERS method analyses the significance of the local sub-model contribution δ_i, in which the fuzzy rule contribution looms as the magnitude of the local sub-model.

4.2.5 Scaffolding Concept (Problematizing Item) : Rule Forgetting Mechanism

We first elaborate the problematizing item of the scaffolding concept by considering an out-dated fuzzy rule, which is no longer relevant for capturing up-to-date data distribution due to the regime-drifting or shifting

property. This rule should be deactivated to reduce computational complexity. Nevertheless, this rule is subject to the future rule recall mechanism to surmount the recurring concept drift. To this end, the so-called T2P+ method is mounted in the learning engine, where the crux is to appraise the cluster evolution in connection with other training samples in the feature space. The T2P+ method is written as follows:

$$\chi_i^n = \sqrt{\frac{1}{1+\sum_{n=1}^{N}\sum_{j=1}^{u}\frac{(x_{i,j}^{weight,n}-c_i j)^2}{(N-1)}}} \tag{44}$$

This formula needs revisiting all previously seen data streams and is thus not in line with the spirit of incremental learning. Therefore, we need to derive the recursive formula as done in Pratama et al (2014c). Because of the identical centroid between the upper and lower fuzzy rules $\bar{c}_{i,j} = \underline{c}_{i,j}$, we arrive at the final expression of the T2P+ method as follows:

$$\chi_i^N = \sqrt{\frac{(N-1)\chi_{N-1,i}^2}{2\chi_{N-1,i}^2 + 2\chi_{N-1,i}^2\sum_{j=1}^{u}(x_{i,j}^{weight,N-1}-c_{i,j})^2+(N-2)}} \tag{45}$$

Use of the T2P+ method is seemingly appropriate for probing the obsolete fuzzy rules, because it quantifies a cluster's density. One cluster is deemed to be out-dated if its presence is no longer able to apprehend the recent data trend. This hints at a changing data pattern, moving away from the cluster influence zone. It is worth noting that the T2P+ method is distinguishable from its predecessor in Pratama et al (2014b), because it is specifically devised to accommodate the multivariable interval type-2 fuzzy rule. Furthermore, the T2P+ method is driven by the Euclidean distance in lieu of Mahalanobis distance to simplify the mathematical derivation. The fuzzy rule is deactivated when the following condition is encountered.

$$\chi_i^N \leq mean(\chi_i^N) - std(\chi_i^N),\ mean(\chi_i^N) = \frac{\sum_{i=1}^{P}\chi_i^N}{P},\ std(\chi_i^N) = \sqrt{\frac{\sum_{i=1}^{P}(\chi_i^N - mean(\chi_i^N))^2}{P-1}} \tag{46}$$

Cyclic concept drift may occur in the data streams, which consequently drives the previously pruned fuzzy rules to become valid again. In the realm of a truly evolving and adaptive system, appending a completely new fuzzy rule without the ability to trigger the rule recall mechanism, or retrieving the fuzzy rules discarded in the earlier training episodes, undermines the remembering facet of the human being (Bartlett, 1932). Formally speaking, the fuzzy rules dispossessed by (25) are not permanently deprived and are able to be reignited in the future provided that their potential, designated by (25), is highly entailed by the model as follows:

$$\max_{i*=1,..,P*}(\chi_{i*}) > \max_{i=1,..,P+1}(DQ_i) \tag{47}$$

where P^* indicates the fuzzy rules deactivated by the T2P+ method thus far. It is worth emphasizing that the fuzzy rule is curbed to merely cultivate (45) without being engaged in a learning scenario. This strategy guarantees that the computational burden is absolutely decimated. Should the fuzzy rule comply (47), the old fuzzy rule is revived as follows:

$$C_{P+1} = C_{i*}, \tilde{\Sigma}_{P+1}^{-1} = \tilde{\Sigma}_{i*}^{-1}, \tilde{\gamma}_{P+1} = \tilde{\gamma}_{i*}, \tilde{\Omega}_{P+1} = \tilde{\Omega}_{i*} \tag{48}$$

4.2.6 Scaffolding Concept (Fading Item) : Rule Merging Mechanism

Another fading scenario is provided by the rule merging scenario, because the fuzzy rule is definitely inconsequential when it has a strong similarity to another fuzzy rule in the rule base. The rule merging procedure is amalgamated in the ST2Class learning engine to coalesce identical fuzzy rules, thereby easing the

complexity while affirming the rule semantics. The notion of the rule merging scenario in the evolving interval type-2 fuzzy system was pioneered in Juang and Chen (2013) and Tung, Quek and Guan (2013), in which they either rely on the shape-based similarity measure or the distance-based similarity check without solving these two issues together in a joint formula. In addition, these works overlook the so-called blow up effect, in which the inexact representation of the merged cluster is incurred as a consequence of fusing non-homogenous clusters. The merged cluster risks the cluster delamination situation because it has a large coverage span. Note that this issue is more obvious in the non-axis-parallel ellipsoidal cluster type, since it creates the opportunity for a cluster to arbitrarily rotate in any direction. To correct these problems, the rule merging scenario of ST2Class is devised in respect to the two problems. The first issue is deciphered by employing the vector similarity measure concept (Wu & Mendel, 2008; 2009), which encompasses both the shape and distance of two clusters to determine their similarity. The volume of the merged cluster is also checked to cope with the second issue.

The crux of the vector similarity measure is to conceive the similarity of the cluster based on the shape and distance of two clusters as follows:

$$s_{v,j}(win,i) = s_{1,j}(win,i) \times s_{2,j}(win,i) \tag{49}$$

where $s_{1,j}(win,i) \in [0,1]$ denotes the shape-based similarity measure and $s_{2,j}(win,i) \in [0,1]$ labels the distance-based similarity measure, whereas the resultant similarity check is elicited by the min *t-norm* operator. The peculiar property of the vector similarity measure can be seen in its alignment procedure in quantifying shape-based similarity, since the distance-based similarity measure is accomplished by a separate similarity measure. That is, one or both $\widetilde{win}$ and $\widetilde{i}$ are aligned, such that $c_{win,j}$ and $c_{i,j}$ coincide $c_{win,j} = c_{i,j}$. Furthermore, the shape-based similarity measure is undertaken by the extended Jaccard similarity measure, exploiting the average cardinality principle to quantify the overall similarity of the upper and lower fuzzy sets.

$$s_{1,j}(win,i) = \frac{M(\underline{\mu}_{win,j} \cap \underline{\mu}_{i,j}) + M(\overline{\mu}_{win,j} \cap \overline{\mu}_{i,j})}{M(\underline{\mu}_{win,j} \cup \underline{\mu}_{i,j}) + M(\overline{\mu}_{win,j} \cup \overline{\mu}_{i,j})} \tag{50}$$

where $\cap and \cup$ indicate the intersection and union of two fuzzy sets $\tilde{\mu}_{win}, \tilde{\mu}_i$. The union of two fuzzy sets can be arranged as follows:

$$M(\underline{\mu}_{win,j} \cup \underline{\mu}_{i,j}) = M(\underline{\mu}_{win,j}) + M(\underline{\mu}_{i,j}) - M(\underline{\mu}_{win,j} \cap \underline{\mu}_{i,j}) \tag{51}$$

where $M(.)$ stands for the area of the fuzzy set and (51) prevails for the upper fuzzy set as well $M(\overline{\mu}_{win} \cup \overline{\mu}_i)$. Note that the computation of the area of the Gaussian function is highly complex, because it features a highly nonlinear contour. We gauge the size of the Gaussian function utilizing the triangular membership function to resemble the Gaussian function, as exemplified in Juang and Lin (1998; Chao et al (1996), as follows:

$$M(\underline{\mu}_{win,j}) = \int_{-\infty}^{\infty} \exp(-\frac{(x-c)^2}{2\sigma^2}) dx = \sigma_{win,j}{}^1\sqrt{2\pi} \tag{52}$$

We can further simplify the similarity measure $M(\underline{\mu}_{win,j} \cap \underline{\mu}_{i,j})$ in Juang and Lin (1998) and Chao et al (1996) due to the alignment procedure as follows:

$$M(\underline{\mu}_{win,j} \cap \underline{\mu}_{i,j}) = \frac{h^2}{2} + \frac{h^2((\underline{\sigma}_{win,j} - \underline{\sigma}_{i,j}))}{2(\underline{\sigma}_{i,j} - \underline{\sigma}_{win,j})} - \frac{h^2(\underline{\sigma}_{win,j} + \underline{\sigma}_{i,j})}{2(\underline{\sigma}_{win,j} - \underline{\sigma}_{i,j})} \tag{53}$$

where $h = \max[0, x]$. Performing equivalent subsequent operations for $M(\overline{\mu_{win,j}} \cap \overline{\mu_{i,j}})$ in (53) and $M(\overline{\mu_{win,j}} \cup \overline{\mu_{i,j}})$ in (51), we can generate the shape-based similarity measure in (50).

An extended kernel-based metric approach, which was devised to cater for the type-1 fuzzy set in Lughofer, Bouchot and Shaker (2011(d) and Pratama et al (2013), is explored to analyse the proximity-based similarity $s_{2,j}(win,i) \in [0,1]$. In principle, we amend the traditional kernel-based metric method with the use of the average cardinality principle to assess the similarity of the interval type-2 fuzzy set. This mechanism represents shape-based similarity component of (49). The extended kernel-based metric method is defined as follows:

$$S_{2,j}(win,i) = \frac{\exp(-A) + \exp(-B)}{2} \tag{54}$$

$$A = \left| c_{win,j}{}^{1} - c_{i,j}{}^{1} \right| - \left| \overline{\sigma}_{win,j} - \overline{\sigma}_{i,j} \right| \quad B = \left| c_{win,j}{}^{2} - c_{i,j}{}^{2} \right| - \left| \underline{\sigma}_{win,j} - \underline{\sigma}_{i,j} \right|$$

The reliability of the kernel-based method is borne out by its appealing properties as follows:

$$S_{2,j}(A,B) = 1 \Leftrightarrow |C_A - C_B| + |\sigma_A - \sigma_B| = 0 \Leftrightarrow C_A = C_B \wedge \sigma_B = \sigma_B, S_{2,j}(A,B) < \varepsilon \Leftrightarrow |C_A - C_B| > \delta \vee |\sigma_A - \sigma_B| > \delta \tag{55}$$

Executing (54) for each input feature $s_{2,j}(win,i)$ outputs the distance-based similarity measure of the fuzzy rule. Henceforth, this result is consolidated with the shape-based similarity measure $s_{1,j}(win,i)$ to commit (49), which produces the vector similarity measure. The vector similarity for each input dimension is combined with the min *t-norm* operator to conclude the resultant similarity of two rules. Two fuzzy rules are deemed to be similar if the following condition is satisfied.

$$S_v \geq \rho_3, S_v = \min_{j=1,..,u}(s_{v,j}) \tag{55}$$

where ρ_3 denotes the similarity threshold, which steers the merging frequency. It is stipulated at 0.8 in all our simulations. In principle, the higher the value assigned to this parameter, the more complex the fuzzy rule base will be. In contrast, the smaller the value assigned to this parameter, the more frugal fuzzy rule base will be. The sensitivity of this parameter was examined in Pratama et al (2014e).

The blow-up effect is intrinsically attributed by merging clusters of different orientation, where an oversised cluster can be induced by the merging process. Merging such clusters is undesirable and is counterproductive to the classifier's generalization owing to the strong contingency of the cluster delamination, even though the two clusters signify a close relationship. Hence, the volume of the merged cluster is assessed by comparison with the total volume of standalone clusters. The merging process is cancelled when the volume of merged clusters surpasses the maximum allowable volume as follows:

$$\overline{V}_{merged} + \underline{V}_{merged} \leq u((\overline{V}_{win} + \overline{V}_i) + (\underline{V}_{win} + \underline{V}_i)) \tag{56}$$

It is worth emphasizing that we include the input dimension *u* to stave off the calculation bias resulting from the curse of dimensionality issue. The rule merging strategy is activated provided that (55) and (56) are fulfilled, in which the merging scenario is carried out as follows:

$$C_{merged}{}^{new} = \frac{C_{win}{}^{old} N_{win}{}^{old} + C_i{}^{old} N_i{}^{old}}{N_{win}{}^{old} + N_i{}^{old}} \tag{57}$$

$$\tilde{\Sigma}^{-1}{}_{merged}{}^{new} = \frac{\tilde{\Sigma}^{-1}{}_{win}{}^{old} N_{win}{}^{old} + \tilde{\Sigma}^{-1}{}_{i}{}^{old} N_i{}^{old}}{N_{win}{}^{old} + N_i{}^{old}}, \ \tilde{\Sigma}_i{}^{-1} = [\underline{\Sigma}_i{}^{-1}, \overline{\Sigma}_i{}^{-1}] \tag{58}$$

$$N_{merged}{}^{new} = N_{win}{}^{old} + N_i{}^{old}, \ N_{merged,o}{}^{new} = N_{win,o}{}^{old} + N_{i,o}{}^{old} \tag{59}$$

$$\tilde{\Omega}_{merged}^{new} = \frac{\tilde{\Omega}_{win}^{old} N_{win}^{old} + \tilde{\Omega}_i^{old} N_i^{old}}{N_{win}^{old} + N_i^{old}} \ , \ \tilde{\Omega}_i = [\overline{\Omega}_i, \underline{\Omega}_i] \tag{60}$$

It can be observed that our rule merging scenario is motivated by the weighted average strategy, where each rule is weighted by its accumulated supports. The cluster occupying a more populated area should impact more on the eventual shape and orientation of the merged cluster. This facet is noteworthy, since the merged cluster should be able to represent an underlying data distribution and cover the majority of both cluster populations, unless a loss of cluster support is imposed. We simply take into account the similarity measure of the winning cluster to relieve the computational burden, because this cluster is the most susceptible to overlap with other clusters owing to the rule premise adaptation in (38)-(40). Albeit scattered properly in the initialization phase, the fuzzy rule will be redundant in the next training episodes, because the streaming data may fill the gap between two neighbouring clusters.

