Knowledge-Based Systems 152 (2018) 136–162

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Recent advances in neuro-fuzzy system: A survey

K.V. Shihabudheen*, G.N. Pillai

Department of Electrical Engineering, Indian Institute of Technology, Roorkee, India

ARTICLE INFO

Article history:
Received 27 September 2017
Revised 1 March 2018
Accepted 10 April 2018
Available online 11 April 2018

Keywords:
Neuro-fuzzy systems
Self organizing
Support vector machine
Extreme learning machine
Recurrent

ABSTRACT

Neuro-fuzzy systems have attracted the growing interest of researchers in various scientific and engineering areas due to its effective learning and reasoning capabilities. The neuro-fuzzy systems combine the learning power of artificial neural networks and explicit knowledge representation of fuzzy inference systems. This paper proposes a review of different neuro-fuzzy systems based on the classification of research articles from 2000 to 2017. The main purpose of this survey is to help readers have a general overview of the state-of-the-arts of neuro-fuzzy systems and easily refer suitable methods according to their research interests. Different neuro-fuzzy models are compared and a table is presented summarizing the different learning structures and learning criteria with their applications.

1. Introduction

Concerns over computational speed, accuracy and complexity of design made researchers think about soft computing techniques for modelling, prediction and control applications of dynamic nonlinear systems. Artificial neural networks (ANN) and fuzzy logic systems are commonly used soft computation techniques. Fusion of these two techniques is proliferating into many scientific and engineering fields to solve the real world problems. Use of fuzzy logic can directly improve the reasoning and inference in a learning machine. The qualitative, albeit imprecise, knowledge can be modeled to enable the symbolic expression of machine learning using fuzzy logic. Use of neural networks incorporates the learning capability, robustness and massive parallelism into the system. Knowledge representation and automated learning capability of the neuro-fuzzy system make it a powerful framework for machine learning problems [1].

Takagi–Sugeno–Kang (TSK) inference system is the most useful fuzzy inference system and is a powerful tool for modeling of nonlinear dynamic systems. The main advantage of TSK system modeling is that it is a 'multimodal' approach which can combine linear submodels to describe the global behavior of complete complex nonlinear dynamic system [2]. One of the popular neuro-fuzzy approaches, adaptive neuro-fuzzy inference system (ANFIS) has been utilized by researchers for regression, modeling, prediction and control problems [3,4]. ANFIS uses TSK type fuzzy inference system in a five layered network structure. ANFIS defines two sets of parameters namely premise parameters and consequent parameters. The fuzzy if–then rules define the relationship between the two sets of parameters. The main drawback of ANFIS is that it is computationally intensive and generates complex models for even relatively simple problems.

Nowadays learning methods and network structure in conventional neuro-fuzzy networks are improved to achieve better results in terms of accuracy and learning time. A highly efficient neuro-fuzzy system should have the following characteristics, (1) fast learning, (2) on-line adaptability, (3) self-adjusting with the aim of obtaining the small global error possible, and (4) small computational complexity. This paper surveys different improved neuro-fuzzy systems available in literatures based on their learning criteria, adaptation capability, and network structure.

Different survey papers are available in the literature. J. Jang [5] gives different learning methods of ANFIS and its application in control systems. Case studies are included to support the study. In early 2000, an exhaustive survey on neuro-fuzzy rule generation has been performed in [1]. It explains different ways of hybridization of neural networks and fuzzy logic for rule generation. Neuro-fuzzy rule generation using evolutionary algorithm also explained. R. Fuller [6] presents a survey of neuro-fuzzy systems and explained different types of methods to build a neuro-fuzzy system. The preliminaries of different modeling and identification techniques using the neuro-fuzzy system are detailed in [7]. In 2001, A. Abraham [8] presented different concurrent models, cooperative models and fully fused models of neuro-fuzzy systems including the advantages and deficiencies of each model. In 2004, a comparative study of different hybrid neuro-fuzzy systems such as ANFIS, FALCON, GARIC, SONFIN and EFuNN was presented [9].

* Corresponding author.
E-mail addresses: shih1dee@iitr.ac.in, shihab4806@gmail.com (K.V. Shihabudheen), gnathfee@iitr.ac.in (G.N. Pillai).

https://doi.org/10.1016/j.knosys.2018.04.014
0950-7051/

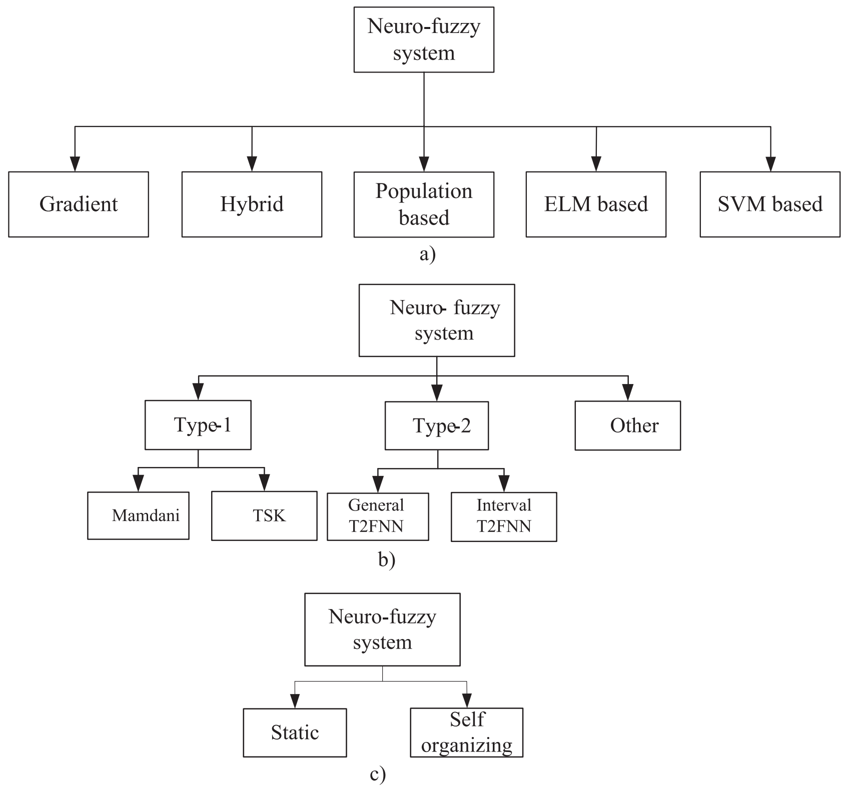

Fig. 1. Classification of neuro-fuzzy techniques based on (a) algorithm, (b) fuzzy method, (c) structure.

But all the explained techniques use gradient descent techniques to tune its parameters, which will create issues related to local minima trapping and convergence. In [10], a survey of hybrid expert systems including the neuro-fuzzy systems is presented. The different applications of neuro-fuzzy systems such as student modeling system, electrical and electronics system, economic system, feature extraction and image processing, manufacturing, forecasting, medical system and traffic control are briefed in [11]. The current trends of the hardware implementation of the neuro-fuzzy system are also explained [12]. A detailed survey of neuro-fuzzy applications in technical diagnostics and measurement is presented in [13]. Applications of neuro-fuzzy systems in the business systems are presented in [14]. Neuro-fuzzy systems are applied in variety of applications like nonlinear modeling [15], control systems [16–18], power systems [19], biometrics [20–22] etc.

To date, there has been no detailed and integrated categorization of the various neuro–fuzzy models based on its structure and learning principle. Classification of neuro-fuzzy techniques based on learning algorithm is shown in Fig. 1(a). Gradient type learning techniques, hybrid techniques, population based computational techniques, extreme learning techniques and support vector machine learning techniques are the major categorization used in neuro-fuzzy systems. The hybrid technique includes hybridization of different techniques such as back-propagation, least square method, clustering algorithm, Kalman filter (KF), etc. As shown in Fig. 1(b) neuro-fuzzy techniques can be classified into Type-1 and Type-2 according to fuzzy methods used. In Mamdani type system, consequents and antecedents are related by the min operator or generally by a t-norm, whereas in TSK system, the consequents are functions of inputs. Other logical type reasoning methods are also used in the neuro-fuzzy system, where consequents and antecedents are related by fuzzy implications. Based on the structure, neuro-fuzzy techniques can be classified into static and self organizing fuzzy neural networks (Fig. 1(c)). In static fuzzy neural networks, the structure of the networks is not changing at the time of training, whereas in self organizing fuzzy neural networks, structure and rules of the fuzzy neural network are self evolving based on adaptation algorithm.

This paper mainly focuses the survey of classification of neuro-fuzzy systems based on their learning algorithm, fuzzy method and structure from 2000 to till date. Firstly, the paper describes different neuro-fuzzy systems based on the classification in Fig. 1(a). After that, it explains details of neuro-fuzzy systems based on the classification given in Fig. 1(b) and (c). Section 2 explains different neuro-fuzzy systems based on the learning algorithm used. It clearly explains the diverse collection of gradient based, hybrid, population, ELM and SVM based neuro-fuzzy systems. The neuro-fuzzy systems based on fuzzy type are detailed in Section 3. This section mainly focuses on the details of Type-2 neuro-fuzzy systems. In Section 4, neuro-fuzzy systems based on the structure are described, in which different self-organizing neuro-fuzzy systems are explained. Detailed comparative simulation study of different type-1 ELM based neuro-fuzzy systems are proposed in Section 5. Section 6 gives comments and remarks of different neuro-fuzzy systems and conclusions are drawn in Section 7.

2. Neuro-fuzzy systems based learning algorithms

This section explains different neuro-fuzzy systems whose training is performed by gradient, hybrid, population, ELM and SVM based techniques.

2.1. Gradient based on neuro-fuzzy systems

Gradient technique is based on the steepest descent which finds optimum based on the negative of the gradient of the objective function. Back-propagation based learning techniques are one of the important sections of gradient techniques. S. Horikawa [23] proposes a modeling approach using fuzzy neural networks, where parameters are identified by the back-propagation algorithm. Three types of fuzzy models are developed for obtaining reasoning capability. Firstly, a zero order TSK model is considered where consequent parameters are taken as constant. In the second model, a first order based TSK model has been developed, which uses first order linear equation in consequent part. In case of third model, Mamdani based fuzzy system is used with fuzzy variables in consequent part. In all the three models, back-propagation algorithm is used to find the parameters.

Y. Shi [24] proposes a new approach for training of neuro-fuzzy system using back-propagation algorithm, that fuzzy rules or membership functions can be learned without changing the form of fuzzy rule table used in usual fuzzy applications. A multi-input multi-output (MIMO) neuro-fuzzy network is proposed for forecasting electrical load time series [25]. It uses modified back-propagation algorithm for training the parameters including adaptive learning rate and oscillation control.

A TSK based neuro-fuzzy system is developed for high interpretability [26,27]. In both works, the gradient descent algorithm is used to adjust the parameters of the fuzzy rule and consequent part. An ensemble neuro-fuzzy technique utilizing AdaBoost and the back-propagation algorithm are presented in [28]. A modified gradient descent algorithm is developed for efficient tuning of neuro-fuzzy system in [29]. In this work, the conventional setting of the error function and the learning formula of gradient descent algorithm are modified in zero-order Takagi–Sugeno inference systems by taking the reciprocals of the widths of the corresponding Gaussian membership functions.

In [30], a recurrent fuzzy neural network with online supervised learning is proposed. The number of rules is created with structure learning using aligned- clustering-based partition scheme. All the parameters of the membership function, feedback structure, and consequent part are calculated using ordered derivative based back-propagation. In [31], a TSK based self-organizing neuro-fuzzy scheme is developed. It includes efficient feature selection scheme and pruning scheme which prune incompatible rules, zero rules, and less used rules. The parameters are trained by error back-propagation algorithm. A gradient learning based growing and pruning neuro-fuzzy algorithm called GP-FNN is proposed in [32]. All the parameters are learned using the supervised gradient descent method. Wide ranges of simulations are conducted in the area such as, Mackey–Glass chaotic time series prediction, nonlinear identification and oxygen concentration control in wastewater treatment processes.

K. Subramanian proposed a complex-valued TSK based neuro-fuzzy system in [33]. A meta-cognitive complex-valued neuro-fuzzy inference system using TSK which utilizes complex valued neural networks was proposed in [34]. Both neuro-fuzzy systems use Wirtinger calculus based fully complex-valued gradient-descent algorithm for parameter extraction.

W. Zhao [35] proposes a fuzzy-neuro system that utilizes the idea of local learning. It uses TSK fuzzy system where the consequent parameters are transformed into a dependent set on the premise parameters which is then tuned by gradient descent method. Simulations on modeling and prediction show that proposed neuro-fuzzy techniques produce effectiveness and more interpretability compared to other neuro-fuzzy systems. A neuro-fuzzy system using Mamdani type inference is developed based on forward recursive input-output clustering and gradient descent algorithm in [36].

It has been shown that gradient descent-based learning methods are generally very slow due to improper learning steps and may easily converge to local minima. Many iterative learning steps may be required for such learning algorithms in order to obtain better learning performance. It was also proved that training using gradient descent technique alone may possess weak firing strength and initial setting of fuzzy rule is difficult for large training data [37]. Moreover, the gradient-based approaches for training recurrent neural networks may cause stability problems and may be insensitive to long-term time dependencies [38].

Different gradient based neuro-fuzzy inference systems including its type of membership, learning criteria, fuzzy technique used and applications are shown in Table 1.

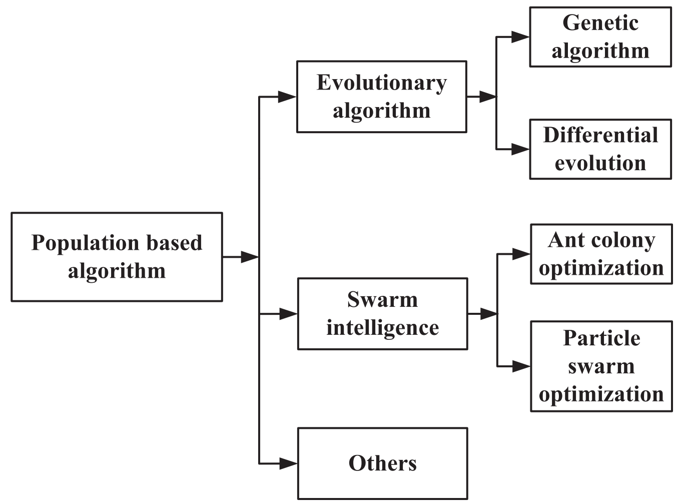

Fig. 2. Classification of the population based techniques.

2.2. Hybrid neuro-fuzzy systems

Neuro-fuzzy systems with single learning technique such as back-propagation require lots of burdens to train the parameters if the network structure and parameters become large. In such cases, a single learning technique may not get the satisfactory result especially if large training data are used. Hybrid neuro-fuzzy system combines two or more learning technique to find the parameters, which is used to achieve stable and fast convergence. Most of the neuro-fuzzy systems that use hybrid learning technique will come under the category of self-organizing neuro-fuzzy systems. Hence the detailed study of neuro-fuzzy systems that uses hybrid learning technique is explained in Section 4.

2.3. Population based neuro-fuzzy systems

Like back-propagation and other supervised learning algorithms using the gradient based technique, neuro-fuzzy systems also suffer from the problem of local minima and overfitting. The population based algorithms do not require nor use information about differentials, and hence, they are most effective in case where the derivative is very difficult to obtain or even unavailable. The classification of the population based technique is shown in Fig 2. This section mainly describes the genetic algorithm (GA), differential evolution (DE), ant colony optimization (ACO) and particle swarm optimization (PSO) based neuro-fuzzy techniques. Neuro-fuzzy systems with artificial bee colony (ABC), simulated annealing, Cuckoo search algorithm and Tabu search algorithm are also included.

2.3.1. Genetic algorithm

GA is a biologically inspired problem solving mechanisms [39,40]. The GA is based on the genetic process such as selection, mutation and crossover. GA performs the search on the coding of the data instead of directly on the raw data. This feature allows GAs to be independent of the continuity of the data and the existence of derivatives of the functions as needed in some conventional optimization algorithms. The coding method in GAs allows

Table 1
Details of different gradient based neuro-fuzzy inference systems.

	Membership	Structure	Learning criteria	Application	Consequent
S. Horikawa [23]	Bell	Static	Back-propagation algorithm	Nonlinear system modeling	TSK/Mamdani
C. Juang [30]	Gaussian membership function	Recurrent self-organizing	Ordered derivative based back-propagation	Online system identification, inverse control	Mamdani
Y. Shi [24]	Gaussian/triangular-type	Static	Back-propagation algorithm	Nonlinear function identification	TSK
A. K. Palit [25]	Gaussian	Static	Back-propagation algorithm with momentum	Forecasting Electrical Load Time Series	TSK
G. Castellano [26]	Gaussian	Static	Back-propagation algorithm	Nonlinear modeling, real world regression	TSK
D. Chakraborty [31]	Gaussian	Self-organizing	Back-propagation algorithm	Classification	TSK
GP-FNN [32]	RBF	Self-organizing	Supervised gradient descent method.	Time series prediction, identification and control	TSK
MCNFIS [34]	Gaussian	Self-regulate	Fully complex-valued gradient-descent algorithm	Function approximation, wind speed prediction, multi class classification	TSK
Local learning neuro-fuzzy [35]	Gaussian	Static	Integrated gradient descent	Nonlinear dynamic modeling, prediction of concrete compressive strength	TSK
J. Qiao [36]	Gaussian	Self-organizing	Gradient descent	Dynamical system modeling, function approximation	Mamdani

them to easily handle optimization problems with multi-parameter or multi-model, which is rather difficult or impossible to be treated by classical optimization methods. The parameter tuning of neuro-fuzzy systems can easily achieve using GA.

In [41], a recurrent based neuro-fuzzy system (TRFN) is proposed where parameters are optimized through GA. The new rules are inserted based on fuzzy clustering mechanism and the parameters are tuned through GA. The tuning TRFN is then extended by hybridizing GA and PSO [42]. In [43], a radial basis function neuro-fuzzy network with GA based parameter identification technology is designed for providing robust control and transient stability in power system. GA tuned self organizing TSK based fuzzy neural network is provided in [44]. Simulations on classification problems show that proposed technique gives higher classification rate compared benchmark neuro-fuzzy systems. An evolutionary based static neuro-fuzzy system utilizing GA is proposed for surface roughness evaluation [45]. An optimally tuned ANFIS using subtractive clustering and GA is proposed for electricity demand forecasting [46]. The complexity of ANFIS is reduced by application of GA which is used to find optimum value of cluster radius and then provides an optimum number of rules and minimum error. The utility of the proposed method is verified through forecasting industrial electricity consumption in Iran by comparing ANFIS technique. It has been noted only less number of researches are reported in the area of self organizing neuro-fuzzy system through GA.

2.3.2. Differential evolution

Differential evolution (DE) is parallel search method utilized by bio-inspired mechanisms, crossover, mutation and selection as like GA [47,48]. For every element in the population, a mutant element is generated by adding difference between two independent elements. The main advantages of DE over GA are that, it has fewer tendencies to premature convergence, it does not require genotype–phenotype data transformations and it has more ability to reach the good solution without local search. Aliev et al. [49] proposes a recurrent neuro-fuzzy architecture where the parameters are identified by DE. Experiments on prediction and identification prove that proposed network gives excellent performance with non-population based neuro-fuzzy structures. A new neuro-fuzzy system hybridized with differential evolution and local information (DELI) is proposed in [50,51]. It is a self organizing scheme where the structure is updated by online clustering algorithm and parameters are tuned by DELI. A modified version of DE called MODE is applied in a neuro-fuzzy system for efficient performance [52]. The proposed neuro-fuzzy system uses clustering based mutation scheme which prevents training from traps into local optima. Experiments on control of inverted pendulum show that proposed neuro-fuzzy controller has obtained more stable performance than conventional control scheme. M. Su [53] gives idea self organizing neuro-fuzzy with rule-based symbiotic modified differential evolution, where the structure learning is performed using fuzzy clustering based entropy measure and parameter learning is conducted using subpopulation symbiotic evolution and a modified differential evolution. In [54], the differential-evolution-based symbiotic cultural algorithm (DESCA) is applied in the neuro-fuzzy system. A recurrent self evolving neuro-fuzzy system using differential harmony search is proposed in [55]. Differential harmony search is a technique uses mutation scheme of DE in the pitch adjustment operation for harmony improvisation process. The recurrent feedback is introduced in fuzzy rule layer and the output layer. The performance analysis of proposed technique is conducted in stock price prediction and obtained the reasonable result. A hybrid multi-objective ANFIS technique is proposed in [56] using singular value decomposition (SVD) and the DE algorithm, which is utilized for finding premise and consequent parameters of system. All the above studies have shown that DE is an able and competent technique that can apply efficiently in the field of neuro-fuzzy systems.

2.3.3. Ant colony optimization

Ant colony optimization (ACO) is a swarming intelligence type meta-heuristic method based on the behaviour of ants seeking their food and colony [57,58]. It is currently using various aspects of numerical problems having multiple optimum points [59]. A simple neuro-fuzzy controller using ACO is proposed in [60]. It is an RBF based structure where the parameters are tuned using ACO and then applied to stability control of an inverted pendulum. A Mamdani type neuro-fuzzy structure is developed for efficient interpretability [61]. The proposed system is a four layer structure combining functional behavior of RBF networks and TSK fuzzy inference system. The clustering method is used for tuning the membership parameters and ACO is used to determine consequent parameters. An ACO based neuro-fuzzy system controller is developed for real-time control of an inverted pendulum in [62]. An ANFIS optimized by ACO for continuous domains is presented for prediction of water quality parameters in [63]. Case Study on water quality of Gorganrood River shows that ANFIS-ACO technique gets a comparable performance with other evolutionary algorithms. A hybrid technique including ACO and ANFIS is implemented in optimal feature selection for predicting soil cation exchange capacity (CEC) [64]. Another ANFIS model optimized by ACO is proposed for modeling CO2-crude oil minimum miscibility pressure [65]. An ensemble strategy using neuro-fuzzy and ACO are introduced for flood susceptibility mapping [66]. This ensemble strategy has been obtained good performance for flood susceptibility along with satisfactory classification performance.

2.3.4. Particle swarm optimization

The concept of PSO is based social interaction using emergent behaviors such as bird flocking [67,68]. PSO is a popular meta-heuristic optimization method which is popular due to its simplicity in implementation, lesser memory requirements and ability to converge optimal solution quickly [69,70]. Compared to other search methods of optimization, PSO can easily to implement and lesser parameters to adjust [71]. A fuzzy neural network is developed for motion control of the robotic system using PSO [72]. The PSO technique is used to tune the consequent parameters of the proposed neuro-fuzzy system. An ANFIS based on PSO technique is used in different applications such as monitoring sensors in nuclear power plant [73], time series prediction [74,75], business failure prediction [76] and channel equalization [77]. A hybrid method is presented for tuning the ANFIS, where PSO is used for training the antecedent part and EKF for training the consequent part [78]. A hybrid neuro-fuzzy method combining ANFIS, wavelet transform and PSO is developed for accurate electricity price forecasting [79] and wind power forecasting [80]. Later above method is extended using quantum-behaved PSO for financial forecasting [81]. A TSK based neuro-fuzzy system utilizing linguistic hedge concept and PSO based parameter optimization have been developed [82]. Linguistic hedges concept is used for fine tuning of piecewise membership functions. Advanced PSO techniques like immune-based symbiotic PSO [83] and adaptive PSO [84] is applied in a neuro-fuzzy system for efficient parameter identification. An intuitionistic neuro-fuzzy network is developed with PSO adaptation for accurate regression analysis [85]. An interval type-2 based neuro-fuzzy system with PSO based parameter extraction is utilized for accurate stock market prediction [86].

PSO is also utilized in some self-organizing scheme neuro-fuzzy techniques. A hybrid learning technique is designed for a neuro-fuzzy system for efficient and adaptive operation [87]. Proposed technique uses fuzzy entropy clustering (FEC) for structure optimization and hybrid PSO and recursive singular value decomposition (RSVD) for parameter optimization. Modified PSO including local approximation and multi-elites strategy (MES) is developed for obtaining the optimal antecedent parameters of fuzzy rules. In [88], a self-evolving fuzzy neuro system is designed using PSO. In proposed strategy, subgroup symbiotic evolution is utilized for fuzzy rule generation and cultural cooperative particle swarm optimization (CCPSO) is used for parameter learning. This hybrid prediction technique is then extended to the functional-link-based neuro-fuzzy network (FLNFN) [89]. A self-organizing neuro fuzzy is implemented for efficient breast cancer classification [90]. The network structure is tuned using self constructing clustering algorithm and parameters are tuned by PSO.

2.3.5. Others

Artificial bee colony (ABC) optimization is a popular meta-heuristic algorithm which inspired from the foraging behaviour of honey bee swarm [91]. D. Karaboga [92] proposed an ANFIS architecture tuned using ABC. The premise parameters and consequent parameters are considered as variables of ABC and then efficiently tuned. Simulations of function approximations show that ABC optimizes the ANFIS better than back-propagation algorithm [93]. An ANFIS model is efficiently used for wind speed forecasting whose parameters are optimized using ABC [94]. An improved ABC algorithm called hybrid artificial bee colony optimization (HABC) is proposed for ANFIS training [95]. Two parameters such as, arithmetic crossover rate, and adaptivity coefficient are added to fasten the convergence rate of ABC. Simulations on benchmark problems show that HABC-ANFIS give more efficient performance than ABC-ANFIS.

In [96], A superior forecasting method is developed with neuro-fuzzy systems using simulated annealing (SA). A fuzzy hyper rectangular composite neural network (FHCNN) is developed and integrated chaos-search genetic algorithm (CGA) with SA is used for parameter identification. Proposed method is applied for short term load forecasting and obtained a higher degree of accuracy compared to popular forecasting methods. An ANFIS networks is proposed for efficient PID tuning with orthogonal simulated annealing (OSA) algorithm in [97]. The proposed approach provided high quality PID controller with good disturbance rejection performance.

Cuckoo search algorithm (CS) is another interesting evolutionary method inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds [98]. A hierarchical adaptive neuro-fuzzy inference system (HANFIS) approach is proposed for predicting student academic performance using CS algorithm [99]. CS algorithm is used to optimize the clustering parameters of HANFIS for proficient rule base formulation. In [100], an ANFIS based traffic signal controller is developed whose parameters are optimized by CS algorithm. The controller is tuned to set green times for each traffic phase using CS algorithm. X. Ding [101] presented a TSK model whose parameters are estimated using heterogeneous cuckoo search algorithm (HeCoS). In [106], -a hybrid algorithm is introduced for multiple mobile robots navigation using ANFIS and CS algorithm. The premise parameters are estimated using CS algorithm and consequent parameters are calculated by least square estimation.

Tabu search algorithm (TSA) is one of the metaheuristic algorithm employing local search methods used for mathematical optimization [102,103]. In [104], a TSK based neuro-fuzzy system is presented and its parameters are optimized by TSA optimization. H. Khalid [105] introduces an ANFIS for fault diagnostic whose parameters are estimated using TSA technique. It is shown that proposed TS-SC-ANFIS model has predicted most fault levels correctly.

Different population based neuro-fuzzy inference systems including its type of membership, learning criteria, fuzzy technique used and application are shown in Table 2.

2.4. Support vector neuro-fuzzy systems

Ever since Vapnik and Cortes [107] put forward the idea of "Support Vector Networks", SVMs have been the premier classification algorithm. In contrast to the existing algorithms which are based on minimizing the squared error, support vector machines and its variants [108–110] employed a revolutionary strategy of finding support vector points which guide the decision boundary. The problem was thus transformed to find a decision surface that maximized the boundary of separation, while at the same time not leading to a large number of misclassifications. This tradeoff between maximizing the boundary without the repercussions of loss in accuracy was controlled by the regularization parameter. The SVM is further enabled to solve regression problems by employing the ε-insensitive loss function [111]. This results in a convex inequality constrained optimization problem which is then solved as a quadratic programming problem. However, this approach means that a large number of parameters have to be optimized in order to satisfactorily solve the constraint satisfaction problem [112]. This led to the advancements in the SVM theorization leading to the development of LS-SVM [108]. The least squared SVM (LS-SVM) formulated the convex constraint satisfaction problem instead in the form of equality constraints rather than inequality constraints. This opens the problem to be further simplified by the use of the Karush–Kuhn–Tucker (KKT) conditions. On the other hand, the proximal support vector machine [109,110] or PSVM, goes about solving the problem by assigning points to one of the hyperplanes rather than the disjoint regions which again leads to a set of linear equation that needs to be solved. SVM in its original form works best with data that is linearly separable. For utilizing SVMs on complex datasets the input has first to be transformed by making use of kernels [113,114]. Kernels are used to transform the input into a higher dimensional feature space where the data becomes linearly separable.

Fuzzy support vector machine (FSVM) [115–117,118] is emerged as an area in machine learning classification where fuzzy membership is assigned to each input point and then reformulated mathematics of SVM. Hence different input points are made different contributions in decision surface based on fuzzy membership. Application of FSVM can reduce the effect of outliers and noises in the data [119]. Later FSVM is extended for regression problems [120,121]. A new FSVM is introduced for credit risk assessment [122]. It assigns different membership values for each positive and negative samples, which improves generalization ability and more insensitive to outliers.

A FSVM with fuzzy clustering algorithm is developed for minimizing the number of training data and time elapsed for constrained optimization [123,124]. A novel positive definite fuzzy classifier (PDFC) [125] is designed for accurate classification whose rules are extracted from SVM formulation. Proposed technique can easily avoid the curse of dimensionality problems because rules created from SVM are much less compared to the number of training samples. A kernel fuzzy clustering based FSVM [126] is implemented for effective classification problems with outliers or noises. An FSVM [127] using the concept of information granulation is available in the literature, which uses fuzzy C-means (FCM) to remove non-support vectors and k-nearest neighbor algorithm to remove noises and outliers. Proposed techniques have less computational complexity and more prediction accuracy compared conventional FSVM. An FSVM with minimum within-class scatter [128] is developed by using fisher discriminant analysis (FDA). Propose techniques make almost insensible to outliers or noises. A multi-class FSVM with PSO is utilized for fault diagnosis of wind turbine [129]. A hybrid tuning of FSVM by a combined bacterial forging-PSO algorithm and empirical mode decomposition (EMD) is applied for fatigue pattern recognition using EMG signal [130]. In all the above method, the number of fuzzy rules is equal to the number of support vectors. So, the large number of fuzzy rules is created for complex problems.

A new method for integrating SVM and FNN are proposed (SVFNN) [131], which uses adaptive kernel function in SVM. In the proposed system, the sufficient numbers of fuzzy rules are constructed by fuzzy clustering method. Then the parameters of SVFNN are identified by SVM theory formulation through the use of fuzzy kernel [132]. The unnecessary fuzzy rules are pruned by sequential optimization proposed in [131]. A TSK based fuzzy neuro system using SVM theory is proposed in [133]. The premise parameters of the proposed system are tuned by fuzzy clustering method and consequent parameters are obtained through SVM theory. The number of fuzzy rules generated is equal to the number of clusters, which make the number of fuzzy rules small. The proposed technique is successfully applied for face detection in color images [134]. The fuzzy weighted support vector regression (FWSVR) is proposed in [135], where the input is partitioned into clusters using fuzzy c-means algorithms and learned each partition locally using SVR. A recurrent fuzzy neural network with SVR (LRFNN-SVR) [136] is proposed for efficient dynamic system modeling. One pass clustering algorithm is implemented for antecedent learning and iterative linear SVR is used for feedback and consequent parameter learning. A novel SVR based TSK fuzzy system is designed for structural risk minimization [137]. Structural learning is performed using self-splitting rule generation (SSRG) which automatically designs the required number of rules. Both antecedent and consequent parameters are found by iterative linear SVR (ILSVR) learning.