4.2.7 Scaffolding Concept (Problematizing Item) : Rule Splitting Mechanism

One can envision that the incremental concept drift reveals a case in which the streaming data scattered in one local region grows incrementally, causing the cluster to uncontrollably expand its zone of influence. This is confirmed by the likelihood of concept drift in the variance of the data clouds (Shaker & Lughofer, 2014), which leads to cluster delamination. That is, the cluster delamination can jeopardise the classifier's generalization, because a cluster covers two or more distinct data clouds due to an over-sized fuzzy region. To remedy this technical flaw, a cluster-splitting technology is assembled here which essentially monitors the cluster's volume to hedge the cluster delamination. If this cluster is oversized, it is halved into two dissimilar clusters as follows:

$$(\overline{V}_{win} + \underline{V}_{win}) \geq \rho_1 (\sum_{i=1}^{P} \overline{V}_i + \underline{V}_i) \tag{61}$$

where $\rho_1 = [0.5, 0.9]$ stands for the splitting threshold, governing the frequency of the splitting mechanism. We scrutinise only the volume of the winning rule because it receives the rule resonance. The rule resonance is the primary rationale for the widening of the cluster's spread, which eventually ends up with the cluster delamination situation. If (61) is satisfied, the fuzzy rule is split into two distinct clusters as follows:

$$C_{win} = C_{win} + fac \ , C_{P+1} = C_{win} - fac \ , \ fac = \frac{1}{2}(\overline{g}_{\max}\sqrt{\overline{\alpha}_{\max}} + (q\sqrt{\overline{\alpha}_{\max}})I^{1\times u}) + (\underline{g}_{\max}\sqrt{\underline{\alpha}_{\max}} + (q\sqrt{\underline{\alpha}_{\max}})I^{1\times u}) \tag{62}$$

$$\tilde{\Sigma}^{-1}{}_{win} = \tilde{\Sigma}^{-1}{}_{P+1} = \tilde{\Sigma}^{-1}{}_{win} - ((\tilde{g}_{\max}\sqrt{\tilde{\alpha}_{\max}})^T (\tilde{g}_{\max}\sqrt{\tilde{\alpha}_{\max}}))^{-1} \tag{63}$$

$$\tilde{\Omega}_{P+1} = \tilde{\Omega}_{win}, \tilde{\gamma}_{P+1} = \tilde{\gamma}_{win}, \ N_{win} = N_{P+1} = N_{win}/2 \tag{64}$$

where $\tilde{\alpha}_{\max} = [\underline{\alpha}_{\max}, \overline{\alpha}_{\max}]$ denotes the largest eigenvalue of the non-diagonal covariance matrix $\tilde{\Sigma}_{win}^{-1} = [\underline{\Sigma}_{win}^{-1}, \overline{\Sigma}_{win}^{-1}]$ and $\tilde{g}_{\max} = [\underline{g}_{\max}, \overline{g}_{\max}]$ stands for its corresponding eigenvector. q is a splitting factor that governs the proximity of two clusters after a rule splitting scenario, and is fixed at *q=0.125*. Because we deal with the arbitrarily rotated ellipsoidal cluster, its largest eigenvalue intrinsically abstracts its underlying cluster expansion. Accordingly, the term $\tilde{g}_{\max}\sqrt{\tilde{\alpha}_{\max}}$ is intended to capture most cluster supports, which mostly lies in the underlying cluster orientation. By extension, $q\sqrt{\tilde{\alpha}_{\max}}$ is useful for ensuring that two clusters will be adequately far apart from each other to forestall the re-merging of these clusters by the rule merging item immediately after the cluster-splitting module has been executed. The output parameters of the new cluster are allocated in the same way as those of the winning cluster, because this cluster should indicate the pertinent data

trend, although it will differ after experiencing future rule adaptation. Fig.3 depicts the various concept drifts discussed in this paper.

4.2.8 Passive Scaffolding Concept : Adaptation of Rule Consequent Mechanism

This section articulates the passive supervision of the scaffolding theory, which is reliant on the predictive quality of the classifier. The underlying notion is to adapt the model in accordance with the classifier's error, thereby performing a sort of action-consequence mechanism. First, this learning facet is developed by the adaptation of the rule consequent by virtue of the Fuzzily Weighted Generalized Recursive Least Square (FWGRLS) method (Pratama et al, 2014b), which forms a local learning version of the GRLS method (Xu, Wong & Leung, 2006). The merit of this method over the classical RLS method is in its explicit weight decay term, which is capable of retaining the weight vector in a small bounded interval. Consequently, a higher classifier generalization can be attained while at the same time achieving a more compact and parsimonious rule base. Since the local sub-system fluctuates in the small values, the inactive fuzzy rules, carrying minor weight vectors, can be easily discovered by (42). The advantage of this approach over the omnipresent adaptation mechanism for the interval type-2 fuzzy system as presented in Juang and Tsao (2008) is its local learning perspective, which offers a more attractive and interpretable rule semantic. The local learning approach also has immense flexibility for the evolution of learning, because the rule-growing, pruning and learning mechanisms of a particular rule only have a subtle effect on the convergence and stability of other rules. In the FWGRLS method, each rule is adjusted separately, and the same strategy is also applicable for the upper and lower fuzzy rules. This method is defined as follows:

$$\tilde{\psi}(n) = \tilde{\gamma}_i(n-1)F(n)\left(\frac{\zeta_i \Delta(n)}{\tilde{R}_i(n)} + F(n)\tilde{\gamma}_i(n-1)F^T(n)\right)^{-1} \tag{65}$$

$$\tilde{\gamma}_i(n) = \tilde{\gamma}_i(n-1) - \tilde{\psi}(n)F(n)\tilde{\gamma}_i(n-1) \tag{66}$$

$$\tilde{\Omega}_i(n) = \tilde{\Omega}_i(n-1) - \varpi\tilde{\gamma}_i(n)\nabla\kappa(\tilde{\Omega}_i(n-1)) + \tilde{\gamma}_i(n)(t(n) - \tilde{y}_i(n)) \tag{67}$$

$$\tilde{y}_i^o = \phi_i(X_N)\tilde{\Omega}_{i,o} \text{ and } F(n) = \frac{\partial \tilde{y}(n)}{\partial \tilde{\Omega}_i(n)} = \phi_i(X_N) \tag{68}$$

where $\tilde{R}_i = [\underline{R_i}, \overline{R_i}]$ denotes the firing strength of *i-th* fuzzy rule and ζ_i expresses the local forgetting factor, obtained by the local drift handling mechanism (23), whereas $\tilde{\psi}(n) = [\underline{\psi}(n), \overline{\psi}(n)]$ labels the Kalman gain and $\tilde{\gamma}_i(n) = [\underline{\gamma_i}(n), \overline{\gamma_i}(n)]$ stands for the output covariance matrix. Meanwhile, $\Delta(n) \in \Re^{u \times u}$ and $\nabla\kappa(\tilde{\Omega}_i(n-1))$, respectively, illustrate the covariance matrix of the modelling error and the gradient of the weight decay function. The covariance matrix of the modelling error is managed as a Hessian matrix for the sake of simplicity, while the gradient of the weight decay function can be determined as any nonlinear function, which may not be differentiable. Hence, we approximate the gradient to the *n-1* time step whenever it is laborious to obtain the gradient solution. The quadratic weight decay function $\kappa(\tilde{\Omega}_i(n-1)) = \frac{1}{2}(\tilde{\Omega}_i(n-1))^2$ is chosen, because it is capable of proportionally shrinking the weight vector to its current value, whereas $\varpi \approx 10^{-15}$ indicates a case-insensitive predefined constant.

The passive supervision of the scaffolding theory is also actualised by the adaptation mechanism of the design coefficients $[q_l^{i,o}, q_r^{i,o}]$ and of the translation and dilation parameters of the wavelet function $[a_{i,j}, b_{i,j}]$, in which it is driven by the Zero-Error Density Maximization (Z-EDM) principle (Silva, Alexandrem & Mardues de Sa, 2005). This adjustment mechanism is deemed to be more efficacious than the conventional

Mean Square Error (MSE)-based method because it is capable of minimizing the error entropy. This perspective leads to the suppression of the proximity between the probability distribution of the target class and the decision boundary, thus guiding the error entropy to culminate at origin. Since it is troublesome to fix the data distribution of the error entropy, the cost function is calculated by the Parzen window estimation method as follows:

$$\hat{f}(0)=\frac{1}{Nh\sqrt{2\pi}}\sum_{n=1}^{N}\exp(-\frac{{e_{n,o}}^2}{2\Gamma^2})=\frac{1}{Nh\sqrt{2\pi}}\sum_{n=1}^{N}K(\frac{-{e_{n,o}}^2}{2\Gamma^2}) \tag{69}$$

where N denotes the number of training observations encountered thus far and Γ designates the smoothing parameter, which is simply specified as 1. $e_{n,o}$ labels the predictive error in the *n-th* training cycle of the *o-th* class. In this method, the gradient descent method is used for an optimization process as follows:

$${q_{l,r}}^{i,o}(N)={q_{l,r}}^{i,o}(N-1)+\eta_o\frac{\partial\hat{f}(0)}{\partial {q_{l,r}}^{i,o}}={q_{l,r}}^{i,o}(N-1)-\eta_o\frac{1}{N\sqrt{2\pi}}\sum_{n=1}^{N}K(-\frac{{e_{n,o}}^2}{2})\frac{\partial E}{\partial {q_{l,r}}^{i,o}} \tag{70}$$

$${a_i}^j(N)={a_i}^j(N-1)+\eta_a\frac{\partial\hat{f}(0)}{\partial {a_i}^j}={a_i}^j(N-1)-\eta_a\frac{1}{N\sqrt{2\pi}}\sum_{n=1}^{N}K(-\frac{{e_{n,o}}^2}{2})\frac{\partial E}{\partial {a_i}^j} \tag{71}$$

$${b_i}^j(N)={b_i}^j(N-1)+\eta_b\frac{\partial\hat{f}(0)}{\partial {b_i}^j}={b_i}^j(N-1)-\eta_b\frac{1}{N\sqrt{2\pi}}\sum_{n=1}^{N}K(-\frac{{e_{n,o}}^2}{2})\frac{\partial E}{\partial {b_i}^j} \tag{72}$$

where $\eta_{a,o}$, $\eta_{b,o}$, $\eta_{q,o}$ are the adaptive learning rates. It can be seen that (70)-(72) are not in line with the spirit of the sequential learning concept, because it is necessary to revisit the preceding training samples in the current training episode. Therefore, we amend these expressions as follows:

$$\sum_{n=1}^{N}\exp(\frac{-{e_n}^2}{2})=A_N=A_{N-1}+\exp(\frac{-{e_{N,o}}^2}{2}) \tag{73}$$

$$\frac{\partial\hat{f}(0)}{\partial {q_{l,r}}^{i,o}}=\frac{A_N}{N\sqrt{2\pi}}\frac{\partial E}{\partial {q_{l,r}}^{i,o}},\frac{\partial\hat{f}(0)}{\partial {a_i}^j}=\frac{A_N}{N\sqrt{2\pi}}\frac{\partial E}{\partial {a_i}^j},\frac{\partial\hat{f}(0)}{\partial {b_i}^j}=\frac{A_N}{N\sqrt{2\pi}}\frac{\partial E}{\partial {b_i}^j} \tag{74}$$

The gradient terms $\partial E/\partial {q_{l,r}}^o$, $\partial E/\partial {a_i}^j$, $\partial E/\partial {b_i}^j$ can be gained with the help of the chain rule as follows

$$\frac{\partial E}{\partial q_o^l}=(y_o-t(n))(\frac{\sum_{i=1}^{P}\overline{R}_i\underline{y}_{i,o}}{\sum_{i=1}^{P}\overline{R}_i}-\frac{\sum_{i=1}^{P}\underline{R}_i\underline{y}_{i,o}}{\sum_{i=1}^{P}\underline{R}_i}),\frac{\partial E}{\partial q_o^l}=(y_o-t(n))(\frac{\sum_{i=1}^{P}\overline{R}_i\overline{y}_{i,o}}{\sum_{i=1}^{P}\overline{R}_i}-\frac{\sum_{i=1}^{P}\underline{R}_i\overline{y}_{i,o}}{\sum_{i=1}^{P}\underline{R}_i}) \tag{75}$$

$$\frac{\partial E}{\partial a_{i,j}}=(y_o-t(n))(\sum_{j=1}^{u}\frac{(x_j^{weight}-a_{i,j})}{{b_{i,j}}^2}\exp(-\frac{(x_j^{weight}-a_{i,j})^2}{2{b_{i,j}}^2})(3-(\frac{x_j^{weight}-a_{i,j}}{b_{i,j}})^2)\underline{w}_{i,j,o}(\frac{\sum_{i=1}^{P}\underline{R}_i(1-{q_l}^o)}{\sum_{i=1}^{P}\underline{R}_i}+\frac{\sum_{i=1}^{P}\overline{R}_i q_l^o}{\sum_{i=1}^{P}\overline{R}_i}) \tag{76}$$

$$+\sum_{j=1}^{u}\frac{(x_j^{weight}-a_{i,j})}{{b_{i,j}}^2}\exp(-\frac{(x_j^{weight}-a_{i,j})^2}{2{b_{i,j}}^2})(3-(\frac{x_j^{weight}-a_{i,j}}{b_{i,j}})^2)\overline{w}_{i,j,o}(\frac{\sum_{i=1}^{P}\underline{R}_i(1-{q_r}^o)}{\sum_{i=1}^{P}\underline{R}_i}+\frac{\sum_{i=1}^{P}\overline{R}_i q_r^o}{\sum_{i=1}^{P}\overline{R}_i}))$$

$$\frac{\partial E}{\partial a_{i,j}}=(y_o-t(n))((\frac{(x_j^{weight}-a_{i,j})^2}{{b_{i,j}}^2}\exp(-\frac{(x_j^{weight}-a_{i,j})^2}{2{b_{i,j}}^2})(3-\frac{(x_j^{weight}-a_{i,j})^2}{{b_{i,j}}^3})\underline{w}_{i,j,o}(\frac{\sum_{i=1}^{P}\underline{R}_i(1-{q_l}^o)}{\sum_{i=1}^{P}\underline{R}_i}+\frac{\sum_{i=1}^{P}\overline{R}_i q_l^o}{\sum_{i=1}^{P}\overline{R}_i}) \tag{77}$$

$$+\frac{(x_j^{weight}-a_{i,j})^2}{{b_{i,j}}^2}\exp(-\frac{(x_j^{weight}-a_{i,j})^2}{2{b_{i,j}}^2})(3-\frac{(x_j^{weight}-a_{i,j})^2}{{b_{i,j}}^3})\overline{w}_{i,j,o}(\frac{\sum_{i=1}^{P}\underline{R}_i(1-{q_r}^o)}{\sum_{i=1}^{P}\underline{R}_i}+\frac{\sum_{i=1}^{P}\overline{R}_i q_r^o}{\sum_{i=1}^{P}\overline{R}_i}))$$

In principle, the learning rates $\eta_{a,o}$, $\eta_{b,o}$, $\eta_{q,o}$ are tailored in such a way as to speed up the convergence as follows:

$$\eta_{a,b,qo}(N) = \begin{cases} \rho_5 \eta_{a,b,q,o}(N-1), \hat{f}(0)^N \geq \hat{f}(0)^{N-1} \\ \rho_4 \eta_{a,b,q,o}(N-1), \hat{f}(0)^N < \hat{f}(0)^{N-1} \end{cases}, \text{where } 0 < \rho_4 < 1 < \rho_5 \tag{78}$$

where $\rho_5 \in (1,1.5]$, $\rho_4 \in [0.5,1)$ label the learning rate factors, which steer the increase and decrease of the learning rates to attain convergence instantaneously. Moreover, these parameters are case-insensitive and are assigned as $\rho_4 = 1.1$, $\rho_5 = 0.9$ (Pratama et al, 2015(a)). The plausible range of the learning rates plays an important role in the learning process to ensure rapid convergence. Otherwise, the numerical instability to impair the evolution of the model may be imposed. In the following, the Lyapunov stability criterion is utilised to canvass the suitable range of learning rates.

Theorem 1: the asymptotic convergence is guaranteed, should the learning rate $\eta_{a,b,q}{}^o$ be set in the interval $0 < \eta_{a,b,q}{}^o < \frac{2N\sqrt{2\pi}}{(P_{o,\max})^2 A_N}$, where $P_{o,\max}$ represents the maximum gradient of the output with respect to gradient parameters $P_{o,\max} = \max\limits_{n=1,..,N} \frac{\partial \hat{y}(n)^o}{\partial Z_i{}^o}$ and the gradient parameters are formed as the vector $Z = [a_{i,o}, b_{i,o}, q_l{}^{i,o}, q_r{}^{i,o}]$.

Proof: we define the Lyapunov function $V(k) = e^2(k)/2$, and the rate of the Lyapunov function is expressed as follows:

$$\Delta V(k) = V(k+1) - V(k) = \frac{1}{2}(e^2(k+1) - e^2(k)) = \frac{1}{2}(e(k+1) + e(k))(e(k+1) - e(k)) \tag{79}$$

$$= \frac{1}{2}(e(k+1) + e(k))\Delta e(k) = (e(k) + \frac{1}{2}\Delta e(k))\Delta e(k) \tag{80}$$

The rate of the system error can be further derived as follows:

$$\Delta e(k) = e(k+1) - e(k) = \frac{\partial e(k)}{\partial Z_i{}^o}^T \Delta Z_i{}^o \tag{81}$$

$$\Delta \gamma_i{}^o = -\eta_o \frac{A_N}{N\sqrt{2\pi}} \frac{\partial E}{\partial Z_i{}^o} = -\eta_o \frac{A_N}{N\sqrt{2\pi}} e(n)^o \frac{\partial \hat{y}^o}{\partial Z_i{}^o} \tag{82}$$

Therefore, the rate of the Lyapunov function can be established as follows:

$$\Delta V(k) = -\frac{1}{2}\left\|P_o\right\|^2 \eta_o \frac{A_N}{N\sqrt{2\pi}} e_o(n)^2 (2 - \left\|P_o\right\|^2 \frac{A_N}{N\sqrt{2\pi}} \eta_o) = -e_o(n)^2 \vartheta \tag{83}$$

Noticeably, the asymptotic convergence can be attained when $0 < \vartheta$. This condition thus results in $0 < \eta_{a,b,q}{}^o < \frac{2N\sqrt{2\pi}}{(P_{o,\max})^2 A_N}$ and accomplishes the proof at once. On the other hand, the construct of the scaffolding theory is in line with the recent finding of the connectionist system (Roy, 2000), which can be stimulated by a non-local learning mode and even by external sources.

4.3 When-to-Learn

This learning module mainly functions to end the training process, which is completed by the sample reserved procedure. The streaming data are appointed as the reserved samples if the rule-growing and adaptation scenarios formulated in (26),(27),(37) are not complied with. These samples are placed in the rear of the sample stack to be investigated when the system is idle or the underlying samples have been fully depleted. Ideally, the

training process is terminated if the reserved samples are fully spent. In practice, this prerequisite is impossible to satisfy because we are dealing with streaming data, where prior knowledge of the total amount of streaming data is unknown. In lieu of the strict requirement, the training process is terminated when the number of reserved samples remains constant. Note that the positive impact of the sample reserved scenario is to fill gaps uncharted by the regular training patterns in finalizing the training process. This mechanism can apparently refine the generalization of the model and the completeness of the rule base. Fig. 4 depicts the working principles of ST2Class.

4.3 Computational Complexity Analysis

The resultant computational complexity of ST2Class is induced by an individual learning module, constructed in the context of the meta-cognitive scaffolding theory. The what-to-learn component, governed by the so-called uncertainty measure, bears a computational burden in the order of $O(P)$. The when-to-learn mechanism, driven by the sample reserved strategy, incurs $O(NS)$, because the key idea is to train the model using from the reserved samples $(X_{ns}, T_{ns})_{ns=1,\ldots,NS+1}$. Furthermore, the how-to-learn constituent presents a synergy of the schema and scaffolding theories, which costs $O(7P + P^* + u^2 + u + u^2P + 2uP + PM)$. To sum up, ST2Class possesses a computational load in the order of $O(P + NS + \rho(7P + u^2P + u^2 + u + P^* + 2uP + PM))$, where ρ signifies the probability of the streaming data to be admitted for the training process. In relation to the computational burden of the state-of-the art classifiers, the computational complexity of ST2Class is comparable. For example, rClass in Pratama et al (2014b) and gClass in Pratama et al (2014a), which adopt the meta-cognitive scaffolding learning method, solicit the computational complexity of $O(\rho(m^2u^2 + mu + m(2P+1)^2 + 6P + P^* + 2u) + NS + 2mP)$ and $O(\rho(m(2P+1)^2 + m^2 + U + 5P + m + U^2 + P^*) + NS + 2mP)$ respectively.

Conversely, the cognitive constituent of ST2Class, exploiting the multivariable interval type-2 fuzzy neural network, has rule base parameters in the order of $O((2P \times (u \times u)) + (u \times P) + 2P \times m \times u + (u \times P))$, in which this rule base burden is commensurate with its counterparts of the interval type-2 fuzzy neural networks. For example, eT2Class of Pratama et al (2015b) charges $O((2P \times (u \times u)) + (u \times P) + 2P \times m \times (2u+1))$, whereas Simplified Interval Type-2 Fuzzy Neural Network (SIT2FNN) draws the complexity of $O((P \times u) + 2(u \times P) + 2P \times m \times (u+1))$. Moreover, the rule base cost of ST2Class is supposed to be more economical than its counterparts, because this network topology is anticipated to scatter fewer fuzzy rules as a consequence of a more appropriate cluster shape and contour crafted by the generalized fuzzy rule.

5 Proof of Concepts

This section demonstrates the leverage of the individual learning component on the final learning performance of ST2Class to analyse to what extent each learning module affects ST2Class learning performance. To this end, four numerical studies, containing various regime-drifting properties, are explored to scrutinise the impact of the meta-cognitive scaffolding learning theory, where the semi-artificial data streams of Minku, White and Yao (2010), namely iris and car, are exploited. These problems are drawn from the machine learning repository at the University of California, Irvine (http://www.ics.uci.edu/mlearn/MLRepository.html), and have recently been customised in Minku, White and Yao (2010) to engage the non-stationary component in data streams. Alongside these data streams, two time-varying synthetic studies, namely line and sin, emanating from the DDD database (Minku & Yao, 2012) are exploited to numerically validate the efficacy of the ST2Class learning modules. The characteristic of these problems is detailed in Section 5. The experimental procedure to

produce the numerical results relies on the periodic hold-out process, simulating the training and testing processes in real time, while our computational resources are an Intel (R) core (TM) i7-2600 CPU @3.4 GHz processor and 8 GB memory. Note that the periodic hold-out process partitions the number of data into the predetermined number of mutually exclusive time stamps to form the sub-problems. Every sub-problem comprises the training and testing phases and the classifier performance is concluded from the average of the numerical results in all sub-problems. Meanwhile, ST2Class is appraised from five evaluation standpoints, viz., classification rate, number of hidden nodes, runtimes, number of training samples, and number of parameters. The classification rate fathoms the classifier's generalization, which is elicited as the rate of correctly classified testing samples. The number of hidden nodes is employed to abstract the efficiency of ST2Class, and is obtained in the same way as the fuzzy rules generated in the training process. The rule base complexity can be assessed by the number of rule base parameters, which is dependent on the cognitive constituent inferring the classification decision. The rule base parameter of ST2Class is acquired as presented in Section 4.4. The number of samples is sufficient to pinpoint the potency of the what-to-learn component in extracting the effective training stimuli. Conversely, the computational burden of ST2Class can be experimentally exhibited by its runtime, which refers to the time required to conclude the training process. The numerical results are summarized in Table 1.

Our empirical studies encompass four underlying tasks detailed as follows: 1) the cognitive component of ST2Class is the first point of the investigation, where we benchmark ST2Class with the architecture of the multivariable interval type-2 fuzzy neural network as exemplified in Pratama et al (2015b), and the standard network topology of the interval type-2 fuzzy neural network as shown in Lin, Liao, Chang and Lin (2014). This experiment is articulated in Section A of Table 2. 2) The purpose of the second part is to study the potency of the what-to-learn component, where the ST2Class numerical result with the absence of the what-to-learn module is canvassed. The numerical results are summed up in Section B of Table 2. 3) We study the learning performance of a learning configuration of ST2Class, in which the complexity reduction component is switched off in another setting. The numerical results are summarized in Section C of Table 2. 4) Our study continues by inspecting the contribution of the problematizing aspect of scaffolding theory, in which the ST2Class performance without this module is investigated, and the numerical results are reported in Section D of Table 2. 5) The efficacy of the fading component of scaffolding theory is displayed in Section E of Table 2, in which the impact of the fading item of scaffolding theory is probed by discharging this learning module in the training process.

Clearly, ST2Class with its resultant learning configuration delivers the most encouraging numerical results, since it prevails in all numerical studies in almost all learning criteria. The use of Chebyshev function or first order TSK function in the rule output results in deterioration in the classifier generalization and retards the runtime, as displayed in Section A. The results justify the multi-resolution property of the wavelet function, which captures the temporal change in the system dynamic. Because the wavelet polynomial scatters a fewer number of network parameters, it is capable of shortening the training time. The absence of the what-to-learn module slackens the training process, because all data streams have to be consumed. This facet aggravates the over-fitting problem, which reduces the generalization ability. The learning performance of ST2Class worsens when the complexity reduction module is switched off, because it cannot cope with the curse of dimensionality. The efficacy of the fading facet is confirmed in Section D, where it is fruitful to simplify the network structure.

It can be seen that network complexity increases dramatically when the fading component is deactivated. On the other hand, the problematizing module contributes deciphering the concept drifts, especially the incremental and gradual concept drifts, which are uncharted by the Schema part. The absence of the problematizing module directly affects the generalization of the classifier and the sample consumption.

6 Benchmark with Prominent Classifiers

This section presents numerical evaluation of ST2Class in 13 non-stationary data streams and experimental comparisons of ST2Class with state-of-the art classifiers in the literature, which are followed by comprehensive statistical tests to arrive at statistically valid conclusions of learning performance of consolidated classifiers. In addition, the learning characteristics of benchmarked algorithms are theoretically compared and some open problems, which can become future research direction, are also explained.

6.1 Numerical Comparison with State-of-the Art Classifiers

In this section, the learning efficacy of ST2Class in processing dynamic and evolving streaming data is profoundly contrasted with other cutting-edge classifiers recently published in the literature. Three prominent meta-cognitive classifiers, namely Generic Classifier (gClass) of Pratama et al (2014b), Recurrent Classifier (rClass) of Pratama et al (2015a), and Meta-Cognitive Fuzzy Inference System (McFIS) of Subramanian, Suresh and Sundarajanan (2013), are consolidated to compete with ST2Class. ST2Class is also compared with its counterparts, Evolving Type-2 Classifier (eT2Class) of Pratama et al (2015b) and Parsimonious Classifier (pClass) of Pratama et al (2014c). Although it neglects the meta-cognitive learning scenario in the training process, eT2Class embodies a holistic concept of the evolving system in the context of the multivariable interval type-2 neuro-fuzzy system, whereas pClass is typical of the evolving system under the framework of the generalized type-1 fuzzy neural network. The setting of the experiment, the evaluation criteria, and the computational resources, are similar to those in Section IV. The consolidated classifiers are evaluated by exploiting 13 non-stationary synthetic and real-world streaming data, serving various concept drifts. Note that it is relevant to involve the artificial data streams in the numerical studies because we can precisely determine the exact drift type, and the time period in which the concept drift is active, by means of the synthetic data streams.

The numerical problems, namely circle, sin, sinh, line, and Boolean, emanating from Minku and Yao (2012), are explored. All problems characterise a binary classification problem and the sudden concept drift. There exist three types for each problem in the DDD data base, where the most complicated version is employed here. The prominent benchmark problem, namely SEA data stream introduced in in Street and Kim (2001), is exploited. We do not use the original version of the SEA problem, and the version customised in Ditzler and Polikar (2014) is utilised in lieu of the traditional variant. This new version is slightly different to the classical version, since it has been amended to conceive the class imbalance problem and the cyclical concept drift. The numerical study is also conducted using the Gaussian data stream, designed in Elwell and Polikar (2011). This study poses a binary classification problem in which class 1 represents a majority class, and class 2 is a minority class. Each class undergoes a gradual and independent drift, which is achieved by varying the means and variance of the parametric equations. The semi-artificial data-streams, namely iris and car, are deployed here and have also been exploited in Section V. The noise corrupted data stream, which has its root in Pratama et al (2014c), is used in this paper. Apart from snapshotting various concept drifts and the multi-category classification problems, this study is worth attempting because it features a noisy property and dynamic class attributes. That is, the class labels are dynamically evolved and pruned during the training process to intensify the complexity. In addition to 10 synthetic data streams, ST2Class is tested by three real world problems, well-known for their non-stationary

components: electricity pricing (Harries, 1999), weather (ftp://ftp.ncdc.noaa.gov/pub/data/gsod/), and wine. The characteristics of all data streams are summarized in Table 2. Note that the Imbalanced Factor (IF) is computed as (Subramanian et al, 2014).