Different advanced versions of SVM are incorporated in the neuro-fuzzy structure. A PSVM with fuzzy structure is proposed in [138]. Q. Wu [139] proposes a new FSVM by hybridizing triangular fuzzy number and v-support vector machine (v-SVM) [140], which uses PSO to find optimal parameters. Later this approach is extended by using GA method [141]. A new technique by hybridizing fuzzy and Gaussian loss function based SVM (g-SVM) is proposed [142]. An LSSVM is applied in local neuro-fuzzy models for nonlinear and chaotic time series prediction [143] and it uses hierarchical binary tree (HBT) learning algorithm for parameter updation. An online LSSVM is utilized for parameter optimization in ANFIS structure which is then applied for adaptive control of nonlinear system [144]. An ANFIS based controller is designed for nonlinear control bioreactor system [145]. SVM is used to extract the gradient information for updating the parameters of ANFIS controller. An ANFIS based system utilizing SVR tuning is suggested in the literature for industrial melt index prediction [146]. Some of the SVM based type-2 neuro-fuzzy systems are also explained in Section 3.

Different SVM based neuro-fuzzy inference systems including its type of membership, learning criteria, fuzzy technique used and application are shown in Table 3.

2.5. ELM based neuro-fuzzy systems

Extreme Learning Machine (ELM) is an emerging neural network technique which is widely used due to concerns of computational time and generalization capability compared to feedforward neural networks [147–152]. In ELM, the parameters of hidden neurons are selected randomly, whereas parameters of output neurons are evaluated analytically using minimum norm least-squares estimation method. The main advantages of ELM that, it produces faster computational time compared conventional training scheme without any iteration. Unlike the traditional neural net-

Table 2
Details of different population based neuro-fuzzy inference systems.

Algorithm	Membership	Structure	Learning criteria	Application	Fuzzy method
TRFN [41]	Gaussian	Recurrent	GA	System identification, system control	TSK
HGAPSO [42]	Gaussian	Recurrent	Hybrid GA and PSO	Temporal sequence generation, MISO Control	TSK
GO-RBFNN [43]	Ttriangular	Static	GA	Intelligent adaptive controllers for power system	TSK
GA-TSKFNN [44]	Trapezoidal	Self-organizing	GA	Classification	TSK
ANFIS-GA-SC [46]	Bell	Static	Subtractive clustering and GA	Electricity demand forecasting	TSK
ENFS [45]	Bell	Static	GA	Surface roughness evaluation	TSK
EARFNN [49]	Triangle	Self-organizing	DE	Non-linear system identification, predicition	TSK
NFS-DELI [50,51]	Gaussian	Self-organizing	DE with local information	Time-series prediction problem	TSK
ANFN-MODE [52]	Gaussian	Self-organizing	Modified DE	Control system	TSK
RSMODE-SONFS [53]	Gaussian	Self-organizing	Modified DE	Online identification and control	TSK
DESCA-NFS [54]	Gaussian	Self-organizing	Differential-evolution-based symbiotic cultural algorithm	Control system	TSK
SERNFIS [55]	Gaussian	Self-evolving	Modified differential harmony search (MDHS)	Stock price prediction	TSK
ANFIS-DE/SVD [56]	Gaussian	Static	Singular decomposition value (SVD) and DE	Modeling scour at pile groups	TSK
NFC-ACO [60]	Generalized Gaussian function	Static	ACO	Control of inverted pendulum	TSK
RBFNF-ACO [61]	Gaussian	Static	ACO	Time series prediction	Mamdani
FNN-PSO [72]	Static	Static	PSO	Motion control of robot	TSK
LHBNFC [82]	Triangular or trapezoidal	Static	PSO	Classification	Zero order TSK
FNN-PSO-RSVD [87]	Gaussian	Self-organizing	PSO and SVD	Non linear identification, time series prediction	TSK
FNN- ISPSO [83]	Gaussian	Static	Immune-based symbiotic PSO	Image processing, time series prediction	Mamdani
NFIS-SEELA [88]	Gaussian	Self-evolving	Cultural cooperative particle swarm optimization (CCPSO)	Time series prediction and control	TSK
FLNFN-SEELA [89]	Gaussian	Self-evolving	CCPSO	Time series prediction	TSK
ANFIS-PSO [74]	Bell	Static	PSO	Time series prediction	TSK
ANFIS-PSO-EKF [78]	Bell	Static	PSO and EKF	System identification, Time series prediction	TSK
Wavelet-ANFIS-PSO [79,80]	Bell	Static	PSO	Electricity price forecasting and wind power forecasting	TSK
Wavelet-ANFIS-QPSO [81]	Bell	Static	PSO	Forecasting, pattern extraction	TSK
FLIT2FNS-PSO [86]	Gaussian	Static	PSO	Stock market prediction	Type-2
SCNF-PSO [90]	Gaussian	Self-organizing	PSO	Breast cancer classification	TSK
NFRBN-HFC-APSO [84]	Gaussian	Static	HFC-APSO	Modeling automotive engine	TSK
INFS-PSO [85]	Gaussian	Static	PSO	Modeling of bench mark problems	TSK
ABC-ANFIS [92]	Gaussian	Static	ABC	Function approximation	TSK
HABC-ANFIS [95]	Gaussian	Static	HABC	Bench mark regression problems	TSK
FHCNN-SA-CGA [96]	Triangular	Static	SA and GA	Short term load forecasting	Mamdani
ANFIS-OSA [97]	Gaussian	Static	OSA	PID design	TSK
ANFIS-CS [100]	Gaussian	Static	CS	Traffic signal controller	TSK
CS-ANFIS [106]	Gaussian	Static	CS and Least square estimation	Mobile robots navigation	TSK
TS-SC-ANFIS [105]	Gaussian	Static	TSA	Fault detection	TSK

Table 3
Details of different SVM based neuro-fuzzy inference systems.

	Membership	Structure	Learning criteria	Application	Consequent
PDFC-SVM [125]	Gaussian	Self-evolving	SVM theory	Classification	Fuzzy singleton type.
FSVM-SCM [123]	RBF	Static	SVM theory	Classification	Fuzzy singleton type.
B-FSVM [122]	Credit scoring methods	Static	SVM theory	Credit risk assessment	Fuzzy singleton type.
SVFNN [131]	Gaussian	Self-organizing	SVM theory	Pattern classification	Fuzzy singleton type.
SOTFN-SV [133]	Gaussian	Self-organizing	SVM theory	Classification, image processing	TSK
FWSVR [135]	Fuzzy c-means	Static	SVM theory	System identification and regression	TSK
ANFIS-SVM [145]	Bell	Static	Gradient +SVM	Control of bioreactor system	TSK
Fv-SVM [139]	Triangular fuzzy function	Static	SVM + PSO	Car sale series forecast	Fuzzy singleton type.
Fg-SVM [142]	Triangular fuzzy function	Static	SVM + GA	Car sale series forecast	Fuzzy singleton type.
LRFNN-SVR [136]	Gaussian	Recurrent	One pass clustering algorithm + SVM	System identification and time series prediction	TSK
FS-RGLSVR [137]	Gaussian	Self-organizing	Iterative linear SVR	Function approximation and time series prediction	TSK
WCS-FSVM [128]	Affinity based function [128]	Static	Support vector data description (SVDD)	Regression	Fuzzy singleton type.
FSVM-FIG [127]	Fuzzy c-means	Static	SVM theory	Classification	Fuzzy singleton type.

works, the learning in ELM aims in simultaneously reaching the smallest training error and smallest norm of output weights. The ELM has become a superior choice for function approximation and modelling of nonlinear systems, due to its universal approximation capabilities [153]. Since the introduction of classic ELM, various variants of ELMs like online sequential, incremental type and optimally pruned ELMs were proposed in the literature [154–156].

The merits of ELM and the fuzzy logic system can be integrated by introducing ELM based neuro-fuzzy approach. Here conventional neural networks are replaced by ELM, which gives more adaptability to the networks. Introduction of ELM leads to increase in generalization accuracy and reduction of complexity in learning algorithm [157]. In most of the ELM based neuro-fuzzy techniques [158–164], the membership function of the fuzzy system can be tuned by ELMs which can be used as the decision-making system in the fuzzy control structure. Even though fuzzy logic is able to encode experience information directly using rules with linguistic variable, the design and tuning of the membership functions quantitatively describing these variables, takes a lot of time. ELM learning technique is able to automate this process, substantially reducing development time and cost, while improving the performance.

A neuro-fuzzy TSK inference system with ELM technique called E-TSK was introduced using K-means clustering to pre-process the data [158]. The firing strength of each fuzzy rule is found by ELM and consequent part ('then' part) of the fuzzy rule is determined using multiple ELMs. In ETSK, number of ELMs used in consequence part will depend on the number of clusters used for pre-processing the data. When more number of clusters uses, the complexity of the ETSK increases. OS-Fuzzy-ELM [159], is a hybrid approach using online sequential ELM in which the parameters of the membership functions are generated randomly, which do not dependent on the data set used for training, and then, consequent parameters are identified analytically. The learning is done with the input data, either coming in a one-by-one mode or a chunk-by-chunk mode. Using the functional equivalent of fuzzy inference system (FIS) together with the recently developed optimally pruned ELM [156], a hybrid structure called evolving fuzzy optimally pruned ELM (eF-OP-ELM) is proposed in [160, 165]. Y. Qu [161] develops an evolutionary neuro-fuzzy algorithm called EF-ELM by using hybridization of differential evolution (DE) technique and extreme learning concept. A hybrid GA based fuzzy ELM (GA-FELM) is proposed for classification [166]. Since above techniques are based on iterative method, training using EF-ELM and GA-FELM are computationally expensive [167].

A new complex neuro-fuzzy system is proposed using adaptive neuro-complex fuzzy inference system (ANCFIS) and online sequential ELM in [168]. It uses complex sinusoidal membership function for accurate time series prediction. The sinusoidal membership parameters are randomly selected from uniform distribution and consequent parameters calculated by recursive least squares method. Recently, a novel neuro-fuzzy system combining ANFIS and ELM is proposed [164]. The premise parameters of the proposed system are randomly selected with constraints to reduce the effect of randomness. The constraints on the premise parameter also improve the interpretability of fuzzy system [169]. The consequent parameters are calculated by ridge regression utilizing the effect of regularization.

Since ELM based neuro-fuzzy techniques give greater generalization capability and less computational speed, it can be applied in practical applications such as weather prediction, medical applications, image processing etc. K. B. Nahato [170] uses fuzzy ELM for classifying clinical dataset for accurate analysis. It uses heart disease data and diabetes datasets from the University of California Irvine for accurate medical treatment. In [171], ELM is integrated with fuzzy K-nearest neighbor approach for efficient gene selection in cancer classification systems. It is shown that proposed hybrid approach efficiently finds the gene for distinguishing different cancer subtype. An ELM based ANFIS is utilized for efficient prediction of ground motion earthquake parameters such as peak ground displacement, peak ground velocity and ground acceleration [163]. It is obtained that hybridization of ELM and ANFIS gives best prediction result with minimum computational time. A hybrid ANFIS with ELM structure is utilized for precise model based control applications [18, 172]. Prediction of landslide displacement is performed using ensemble method combining the ELM-ANFIS and empirical mode decomposition (EMD) [173,174]. This work shows that combination of EMD and ELM-ANFIS will get the best performance in terms of error prediction accuracy and error percentage.

Table 4
Details of different ELM based neuro-fuzzy inference systems.

	Membership	Structure	Learning criteria		Application	Fuzzy method
			Premise	Consequent		
ETSK [158]	ELM	Static	ELM	ELM	Function approximation	TSK
OS-Fuzzy-ELM [159]	Gaussian	Static	Random	Recursive least squares (RLS)	Regression and classification	TSK
eF-OP-ELM [178]	Gaussian	Static	Random	Recursive least squares	Real world regression, time series prediction and control	TSK
EF-ELM [161]	Gaussian	Static	Random	DE	Mammographic risk analysis	TSK
GA-FELM [166]	Gaussian	Static	Random	GA	Classification	TSK
McFELM [180]	Gaussian	Self-organizing	Random	Least square	Real world regression, non linear system identification	TSK
ANCFIS-ELM [168]	Sinusoidal functions	Static	Random	Recursive least squares	Real world prediction	TSK
RELANFIS [164]	Bell	Static	Random	Ridge regression	Regression and classification	TSK

A neuro-fuzzy system with efficient pattern classification is proposed in [175]. It uses neighbourhood rough set (NRS) theory for feature extraction from input pattern and forms a fuzzy feature matrix. The fuzzy features then processed through ELM for efficient classification. The superiority of fuzzy ELM is well tested in both completely and partially labeled data including remote sensing images. An adaptive non symmetric activation fuzzy function (ANF) is designed in ELM for accurate face recognition applications [176]. It is shown that application of ANF will improve the performance of face identification. An ensemble fuzzy ELM is used for efficient prediction of Shanghai stock index and NASDAQ [177]. An ELM based type-2 fuzzy system is utilized for modeling the permeability of carbonate reservoir [178]. It is shown that proposed technique is a good effort for considering the uncertainties in reservoir data and obtained better modeling results. An electricity load forecasting from Victoria region, Australia has been performed using IT2FNN with ELM [179].

Different ELM based neuro-fuzzy inference systems including its type of membership, learning criteria, fuzzy technique used and application are shown in Table 4.

3. Neuro-fuzzy systems based on fuzzy methods

Neuro-fuzzy systems are classified into Type-1 and Type-2 neuro-fuzzy networks based on the fuzzy sets used. This session mainly explained different Type-2 neuro-fuzzy systems available in the literature.

3.1. Type-1 neuro-fuzzy systems

Type-1 fuzzy system is a one dimensional fuzzy set which can handle the uncertainty associated with input and output data. Most of the neuro-fuzzy systems explained in earlier sections utilize type-1 fuzzy set, which will come under the category of type-1 neuro-fuzzy systems.

3.2. Type-2 neuro-fuzzy system

In type-1 fuzzy system, uncertainties related to input and output are modelled easily using precise and crisp membership function. Once the type-1 membership function has been chosen, the fact that the actual degree of membership itself is uncertain which no longer modelled in type-1 fuzzy sets. The uncertainties associated with the rules and membership functions cause difficulty in determining the exact and precise antecedents and consequent during fuzzy logic design [181]. The designed type-1 system cannot give optimal performance when the parameters and operating conditions are uncertain. Hence type-2 fuzzy systems are developed especially for dealing with uncertainties which are generally more robust that type-1 fuzzy system [182]. With the unique structure in membership function, type-2 fuzzy can effectively model the uncertainties in rules and membership functions [183,184]. Interval type-2 fuzzy is as simplest and most commonly used type-2 systems which provide smoother control surface and robust performance [185,186]. The detailed theory and design of interval type-2 fuzzy system are performed in [187]. Different type reduction algorithms for interval type-2 fuzzy sets, such as the Karnik–Mendel (KM) algorithm, the enhanced KM (EKM) algorithm, or the enhanced iterative algorithm with stop condition (EIASC) [185, 187–189] are widely used.

In type-2 neuro-fuzzy system, type-2 fuzzy set is embedded in antecedent part or consequent part or both. The major challenges for the design of type-2 neuro-fuzzy system are based on optimal structure selection and parameter identification selection. Training algorithm will be derivative or derivative free or hybrid [190]. Most of the proposed type-2 neuro-fuzzy systems use the hybrid technique for parameter identification. Type-2 neuro-fuzzy systems are classified into interval type-2 neuro-fuzzy systems (IT2NFS) and general type-2 neuro-fuzzy systems (GT2NFS).

3.2.1. Interval type-2 neuro-fuzzy system

Interval type-2 fuzzy neural network (IT2NFS or IT2FNN) is firstly implemented in the literature [191]. It consists of two layer interval type-2 Gaussian MF as antecedent part and two layer neural networks as consequent part. GA is used for antecedent parameter training and back-propagation is used for consequent parameter training. Hybrid learning composed of back-propagation (BP) and recursive least-squares (RLS) for training T2FNN is proposed in [192]. A self evolving interval type-2 fuzzy neuro system (SEIT2FNN) [193] is proposed which have both structure and parameter learning capability. The structure learning is performed using clustering method and parameter learning is done by a hybrid method utilizing gradient descent algorithms and Kalman filter algorithm. The theoretical insight of IT2FNN for function approximation capability is discussed in [194]. It also introduced the idea of interval type-2 triangular fuzzy number. An IT2FNN system utilizing fuzzy clustering algorithm and evolutionary algorithm (DE) is presented in [195]. C. Li [196] proposes the theoretical analysis of monotonic IT2FNN, which uses hybrid strategy with least squares method and the penalty function-based gradient descent algorithm for parameter estimation. In [197], a self organizing IT2FNN is designed, where the learning rate is adaptively calculated by fuzzy rule formulation. In order to improve the discriminability and reduce the parameters the number of parameters in consequent part, vectorization-optimization method (VOM)-based type-2 fuzzy neural network (VOM2FNN) [198] is developed. Interval type-2 fuzzy sets are incorporated in antecedent part and VOM technique is used to tune the consequent parameters optimally. A self-evolving compensatory IT2FNN [199] is developed using the compensatory operator based type-2 fuzzy reasoning mechanism for efficiently handling of time-varying characteristic problem. A simplified IT2FNN is proposed with a novel type reduction process without the use of computationally expensive K-M procedure [200]. It is shown that application of above technique in system identification and time series prediction will effectively reduce the computational burden of IT2FNN.

In [201], a novel fuzzy neuro-system called DIT2NFS-IP is proposed for best model interpretability and improved accuracy. The rules sets are generated by clustering algorithm and parameters are updated through gradient descent and rule-ordered recursive least squares algorithms. Most of the above explained technique does not have rule pruning mechanism, which may create complex and ever-expanding rule base. A Mamdani type-2 fuzzy neuro system is proposed [202]. The obsolete rules are removed if its certainty factor is below the threshold value. Proposed technique obtained higher accuracy and interpretability for online identification and time series prediction. A new type-2 neuro-fuzzy system (IT2NFS-SIFE) [203] for simultaneous feature selection and system identification is implemented. A hybrid technique using gradient descent algorithm and rule-ordered Kalman filter algorithm is used for parameter identification. Poor features are discarded using the concept of a membership modulator which invariably reduces the number of rules.

In [204], a metacognitive IT2FNN (McIT2FIS) is proposed. Cognitive component consists of TSK based type-2 fuzzy system and meta-cognitive component constitutes self-regulatory learning mechanism. But above structure proposed to have more complex, gives narrow footprint of uncertainty and may lead to overfitting [205]. A parallel evolutionary based IT2FNN (IT2SuNFIS) [206] is implemented by inserting subsethood measure between fuzzy input and antecedent weight to understand the degree of overlap or contaminant between them.

3.2.1.1. Recurrent interval type-2 neuro-fuzzy system. A recurrent interval type-2 FNN (RSEIT2FNN) [207] is proposed by providing internal feedback in rule firing strength. Structure learning is performed using online clustering algorithm and parameter learning is performed using gradient descent algorithm and Kalman filter algorithm. In [208], mutually recurrent type-2 neuro-fuzzy system (MRIT2NFS) is discussed. It provides local internal feedback in rule firing strength and mutual feedback between the firing strength of each rule. The rule is updated based ranking of spatial firing strength and parameter is optimized by a hybrid method comprising gradient descent learning and rule-ordered Kalman filter algorithm. Simulations on system identification show that proposed techniques outperformed other states of art interval type-2 neuro-fuzzy systems. A novel recurrent type-2 neuro-fuzzy system is presented by giving double recurrent connections [209]. First feedback loop is inserted in rule layer and second is included in the consequent layer. The type-2 fuzzy system incorporated in antecedent layer and wavelet function is used in consequent part. Rule pruning is done using relative mutual information and dimensionality reduction is performed using sequential Markov blanket criterion (SMBC) [210]. In [211], a recurrent hierarchical IT2FNN is proposed for efficient synchronization of chaotic systems and time series prediction. The learning parameters of the proposed system are identified by square-root cubature Kalman filters (SCKF) algorithm.

3.2.1.2. SVM based interval type-2 neuro-fuzzy system. In order to improve the accuracy, some researchers hybridize the SVM concept with IT2FNN. Use of SVM technique in neuro-fuzzy system reduces the structure complexity [212]. A TSK base IT2FNN is proposed using SVR. An online clustering algorithm is used for structure learning and linear SVR is employed for parameter learning. It is noted the above technique is only applicable to regression problems and its antecedent parameters are not optimized. A novel SVM based interval type-2 neuro-fuzzy system is proposed for human posture classification application to improve the generalization ability conventional IT2FNN [213]. The premise parameters are tuned using margin-selective gradient descent (MSGD) and consequent parameters are identified by SVM theory. The experimental test shows that proposed technique gives fewer rules with a simple structure and higher accuracy for noisy attributes.

3.2.1.3. ELM based interval type-2 neuro-fuzzy system. ELM based Type-2 neuro-fuzzy have also emerged in the literature. Z. Deng [214] proposed a fast learning strategy for IT2FNN system using extreme learning strategy named as T2FELA. The antecedent parameters of proposed T2FELA are randomly selected and consequent parameters are calculated by optimum estimation method. Significant improvement in training speed has been found in benchmarking real-world regression dataset compared state-of-art IT2FNN techniques. A novel meta cognitive type-2 neuro-fuzzy system utilizing extreme learning concept is implemented in literature [215,216]. The new rule is added based on Type-2 Data Quality (T2DQ) method and rules are pruned through Type-2 relative mutual information. The parameters of the networks are learned through extreme learning concept. S. Hassan [217] proposed a new IT2FNN system whose antecedent parameters are tuned by non-dominated sorting genetic algorithm (NSGAII) and consequent parameters are identified through extreme learning concept. Another IT2FNN is used for modeling of chaotic data which uses hybrid learning technique, the combination of artificial bee colony optimization (ABC) and extreme leaning concept to tune antecedent and consequent parameters [218].

3.2.1.4. Interval type-2 neuro-fuzzy for control application. Type-2 fuzzy neural network is applied in lots of real world control applications. An adaptive IT2FNN is proposed for robust control of linear ultrasonic motor based motion control [219]. In this work, an adaptive training algorithm is designed using Lyapunov stability analysis. An IT2FNN network is used for adaptive motion control of permanent-magnet linear synchronous motors [220]. An online learning based sliding mode control of servo system using IT2FNN is proposed in [221]. Stability of proposed control is preserved using the Lyapunov method. An adaptive dynamic surface control of nonlinear system under uncertainty environment has been performed using IT2FNN network [222]. The performance of proposed technique in control algorithm is experimentally verified in ball and beam system and the stability is confirmed using Lyapunov theory. Two mode adaptive control of the chaotic nonlinear system is developed using hierarchical T2FNN [223]. Proposed control gives least prediction error along with less computational effort.

3.2.2. General type-2 neuro-fuzzy system

Interval type-2 fuzzy systems (IT2FS) have received the most attention because the mathematics that is needed for such sets is primarily *Interval arithmetic*, which is much simpler than the mathematics that is needed for general type-2 fuzzy systems (GT2FS).

So, the literature about IT2FS is large, whereas the literature about GT2FS is much smaller. In simple terms, uncertainty in a GT2FS is represented by a 3D volume, while uncertainty in an IT2FS is represented by a 2D area. It is not until recent years that research interest has gained momentum for GT2FSs, but uses of easier type reduction methods will encourage more and more research papers in GT2FS. Liu [224] introduced a type reduction method based on an α-plane representation for general type-2 fuzzy sets. Centroid-Flow algorithm [225] and extended α-cuts decomposition [226] have also emerged for type reduction in general type-2 fuzzy systems.

Like IT2FNN, general type-2 neuro-fuzzy systems (GT2FNN) also account for MF uncertainties, but it weights all such uncertainties non-uniformly and can therefore be thought of as a second-order fuzzy set uncertainty model. GT2FNN is described in more parameters than IT2FNN models. A general type-2 neuro-fuzzy system is developed [227] using α-cuts to decomposition. Fuzzy clustering and linear least squares regression are used for structure identification and hybrid method using PSO and RLS are used for parameter identification. Another general type-2 neuro-fuzzy system [228] uses hybrid learning method using PSO and RLE.

3.3. Other neuro-fuzzy systems

Neuro-fuzzy systems having logical type reasoning methods are available in literature, where consequents and antecedents are related by fuzzy implications. L. Rutkowski [229] proposed flexible neuro-fuzzy system (FLEXNFIS) which incorporates flexibility into the designing of fuzzy method. Using the training data, the structure of fuzzy system (Mamdani or logical) is learned along with fuzzy parameters. This technique is based on definition of H-function which becomes a T-norm or S-norm depending on a certain parameter which can be found in the process of learning. In case of AND-type flexible neuro-fuzzy system, fuzzy inference is characterized by the simultaneous appearance of Mamdani-type and logical-type reasoning [229–231]. A new method of FLEXNFIS is designed that allow to reduce number of discretization points in the defuzzifier, number of rules, number of inputs, and number of antecedents [232,233]. In evolutionary flexible neuro-fuzzy system, GA is used to develop the flexible parameters [234,235]. An online smooth speed profile generator used in trajectory interpolation in milling machines was proposed using FLEXNFIS in [17]. Another FLEXNFIS is presented for nonlinear modeling which allow automatic way to choose the types of triangular norms in the learning process [236,237]. An evolutionary based learning is performed and a fine-tuned structure is obtained for benchmark problems. A class of neuro-fuzzy system utilizing the quasi-triangular norms is developed by adjustable quasi-implications [238].

The fuzzy sets can be combined with the rough set theory to cope with the missing data and such system is termed as rough neuro-fuzzy systems [239,240]. An ensemble of neuro-fuzzy systems is trained with the AdaBoost algorithm and the back-propagation [241,242]. Then rules from neuro-fuzzy systems constituting the ensemble are used in a neuro-fuzzy rough classifier. A gradient learning of the rough neuro-fuzzy classifier (RNFC) parameters using a set of samples with missing values is proposed in [243]. The proposed technique obtained reasonable classification capability for missing data.

Different type-2 based neuro-fuzzy inference systems including its type of membership, learning criteria, fuzzy technique used and application are shown in Table 5.

4. Neuro-fuzzy systems based on structure

Neuro-fuzzy systems are classified into static and self-organizing neuro-fuzzy networks based on its structure. This session mainly explained different self-organizing neuro-fuzzy systems available in the literature.

4.1. Static neuro-fuzzy systems

In static neuro-fuzzy systems, the structure of the systems remains constant during the training. The number of fuzzy rules used, number of the premise and consequent parameters and the number of inputs, rule nodes and/or input /output-term nodes are fixed during the neuro-fuzzy operation. Most of the above explained gradient, population, SVM and ELM based neuro-fuzzy systems have come under the category of static neuro-fuzzy systems.

4.2. Self-organizing neuro-fuzzy systems

In self organizing neuro-fuzzy systems, both structure and parameters are tuned during the training process. Along with parameter tuning algorithm, structural learning algorithm also is specified to self-organize the fuzzy neural structure. The rule nodes and/or input/output-term nodes are created dynamically as learning proceeds upon receiving on-line incoming training data. During the learning process, novel rule nodes and input/output-term nodes will be added or deleted according to structural learning algorithms.

Self-constructing neural fuzzy inference network (SONFIN) is a TSK fuzzy inference based neuro-fuzzy system whose rules are updated through online adaptation algorithm [246]. The structure of SONFIN adaptively updated by aligned clustering based algorithm. Afterwards, the input variables are selected via a projection-based correlation measure for each rule will be added to the consequent part incrementally as learning proceeds. The preconditioning parameters are tuned by back-propagation algorithm and consequent parameters are updated by the recursive least square algorithm. The performance analysis of SONFIN is tested for online identification of dynamic system, channel equalization, and inverse control applications. But in SONFIS, the rules of fuzzy system grow according to the partition of input-output space which is not an efficient rule management [247]. J. S. Wang [247] proposed a self adaptive neuro-fuzzy inference system (SANFIS) where it can able to self adapting and self organizing its structures to achieve best rule base. A mapping constrained agglomerative (MCA) clustering algorithm is implemented to develop the required number of rules which is considered as initial structure of SANFIS.The parameters of SANFIS is identified by recursive least squares and Levenberg–Marquardt (L–M) algorithms. The performance analysis of SANFIS is tested in classification for the wide range of real world regression problems. Although this algorithm is sequential in nature, it does not remove the fuzzy rules once created even though that rule is not effective. This may result in a structure where the number of rules may be large.

4.2.1. Dynamic fuzzy neural networks

Dynamic fuzzy neural networks (D-FNN) [248] are a hybrid neuro-fuzzy technique using TSK fuzzy system and radial basis neural networks. A self organizing learning technique is used to update the parameters using Hierarchical Learning. The neurons are inserted or deleted according to system performance. Using neuron pruning, the significant amount of rules and neurons are selected to achieve the best performance. Later G. Leng [249] proposed a self organizing fuzzy neural network (SOFNN), used adding and pruning techniques based on the geometric growing criterion and recursive least square algorithm to extract the fuzzy rule from input-output data. However, in these two algorithms, the pruning criteria need all the past data received so far. Hence, they are not strictly sequential and further require increased memory for storing all the past data. N. Kasabov [250] proposes dynamic evolving

Table 5
Details of different type-2 based neuro-fuzzy inference systems.

	Membership	Structure	Learning criteria		Application	Fuzzy method
			Premise	Consequent		
T2FNN [191]	Interval type-2 Gaussian MF	Static	GA	Back-propagation	System identification and control	Interval type-2
T2FNN-RLS-BP [192]	Interval type-2 Gaussian MF	Static	Back-propagation (BP)	Either recursive least-squares (RLS) or square-root filter (REFIL) method.	Temperature prediction	Interval type-2
IT2-TSK-FIS [244]	Interval type-2 Gaussian MF	Static	Back-propagation	Adaptive learning rate back-propagation	Online identification, time series prediction	TSK
SEIT2FNN [193]	Interval type-2 Gaussian MF	Self-organizing	Gradient descent algorithms	Kalman filter algorithm	Plant modeling, adaptive noise cancellation, and chaotic signal prediction	Interval type-2 TSK
AIT2FNN [219]	Type-2 Gaussian MF	Static	GA	Lyapunov stability theorem	Adaptive control	Interval type-2 TSK
RSEIT2FNN [207]	Type-2 Gaussian MF	Recurrent self-evolving	Gradient descent algorithm.	Kalman filter algorithm	Dynamic system identification, time series prediction	Type-2 TSK
GT2FNN [227]	General type-2 Gaussian	Self-organizing	PSO	Recursive least squares (RLS)	Function approximation	Type-2 TSK
T2FNN-HLA [228]	General type-2 Gaussian MF	Self-organizing	PSO	Least squares estimation (LSE)	Function approximation	TSK
T2FINN-DE [195]	Type-2 triangular MF	Self-organizing	DE	DE	System identification	TSK
MT2FNN [196]	Interval type-2 Gaussian MF	Static	Least squares method	Penalty function-based gradient descent algorithm	Thermal comfort prediction	TSK
MRIT2NFS [208]	Interval type-2 Gaussian MF	Recurrent self-evolving	Gradient descent learning	Rule-ordered Kalman filter algorithm	Dynamic system identification, time series prediction	TSK
IT2TSKFNN [197]	Interval type-2 Gaussian MF	Self-evolving	Gradient descent method	Gradient descent method	Nonlinear system identification	TSK
VOM2FNN [198]	Gaussian MF.	Static	VOM	VOM	Noisy data classification	TSK
DIT2NFS-IP [201]	Interval type-2 Gaussian MF	Self-organizing	Gradient descent	Rule-ordered recursive least squares	System identification with noise, time series prediction	Zero-order TSK
eT2FIS [202]	Interval type-2 Gaussian MF	Self-evolving	Gradient descent approach	Gradient descent approach	Online identification and time series prediction	Mamdani
TSCIT2FNN [199]	Interval type-2 Gaussian MF	Self-organizing	Gradient descent algorithm	Variable-expansive Kalman filter algorithm	System identification, adaptive noise cancellation, time-series prediction	TSK
SIT2FNN [200]	Interval type-2 Gaussian MF	Self-evolving	Gradient descent algorithm	Gradient descent algorithm	Identification and time series prediction	TSK
HT2FNN [223]	Interval type-2 Gaussian MF	Static	Lyapunov stability	Lyapunov stability	Two mode adaptive control	TSK
McIT2FIS [204]	Interval type-2 Gaussian MF	Self-evolving	Projection based learning algorithm	Projection based learning algorithm	Classification	TSK
McIT2FIS [245]	Interval type-2 Gaussian MF	Self-evolving	Extended Kalman filtering	Extended Kalman filtering	Benchmark prediction/forecasting	TSK
eT2RFNN [209]	Type-2 multivariate Gaussian function	Self-evolving	Sequential maximum likelihood estimation	Fuzzily weighted generalized recursive least square (FWGRLS)	Real world regression problems	Wavelet function
SRIT2NFIS [205]	Gaussian membership function	Self-evolving	Regularized projection-based learning algorithm.	Regularized projection-based learning algorithm.	Handling intersession nonstationarity of EEE signal	TSK
IT2SuNFIS [206]	Type-2 multivariate Gaussian function	Static	DE	DE	Function approximation, time series prediction and control	TSK