Our numerical study benefits from the data stream generator, which is downloadable at (www.cs.bham.ac.uk/~flm/opensource/DriftGenerator.zip). The numerical results are reported in Table 3. Fig.5(a) shows the fuzzy rule evolution, and Fig.5(b) depicts the local forgetting evolution of fuzzy rule ζ_i; these two figures are produced by the electricity pricing problem. Fig.5(c) delineates the feature weight evolution and Fig.5(d) visualises the system error; 5(c)-(d) are generated by the sin problem. Fig.6(a) pictorially exhibits the trace of classification rate of consolidated classifiers, and Fig. 6(b) visualises the trace of the fuzzy rule of consolidated classifiers. The trace of runtime of consolidated classifiers is depicted by Fig. 6(c) and the trace of number of accepted training samples of consolidated algorithms is portrayed by Fig. 6(d). All figures 6(a)-(d) are produced using the line data stream. The rule base parameters of ST2Class and eT2Class are quantified as in Section 4.4, while the rule base parameter of gClass is specified as $O(p\times m\times(2u+1)+p\times(u\times u)+p\times u)$ because gClass is driven by the multivariate Gaussian function in the hidden layer and the Chebyshev polynomial in the output layer. By extension, the number of rule base parameters of rClass is computed as $O(m\times p\times(2u+1)+p\times(u\times u)+p\times u+p\times m)$, because the output is inferred by the generalized local recurrent network topology. McFIS is constructed by the traditional zero order TSK fuzzy system, which imposes $O(mp+UP+P)$ parameters. pClass is built upon the generalized first-order TSK fuzzy system incurring $O(p\times m\times(u+1)+p\times(u\times u)+p\times u)$ parameters. The predefined parameters of the classifiers are allocated as the rule of thumb in their original publications.

Referring to Table 2, ST2Class clearly outperforms other benchmarked classifiers on three criteria, viz., classification rate, number of fuzzy rules and number of parameters. ST2Class is clearly superior in the aspect of classification accuracy, because it produces the most accurate prediction in 12 of 13 study cases. Notwithstanding that it is built upon the multivariable interval type-2 neuro-fuzzy architecture, ST2Class in fact contains fewer parameters to be stored in memory because it evolves the most compact fuzzy rule, and consequently alleviates the rule base load. It is worth noting that McFIS constitutes a fully supervised classifier, which requires all data streams to be labelled by operator, although it consumes the least number of samples in Table 2. ST2Class is not only a semi-supervised classifier, but also capable of suppressing operator annotation effort to a minimum level, because the labelling process for accepted data streams can be automatically done with the absence of manual intervention. It can be observed as well that although SIT2Class is constructed by the metacognitive scaffolding algorithm, encompassing a significant number of training scenarios, SIT2Class's runtime is generally speaking comparable with other consolidated algorithms. It is as a result of its fully sequential working principle and the online active learning process in the what-to-learn part. The online active learning scenario is capable of discarding inconsequential samples from the training process, thus speeding up the runtime. This component renders ST2Class's execution time faster than non-meta-cognitive classifiers: pClass, eT2Class.

Inevitably, ST2Class adopts an open rule base principle in which the fuzzy rules can be augmented, refined, pruned, merged and recalled on demand in accordance with the given learning context, as shown by Fig.5(a). Fig.5(b) confirms the local forgetting mechanism of ST2Class, which outputs a unique forgetting level ζ_i to

unravel different drift intensities in the respective local fuzzy regions. Meanwhile, superfluous input features can be softly eliminated from the learning process by supplying a minuscule weight, thereby smoothly minimizing their influence on the model updates. In other words, the online feature weighting mechanism can be considered a soft feature reduction approach which is capable of surmounting the curse of dimensionality problem over time without explicitly pruning superfluous input features. By extension, the system errors are stable in a small bounded range during the training phase, which corroborates the effectiveness of the passive scaffolding component. The accuracy of consolidated classifiers is illustrated in Fig. 6(a), where ST2Class delivers the most reliable trend of the classification rate. The efficacy of ST2Class can be seen in Fig.6(b),(c), where it crafts the most compact and parsimonious network structure. ST2Class also consumes a low number of training samples, which is merely inferior to that of McFIS. Nevertheless, one should keep in mind that McFIS is a fully supervised classifier, costly in practise.

6.2 Statistical Tests

This section aims to statistically validate the efficacy of the ST2Class to reach a firm conclusion about the numerical results. The classifier's rankings are tabulated in Table 5. ST2Class noticeably outstrips other classifiers in three evaluations with respect to the classification rate, the number of parameters and the number of fuzzy rules, and it is only inferior to rClass and McFIS in the runtime category. We overlook the number of sample criteria in our statistical test, because the consolidated algorithms utilise different what-to-learn concepts and thus cannot be compared on an 'apple-to-apple ' basis. We depart from our statistical analysis by means of the Friedman statistical test, which is a prominent statistical tool for considering whether or not there is a substantial performance difference between the consolidated algorithms. As shown in Demsar (2006), we land on $\chi_F^2 = 39.2,36.5,32.5,30.5$, and the critical value of $\alpha = 0.1$ with 5 Degrees of Freedom (DoF) is 9.24. Hence, the null hypothesis is rejected outright for all evaluation criteria. Nevertheless, the Friedman statistical test is too conservative, which encourages us to employ the Anova test. This test is akin to the Friedman test and investigates performance difference between the benchmarked classifiers. The critical value of $\alpha = 0.05$ with (5,60) DoF is 2.38 and we arrive at $F_F = 20.7,17.1,13,11.4$, which consequently means a rejection of the null hypothesis.

These two tests do not reveal the performance dissimilarity between a classifier pair, thus, it is not yet possible to deduce the superiority of ST2Class against other classifiers. To this end, we address this issue using a post-hoc Bonferrroni-Dunn test, in which the performance difference between two classifiers is deemed to be major if their difference transcends the critical difference. In the following, the critical difference with $\alpha = 0.1$ is $CD = 1.95$, while the performance difference between two benchmarked classifiers is quantified as $z = (R_i - R_j)\sqrt{6Q} \Big/ \sqrt{(M+1)M}$, where Q denotes the number of numerical examples deployed in our numerical study and M labels the number of consolidated classifiers. Table 6 tabulates the performance difference between ST2Class and its counterparts. It would appear that ST2Class excels compared to other algorithms in terms of classification accuracy. From the viewpoint of the evolved fuzzy rules and the rule base parameter, it is statistically better than all other classifiers except eT2Class. In contrast, McFIS surpasses ST2Class in runtime, while rClass outstrips ST2Class in runtime, yet the difference is subtle.

6.3 Theoretical Comparison with State-of-the Art Classifiers

ST2Class makes use of the interval type-2 multivariate Gaussian function in the hidden layer and employs the interval-valued Wavelet polynomial in the output layer. This rule variant supports more accurate input space partition than that of the traditional interval type-2 Gaussian function. Accordingly, ST2Class requires the least number of fuzzy rules to guarantee the completeness of the rule base as shown in our experiment, where ST2Class is capable of generating the most compact and parsimonious network structure. The use of the wavelet function in the output layer is effective for boosting the local mapping aptitude of the classifier while sustaining low complexity. The amendment of the local output mapping property enables the production of more precise predictions than the first order TSK output layer, as exemplified in pClass and McFIS. By extension, the wavelet function is more practical than the Chebyshev polynomial-based output layer, which exists in rClass, gClass and eT2Class, because this rule consequent variant does not impose a higher degree of freedom, ultimately inducing more parameters to be accumulated in the memory.

The what-to-learn aspect of ST2Class not only extracts relevant streaming data in the semi-supervised learning mode, but also reduces the manual intervention required to label the streaming data. Although McFIS spends the least number of rule base parameters in our numerical study, it is based on the fully supervised hinge loss function to foresee the contribution of the streaming data. The major difference between ST2Class and its predecessors is in its self-labelling process, which overcomes the lack of ground truth situations in annotating unlabelled streaming data. ST2Class is also endowed with the online feature weighting mechanism, which is carried out in an unsupervised manner. It is unlike the online feature weighting mechanisms in pClass, rClass and gClass, which violate the semi-supervised learning principle because they are devised from the Fisher separability criterion. Online feature selection is absent in McFIS, and is definitely inadequate for confronting the curse of dimensionality problem. It is worth stressing that ST2Class puts forward the type-2 meta-cognitive scaffolding mechanism, leading to the plug-and-play learning perspective, where all learning modules (rule pruning, merging, split, drift handling, feature selection) are coupled in a single dedicated learning process. The properties of the consolidated algorithms are summarized in Table 4.

6.4 Open Problems

This paper presents in-depth study of the type-2 fuzzy neural network classifier built upon the psychologically sound theories of the metacognitive scaffolding theory. Although this study has actualised majority of the metacognitive scaffolding components in the machine learning context, three facets at least remain open issues: 1) ST2Class is still designed in the conventional feed-forward network architecture, which has some limitations in dealing with the temporal system behaviour and the absence of system order; 2) the local forgetting mechanism described in Section 4.2.2 is evaluated using the rate of local error in respect to the previous step. This approach is deemed to be inaccurate to reflect the evolution of changing system dynamic, because the window size is too narrow. On the other hand, a suitable choice of window size is usually time-consuming and problem-dependent. In addition, the local error is not an accurate measure for the gradual concept drift, because it is sensitive to the over-fitting and under-fitting issues; 3) the feature weighting scenario in Section 4.2.1 does not substantially alleviate the complexity, because superfluous features are still kept in the memory and only their impacts to the learning process are minimised.

7 Conclusion and Further Study

To correct several shortcomings of the existing machine learning variants, an incremental type-2 meta-cognitive scaffolding algorithm, namely ST2Class, is put into perspective. The contribution of ST2Class lies in four aspects: 1) ST2Class features the meta-cognitive model of Nelson and Narens (1990), where the how-to-

learn component is specifically derived from a synergy between the Schema and Scaffolding theories. It can be perceived as an enhanced version of Pratama et al (2015a); 2) ST2Class is driven by a multivariable interval type-2 fuzzy rule, which has an interval multivariate Gaussian function in the rule layer and the wavelet function in the consequent layer. 3) ST2Class is equipped with a new online active learning module, which features an autonomous labelling property. 4) The online input weighting mechanism is derived from mutual information estimation, which approximates the significance of input attributes in the unsupervised manner. All ST2Class learning modules are designed in accordance with incremental and local design concepts to attain the greatest possible efficiency for learning from data streams and are embedded in the single learning process to render the plug-and-play learning paradigm. The plug-and-play learning perspective disallows pre-and-post-training processes, which harm the notion of the truly online learning concept.

ST2Class delivers a novel prospect for big scalable data analytics, and can deal with various concept drifts and uncertainties in data streams. It also takes a step closer to rapprochement between machine learning theories and cognitive psychology. The efficacy of ST2Class has been circumspectly tested by 13 synthetic and real-world problems and has been further confirmed through rigorous statistical tests, in which ST2Class delivers the soundest performance in achieving a trade-off between accuracy and complexity. We envision the enhanced recurrent type-2 meta-cognitive scaffolding theory as our future work to deal with temporal problems and ill-defined system, where prior knowledge of the system order is unavailable. We will also apply ST2Class to perform tool wear prognosis in the ball-nose end-milling process.

ACKNOWLEDGEMENTS

The work presented in this paper is supported by the Australian Research Council (ARC) under Discovery Project DP140101366 and the La trobe university start-up grant. The third author acknowledges the support of the Austrian COMET-K2 programme of the Linz Center of Mechatronics (LCM), funded by the Austrian federal government and the federal state of Upper Austria. This publication reflects only the authors' views.

REFERENCES

Abiyev,R.H.,Kaynak,O. (2008). Fuzzy Wavelet Neural Networks for Identification and Control of Dynamic Plants-A Novel Structure and a Comparative Study. *IEEE Transactions on Industrial Electronics*,55(8),3133-3140.

Abiyev,R.H.,Kaynak,O.(2010). Type-2 fuzzy neural structure for identification and control of time-varying plants. *IEEE Transactions on Industrial Electronics*.57(12),4147-4159.

Abiyev,R.H.,Kaynak,O.,Kayacan,K.(2012). A type-2 fuzzy wavelet neural network for system identification and control. *Journal of the franklin institute*,550,1658-1685.

Angelov,P.P.,Filev,D.(2004).An approach to online identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*: *Cybernetics*, 34, 484-498.

Angelov,P.P.,Zhou,X.(2008).Evolving fuzzy-rule-based classifiers from data streams. *IEEE Transactions on Fuzzy Systems*, 16(6), 1462-1475.

Angelov,P.P.,Lughofer,E.,Zhou,X.(2008).Evolving fuzzy classifiers using different model architectures. *Fuzzy Sets and Systems*,159(23), 3160-3182.

Angelov,P.P. (2011). Fuzzily connected multi-model systems evolving autonomously from data streams. *IEEE Transactions on Systems, Man, and Cybernetics, Part B:Cybernetics*, 41(4), 898-910.

Bartett,F.C.(1932). Remembering :A study in Experimental and Social Psychology. U.K:Cambridge.

Baruah,R.D.,Angelov,P.P.,Zhou,X.(2011). *In: Proceedings of 2011 IEEE International Conference on Systems, Man and Cybernetics, SMC 2011*, 2249-2254.

Babu,B.S.,Suresh,S.(2013).Sequential projection-based metacognitive learning in a radial basis function network for classification problems. *IEEE Transactions on Neural Networks and Learning Systems*, 24(2), 194-206.

Bustince,H., Fernandez, J., Hagras, H., Herrera,F.(2015) Interval Type-2 Fuzzy Sets are generalization of Interval-Valued Fuzzy Sets: Towards a Wider view on their relationship. *IEEE Transactions on Fuzzy Systems*, (2015) doi: 10.1109/TFUZZ.2014.2362149, in press.

Bouchachia,A.,Vanaret,C.(2014). GT2FC: An Online Growing Interval Type-2 Self-Learning Fuzzy Classifier. *IEEE Transactions on Fuzzy Systems*, 22(4),999-1018.

Bortman,M.,Aladjem,M.(2009).A growing and pruning method for radial basis function networks. *IEEE Transactions on Neural Networks*, 20(6), 1039-1045.

Bose,R.P.J.C., van der Aalst,W.M.P., Zliobaite.I,Pechenizkiy,M.(2014).Dealing with concept drifts in process mining. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1), 154-171.

Chao, C.T., Chen,Y.J., Teng, C.C.(1996). Simplification of fuzzy-neural systems using similarity analysis. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*,26(2),344–354

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

Demsar,J.(2006). Statistical Comparisons of Classifiers over Multiple Datasets. *Journal of Machine Learning Research*, 7, 1-30.

Elwell,R.,Polikar,R.(2011). Incremental learning of concept drift in non-stationary environments. *IEEE Transactions on Neural Networks*, 22(10), 1517-1531.

Flavell,J.H.(1996).Piagiet's legacy.*Psychological Science*, 7(4) ,200-203

Harries,M.(1999).NewSouthWales.Splice-2 Comparative Evaluation : Electricity Pricing. Available from: ⟨ftp://ftp.cse.unsw.edu.au/pub/doc/papers/UNSW/ 9905.pdf.⟩.
Isaacson, R., Fujita, F.(2006). Metacognitive knowledge monitoring and self-regulated learning: Academic success and reflection on learning. *Journal of the Scholarship of Teaching and Learning*, 6(1), 39-55.
Juang, C.F., Lin,C.T.(1998). An on-line self-constructing neural fuzzy inference network and its applications. *IEEE Transactions on Fuzzy Systems*, 6(1), 12-32.
Juang, C.F., Tsao,Y.W.(2008). A self-evolving interval type-2 fuzzy neural network with on-line structure and parameter learning. *IEEE Transactions on Fuzzy Systems.* 16(6),1411-1424.
Juang,C.F.,Chen,C.Y.(2013).Data-driven interval type-2 neural fuzzy system with high learning accuracy and improved model interpretability. *IEEE Transactions on Cybernetics*.43(6).1781-1795.
Joysula,D.P.,Vadali.H., Donahue, B.J., Hughes,F.C.(2009).Modeling metacognition for learning in artificial systems. *In proceeding of World Congress on Nature and Biologically Inspired Computing*.1419-1424.
Karnik,N.N.,Mendel,J.M.,Liang,Q.(1999).Type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems.*7(6),643-658.
Lemos,A.,Caminhas,W.,Gomide,F.(2013). Adaptive fault detection and diagnosis using an evolving fuzzy classifier. *Information Sciences*,220,64-85
Lewis,D.,Catlett,J.(1994).Heterogeneous uncertainty sampling for supervised learning. *In Proceedings of the International Conference on Machine Learning*, 148–156.
Lin,Y.Y.,Chang,J.Y.,Lin,C.T.(2014(a)). A TSK-Type-Based Self-Evolving Compensatory Interval Type-2 Fuzzy Neural Network (TSCIT2FNN) and its applications. *IEEE Transactions on Industrial Electronics*, 61(1),447-459.
Lin,Y.Y.,Liao,S.H,Chang,J.Y.,Lin,C.T.(2014(a)).Simplified Interval Type-2 Fuzzy Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5),959-969.
Liang,Q., Mendel,J.M.(2000). Interval Type-2 Fuzzy Logic Systems: Theory and Design. *IEEE Transactions on Fuzzy Systems*, 8(5), 535-550.
Liu.F.,Quek.C.,Ng.G.S.(2007). A Novel Generic Hebbian Ordering-Based Fuzzy Rule Base Reduction Approach to Mamdani Neuro-Fuzzy System. *Neural computation*, 19(6). 1656-1680.
Lughofer, E.(2006).Process Safety Enhancements for Data-Driven Evolving Fuzzy Models. *In proceeding of International Symposium on Evolving Fuzzy Systems*,42-48.
Lughofer,E.(2011(a)). On-line incremental feature weighting in evolving fuzzy classifiers. *Fuzzy Sets and Systems*,163(1),1-23.
Lughofer,E.(2011(b)). *Evolving Fuzzy Systems --- Methodologies, Advanced Concepts and Applications*, Springer, Heidelberg.
Lughofer,E.,Bouchot,J.L.,Shaker.A.(2011). On-line elimination of local redundancies in evolving fuzzy systems. *Evolving Systems*, 2(3),380-387.
Lughofer,E.(2012). A Dynamic Split-and-Merge Approach for Evolving Cluster Models. *Evolving Systems.*3(3),135-151.
Lughofer,E.(2008). FLEXFIS: A robust incremental learning approach for evolving Takagi.–Sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems*,16(6),1393-1410.
Lughofer,E.,Buchtala,O.(2013). Reliable All-Pairs Evolving Fuzzy Classifiers. *IEEE Transactions on Fuzzy Systems*, 21(4),625-641.
Mitra,P.,Murthy,C.A.,Pal,S.K.(2002).Unsupervised feature selection using feature similarity, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3),301-312.
Martin,T.(2005). Fuzzy sets in the fight against digital obesity. *Fuzzy Sets and Systems*, 156(3), 411-417.
Mendel,J.M.,John,R.I.(2002). Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems*, 10(2),117-127.
Minku,L.L.,White,A.P.,Yao,X.(2010). The Impact of Diversity on Online Ensemble Learning in The Presence Concept of Drift. *IEEE Transactions on Knowledge and Data Engineering*, 22(5),730-742.
Minku,L.L.,Yao,X.(2012). DDD: A New Ensemble Approach for Dealing with Drifts. *IEEE Transactions on Knowledge and Data Engineering*, 24(4), 619-633.
Nelson,T.O.,Narens,L.(1990). Metamemory: A theoretical framework and new findings. *Psychology of Learning and Motivation*, 26,125-173.
Pratama.M.,Anavatti,S.,Angelov,P.P.,Lughofer,E.(2013), PANFIS: A novel incremental learning machine. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1),55-68.
Pratama,M.,Anavatti,S.G.,Lughofer,E.(2014(a)). GENEFIS: towards an effective localist network. *IEEE Transactions on Fuzzy Systems*, 22(3),547-562.
Pratama,M.,Meng Joo,Er.,Anavatti,S.G,Lughofer,E.(2014(b)). A novel meta-cognitive scaffolding classifier to sequential non-stationary classification problems. *In proceeding of 2014 IEEE Conference on Fuzzy Systems*,369-376.
Pratama,M.,Anavatti,S.,Lughofer,E.(2014(c)). pClass: an effective classifier to streaming examples. *IEEE Transactions on Fuzzy Systems*, in press(10.1109/TFUZZ.2014.2312983).
Pratama,M.,Anavatti,S.,Lu,J.(2015(a)).Recurrent classifier based on an incremental meta-cognitive scaffolding algorithm. *IEEE Transactions on Fuzzy Systems*, in press.
Pratama,M.,Lu,Jie.,Zhang,G.,Anavatti.S.(2015(b)). Evolving Type-2 Fuzzy Classifier. *submitted to IEEE Transactions on Fuzzy Systems.*
Rong,H.J,Sundararajan,N, Huang, G.B, Zhao, G.S.(2011). Extended Sequential Adaptive Fuzzy Inference System for Classification Problems. *Evolving Systems*, 2(2),71-82.
Roy,A.(2000).On connectionism, rule extraction and brain-like learning. *IEEE Transactions on Fuzzy Systems*, 8(2), 222-227.
Reiser,B.J.(2004). Scaffolding complex learning: The mechanisms of structuring and problematizing student work. *Journal of Learning Sciences*,13(3),273-304.
Savitha,R.,Suresh,S.,Sundararajan,N.(2012). Metacognitive Learning in a Fully Complex-Valued Radial Basis Function Neural Network. *Neural computation*, 24(5), 1297-1328.
Shaker,A.,Lughofer,E.,(2014). Self-adaptive and local strategies for a smooth treatment of drifts in data streams. *Evolving Systems*, 5(4), 239-257.
Subramanian,K.,Suresh,S., Sundararajan,N.(2014).A Meta-Cognitive Neuro-Fuzzy Inference System (McFIS) for sequential classification systems. *IEEE Transactions on Fuzzy Systems*, 21(6), 1080-1095.
Subramanian,K.,Das,A.K,Suresh,S.,Savitha,R.(2014). A meta-cognitive interval type-2 fuzzy inference system and its projection based learning algorithm. *Evolving Systems*, 5(4),219-230.
Suresh,S., Dong, K., Kim.H.(2010),A sequential learning algorithm for self-adaptive resource allocation network classifier," *Neurocomputing*, 73(16), 3012–3019.
Silva,L., Alexandrem,A., Mardues de Sa,J. (2005). Neural Network Classification: Maximizing Zero-Error Density. *Lecture Notes in Computer Science: Pattern Recognition and Data Mining*, 3686,127-135.
Street, W.N., Kim, Y.(2001).A streaming ensemble algorithm SEA for large- scale classification", in the proceeding of 7th ACM SIGKDD, 377-382.

Tung,S.W., Quek,C., Guan,C. (2013).eT2FIS: An Evolving Type-2 Neural Fuzzy Inference System. *Information Sciences,* 220,124-148.
Vygotsky,L.S. (1978). *Mind and Society: The Development of Higher Psychological Processes*, Cambridge, U.K: Harvard University Press.
Vigdor,B.,Lerner.B.(2007).The Bayesian ARTMAP," *IEEE Transactions on Neural Networks*, 18(6).1628–1644.
Wood,D.(2001).Scaffolding contingent tutoring and computer-based learning", *International Journal of Artificial Intelligence in Education,* 12(3), 280-292.
Wu,D., Mendel,J.M.(2009) .A comparative study of ranking methods, similarity measures and uncertainty measures for interval type-2 fuzzy sets", *Information Sciences,* 179, 1169-1192.
Xiong,S.,Azimi,J.,Fern,X.Z.(2014). Active Learning of Constraints for Semi-Supervised Clustering. . *IEEE Transactions on Knowledge and Data Engineering.*26(1),43-54.
Yu,L.,Liu,H.(2004). Efficient Feature Selection via Analysis of Relevance and Redundancy. *Journal of Machine Learning Research*, 5, 1205-1224.
Zadeh, L.A.(1975).The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences*, 8(3), 199–249
Zliobaite,I., Bifet,A., Pfahringer.B., Holmes,B.(2014).Active Learning with Drifting Streaming Data.*IEEE Transactions on Neural Networks and Learning Systems,* 25(1), 27-39.