(continued on next page)

Table 5 (*continued*)

	Membership	Structure	Learning criteria		Application	Fuzzy method
			Premise	Consequent		
RHT2FNN [211]	Interval type-2 Gaussian MF	Self-evolving	Square-root cubature Kalman filters (SCKFs)	SCKF	Synchronization of chaotic systems, time series prediction	TSK
IT2NFS-SIFE [203]	Interval type-2 Gaussian MF	Self-evolving	Gradient descent algorithm	Rule-ordered Kalman filter algorithm	System identification and real world regression	TSK
IT2FNN-SVR [212]	Interval type-2 Gaussian MF	Self-evolving	Clustring	SVM	Real world regression, time series prediction and control	TSK
IT2NFC- SMM [213]	Interval type-2 Gaussian MF	Self-organizing	Gradient descent algorithm	SVM	Human posture classification	TSK
T2FELA [214]	Interval type-2 Gaussian MF	Static	Random	Least square	Real-world regression	TSK
eT2ELM [215]	Interval type-2 multivariate Gaussian function	Self-evolving	Random	Generalized recursive least square (GRLS)	Classification	TSK
IT2FLS-ELM [217]	Type-2 Gaussian MF	Static	Non-dominated sorting genetic algorithm (NSGAII)	ELM concept	Forecasting of nonlinear dynamic	TSK
IT2FLS-ELM-ABC [218]	Type-2 Gaussian MF	Static	Artificial bee colony optimization (ABC)	Extreme leaning	Time series preditction	TSK

neural-fuzzy inference system (DENFIS) in which output is formulated based on mostly activated rules which are dynamically chosen from a set of fuzzy rules. It uses evolving clustering methods to partition the input data for both online and offline identification and recursive least square method is used for consequent parameters identification. But DENFIS require the training data several times (more passes) to lean and this may produce more iterative time [9]. The authors of ASOA-FNN [251] proposed a hybrid fuzzy neural learning algorithm with fast convergence for ensuring best modeling performance. It composed of both adaptive second-order algorithm (ASOA) and adaptive learning rate strategy for efficient training of fuzzy neural networks (FNN).

4.2.2. Using singular value decomposition

S. Lee and C. Ouyang [252] developed self-constructing rule generation based neuro-fuzzy system using singular value decomposition. Initially, the data is partitioned several clusters using input-similarity and output-similarity tests. Then fuzzy rules are extracted from each cluster. A hybrid learning algorithm using recursive gradient descent method and singular value decomposition-based least squares estimator is developed for efficient estimation of parameters. The data used in SCRD-SVD might be consist of anomalies and should be correlated. However, data patterns may not be properly described due to the removal of such fuzzy rules.

4.2.3. Sequential and probabilistic networks

Sequential Adaptive Fuzzy Inference System called SAFIS [253] is developed based on the principle of functional equivalent between fuzzy inference system and RBF networks. SAFIS include or remove the new rules using the concept of the *influence* of a fuzzy rule by contribution to the system output in a statistical sense. The parameters of SAFIS are determined by Euclidean distance winner strategy and updated by extended Kalman filter (EKF) algorithm. However, the exact calculation of this measure is not practically feasible in a truly sequential learning scheme. The algorithm assumes a uniform distribution for the input data space to simplify the resulting formulation. Later SAFIS is extended using the concept of Weighted Rule Activation Record (WRAR) [254]. The SAFIS is also extended to classification problems using Recursive Least Square Error (RLSE) scheme [255]. A probabilistic fuzzy neural network (PFNN) [256] is proposed to accommodate the complex stochastic and time-varying uncertainties. A probabilistic fuzzy logic system [257] with 3-D membership function is used for fuzzification. Mamdani and Bayesian inference methods are used to process the fuzzy and stochastic information in the inference stage. A hybrid learning algorithm composed of SPC-based variance estimation and gradient descent algorithm is utilized for estimation parameters. But it requires high computational complexity for a better modeling performance.

4.2.4. Growing and pruning

A self-organizing scheme for parsimonious fuzzy neural networks (FAOS-PFNN) [258] is proposed for a fast and accurate prediction using growing and pruning of rules based error criteria and distance criteria. All parameter of FAOS-PFNN is updated by extended Kalman filter (EKF) method. Unfortunately, the FAOS- PFNN cannot prune the redundant hidden neurons; besides, the convergence of the algorithm is not analyzed. For extracting the rules for neuro-fuzzy system effectively from a large numerical database, an online self-organizing fuzzy modified least-square (SOFMLS) [259] was proposed. The network can generate the new rule based on distance criteria with all the existing rules are more than a prespecified radius. A pruning algorithm is developed based on density criteria and the modified least-square algorithm is used for updating antecedent and consequent part. However, the criterion used to determine the rules in the SOFMLS network is only based on new data and the current density value of the rules. Another problem is that every step of the structure adjusting strategy is only capable of adding or pruning a neuron from the network. Another growing and pruning neuro-fuzzy algorithm called GP-FNN [32] is proposed using a sensitivity analysis (SA) of the output from the model. The significance of fuzzy rule is calculated by Fourier decomposition of the variance of the network output. Simulation studies have shown GP-FNN is a better method for growing and pruning. It is shown that the proposed method produces compact and high performance fuzzy rules and greater generalization capability compared to other self organizing fuzzy neural networks.

4.2.5. Meta-cognitive neuro-fuzzy

Meta-cognitive ideas in the neuro-fuzzy system have emerged in the literature. In McFIS [260,261], a meta-cognitive neuro-fuzzy system using self regulatory approach has been explained. It includes cognitive component consists of TSK based type-0 neuro-fuzzy inference system and meta-cognitive component composed of self regulatory learning mechanism for training of cognitive component. When a new rule is added, the parameters are assigned based on overlapping between the adjacent rule and localization property of the Gaussian rules. W. Zhao [35] proposes a fuzzy-neuro system utilizing the idea of local learning, which mainly focus on the fuzzy interpretability rather than improving the accuracy. Later meta-cognitive learning in neuro-fuzzy technique is extended using the idea of Schema and Scaffolding theory [262]. The technique MCNFIS [34], meta-cognitive complex-valued neuro-fuzzy inference system using TSK which utilizes complex valued neural networks. Both premise part and consequent part are fully complex.

4.2.6. Recurrent

The recurrent property is coming by feeding internal variables from fuzzy firing strength; back either input or output or both layers. Through these configurations, each internal variable feedback is responsible for memorizing the history of its fuzzy rules. J. Zhang [263] invented the idea of recurrent structure in fuzzy neural networks. It uses recurrent radial basis network with Levenberg–Marquardt algorithm for parameter estimation. A self organizing recurrent fuzzy neural network (RSONFIN) [30] with online supervised learning is proposed. RSONFIN uses ordinary Mamdani type fuzzy system. Y. Lin [264] proposes interactively recurrent self organizing fuzzy neural networks (IRSFNN) using the idea of recurrent structure and functional-link neural networks (FLNN). Interaction feedback is composed of local feedback and global feedback that provided by feeding the rule firing strength of each rule to others rules and itself. Consequent part is developed by FLNN which uses trigonometric functions instead of TSK inference system. The antecedent part and recurrent parameters are learned by gradient descent algorithm. The consequent parameters in an IRSFNN are tuned using a variable-dimensional Kalman filter algorithm. Recurrent fuzzy neural network cerebellar model articulation controller (RFNCMAC) [265] is designed for accurate fault-tolerant control of six-phase permanent magnet synchronous motor position servo drive. Incorporation of cerebellar model articulation controller (CMAC) introduces higher generalization capability compared conventional neural networks. An adaptive learning algorithm is designed using Lyapunov stability method for online training. A recurrent self-evolving fuzzy neural network with local feedbacks (RSEFNN-LF) [266] is proposed using TSK fuzzy system. A local rule feedback is inserted to give better accuracy and all the rules are generated online. The premise part of the fuzzy rule and feedback parameters are identified by gradient descent algorithm whereas the consequent part is identified by varying-dimensional Kalman filter algorithm.

4.2.7. Parsimonious network

A parsimonious network based on fuzzy inference system (PANFIS) [267] is proposed whose learning process is based on scratch with an empty rule base. The new rule is added or removed based on statistical contributions of the fuzzy rules and injected datum afterward. The rule is pruned using extended rule significance (ERS) and rule parameters are updated using extended self-organizing map (ESOM) and enhanced recursive least square (ERLS) method. GENEFIS [268] is an extended version of PANFIS. The growing and pruning of fuzzy rule are performed based on statistical contribution composed of extended rule significance (ERS) and Dempster–Shafer theory (DST). The premise parameters are updated by generalized adaptive resonance theory (GART) and consequent parameters are calculated by fuzzily weighted generalized recursive least square (FWGRLS).

4.2.8. Evolving neo-fuzzy neuron

Evolving neo-fuzzy neuron [269] is developed incorporating evolving fuzzy neural modeling into neo-fuzzy neuron (NFN). It uses zero order TSK fuzzy inference system where the parameters are updated using gradient scheme with optimal learning rate. The evolving approach modifies the network structure, the number of membership function and the number of neurons based on evolving neo-fuzzy learning criteria [270,271]. Later above network is improved using adaptive input selection [272].

4.2.9. Wavelet neuro-fuzzy

S. Yilmaz [273] proposed a network called FWNN, where the idea of the fuzzy network is merged with wavelet functions to increase the performance of the neuro-fuzzy system. The unknown parameters of the FWNN models are adjusted by using the Broyden–Fletcher Goldfarb–Shanno (BFGS) gradient method. Later evolving spiking wavelet-neuro-fuzzy system (ESWNFSLS) [274] is proposed**.** The proposed system is developed incorporating evolving fuzzy neural modeling into self-learning spiking neural networks [275] for fast and efficient data clustering. An adaptive wavelet activation function is utilized as the membership function and unsupervised learning is performed by the combined action of 'Winner-Takes-All' rule and temporal Hebbian rule. A hybrid intelligent technology using ANFIS and wavelet transform are proposed [276]. The technique is then applied for short term wind power forecasting in Portugal.

4.2.10. Partition based neuro-fuzzy

Correlated fuzzy neural network, CFNN [277] is proposed that can produce a non-separable fuzzy rule to cover correlated input spaces more efficiently. By considering correlation effect of input data, the number of fuzzy rules generated is considerably reduced. A self-constructing neural fuzzy inference system (DDNFS-CFCM) [278] is proposed using collaborative fuzzy clustering mechanism (DDNFS-CFCM). A Mamdani based forward recursive input-output clustering neuro-fuzzy [36] system is proposed in which forward input-output clustering method is utilized for structure identification. The similar types of fuzzy rules are merged using accurate similarity analysis.

Different self-organizing neuro-fuzzy inference systems including its type of membership, learning criteria, fuzzy technique used and application are shown in Table 6.

5. Comparison study of ELM based neuro-fuzzy systems

The traditional neuro-fuzzy inference systems are tuned iteratively using hybrid techniques and need a long time to learn. In few cases, some of the parameters have to be tuned manually. Gradient based learning techniques may provide local minima and overfitting. In case of ELM, input weights and hidden layer biases are randomly assigned and can be simply considered as a linear system and output weights can be computed through simple generalized inverse operation. Different from traditional learning algorithm, the extreme learning algorithm not only provides the smaller training error but also the better performance. It also seen that, the extreme learning algorithm can learn thousands of times faster than conventional popular learning algorithms for feedforward neural networks [148]. The prediction performance of extreme learning is similar or better than SVM/SVR in many applications. Due to above reasons, ELM based neuro-fuzzy system have emerged in the field of regression and control applications.

Table 6
Details of different self-organizing neuro-fuzzy inference systems.

Algorithm	Membership	Structure	Learning criteria	application	Fuzzy method
SONFIN [246]	Gaussian	A self-constructing	Back propogation algorithm, recursive least square algorithm	Online identification of dynamic system, channel equalization and inverse control applications	TSK
SANFIS [247]	Gaussian	Self-adaptive	Recursive least squares and Levenberg–Marquardt (L–M) algorithms	Classification	TSK
RSONFIN [30]	Gaussian	Recurrent self-organizing	Ordered derivative based back-propagation	Online system identification, inverse control	Mamdani
D-FNN [248]	Gaussian	Self-organizing	Hierarchical learning	Function approximation, nonlinear system identification and time series prediction	TSK
SOFNN [279] [249],	Ellipsoidal basis function (EBF)	Self-organizing	Recursive learning algorithm	Function approximation and system identification	TSK
DENFIS [250]	Evolving clustering method (ECM)	Evolving	Clustering method, recursive least square method	Time series prediction	TSK
SCRG-SVD [252]	Gaussian	Self-constructing	Recursive singular value decomposition-based least squares estimator and the gradient descent method	Function approximation, real world regression	Zero order TSK
SAFIS [253]	Gaussian	Self-organizing	Extended Kalman filter (EKF) scheme	Nonlinear system identification and time series prediction	TSK
PFNN [256]	3-D probabilistic function	Self-organizing	Gradient descent algorithm	Modeling for nonlinear system	TSK
FAOS-PFNN [258]	Gaussian	Self-organizing	Extended Kalman filter (EKF) method.	Modeling, system identification, Time series prediction, real world regression problems.	Sugeno
SOFMLS [259]	Gaussian	Self-organize	Modified least-square algorithm	Nonlinear system identification	Mamdani
GP-FNN [32]	RBF	Self-organizing	Supervised gradient descent method.	Time series prediction, identification and control	TSK
Improved Structure Optimization FNN [280]	Clustering	Self-organizing	Iterative lest squire method and Akaike information criterion (AIC)	SISO and MIMO modeling	TSK
McFIS [260]	Gaussian	Self-organizing	Sample deletion; (2) sample learning; and (3) sample reserve. Extended decoupled Kalman filter	Regression and classification	Zero order TSK
IRSFNN [264]	Gaussian	Recurrent self-evolving	Variable-dimensional Kalman filter algorithm gradient descent algorithm	System identification and time series prediction	TSK or functional-link
Local Learning neuro-fuzzy [35]	Gaussian	Static	Integrated gradient descent	Nonlinear Dynamic Modeling, Prediction of Concrete Compressive Strength	TSK
PANFIS [267]	Multidimensional Gaussian	Self-organizing	Extended self-organizing map (ESOM) theory, enhanced recursive least square (ERLS) method	Nonlinear Modeling, time series Prediction and classification	Zero-order TSK
SOFNN-ACA [281]	Radial basis function (RBF)	Self-organizing growing and pruning	Adaptive computation algorithm (ACA)	Nonlinear dynamic system modeling, time-series prediction, real world prediction	

(*continued on next page*)

Table 6 (*continued*)

Algorithm	Membership	Structure	Learning criteria	application	Fuzzy method
GENEFIS [268]	Gaussian function	Growing and pruning	Generalized adaptive resonance theory (GART) Fuzzily weighted generalized recursive least square (FWGRLS)	Mackey–Glass time series data, real world preidiction	TSK
Evolving neo-fuzzy neuron [269]	Complementary triangular membership functions.	Evolving	Gradient-based scheme with optimal learning rate.	Stream flow forecasting, time-series forecasting, system identification problem	Zero order TSK
ESWNFSLS [274]	Adaptive wavelet activation-membership functions	Evolving (for clustering)	Unsupervised-temporal Hebbian learning algorithm	Image processing	Population coding [274] which is analogous from TSK inference system
CFNN [277]	Multivariable Gaussian fuzzy membership function is	Static (for clustering)	Levenberg–Marquardt (LM) optimization	Function approximation, time series prediction, system identification, real world regression	TSK
DDNFS-CFCM [278]	Gaussian	Evolving (for clustering)	Fuzzy c-means clustering, Collaborative fuzzy clustering	Time series prediction, system identification	TSK
Forward recursive input-output clustering neuro-fuzzy [36]	Gaussian	Evolving (for clustering)	Gradient descent algorithm	Function approximation, system identification	Mamdani fuzzy
SONFIS [282]	Gaussian	Self-organization learning (ANFIS)	Hybrid learning algorithm: LMS and back-propagation	Function approximation, time series prediction	TSK
GENERIC-classifier (gClass) [262]	Multivariate Gaussian functions	Meta-cognitive	Schema and scaffolding theory	Real world classification problems	TSK
RFNCMAC [265]	Gaussian	Recurrent	Lyapunov stability method	Fault-tolerant control of six-phase permanent magnet synchronous motor position servo drive	TSK
ASOA-FNN [251]	Radial basis function (RBF)	Self-organization learning	Adaptive second-order algorithm (ASOA)	Time series prediction, MIMO modeling	TSK
MCNFIS [34]	Gaussian	Self-regulate	Fully complex-valued gradient-descent algorithm	Function approximation, wind speed prediction, multi class classification	TSK
FWNN [273]	Gaussian	Static	Broyden–Fletcher–Goldfarb–Shanno method	System identification, time series prediction	TSK
RSEFNN-LF [266]	Gaussian	Recurrent self-evolving	gradient descent algorithm, Kalman filter algorithm	Chaotic sequence prediction, System identification, time series prediction	TSK

The main advantages of ELM algorithm are that it can train output parameters β_i only with all input parameters w_i are randomly selected. Hence ELM tends to reach the smallest training error with the smallest norm of output weights. It has been shown that SLFNs with smallest training error with the smallest norm of weights produces better generalization performance [148]. Hence the problem can be considered as

$$\text{Minimize}: \|H\beta - D\| \text{ and } \|\beta\|$$

The solution of above can be found by least square estimation method [157].

$$\beta = H^{\oplus} D \tag{1}$$

where $H^{\oplus}$ is called Moore-Penrose generalized inverse of H [153].