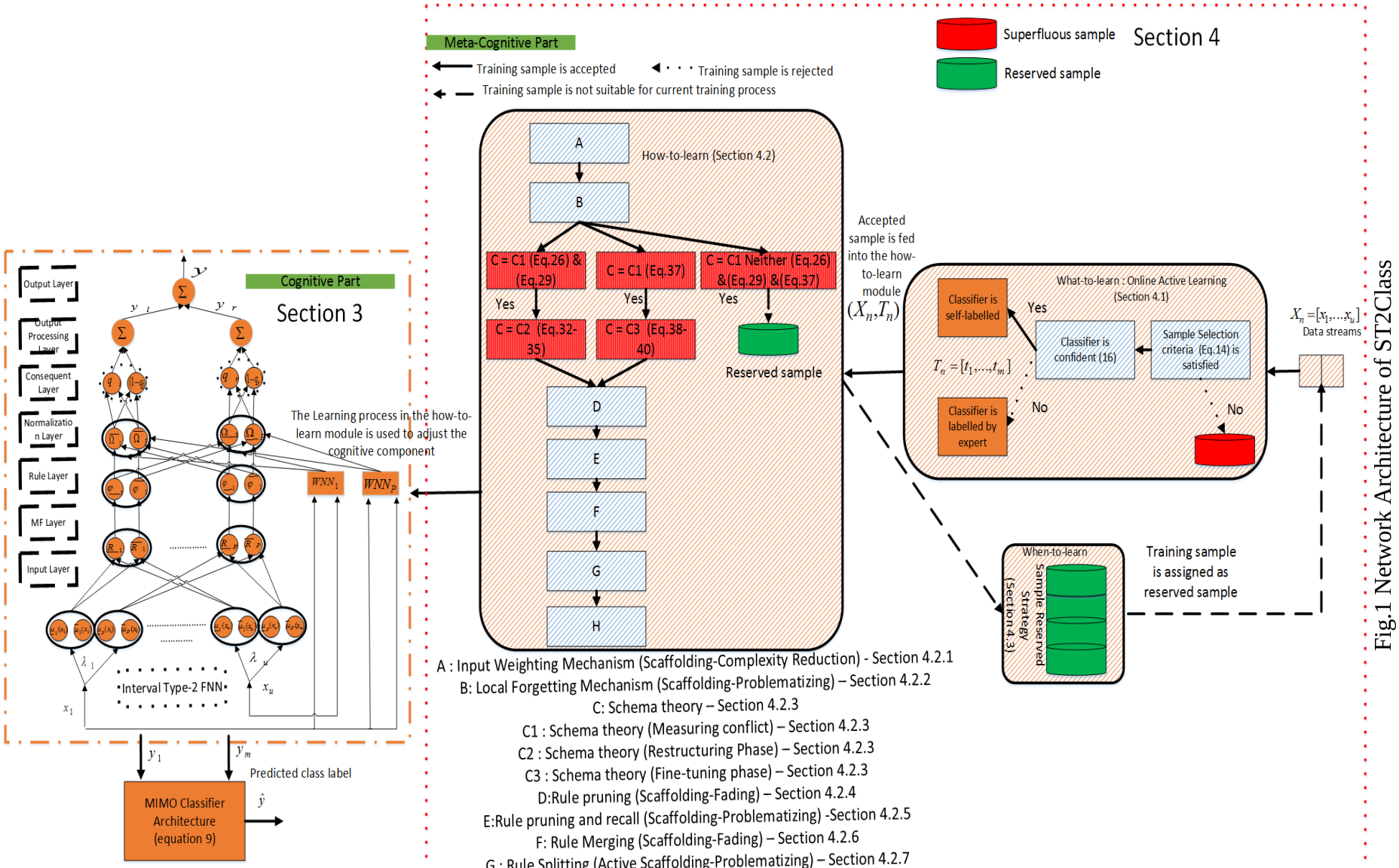

Fig.1 Network Architecture of ST2Class

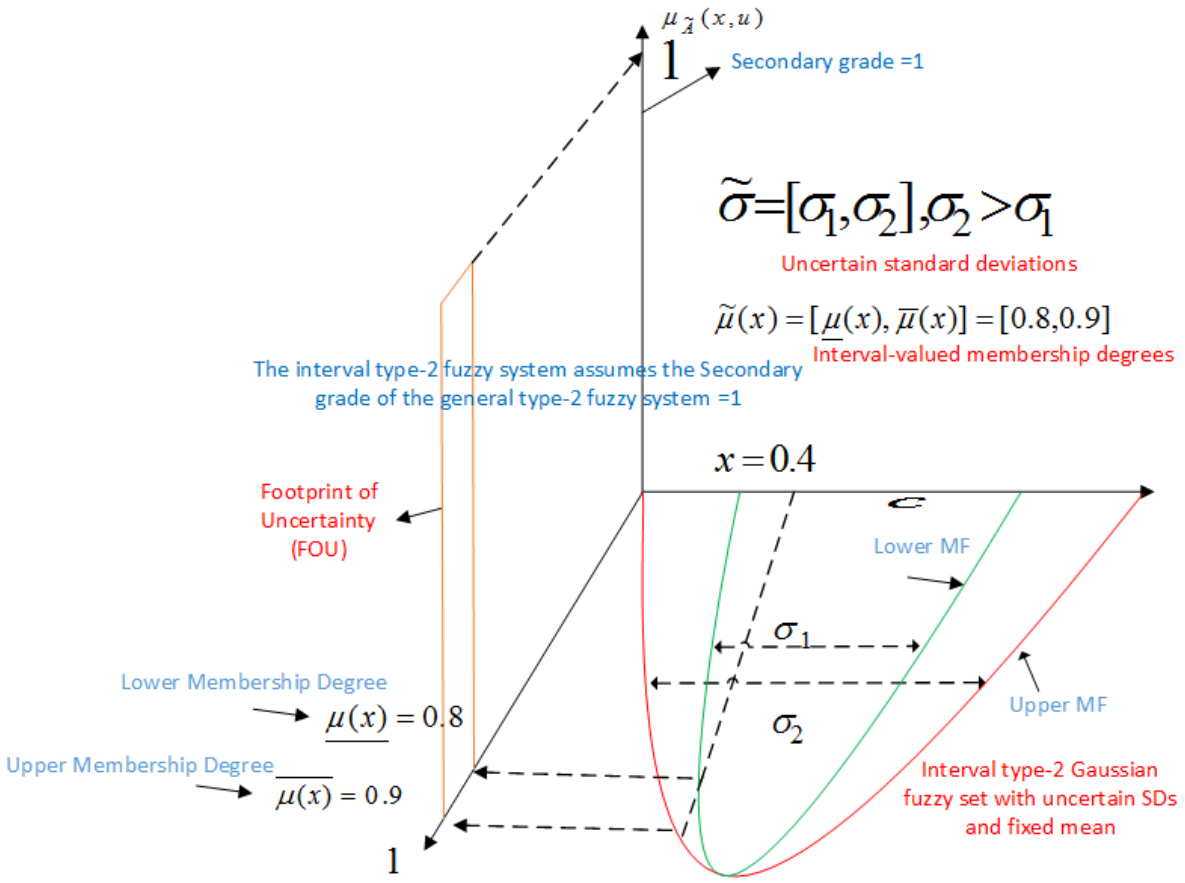

Fig.2 Interval Type-2 Gaussian fuzzy set with uncertain SDs

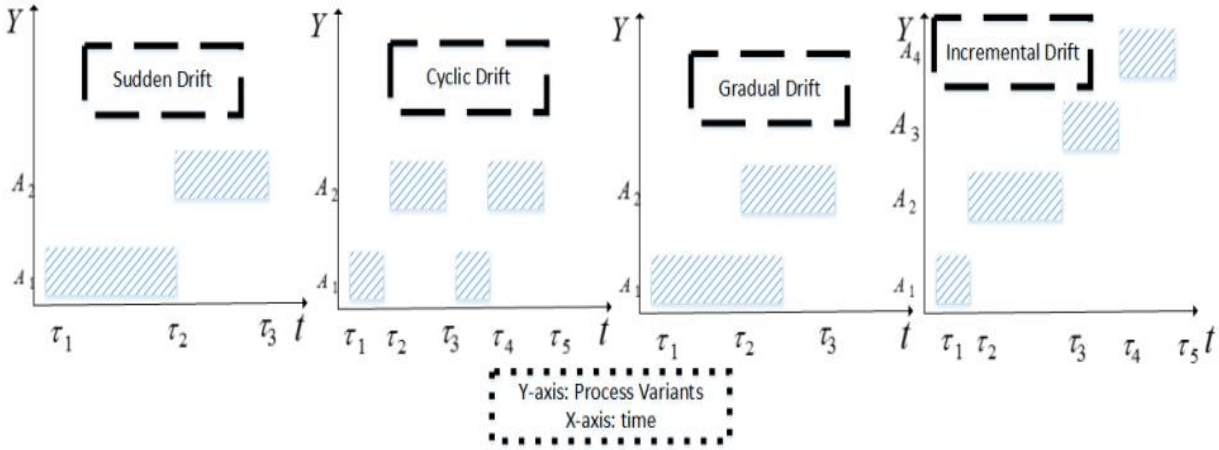

Fig.3 some drift variants

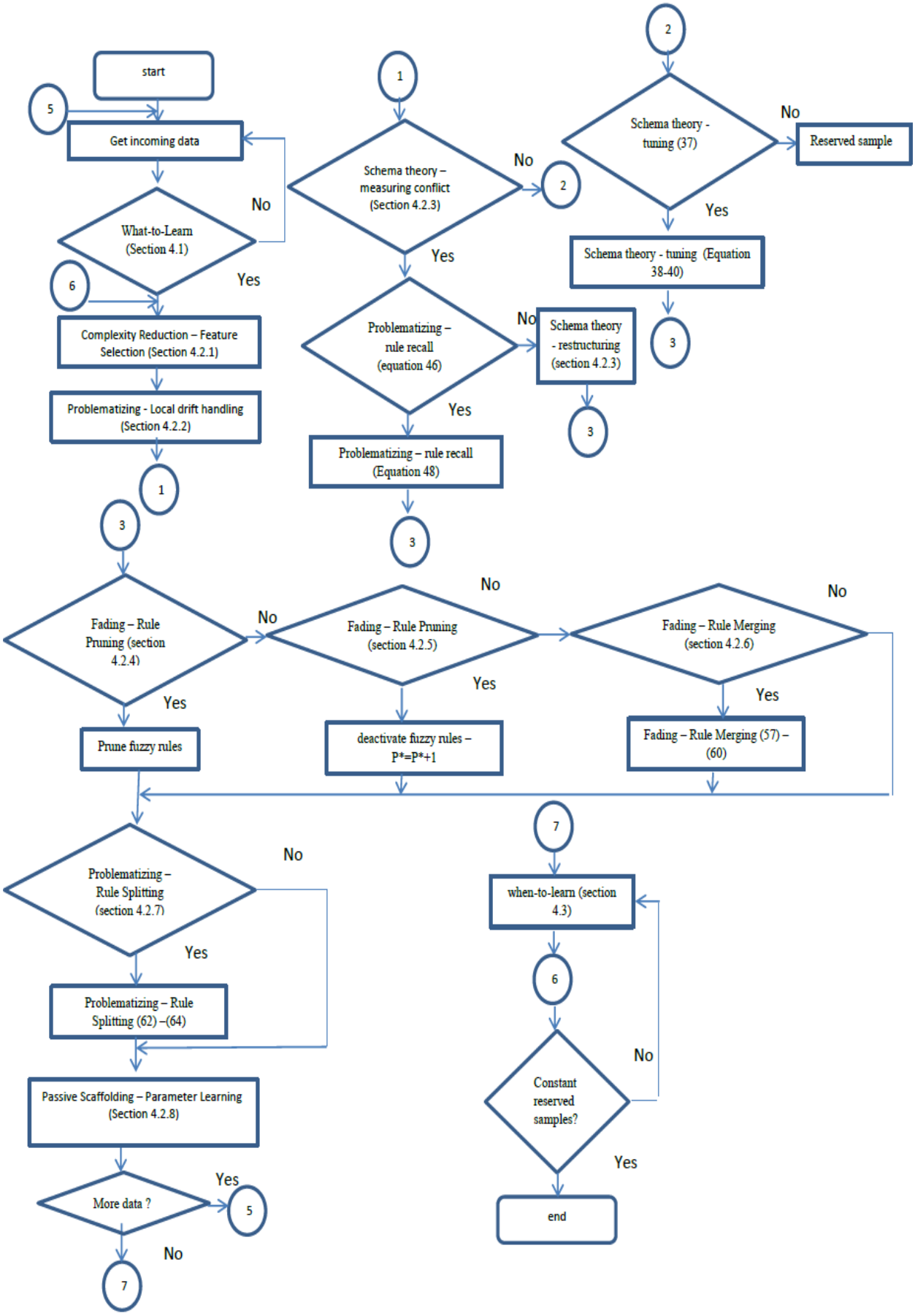

Fig.4 the learning algorithm of ST2Class

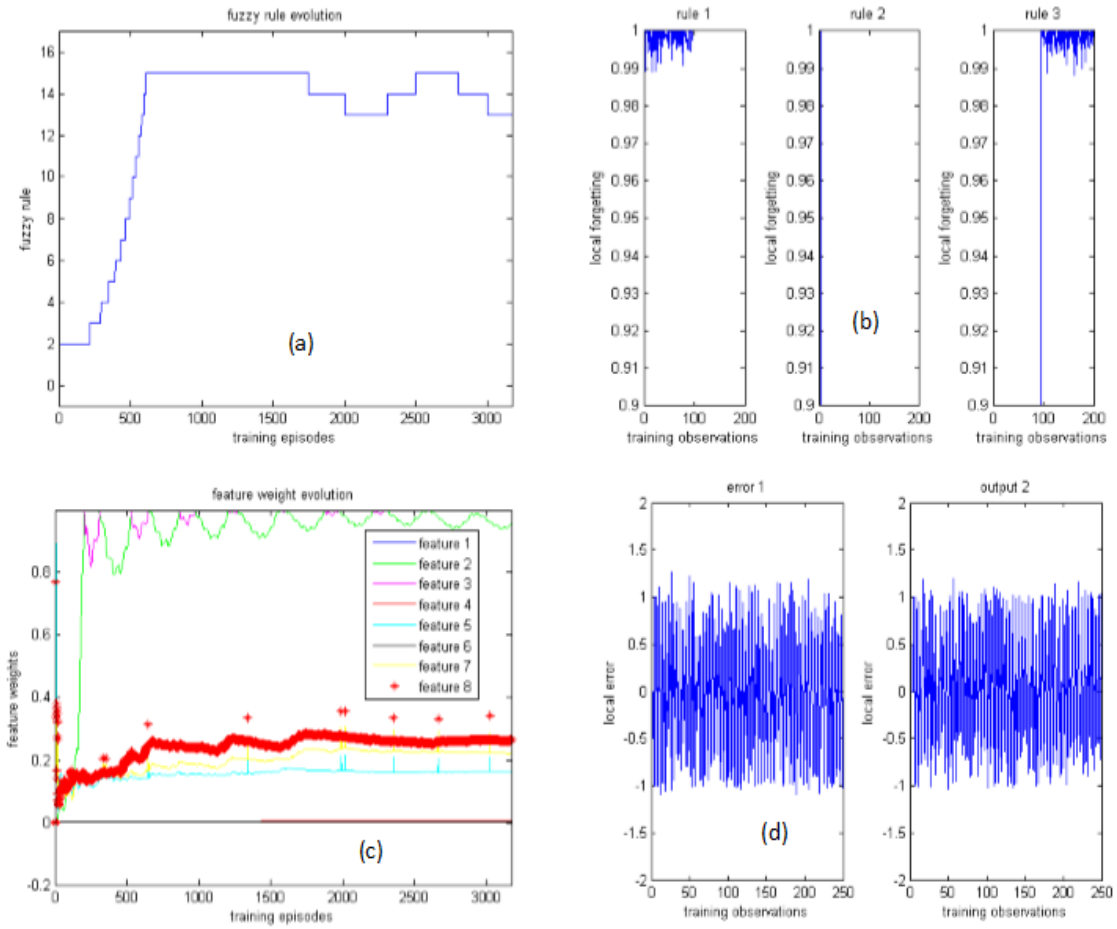

Fig.5(a) fuzzy rule evolution, (b) local forgetting evolution, (c) feature weight evolution, (d)system error