In this section, comparative study of ELM based type-1 neuro-fuzzy system is performed for regression problems. The neuro-fuzzy systems such as ETSK [158], OS-Fuzzy-ELM [159], eF-OP-ELM [160], EF-ELM [161] and RELANFIS [164] are considered for comparative analysis. Initially, basic idea with relevant mathematics of above neuro-fuzzy system is presented. Then comparative studies of above techniques are performed for modelling, time series prediction and real world dataset regression. All these methods use the idea of TSK fuzzy inference system because it can learn and memorize temporal information implicitly [283]. Let there be L rules used for the knowledge representation. A TSK network has rules of the form:

$$\text{Rule R}_i : \begin{array}{l} If\ (x_1\ is\ A_{i1})\ and\ (x_2\ is\ A_{i2}) and \ldots \quad and (x_n\ is\ A_{in}) \\ THEN\ (y_1\ is\ B_{i1})\ ,\ (y_2\ is\ B_{i2}), \ldots, (y_m\ is\ B_{im}) \end{array} \tag{2}$$

where $i = 1, 2, \ldots, L$

where $x = [x_1, x_2, ..., x_n]^T$ and $y = [y_1, y_2, ..., y_m]^T$ are crisp inputs and outputs with n-dimensional input variable and m-dimensional output variable. $A_{ij} (j = 1, 2..., n)$ are linguistic variables corresponding to inputs and $B_{ik} (k = 1, 2, .., m)$ are the crisp variable corresponding to outputs, which is represented as linear combination of input variables, i.e.,

$$B_{ik} = p_{ik0} + p_{ik1}x_1 + p_{ik2}x_2 + \ldots + p_{ikn}x_n \tag{3}$$

where $p_{ikl}(l = 0, 1, 2, ..., n)$ are the real-valued parameters.

The membership grades of the input variable x_j satisfy A_{ij} in the rule i can be represented as $\mu_{A_{ij}}(x_j)$

The firing strength of ith rule can be calculated by

$$w_i(x) = \mu_{Ai1}(x_1) \otimes \mu_{Ai2}(x_2) \otimes ... \otimes \mu_{Ain}(x_n) \tag{4}$$

where $\otimes$ indicates *'and'* operator of the fuzzy logic. Since T-norm (triangular norm) product gives a smoothing effect, it is used to obtain the firing strength of a rule by performing the '*and*' membership grades of premise parameters. The normalized firing strength of the ith rule can be found by

$$\bar{w}_i(x) = \frac{w_i(x)}{\sum_{i=1}^{L} w_i(x)} \tag{5}$$

'THEN' part or consequent part consists of linear neural network with p_{ikl} as weight parameters. The system output can be calculated by weighted sum of the output of each normalized rule. Hence, the output of the system is calculated as,

$$y = \frac{\sum_{i=1}^{L} \mathrm{B}_i w_i(x)}{\sum_{i=1}^{L} w_i(x)} = \sum_{i=1}^{L} \mathrm{B}_i \bar{w}_i \tag{6}$$

where $\mathrm{B}_i = (B_{i1}, B_{i2}, ...B_{im})$

5.1. ELM based TSK system (ETSK)

Z.-L. Sun [158] proposed an ELM based TSK fuzzy inference system called ETSK, which uses ELM techniques to tune 'if' and 'then' parameters of TSK system. The basic structure and algorithm was formulated in [284], in which membership function is generated using neural networks in the premise part and determines the parameters in consequent part. In ETSK, conventional neural networks are replaced by ELM and generates new learning algorithm based on ELM learning techniques. ETSK can explain with four major steps as follows.

Step 1-Division of input space: The input data is divided into training data, validation data and testing data. Then training data is grouped based on k-means clustering algorithm. The training data is partitioned into r clusters and each group of training data is represented by R^s where $s = 1, ..., r$, which is expressed as (x_i^s, y_i^s) where $i = 1, ..., (n_t)^s$ and $(n_t)^s$ is the number of training data in each R^s

Step 2-Identification of 'if' part: ELM is used for the identification of 'if' part. If x_i are the values of input layer, corresponding target w_i^s can be found by

$$w_i^s = \begin{cases} 1, & x_i \in R^s \\ 0, & x_i \notin R^s \end{cases} \quad i = 1, ..., (n_t)^s; s = 1, ..., r \tag{7}$$

Hence, learning of ELM is conducted and the degree of attribution $\hat{w}_i^s$ of each training data item to R^s can be inferred. So the output of 'if' part ELM can be written as

$$\mu_A^s(x_i) = \hat{w}_i^s, i = 1, 2, ..., N \tag{8}$$

Step 3-Identification of 'then' part: Structure consists of r number of ELMs which are used to learn 'then' part of each inference rule. The training input x_i^s and the output value y_i^s, $i = 1, 2, ..., (n_t)^s$ are assigned input-output data pair of the ELMs.

Step 4-Computation of final output: Final output value can be derived as,

$$y_i^* = \frac{\sum_{s=1}^{r} \mu_A^s(x_i) \cdot \bar{u}_s(x_i)}{\sum_{s=1}^{r} \mu_A^s(x_i)}, i = 1, 2, ..., N \tag{9}$$

where $\bar{u}_s(x_i)$ is the inferred value obtained after the training of 'then' part. The above algorithm is repeated for a finite number of iteration and parameters are selected corresponding to the smallest validation error. After the training, output can be found out by giving testing data as input.

5.2. Online sequential fuzzy ELM

Online sequential fuzzy ELM (OS-Fuzzy-ELM) was proposed by H. J. Rong [159], where learning can be done with the input data can be given as one-by-one or chunk-by-chunk. It uses TSK fuzzy inference system with membership function parameters selected randomly and is independent of training data. The learning algorithm is explained below in brief.

Step 1-Initialization phase: Randomly select the membership function parameters. For initialization training set, calculate the initial output matrix Y_0 and hidden mode matrix H_0. Then, find initial parameter matrix $Q_0 = P_0 H_0^T Y_0$, where $P_0 = (H_0^T H_0)^{-1}$

Step 2-Learning phase: The parameters of the model are updated for each new set of observation or chunk. The parameter updating equation is given by the following recursive formula

$$\begin{aligned} P_{k+1} &= P_k - P_k H_{k+1}^T \left(I + H_{k+1} P_k H_{k+1}^T\right)^{-1} H_{k+1} P_k \\ Q_{k+1} &= Q_k + P_{k+1} H_{k+1}^T (Y_{k+1} - H_{k+1} Q_k) \end{aligned} \tag{10}$$

5.3. Optimally pruned fuzzy ELM

In order to improve the robustness of conventional ELM, an optimally pruned ELM (OP-ELM) is proposed in [156] where the number of hidden neurons is optimally pruned. It selects the optimum number of nodes by pruning the non-useful hidden neurons. Initially hidden neurons are ranked based on its accuracy, using multi-response sparse regression (MRSR) algorithm [285]. The LOO error is estimated for different number of nodes to find out the optimal number of hidden neurons [286].

In [160], fuzzy logic methods are integrated into OP-ELM strategy discussed above. This method called evolving fuzzy optimally pruned ELM (eF-OP-ELM) uses TSK fuzzy inference methods to construct the fuzzy rules which are fully evolving. The basic steps of algorithms are summarized below.

Step 1: Randomly assign membership function parameters.

Step 2: For a given initialization data set, calculate the hidden node matrix. The best number of rules are selected by ranking the Fuzzy rules of H_0 a based on OP-ELM strategy.

Step 3: For new observation, generate evolving $H(\cdot)$ by applying MRSR and LOO error estimation methods.

Step 4: Find the parameter matrix $Q(\cdot)$ using analytical methods. Hence model output can be computed for any inputs.

5.4. Evolutionary fuzzy ELM (EF-ELM)

In EF-ELM, the differential evolutionary technique is used to tune the membership function parameters whereas the consequent parameters are tuned by Moore–Penrose generalized inverse techniques [161]. Gaussian membership function can be used to represent the membership value, which can be written as

$$\mu_A(x) = \exp\left(\frac{-(x-a)^2}{\sigma^2}\right) \tag{11}$$

where a is the center and σ is the width of $\mu_A(x)$

The DE consists of mutation, crossover and selection. Assume that given a set of parameters $\varphi_{i,G}, i = 1, 2, ..., NP$, where G is the population of each generation. Initial populations are selected in such a way that it can cover the entire parameter space.

Mutation: For a target vector $\varphi_{i,\,G}$, generate mutant vector according to formula

$$v_{i,G+1} = \varphi_{r_1,G} + F\left(\varphi_{r_2,G} - \varphi_{r_3,G}\right) \tag{12}$$

With random integer r_1, r_2, $r_3 \in \{1, 2, ..., NP\}$, and $F > 0$

Crossover: Trail vector

$\alpha_{i,G+1} = (\alpha_{1i,G+1}, \alpha_{2i,G+1}, ..., \alpha_{Di,G+1})$ can be formed, where

$$\alpha_{ji,G+1} = \begin{cases} v_{ji,G+1} if(randb(j) \le C_R) \text{ or } j = nrbr(i) \\ \varphi_{ji,G} if(randb(j) \le C_R) \text{ and } j \ne nrbr(i) \end{cases} \tag{13}$$

where $j = 1, 2, ..., D$, *randb*(*j*) is the *j*th evaluation of a uniform random number with its outcome belong to [0, 1], C_R is the cross over constant belong to [0, 1] and *rnbr*(*j*) is a randomly chosen index belong to [1, *D*]

Selection: The trail vector $\alpha_{i,G+1}$ is compared with target vector $\varphi_{i,\,G}$, if $\alpha_{i,G+1}$ yields a smaller cost function than $\varphi_{i,\,G}$, then $\varphi_{i,G+1}$ is set to $\alpha_{i,G+1}$; otherwise, old value of $\varphi_{i,\,G}$ is retained as $\varphi_{i,G+1}$.

The basic steps of EF-ELM algorithm are summarized below.

Step 1: Randomly generate population individual within the range $[-1 + 1]$. Each individual in the population is composed of premise membership function parameters a and σ.

Step 2: Analytically compute consequent parameters of the TSK fuzzy system using Moore-Penrose generalized inverse.

Step 3: Find out the fitness value of all individual in the population. Fitness function is considered as RMSE on the validation set.

Step 4: Application of three steps of DE such as mutation, crossover and selection to obtain new population. Then above DE process is repeated with new population set, until the required iteration is met.

5.5. Regularized extreme learning ANFIS (RELANFIS)

In RELANFIS, the ELM strategy is applied to tune the parameters [164]. Bell shaped membership function is used for knowledge representation. The premise parameters are generated arbitrarily in a constrained range. These ranges are dependent on the number of membership functions and the span of input. Hence unlike ELM, the randomness of the premise part of RELANFIS is reduced by use of qualitative knowledge embedded in the input membership function part of the fuzzy rules. The consequent parameters are identified by ridge regression [169]. The constrained optimization problem for single output node of RELANFIS can be reformulated by

$$\text{Minimize}: \tfrac{1}{2}\left\|\beta^{+}\right\|^{2} + C\tfrac{1}{2}\sum_{i=1}^{N}\chi_i^2 \tag{14}$$

$$\text{Subjected to}: h(x_i)\beta^{+} = d_i - \chi_i,\ i = 1, 2, \ldots, N \tag{15}$$

where C is the regularization parameter (user specified parameter), χ_i is the training error and $h(x_i)$ is the hidden matrix output with normalized firing strength of fuzzy rule for the input x_i. The consequent parameters of RELANFIS is obtained by

$$\beta^{+} = H^{T}\left(HH^{T} + \frac{I}{C}\right)^{-1} D \tag{16}$$

The training algorithm is summarized below:

Consider N training data, with n attribute inputs $[\ X_1 \quad X_2 \quad . \quad . \quad X_n\ ]_{N\times n}$ and m dimensional outputs $[\ T_1 \quad T_2 \quad . \quad . \quad T_n\ ]_{N\times m}$. Now the range of input can be defined as

$$R_i = MAX\{X_i\} - MIN\{X_i\} \text{ for } i = 1, 2, \ldots n \tag{17}$$

where $MAX\{\cdot\}$ and $MIN\{\cdot\}$ indicates the maximum value and minimum value of the function.

Step 1: Divide the total data in to training, validation and testing data sets.

Step 2: Randomly select the premise parameters (a_j, b_j, and c_j) according to the ranges given below.

$$\frac{a_j^*}{2} \le a_j \le \frac{3a_j^*}{2},\ \text{where } a_j^* = \frac{R_i}{2h - 2} \tag{18}$$

$$1.9 \le b_j \le 2.1 \tag{19}$$

$$\left(c_j^* - \tfrac{d_{cc}}{2}\right) < c_j < \left(c_j^* + \tfrac{d_{cc}}{2}\right) \tag{20}$$

where d_{cc} is distance between two adjacent centres of uniformly distributed membership function. The initial centres ($c_j{}^*$) are selected such that the range of input is divided into equal intervals.

Step 4: Calculate the consequent parameters using (16)

Step 5: The above steps 2 to 4 is run for wide range of C. In this paper the range is selected as $\{2^{-10}, 2^{-9}, ..., 2^{19}, 2^{20}\}$. The best value of parameters is selected by least root mean square error (RMSE) from validation data.

5.6. Performance study

In this sub-section, the performance comparisons of ELM based type-1 neuro-fuzzy algorithms such as ETSK [158], OS-Fuzzy-ELM [159], eF-OP-ELM [160], EF-ELM [161] and RELANFIS [164] are done in the areas of three input modelling, sunspot time series prediction and real world regression. The testing has been done with MATLAB 7.12.0 environment running on an Intel (R) Core (TM) i7-3770 CPU @3.40 GHz. The performance has been compared by considering the root mean square error (RMSE) between actual value and predicted value. The regularization parameter of RELANFIS is selected by grid search method within the range $\{2^{-10}, 2^{-9}, ..., 2^{19}, 2^{20}\}$. In ETSK, according to the suggestions [158], eight sets of input weights are chosen with smallest validation error in 20 trials. Then the final prediction is calculated by taking the average of eight sets taken. The training set is divided into two groups by using the *k*-means clustering method. Number of hidden neurons in ETSK is increased from 1 to 50 and optimal number of hidden neuron is selected corresponding to maximum validation accuracy. The Gaussian membership function is chosen in OS-Fuzzy-ELM with parameters selected at random. Different numbers of rules are given with increment 1 in the range {1,100} and corresponding average cross validation error for 25 trials is computed. Then, the rule and parameters corresponding to the best cross validation error are selected. The block size of OS-Fuzzy-ELM is taken as one. eF-OP-ELM was used with Gaussian kernels along with a maximum of 50 hidden neurons. The parameters of EF-ELM are selected as follows: Population size *NP* is set 100; *F* and C_R are set to 1 and 0.8 respectively. The number of hidden neurons is empirically taken as 50. Total 50 trials have been conducted for all the algorithms and the average results are compared for performance evaluation.

Example 1. Modeling of three input system.

In this example, the above algorithms are tested for a three input-single output function

$$y = \left(1 + x_1^{0.5} + x_2^{-1} + x_3^{-1.5}\right)^2 \tag{21}$$

where *x, y, z* ∈ [1,6]. 512 instances are randomly selected for training and 216 data are used for testing. The performance in terms of RMSE and time taken for training and testing for different ELM based type-1 neuro-fuzzy algorithms are shown in Table 7. As evident from the Table 7, compared to conventional ANFIS, all the ELM based methods have better generalization performance and

Table 7
Performance comparison for three input-single output system (T_R-training, T_S-testing).

		RELANFIS	ANFIS	ETSK	OS-Fuzzy-ELM	eF-OP-ELM	EF-ELM
RMSE	T_R	2.21e−03	0.08591	0.0917	0.019	0.0691	0.095
	T_S	0.02321	0.1072	0.0491	0.036	0.0648	0.102

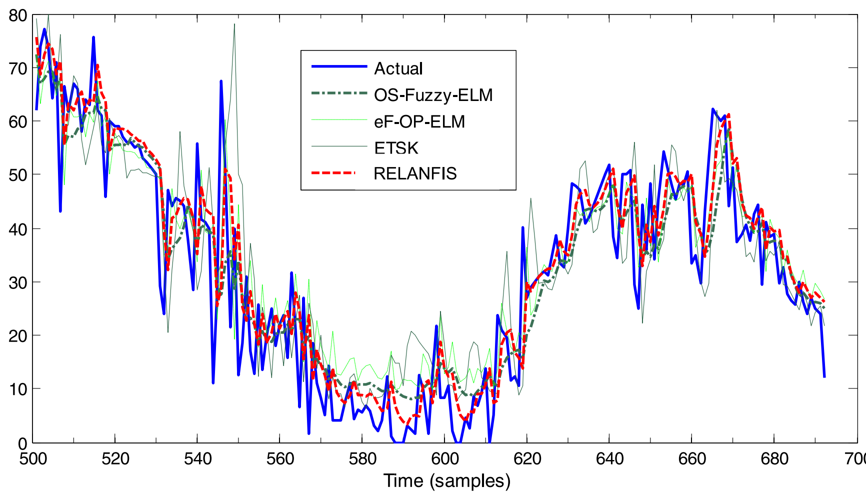

Fig. 3. Solar prediction using different based neuro-fuzzy systems.

Table 8
Performances of ELM based neuro-fuzzy systems for solar prediction.

	RELANFIS	OS-Fuzzy-ELM	eF-OP-ELM	ETSK
RMSE	8.9524	9.7835	9.1981	11.9955
MAE	6.0751	7.5674	6.4563	8.8547
R	0.8992	0.8882	0.8952	0.8675

faster learning speed. It has been also observed that among the ELM based methods, RELANFIS techniques produces better training and testing accuracy without sacrificing the speed. Since EF-ELM is getting arbitrarily higher error compared to other ELM based neuro-fuzzy system, it is omitted for further studies.

Example 2. Time series prediction.

Time series prediction capability of different ELM based type-1 neuro-fuzzy algorithms also investigated using natural solar sunspot activity dataset [287]. The task is to predict the sunspot number $s(t)$ at year t, given as much as 18 previous sunspot numbers. If 4 time lags $s(t-1), s(t-2), s(t-9)$ and $s(t-18)$ are used to predict $s(t)$ then network representation become,

$$\hat{s}(t) = \hat{f}(s(t-1), s(t-2), s(t-9), s(t-18)) \tag{22}$$

Total 700 data are considered, out of these first 70% is used for training and remaining 30% is used for testing. Prediction performance of ELM based neuro-fuzzy techniques are shown in Fig 3. The error index in terms of RMSE and mean average error (MAE) is given in Table 8. It is shown that RELANFIS gives minimum error index (RMSE and MAE) compared other algorithms. This indicates the effectiveness of RELANFIS model to predict time series with minimum error percentage. Correlation coefficient (R) indicates the amount of correlation between actual value and predicted value. The values of R for different forecasting models are also shown in Table 8. There is a well defined presumption that if $|R| > 0.8$, there exists strong correlation between actual value and predicted value [163]. Table 8 indicates that all the four methods produces strong correlation, out of this, RELANFIS gives best correlation among other methods.

Table 9
Data set characteristics.

Data sets	#data	#attributes
Abalone	4176	8
Diabetes	43	2
Servo	167	4
Istanbul Stock Exchange	536	8
Yacht Hydrodynamics	308	7
Auto MPG	398	7

Example 3. real world regression problems.

Here, six real world regression problems [288] are considered for testing the performance of ELM based neuro-fuzzy algorithms. The data set characteristics are tabulated in Table 9. A comparative study of the above mentioned techniques by considering training and testing RMSE are presented in Table 10. Both eF-OP-ELM and OS-Fuzzy-ELM are outperformed ETSK for both training and testing. It has been shown that, RELANFIS produces better training and testing accuracy in terms of RMSE along with good generalization performance than other ELM based neuro-fuzzy approaches. It can be noted among the all five methodologies, the accuracy of RELANFIS algorithm in terms of RMSE for testing is significantly high. Hence RELANFIS can be preferred over other methods for highly accurate modelling.

6. Comments and remarks on various neuro-fuzzy techniques

In this paper, comparisons of various neuro-fuzzy methods are done based on learning technique, fuzzy method used and structural organization. A summary of comparison of neuro-fuzzy systems based on speed of convergence, complexity, classification and regression capability and local minima behaviour are shown in Table 11.

6.1. Convergence speed

It is difficult to rank the convergence speed of various neuro-fuzzy methods in a general way due to several reasons: (1) in the

Table 10
Performance comparison of different ELM based neuro-fuzzy techniques for real world regression problems.

Data set	Algorithm	Training RMSE	Testing RMSE	
			Mean	SD
Abalone	RELANFIS	0.0329	0.0374	0.0399
	ETSK	0.1671	0. 5631	0.0212
	OS-Fuzzy-ELM	0.0712	0.0670	0.0056
	eF-OP-ELM	0.0971	0.0280	0.0798
Diabetes	RELANFIS	0.1071	0.1316	0.1382
	ETSK	0.0941	0.5941	0.0005
	OS-Fuzzy-ELM	0.0339	0.3393	5.73e−06
	eF-OP-ELM	0.0291	0.2903	9.28e−08
Servo	RELANFIS	0.1401	0.1103	0.2178
	ETSK	0.1942	0.3842	0.0758
	OS-Fuzzy-ELM	0.1192	0.3192	0.4384
	eF-OP-ELM	0.1904	0.4904	0.0596
Istanbul Stock Exchange	RELANFIS	3.0614e−04	1.38e−04	1.61e−04
	ETSK	0.0048	0.0059	0.7648
	OS-Fuzzy-ELM	0.00454	0.0045	0.4961
	eF-OP-ELM	0.00461	0.0046	0.0495
Yacht Hydrodynamics	RELANFIS	0.01014	0.04013	0.0472
	ETSK	0.01852	0.0914	0.0572
	OS-Fuzzy-ELM	0.09115	0.0911	0.0288
	eF-OP-ELM	0.09185	0.0919	0.0196
Auto MPG	RELANFIS	0.0217	0.4105	0.7943
	ETSK	0.0923	0.557	0.00739
	OS-Fuzzy-ELM	0.0624	0.771	0.0193
	eF-OP-ELM	0.00146	0.697	0.0138

Table 11
Comparison of neuro-fuzzy technique.

Neuro-fuzzy	Speed of convergence	Algorithm complexity	Classification capability	Regression accuracy	Entrapment in local minima
Gradient based	Medium	Moderate	Moderate	Good	May
Hybrid based	Slow	High	High	Good	May/may not
Population based	Slow	Moderate	Moderate	Good	Never
ELM based	Fast	Simple	High	Good	Never
SVM based	Fast/medium	High	High	Good	Never

literature, neuro-fuzzy schemes are implemented on different software environment, such as matlab, python etc., (2) the computer specifications (RAM, clock speed, etc.) used to simulate the algorithm are different, (3) the effectiveness of neuro-fuzzy systems are verified in different applications such as regression, classification, control systems applications etc., (4) the programming efficiency, i.e. the codes, may or may not be optimized, and (5) the hardware (prototype) used to implement the algorithm is different to each other. Besides that, each researcher defines his own preferable parameters and condition to test the algorithm. Based on these factors, a fair benchmarking for the computational speed is not feasible. Notwithstanding these difficulties, a summary of the convergence for various neuro-fuzzy techniques is presented in Table 11.

Some of the hybrid neuro-fuzzy architectures use the gradient descent backpropagation techniques for the learning its internal parameters. But the learning using gradient descent consumes more time for the training due to improper learning steps and can easily converge to local minima. As in the case of neural networks, for faster convergence, efficient algorithms like conjugated gradient, Levenberg–Marquardt, EKF or recursive least square search must be used in neuro-fuzzy systems also. The main advantages of ELM based neuro-fuzzy methods that, it only requires single iteration to optimize the parameters, hence produces faster computational time compared conventional training scheme.

Different SVM based models also reduce the computational complexity without sacrificing the accuracy. Population based neuro-fuzzy systems algorithms necessitate the huge number of the evaluation of the neuro-fuzzy systems which is generally very slow and time consuming and generally are recommended for offline problems. Memory requirements may also be another disadvantage of these methods.

6.2. Learning complexity

The complexity of different training algorithms used in neuro-fuzzy systems mainly depends on algorithm mathematics and number of iterations. The gradient based method, need some partial derivatives to be computed in order to update the parameters of the neuro-fuzzy systems. This may take more time to update the parameters. In case of hybrid neuro-fuzzy stems, complexity depends on types of the algorithm used. For example, in extended Kalman filter algorithm, the size of covariance matrix is very large when it is used to train the parameters of the neuro-fuzzy system. In Levenberg–Marquardt algorithm, the inverse of a large size matrix is required in each step. In summary, when the parameter search space is too big, these methods start suffering from matrix manipulations.

6.3. Structural complexity

Type-2 based neuro-fuzzy systems are more complex compared to type-1 neuro-fuzzy algorithms. Out of different type-2 neuro-fuzzy system, interval type-2 neuro-fuzzy systems (IT2FNN) have received the most attention because the mathematics that is needed for such sets is primarily interval arithmetic, which is much simpler than the mathematics that is needed for general type-2 neuro-fuzzy systems (GT2FNN).

Table 12
Trade of between interpretability and accuracy in neuro-fuzzy system.

	Interpretability	Accuracy
Premise parameters	Fewer parameters	More parameters
Consequent parameters	Fewer parameters	More parameters
Neuro-fuzzy structure	simple	Complex
No. of rules	Fewer rules	More rules
No. of input feature	Fewer features	More features
Number of linguistic terms	Fewer terms	More terms
Type of fuzzy systems	Mamdani	TSK

6.4. Uncertainty

The detailed review of the type-2 neuro-fuzzy system is performed based on the structure and parameter identification selection. It is shown that type-2 neuro-fuzzy systems outperform most of the type-1 neuro-fuzzy systems and results were promising especially in the presence of significant uncertainties in the system. The parameter identification type-2 neuro-fuzzy systems mainly focused on derivative-based (computational approaches), derivative-free (heuristic methods) and hybrid methods which are the fusion of both the derivative-free and derivative-based methods.

6.5. Interpretability

Interpretability of a fuzzy system means the ability to represent the system behavior in an understandable manner [289]. The Interpretability is a subjective property that depends on various factors like model structure, number of input features, number of rules, number of linguistic terms, type of fuzzy set etc. [290]. The interpretability is not an inherent characteristic of fuzzy inference system and improvement in interpretability leads to the reduction of the accuracy of the neuro-fuzzy systems [291]. Hence when interpretability is important a trade-off between interpretability and accuracy has to be made (Table 12). Traditional data driven neuro-fuzzy system aimed to optimize the accuracy of the fuzzy model. But if accuracy improves, interpretability of fuzzy system may lose. If the membership function obtained after training become distinguishable, it can easily assign the linguistic labels to each of the membership functions. Therefore, the model obtained for the system will be described by a set of linguistic rules, easily interpretable [292]. For better interpretability, learning algorithms should be constrained such that adjacent membership functions do not exchange positions, do not move from positive to negative parts of the domains or vice versa, have a certain degree of overlapping, etc. The other important requirement to obtain interpretability is to keep the rule base small. A neuro-fuzzy model with interpretable membership functions but a very large number of rules is far from being understandable. By reducing the complexity, i.e. the number of parameters of a fuzzy model, not only the rule base is kept manageable but also it can provide a more readable description of the process underlying the data [294].

Most of the neuro-fuzzy systems are implemented based on Takagi–Sugeno type fuzzy inference systems, which is getting less interpretability than the approaches that are based on Mamdani type inference. Neuro-fuzzy systems based on TSK type inference mainly focus on the accuracy rather than the fuzzy interpretability. Taking the ANFIS as an example, it is a two-stage algorithm but the objective is to obtain an accurate overall model output. The overlaps among fuzzy partitions are often so significant such that the resulting local behavior of a responsible fuzzy partition can be dramatically different from the system output. This observation can also be found in fuzzy neural models trained by some heuristic algorithms reported in the literature. For these algorithms, the overall performance of the acquired model is generally satisfactory, but its interpretability is often lost [35].

TSK neuro-fuzzy system can improve interpretability using different techniques. W. Zhao [35] improved the TSK fuzzy systems and proposed a highly interpretable neuro-fuzzy system by utilizing the idea of local learning capability. A zero order TSK based neuro-fuzzy system is developed for high interpretability [26]. Interpretability is improved by generating human-understandable fuzzy rules that work in a parameter space with reduced dimensionality with respect to the space of all the free parameters of the model. M. Velez [27] proposes another TSK based interpretable neuro-fuzzy system which uses overlap area and overlap ratio index to adjust the parameters. A zero-order TSK IT2FNN is developed for high interpretability [201]. Interpretability is maintained using constraint type-2 fuzzy sets. Flexible neuro-fuzzy system is also developed for nonlinear modeling to get more interpretability [291]. A highly interpretable ELM based neuro-fuzzy system is developed for classification purpose [290]. A TSK neuro-fuzzy model is proposed using the constructive cost model (COCOMO) [293]. The parameters of standard COCOMO models are used to initialize the neuro-fuzzy model and therefore accelerate the learning process. The resultant model can be easily interpreted and has good generalization capability. In case of Mamdani based neuro-fuzzy system, interpretability is acknowledged as one of the most appreciated and valuable characteristics of fuzzy system identification methodologies.

6.6. Classification

SVM based neuro-fuzzy systems are popular in solving classification problems. Compared to other neuro-fuzzy techniques, SVM neuro-fuzzy systems optimization delivers a unique solution, since the optimality problem is convex. Since the SVM kernel implicitly contains a non-linear transformation, no assumptions about the functional form of the transformation, which makes data linearly separable, is necessary. The transformation occurs implicitly on a robust theoretical basis and human expertise judgment beforehand is not needed.

6.7. Self-organizing capability

There are numerous kinds of neuro-fuzzy systems proposed in the literature, and most of them are suitable for only off-line cases. In self organizing neuro-fuzzy systems, both structure and parameters are tuned to obtain the reasonable result. Along with parameter tuning algorithm, some structural learning algorithm also specified to self-organize the fuzzy neural structure. In most of the self-organizing neuro-fuzzy systems, prior knowledge of the distribution of the input data is not required for initialization of fuzzy rules. They are automatically generated with the incoming training data. The main advantage of the self-organizing neuro-fuzzy systems has it can be most suitable for online applications.