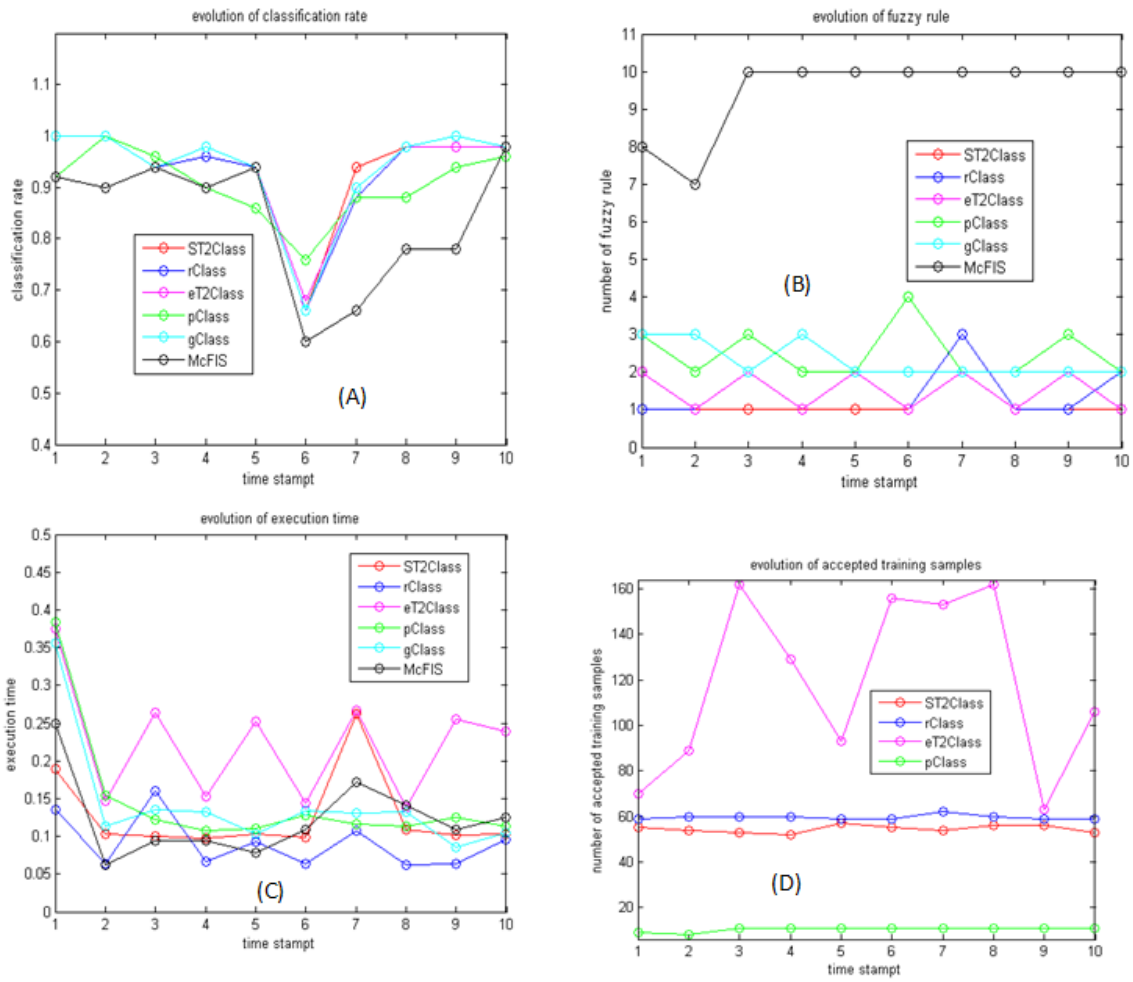

Fig. 6: (a) evolution of classification rate, (b) evolution of fuzzy rule; (c) evolution of execution time; evolution of accepted training samples

Algorithm 1: Pseudo code of ST2Class Classifier

Define: Input attributes and Desired class labels: $(X_n, T_n) = (x_1,\ldots,x_u, t_1,..,t_m)$

Reserved samples: $(XS_n, TS_n) = (xs_{1,n},\ldots,xs_{u,n}, ts_1,\ldots,ts_m)$

Predefined Thresholds: $\rho_1 = 0.5, \rho_2 = 0.01, \rho_3 = 0.8, \rho_4 = 1.1, \rho_5 = 0.9, \varpi = 10^{-15}$

/*Phase 4.1: What to Learn Strategy: Active Learning Strategy/*
For i=1 to P do
Compute the probability of the training sample to belong to existing clusters (10)
End for
Compute the sample entropy (12) and the actual labelling expense (13)
IF (14) Then
Label the data steam using either expert knowledge or self-labeling approach (15)-(17)
End IF
/*Phase 4.2.1: Input Weighting Mechanism /*
For j=1 to u do
For z=1 to u do $j \neq z$
Compute compression index (18)
End For
End For
For j=1 to u do
Assign feature weight to the input feature (20),(21)
End For
/*Phase 4.2.2: Local Drift Handling Strategy /*
For i=1 to P do
Compute the local error and forgetting level (22),(23)
End For
/*Phase 4.2.3: Schema theory /*
For i=1 to P do /*Measuring Conflict /*
Compute the posterior probabilities of the fuzzy rules (36)
Update the T2DQ method for all rules (28)
End For
Determine the winning rule $win = \arg\max_{i=1,\ldots,P} \widehat{P}(R_i|X)$
Compute the T2DS method (25)
For i=1 to P* do
update the T2P+ method for P* rules(45)
End For
IF (47)Then /* Rule recall phase-*
Activate rule recal mechanism (48)
Else IF (26) and (29) Then / *Restructuring Phase/*
Carry out the restructuring phase (30)-(34)
Assign the output parameters as (35), $N_{P+1} = 1, N_{P+1}{}^{o} = 1$
Else IF (37) /*Fine-tuning Phase/*
Update the premise parameters of the winning rule (38)-(40)
and increase the number of populations of winning rule $N_{win} = N_{win} + 1$, $N_{win}{}^{o} = N_{win}{}^{o} + 1$
Else IF
Append the reserved samples with the current sample $(XS_{NS+1}, TS_{NS+1}) = (X_N, T_N)$
End IF
/*Phase 4.2.4: Rule Pruning Strategy /*
For i=1 to P do
Enumerate the T2ERS and T2P+ method (42) and (45)
IF* (43)*Then
Prune the fuzzy rules
End IF
/*Phase 4.2.5: Rule Forgetting Strategy /*
IF (46)Then
Deactivate the fuzzy rules subject to the rule recall mechanism P*=P*+1
End For
/*Phase 4.2.6: Rule Merging Strategy /*
For i=1 to P do
For z=1 to P do
$i \neq z$
For j=1 to U do
Compute the vector similarity measure (49) based on shape-based (50) and distance-based (54) similarity measures
End For
End For
End For
IF (55)(56) Then
Coalesce the fuzzy rules (57)-(60)
End IF
/*Phase 4.2.7: Rule Splitting Mechanism /*
IF (61) Then
Split the winning cluster (62)-(64)
End IF
/*Phase 4.2.8: Passive Scaffolding Theories /*
For i=1 to P do
Adjust the fuzzy rule consequents (65)-(68)
For j=1 to u do
Adjust the dilation and translation parameters (71),(72)
End For
For o=1 to m do
Fine-tune the design factors (70)
End For
End For

Table 1. the efficacy of respective learning modules

ALGORITHMS		Iris+	Car+	Line	Sin	Section
ST2Class	Classification rate	**0.82±0.07**	**0.83±0.03**	**0.95±0.09**	**0.8±0.3**	A
	# of Rules	**1.5±0.1**	**3.2±0.5**	**1.1±0.1**	**1.9±0.7**	
	Time (s)	**0.08±0.02 s**	**0.17±0.07 s**	**0.17±0.05 s**	**0.18±0.03 s**	
	# of samples	**20.5±5**	**50.5±0.1**	**94.8±55.7**	136.3±51.8	
	# of parameters	**83.3**	**300**	**17.6**	30.4	
The generalized fuzzy rule with the non-linear Chebyshev function (model B)	Classification rate	0.8±0.17	0.8±0.11	0.94±0.09	0.77±0.2	
	# of Rule	1.7±0.5	3.7±0.5	1.6±0.7	1.8±0.6	
	Time (s)	0.13±0.02 s	0.24±0.07 s	0.27±0.06 s	0.25±0.05 s	
	# of samples	22.5±5	52.5±1.1	135.6±55.7	**125±51**	
	# of Parameters	122.4	332.8	32	36	
The standard interval type-2 fuzzy neural network with uncertain SDs (model C)	Classification rate	0.7±0.15	0.8±0.11	0.87±0.05	0.76±0.63	
	# of Rules	1.7±0.5	3.7±0.5	1.6±0.7	1.8±0.6	
	Time (s)	0.1±0.01 s	0.18±0.05 s	0.21±0.06 s	0.22±0.06 s	
	# of samples	21.4±4.4	52.5±1.1	135.6±55.7	155±51	
	# of Parameters	95.2	340.4	25.6	28.8	
without what-to-learn	Classification rate	0.8±0.18	0.78±0.13	0.94±0.1	0.77±0.24	B
	# of Rules	1.9±0.7	6±1.05	2±1.05	2.4±0.9	
	Time (s)	0.18±0.01 s	0.43±0.1 s	0.33±0.11 s	0.34±0.11 s	
	# of samples	34	130	200	200	
	# of Parameters	114.8	552	32	38.4	
without the complexity reduction aspect of the scaffolding theory	Classification rate	0.75±0.17	0.8±0.11	0.91±0.1	0.76±0.2	C
	# of Rules	1.7±0.5	3.7±0.5	1.6±0.7	1.8±0.63	
	Time (s)	0.15±0.01 s	0.19±0.07 s	0.23±0.02 s	0.21±0.05 s	
	# of samples	22.5±0.3	52.5±1.1	135.4±55.7	165.1±5.7	
	# of Parameters	104.4	345	25.6	28.8	
Without the problematizing aspect of the scaffolding theory	Classification rate	0.76±0.17	0.8±0.15	0.93±0.09	0.76±0.2	D
	# of Rules	1.7±0.5	3.2±0.8	1.5±0.53	1.7±0.5	
	Time (s)	0.18±0.06 s	0.17±0.06 s	0.27±0.05 s	0.19±0.06 s	
	# of samples	23.5±0.01	57.8±2.04	135.9±55.2	170.1±1.7	
	# of parameters	104.4	300	24	**27.2**	
Without the fading component of the scaffolding theory	Classification rate	0.81±0.17	0.75±0.16	0.93±0.09	0.77±0.25	E
	# of Rules	2.4±0.51	3.6±0.7	1.7±0.7	2.8±0.6	
	Time (s)	0.19±0.05 s	0.16±0.05 s	0.22±0.04 s	0.29±0.7 s	
	# of samples	31.2±0.3	53±2.3	134.7±56.2	143±1.6	
	# of parameters	140.8	336	34	43.2	

Table 2. Characteristic of data streams

Data stream	*Num of input attributes*	*Num of classes*	*Num of data points*	*Num of time stamps*	*Num of training samples in each time stamp*	*Num of testing samples in each time stamp*	*IF*
SEA	*4*	*2*	*100000*	*200*	*250*	*750*	*0.93*
Iris+	*4*	*4*	*450*	*10*	*34*	*11*	*0.25*
Car+	*6*	*2*	*1728*	*10*	*130*	*42*	*0.17*
Wine	*13*	*3*	*178*	*10*	*10*	*7*	*0.19*
Electricity pricing	*8*	*2*	*45312*	*10*	*3172*	*1359*	*0.15*
Weather	*4*	*2*	*60000*	*10*	*1000*	*5000*	*0.26*
Line	*2*	*2*	*2500*	*10*	*200*	*50*	*0*
Circle	*2*	*2*	*2500*	*10*	*200*	*50*	*0*
Sin	*2*	*2*	*2500*	*10*	*200*	*50*	*0*
Sinh	*2*	*2*	*2500*	*10*	*200*	*50*	*0*
Boolean	*3*	*2*	*1200*	*10*	*100*	*25*	*0*
Noise corrupted signal	*1*	*3*	*100K*	*10*	*7000*	*3000*	*0.25*
Gaussian	*4*	*2*	*800K*	*100*	*400*	*7200*	*0.33*

Table 3. the numerical results of consolidated algorithms

ALGORITHMS		ST2Class	eT2Class	McFIS	gClass	pClass	rClass
SEA dataset	Classification rate	**0.92±0.15**	0.88±0.23	0.73±0.11	0.9±0.1	0.89±0.1	0.86±0.2
	# of Rule	**1.09±0.42**	1.5±0.5	9.9±0.4	2.5±0.8	6.6±4.2	3.8±0.6
	Time(s)	0.19±0.03 s	0.34±0.11 s	**0.13±0.03 s**	0.21±0.05 s	0.42±0.3 s	0.25±0.06 s
	# of Parameters	**32.7±12.5**	61.3±21	52.6±1.8	82.3±27.3	157.3±101.9	136.1±20.8
	# of samples	(32.7)113.9	250	**10.9***	161.9	250	130.1
Electricity pricing dataset	Classification rate	**0.79±0.04**	0.77±0.08	0.5±0.1	0.79±0.05	0.68±0.1	0.77±0.07
	# of Rule	12.4±8.8	2.3±0.5	9.6±0.7	3.7±1.3	3.5±2.4	**2±0.82**
	Time(s)	1.96±5.98 s	5.1±1.3 s	**0.5±0.4 s**	2.8±0.3 s	7.1±4.4 s	4.2±1.1 s
	# of Parameters	1909.6±1352.7	354.2±74.4	**104±6.99**	392.2±132.7	226.8±95.6	216±88.8
	# of samples	(159.1)1532.3	3172	**10.6***	2271.4±330.3	3172	663.8±4.7
Weather dataset	Classification rate	**0.81±0.04**	0.8±0.03	0.61±0.14	0.8±0.02	0.8±0.04	0.73±0.07
	# of Rules	1.7±0.8	2.3±0.3	10	**1.3±0.7**	2.3±0.5	15.8±12.8
	Time(s)	1.1±0.2 s	1.8±0.1 s	**0.41±0.08 s**	0.93±0.08 s	1.8±0.22 s	1.7±1.04 s
	# of parameters	261.8±74.4	391±81	**108**	391±82.1	441±276.4	1706.4±1388.5
	# of samples	(204.6)482.6	1000	**11***	980.2	1000	222
Sin dataset	Classification rate	**0.8±0.3**	0.71±0.3	0.76±0.18	0.77±0.2	0.72±0.2	0.75±0.24
	# of Rule	**1.9±0.7**	1.9±1.1	9.1±1.2	2.7±0.8	3.3±1.2	2.7±0.5
	Time(s)	0.18±0.03 s	0.24±0.03 s	**0.08±0.03 s**	0.12±0.02 s	0.17±0.04 s	0.12±0.04 s
	# of Parameters	**30.4±11.8**	38±11.3	58.6±7.18	32.4+1.01	39.6	48.6±14.8
	# of samples	(43.3)93	200	**10.1***	81	200	96.9
Circle dataset	Classification rate	**0.9±0.06**	0.89±0.05	0.8±0.14	0.87±0.06	0.72±0.13	0.85±0.08
	# of Rule	**1.2±0.42**	1.2±0.6	9.8±0.42	2.3±1.5	2.8±1.1	1.8±0.8
	Time(s)	0.1±0.06 s	0.16±0.15 s	0.12±0.05 s	0.16±0.04 s	0.17±0.008 s	**0.08±0.03 s**
	# of Parameters	**19.2±6.7**	24±8.4	62.8±2.5	36.8±23.9	33.6	32.4±14.2
	# of samples	(41.6)93.3	200	**10.8***	179.2	200	79.6
Line dataset	Classification rate	**0.95±0.09**	0.94±0.1	0.84±0.13	0.93±0.1	0.91±0.07	0.94±0.2
	# of Rule	**1.1±0.1**	1.1±0.3	9.4±1	2	2.5±0.71	1.5±0.7
	Time(s)	0.17±0.05 s	0.13±0.04 s	0.1±0.03 s	0.14±0.06 s	0.25±0.0009 s	**0.09±0.02 s**
	# of parameters	**17.6±5.1**	22	60.4±6.4	24	30	27
	# of samples	(40.3)54.5	200	**10.4***	110.4	200	59.8
Sinh dataset	Classification rate	**0.71±0.06**	0.69±0.06	0.64±0.15	0.7±0.06	0.71±0.09	0.7±0.03
	# of Rule	**1.2±0.4**	1.2±0.7	10	4.6±1.3	3.6±1.9	2±0.9
	Time(s)	0.13±0.03 s	0.18±0.05 s	**0.1±0.02 s**	0.23±0.08 s	0.27±0.01 s	0.12±0.05 s
	# of Parameters	**19.2±6.7**	24±8.43	64	73.6±20.2	43.2	36±16.9
	# of samples	(22.2)95.4	200	**11***	155.4	200	115.5±22.2
Iris+ dataset	Classification rate	0.82±0.07	**0.83±0.18**	0.77±0.25	0.68±0.3	0.73±0.18	0.74±0.2
	# of Rule	1.5±0.1	**1.4±0.5**	8.3±2.1	2.3±0.5	4.6±1.9	2.1±0.32
	Time (s)	0.13±0.02 s	0.06±0.02 s	0.03±0.03 s	**0.02±0.008 s**	0.09±0.07 s	0.04±0.002 s
	# of Parameters	**83.3±23.7**	113.4±41.8	113.8±29.04	149.5±31.4	138	147±22.1
	# of samples	(20)22.5	34	**9.3***	20.8	34	26.1
Car+ dataset	Classification rate	**0.82±0.12**	0.77±0.14	0.6±0.2	0.78±0.2	0.77±0.1	0.75±0.14
	# of Rule	**1.2±0.6**	1.5±0.5	9.3±0.9	3.1±1.2	2.5±0.8	3.4±3.9
	Time(s)	0.09±0.02 s	0.16±0.05 s	**0.07±0.01 s**	0.11±0.07 s	0.1±0.02 s	0.11±0.05 s
	# of Parameters	110.4±58.2	156±54.8	**80.4±7.6**	210.8±81.4	140±47.6	238±276.5
	# of samples	(27.7)61.3	130	**10.3***	72.2±36.4	130	73.9

Noise corrupted signal dataset	Classification rate	**0.74±0.12**	0.73±0.10	0.69±0.14	0.73±0.16	0.72±0.12	0.73±0.12
	# of Rule	**1**	1.4±0.5	9.6±3.4	**2**	3±1.2	2.5±0.1
	Time(s)	5.4±0.17 s	10.4±3.1 s	**2.6±1 s**	10.2±0.3 s	6.4±0.7 s	3.9±0.06 s
	# of Parameters	**9**	12.6±4.64	36.2±14.6	**28**	24	36
	# of samples	3404.9±1.9	7000	**10.6***	6907.2	7000	**1439**
Boolean dataset	Classification rate	**0.92±0.2**	0.91±0.15	0.86±0.2	0.89±0.2	0.83±0.2	0.88±0.2
	# of Rules	**1.3±0.5 s**	1.8±0.4	7.4±1.9	2.6±1.1	2.6±0.8	1.4±0.5
	Time (s)	**0.05±0.01**	0.13±0.02 s	0.05±0.05 s	0.07±0.02 s	0.08±0.002 s	0.06±0.01 s
	# of Parameters	**37.7±14**	63±14.8	65.2±15.6	52±21.5	52	39.2±14.4
	# of samples	(24.5)44.7	100	**8.4***	76.1	100	39.3
Gaussian dataset	Classification rate	**0.75±0.1**	0.72±0.13	0.66±0.13	0.74±0.01	0.74±0.2	0.72±0.3
	# of Rules	**1.01±0.1**	1.4±0.5	8.05±1.9	2.9±1	3.3±1.1	2.1±0.3
	Time(s)	1.3±0.06 s	1.6±0.3 s	0.93±0.4 s	**0.7±0.08 s**	0.74±0.05 s	0.9±0.03 s
	# of Parameters	**17.7±1.7**	35.5±12.4	34.2±7.6	60.7±20.6	49.5±17	50.2±6.9
	# of samples	(179.3)185	400	**9.05***	140.1	400	100
Wine dataset	Classification rate	**0.96±0.08**	0.94±0.07	0.69±0.13	0.95±0.05	0.86±0.07	0.87±0.08
	# of Rules	**2**	2	15	2	2.1±0.32	2
	Time(s)	**0.1±0.006**	0.11±0.001	0.11±0.006	0.15±0.006	0.2±0.16	0.12±0.006
	# of Parameters	864	864	**253**	526	470.4	526
	# of samples	51	161	**16**	144.7	161	51.7±0.5

*:data are fully labelled, (): the number of self-labelled data

Table 4. Summary of learning features of benchmarked algorithms

	Hidden Layer	Output Layer	What-to-learn	How-to-learn	When-to-learn	Classifier type
ST2Class	Interval type-2 multivariate Gaussian function	Interval-valued Wavelet functions	Active Learning with self-labelling	Type-2 Schema and Scaffolding theory	Sample reserved	Semi-supervised
eT2Class	Interval type-2 multivariate Gaussian function	Interval-Valued Chebyshev function	N/A	Schema theory	Sample reserved	Fully-supervised
rClass	Recurrent type-1multivariate Gaussian function	Type-1 Chebyshev function	Active Learning with ground truth	Schema theory	Sample reserved	Semi-supervised
gClass	Multivariate type-1 Gaussian function	Type-1 Chebyshev function	Active learning with ground truth	Schema theory+ input weighting	Sample reserved	Semi-supervised
McFIS	Uni-variable type-1 Gaussian function	Type-1 First order TSK	Sample-deletion	Schema theory	Sample reserved	Fully supervised
pClass	Multivariate ype-1 Gaussian functions	Type-1 First order TSK	N/A	Schema theory	N/A	Fully-supervised

Table 5. Classifier ranking according to classification rate

Study cases	ST2Class (CR, R, ET, RB)	eT2Class (CR, R, ET, RB)	McFIS (CR, R, ET, RB)	gClass(CR, R, ET, RB)	pClass(CR, R, ET, RB)	rClass(CR, R, ET, RB)
Sea dataset	(1,1,2,1)	(4,2,5,3)	(6,6,1,2)	(2,3,3,4)	(3,5,6,6)	(5,4,4,5)
Electricity pricing dataset	(1,6,2,6)	(4,2,5,4)	(6,5,1,1)	(2,4,3,5)	(5,3,6,3)	(3,1,4,2)
Weather dataset	(1,2,3,2)	(3,3,4,5)	(6,5,1,1)	(2,1,2,4)	(4,4,6,5)	(5,6,4,6)
Sin dataset	(1,1,5,1)	(6,2,6,3)	(3,6,1,6)	(2,3,2,2)	(5,5,4,4)	(4,3,3,3)
Circle dataset	(1,1,2,1)	(2,2,5,2)	(5,6,3,6)	(3,4,4,5)	(6,5,6,6)	(4,3,1,4)
Line dataset	(1,1,5,1)	(2,2,3,2)	(6,6,1,6)	(4,4,4,3)	(5,5,6,5)	(3,3,1,4)
Sinh dataset	(1,1,3,1)	(5,2,4,2)	(6,6,1,5)	(4,5,5,6)	(2,4,6,4)	(3,3,2,4)
Iris+ dataset	(2,2,6,1)	(1,1,4,2)	(4,6,2,3)	(6,4,1,6)	(5,5,6,4)	(4,3,3,5)
Car dataset	(1,1,2,2)	(4,2,6,3)	(6,6,1,1)	(2,4,5,5)	(3,3,4,3)	(5,5,4,6)
Noise corrupted signal dataset	(1,1,3,1)	(2,2,6,2)	(6,6,1,6)	(4,3,5,4)	(5,5,4,3)	(3,4,2,5)
Boolean dataset	(1,1,1,1)	(2,3,6,5)	(5,6,2,6)	(3,5,5,4)	(6,4,5,3)	(4,2,3,2)
Gaussian dataset	(1,1,5,1)	(4,1,6,3)	(6,6,2,2)	(2,4,1,6)	(4,5,3,4)	(5,3,3,5)
Wine dataset	(1,1,1,4,2)	(3,1,2,4,5)	(6,3,3,1,1)	(2,1,5,3,4)	(5,2,6,2,5)	(4,1,4,3,5)
Average	(**1.1**,**1.6**,3.2,**1.6)**	(3.2,5,3)	(5.4,5.8,**1.4**,3.8)	(3,3.7,3.3,4.5)	(4.4,4.4,5.2,4.2)	(4,3.3,2.8,4.3)

CR=Classification Rate, R=Rule, RB= Rule Base Parameters, ET=Execution Time

Dr. Mahardhika Pratama received his PhD degree from the University of New South Wales, Australia in 2014. He completed his PhD in 2.5 years with a special approval of the UNSW higher degree committee due to his outstanding PhD research achievement. Dr. Pratama is currently working at the Department of Computer Science and IT, La Trobe University, Melbourne, Australia as Lecturer (Assistant Professor). Prior to joining La Trobe University, he was with the Centre of Quantum Computation and Intelligent System, University of Technology, Sydney as a postdoctoral research fellow of Australian Research Council Discovery Project. Dr. Pratama received various competitive research awards in the past 5 years, namely the Institution of Engineers, Singapore (IES) Prestigious Engineering Achievement Award in 2011, the UNSW high impact publication award in 2013 and 2014. Dr. Pratama has published over 30 high-quality papers in journals and conferences, and has been invited to deliver keynote speeches in international conferences. Dr. Pratama is a member of IEEE, IEEE Computational Intelligent Society (CIS) and IEEE System, Man and Cybernetic Society (SMCS), and Indonesian Soft Computing Society (ISC-INA) and severs as a reviewer in some top tier journals such as: *IEEE Transactions on Fuzzy Systems, IEEE Transactions on Cybernetics, Neurocomputing, Soft Computing, and Evolving Systems*. His research interests involve machine learning, computational intelligent, evolutionary computation, fuzzy logic, neural network and evolving adaptive systems.

Professor Jie Lu received her PhD from Curtin University, Australia. She is the Associate Dean Research in the Faculty of Engineering and Information Technology (FEIT) at the University of Technology, Sydney (UTS). Her main research interests lie in the area of decision support systems, recommender systems, knowledge-based prediction and warning systems, fuzzy information processing and e-Service intelligence. She has published six research books (including "Multi-Level Decision Making", 2015; "Cognition-Driven Decision Support for Business Intelligence", 2010; "Multi-objective group decision making with fuzzy set techniques", 2007) and 350 papers in refereed journals and conference proceedings. She has won seven Australian Research Council (ARC) Discovery Project grants and 10 other research grants; has been the supervisor of 25 Ph.D. students. She received the first UTS Research Excellence Medal for Teaching and Research Integration in 2010, and 2012 FLINS Gold Medal (Fuzzy Logic and Intelligent technologies in Nuclear Science). She serves as Editor-In-Chief for Knowledge-Based Systems (Elsevier), Editor-In-Chief for International Journal on Computational Intelligence Systems (Atlantis), Associate Editor for IEEE Trans on Fuzzy Systems, editor for book series on Intelligent Information Systems (World Scientific), and has served as a guest editor of eight special issues for international journals, chairs for ten international conferences as well as having delivered many keynote speeches at international conferences/

Dr. Edwin Lughofer received his PhD. degree from the Department of Knowledge-Based Mathematical Systems, Johannes Kepler University Linz, where he is now employed as senior/key researcher. During the past 10-12 years, he has participated in several research projects on European and national level. In this period, he has published around 130 journal and conference papers in the fields of evolving fuzzy systems, machine learning and vision, data stream mining, active learning, classification and clustering, fault detection and diagnosis, condition monitoring as well as human-machine interaction, including a monograph on 'Evolving Fuzzy Systems' (Springer, Heidelberg) and an edited book on 'Learning in Non-stationary Environments' (Springer, New York). He is associate editor of the international journals IEEE Transactions on Fuzzy Systems (IEEE press), Evolving Systems (Springer), Information Fusion (Elsevier) and Soft Computing (Springer), the general chair of the IEEE Conference on Evolving and Adaptive Intelligent Systems 2014 and Area chair of the FUZZ-IEEE 2015 conference in Istanbul. He serves as program

committee member of several international conferences, and acts as a peer-reviewer for 20+international journals. In 2006 he received the best paper award (as main author) at the International Symposium on Evolving Fuzzy Systems, and in 2013 the best paper award (as co-author) at the IFAC conference in Manufacturing Modeling, Management and Control Conference (800 participants). He is currently key researcher in the national K-Project 'imPACts' (20 partners, coordination: Recendt GmbH).

Professor Guangquan Zhang received the Ph.D. degree in applied mathematics from Curtin University, Australia. He received Queen Elizabeth II (QEII) fellowship in 2005. He is currently an Associate Professor in the Faculty of Engineering and Information Technology at the University of Technology Sydney, Australia. From 1993 to 1997, he was a full Professor in the Department of Mathematics, Hebei University, China. His main research interests lie in the area of multi-objective, bi-level, and group decision making under uncertainty, fuzzy sets, fuzzy measure, fuzzy optimization and fuzzy machine learning. He has published five monographs, five reference books, and over 300 papers including more than 160 refereed journal articles. He has won six Australian Research Council (ARC) Discovery Project grants and many other research grants for about $3M. He has served as a Guest Editor for five special issues of international journals. He has also been invited to serve on the Editorial Boards and Executive Advisory Boards in a number of international journals.

Dr. Sreenatha Anavatti received his Ph.D degree in aerospace engineering from the Indian Institute of Science in 1990, his Bachelor of Engineering degree in mechanical engineering from the Mysore University, India in 1984. He is currently a Senior Lecturer at the School of Aerospace, Civil and Mechanical Engineering (ACME), University of New South Wales at Australian Defence Force Academy (UNSW@ADFA), Australia. His current research interests include control systems, flight dynamics, robotics, aeroelasticity, artificial neural networks, fuzzy systems and unmanned systems.


Mahardhika Pratama


Jie Lu


Edwin Lughofer


Guangquan Zhang


Sreenatha Anavatti