6.8. Comparative study between different population based neuro-fuzzy systems

Population based learning in neuro-fuzzy systems represents highly competitive alternatives to conventional techniques. Population based algorithms are particularly effective in handling problems with various and often incompatible types of objectives and/or constraints, which are very common in identifications and control of robots. Population based algorithms perform well on problems with noise and uncertain parameters, and produces a robust solutions with adaptable to changing environments. It is noted that systems based on gradient learning can be also taught by any meta-heuristic method without much effort.

Out of different population based neuro-fuzzy systems, PSO based neuro-fuzzy systems outperform others with a larger differential in computational efficiency due to following reasons: (1) PSO have less number of parameters to tune compared to others, (2) PSO algorithm is simple, (3) PSO does not have any genetic operators like recombination, mutation or cross over to optimize the objective function, just like as other heuristic optimization techniques. It is difficult to draw a general conclusion regarding the accuracy of population based neuro-fuzzy systems; it purely depends on structure of neuro-fuzzy system and type of application used. For example, P. Hajek [85] proved that the PSO tuned ANFIS has more accurate result than GA tuned ANFIS for benchmark regression problems. DE-ANFIS has outperformed GA-ANFIS in case of modeling scour at pile groups in clear water condition [56].

6.9. Comparative study of ELM neuro-fuzzy systems

A detailed comparative study of ELM based neuro-fuzzy method is performed in the field of nonlinear modelling, time series prediction and real world benchmark regression. It has been shown that compared to conventional ANFIS, all the ELM based methods have better generalization performance and faster learning speed. Various experiments have shown that RELANFIS has the best training and testing performance. Increased generalization performance due to the use of ELM and reduced randomness due to the use of fuzzy explicit knowledge make the RELANFIS algorithm an attractive choice for modelling dynamic systems.

6.10. Hardware implementation

Neuro-fuzzy systems are powerful algorithms with applications in pattern recognition, memory, mapping, etc. Recently there has been a large push toward a hardware implementation of these networks in order to overcome the calculation complexity of software implementations. Neuro-fuzzy systems can be implemented in analog IC chips or CMOS digital circuitry. Recent advances in reprogrammable logic enable implementing large neuro-fuzzy system on a single field-programmable gate array (FPGA) device. The main reason for this is the miniaturization of component manufacturing technology, where the data density of electronic components doubles every 18 months. The recent challenges of hardware implementation include higher speed, smaller size, parallel processing capability, small cost etc. It can be noted that ELM based neuro-fuzzy systems can easily implementable in the digital environment due to less complexity. The details of the hardware implementation of neuro-fuzzy system are depicted in [12].

7. Conclusion

This paper gives a survey of research advances in the neuro-fuzzy systems. Due to the vast number of common tools, it continues to be difficult to compare conceptually the different architectures and to evaluate comparatively their performances. Tabular comparisons of neuro-fuzzy techniques are provided at each section, which will help the researchers to choose a particular neuro-fuzzy technique on the basis of learning criteria, fuzzy method, structure and application. Finally, some comments and remarks of all neuro-fuzzy systems are given based on certain parameters like accuracy, complexity, classification and regression ability, convergence time, self-organizing capability, and hardware implementation. This review is expected to be helpful for the researchers working in the area of development of neuro-fuzzy systems and its applications.

References

[1] S. Mitra, Y. Hayashi, Neuro-fuzzy rule generation: survey in soft computing framework, IEEE Trans. Neural Networks 11 (3) (2000) 748–768.

[2] J. Yen, L. Wang, C.W. Gillespie, Improving the interpretability of TSK fuzzy models by combining global learning and local learning, IEEE Trans. Fuzzy Syst. 6 (4) (1998) 530–537.

[3] J.R. Jang, ANFIS: adaptive-network-based fuzzy inference system, IEEE Trans. Syst. Man, Cybern. 23 (3) (1993) 665–685.

[4] J.S.R. Jang, C.T. Sun, Neuro-fuzzy modeling and control, Proc. IEEE 83 (3) (1995) 378–406.

[5] J.-S.R. Jang, C.-T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing—A Computational Approach to Learning and Machine Intelligence, Prentice Hall, Upper Saddle River, NJ, 1997.

[6] R. Fuller, Neural Fuzzy Systems, Lecture Notes, Abo Akademi University, 1995.

[7] O. Nelles, Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models, Springer-Verlag, Berlin, Heidelberg, New York, 2001.

[8] A. Abraham, Neuro fuzzy systems: state-of-the-art modeling techniques, in: International Work-Conference on Artificial Neural Networks, IWANN 2001, 2001, pp. 269–276.

[9] J. Vieira, F.M. Dias, A. Mota, Neuro-fuzzy systems: a survey, WSEAS Trans. Syst. 3 (2) (2004) 414–419.

[10] S. Sahin, M.R. Tolun, R. Hassanpour, Hybrid expert systems: a survey of current approaches and applications, Expert Syst. Appl. 39 (4) (2012) 4609–4617.

[11] S. Kar, S. Das, P.K. Ghosh, Applications of neuro fuzzy systems: a brief review and future outline, Appl. Soft Comput. 15 (2014) 243–259.

[12] G. Bosque, I. Del Campo, J. Echanobe, Fuzzy systems, neural networks and neuro-fuzzy systems: a vision on their hardware implementation and platforms over two decades, Eng. Appl. Artif. Intell. 32 (1) (2014) 283–331.

[13] Z.J. Viharos, K.B. Kis, Survey on neuro-fuzzy systems and their applications in technical diagnostics and measurement, Meas.: J. Int. Meas. Confed. 67 (2015) 126–136.

[14] S. Rajab, V. Sharma, A review on the applications of neuro-fuzzy systems in business, Artif. Intell. Rev. 49 (4) (2017) 1–30.

[15] Ł. Bartczuk, A. Przybył, K. Cpałka, A new approach to nonlinear modelling of dynamic systems based on fuzzy rules, Int. J. Appl. Math. Comput. Sci. 26 (3) (2016) 603–621.

[16] K. Lapa, K. Cpalka, Flexible fuzzy PID controller (FFPIDC) and a nature-inspired method for its construction, IEEE Trans. Ind. Inf. 14 (3) (2018) 1–10.

[17] L. Rutkowski, A. Przybył, K. Cpałka, Novel online speed profile generation for industrial machine tool based on flexible neuro-fuzzy approximation, IEEE Trans. Ind. Electron. 59 (2) (2012) 1238–1247.

[18] K.V Shihabudheen, G.N. Pillai, Internal model control based on extreme learning ANFIS for nonlinear application, in: IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2015, pp. 1–5.

[19] K.V. Shihabudheen, S.K. Raju, G.N. Pillai, Control for grid-connected DFIG-based wind energy system using adaptive neuro-fuzzy technique, Int. Trans. Electr. Energy Syst. (2018) e2526.

[20] K. Cpałka, M. Zalasiński, L. Rutkowski, A new algorithm for identity verification based on the analysis of a handwritten dynamic signature, Appl. Soft Comput. J. 43 (2016) 47–56.

[21] K. Cpałka, M. Zalasiński, On-line signature verification using vertical signature partitioning, Expert Syst. Appl. 41 (9) (2014) 4170–4180.

[22] K. Cpałka, M. Zalasiński, L. Rutkowski, New method for the on-line signature verification based on horizontal partitioning, Pattern Recognit. 47 (8) (2014) 2652–2661.

[23] S. Horikawa, T. Furuhashi, Y. Uchikawa, On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm, IEEE Trans. Neural Networks 3 (5) (1992) 801–806.

[24] Y. Shi, M. Mizumoto, A new approach of neuro-fuzzy learning algorithm for tuning fuzzy rules, Fuzzy Sets Syst. 112 (1) (2000) 99–116.

[25] A.K. Palit, G. Doeding, W. Anheier, D. Popovic, Backpropagation based training algorithm for Takagi–Sugeno type MIMO neuro-fuzzy network to forecast electrical load time series, in: Proceedings of the 2002 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'02), 2002, pp. 86–91.

[26] G. Castellano, A.M. Fanelli, C. Mencar, A neuro-fuzzy network to generate human-understandable knowledge from data, Cognit. Syst. Res. 3 (2002) 125–144.

[27] M.A. Velez, O. Sanchez, S. Romero, J.M. Andujar, A new methodology to improve interpretability in neuro-fuzzy TSK models, Appl. Soft Comput. 10 (2) (2010) 578–591.

[28] M. Korytkowski, R. Nowicki, L. Rutkowski, R. Scherer, Merging ensemble of neuro-fuzzy systems, in: IEEE International Conference on Fuzzy Systems, 2006, pp. 1954–1957.

[29] W. Wu, L. Li, J. Yang, Y. Liu, A modified gradient-based neuro-fuzzy learning algorithm and its convergence, Inf. Sci. 180 (9) (2010) 1630–1642.

[30] C.-F. Juang, C.-T. Lin, A recurrent self-organizing neural fuzzy inference network, IEEE Trans. Neural Networks 10 (4) (1999) 828–845.

[31] D. Chakraborty, N.R. Pal, A neuro-fuzzy scheme for simultaneous feature selection and fuzzy rule-based classification, IEEE Trans. Neural Networks 15 (1) (2004) 110–123.

[32] H. Han, S. Member, J. Qiao, A self-organizing fuzzy neural network based on a growing-and-pruning algorithm, IEEE Trans. Fuzzy Syst. 18 (6) (2010) 1129–1143.

[33] K. Subramanian, R. Savitha, S. Suresh, B.S. Mahanand, Complex-valued neuro–fuzzy inference system based classifier, in: B.K. Panigrahi, S. Das, P.N. Suganthan, K. Nanda (Eds.), SEMCCO 2012, Lecture Notes in Computer Science Book Series, 7677, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 348–355.
[34] K. Subramanian, R. Savitha, S. Suresh, A complex-valued neuro-fuzzy inference system and its learning mechanism, Neurocomputing 123 (2014) 110–120.
[35] W. Zhao, K. Li, G.W. Irwin, A new gradient descent approach for local learning of fuzzy neural models, IEEE Trans. Fuzzy Syst. 21 (1) (2013) 30–44.
[36] J. Qiao, W. Li, X. Zeng, H. Han, Identification of fuzzy neural networks by forward recursive input-output clustering and accurate similarity analysis, Appl. Soft Comput. 49 (2016) 524–543.
[37] Y. Shi, M. Mizumoto, Some considerations on conventional neuro-fuzzy learning algorithms by gradient descent method, Fuzzy Sets Syst. 112 (1) (2000) 51–63.
[38] M. Han, J. Xi, S. Xu, F. Yin, Prediction of chaotic time series based on the recurrent predictor neural network, IEEE Trans. Signal Process. 52 (12) (2004) 3409–3416.
[39] M. Srinivas, L.M. Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms, IEEE Trans. Syst. Man Cybern.—Part B: Cybern. 24 (4) (1994) 656–667.
[40] J. Zhang, H.S.. Chung, B.. Hu, Adaptive probabilities of crossover and mutation in genetic algorithms based on clustering technique, in: Proceedings of IEEE Congress on Evolutionary Computation (CEC2004), 2004, pp. 2280–2287.
[41] C. Juang, A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms, IEEE Trans. Fuzzy Syst. 10 (2) (2002) 155–170.
[42] C.-F. Juang, A hybrid of genetic algorithm and particle swarm optimization for recurrent network design, IEEE Trans. Syst. Man Cybern. Part B, Cybern. 34 (2) (2004) 997–1006.
[43] S. Mishra, P.K. Dash, P.K. Hota, M. Tripathy, Genetically optimized neuro-fuzzy ipfc for damping modal oscillations of power system, IEEE Trans. Power Syst. 17 (4) (2002) 1140–1147.
[44] A.M. Tang, C. Quek, G.S. Ng, GA-TSKfnn: parameters tuning of fuzzy neural network using genetic algorithms, Expert Syst. Appl. 29 (4) (2005) 769–781.
[45] I. Svalina, G. Simunovic, T. Saric, R. Lujic, Evolutionary neuro-fuzzy system for surface roughness evaluation, Appl. Soft Comput. 52 (2017) 593–604.
[46] M. Shahram, Optimal design of adaptive neuro-fuzzy inference system using genetic algorithm for electricity demand forecasting in Iranian industry, Soft Comput. 20 (12) (2016) 4897–4906.
[47] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31.
[48] S. Das, S.S. Mullick, P.N. Suganthan, Recent advances in differential evolution–an updated survey, Swarm Evol. Comput. 27 (2016) 1–30.
[49] R.A. Aliev, B.G. Guirimov, B. Fazlollahi, R.R. Aliev, Evolutionary algorithm-based learning of fuzzy neural networks. Part 2: recurrent fuzzy neural networks, Fuzzy Sets Syst. 160 (2009) 2553–2566.
[50] C. Lin, M. Han, Y. Lin, S. Liao, J. Chang, Neuro-fuzzy system design using differential evolution with local information, in: IEEE International Conference on Fuzzy Systems, 2011, pp. 1003–1006.
[51] M. Han, C. Lin, J. Chang, Differential evolution with local information for neuro-fuzzy systems optimisation, Knowledge-Based Syst. 44 (2013) 78–89.
[52] C.-H. Chen, C.-J. Lin, C.-T. Lin, Nonlinear system control using adaptive neural fuzzy networks based on a modified differential evolution, IEEE Trans. Syst. Man Cybern.–Part C: Appl. Rev. 39 (4) (2009) 459–473.
[53] M. Su, C. Chen, C. Lin, C. Lin, A rule-based symbiotic modified differential evolution for self-organizing neuro-fuzzy systems, Appl. Soft Comput. 11 (8) (2011) 4847–4858.
[54] C. Chen, S. Yang, Efficient DE-based symbiotic cultural algorithm for neuro–fuzzy system design, Appl. Soft Comput. 34 (2015) 18–25.
[55] R. Dash, P. Dash, Efficient stock price prediction using a self evolving recurrent neuro-fuzzy inference system optimized through a modified differential harmony search technique, Expert Syst. Appl. 52 (2016) 75–90.
[56] H. Azimi, H. Bonakdari, I. Ebtehaj, S. Hamed, A. Talesh, Evolutionary Pareto optimization of an ANFIS network for modeling scour at pile groups in clear water condition, Fuzzy Sets Syst. 319 (2017) 50–69.
[57] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE Comput. Intell. Mag. 1 (4) (2006) 28–39.
[58] C. Blum, Ant colony optimization: introduction and recent trends, Phys. Life Rev. 2 (4) (2005) 353–373.
[59] M. Dorigo, T. Stutzle, Ant colony optimization: overview and recent advances, in: M. Gendreau, J.Y. Potvin (Eds.), Handbook of Metaheuristics, Vol. 57, Springer, US, 2010, pp. 227–263. International Series in Operations Research & Management Science.
[60] Z. Baojiang, Optimal design of neuro-fuzzy controller based on ant colony algorithm, in: Proceedings of the 29th Chinese Control Conference, CCC'10, 2010, pp. 2513–2518.
[61] D.G. Rodrigues, G. Moura, C.M.C. Jacinto, P. Jose, D.F. Filho, M. Roisenberg, Generating Interpretable Mamdani-type fuzzy rules using a neuro-fuzzy system based on radial basis functions, in: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2014, pp. 1352–1359.
[62] Z. Baojiang, L. Shiyong, Ant colony optimization algorithm and its application to neuro-fuzzy controller design, J. Syst. Eng. Electron. 18 (3) (2007) 603–610.
[63] A. Azad, H. Karami, S. Farzin, A. Saeedian, H. Kashi, F. Sayyahi, Prediction of water quality parameters using ANFIS optimized by intelligence algorithms (case study: Gorganrood River), KSCE J. Civil Eng. (2017) 1–8.
[64] H. Shekofteh, F. Ramazani, H. Shirani, Optimal feature selection for predicting soil CEC: Comparing the hybrid of ant colony organization algorithm and adaptive network-based fuzzy system with multiple linear regression, Geoderma 298 (2017) 27–34.
[65] K.-T. Abdorreza, S. Hajirezaie, H.-S. Abdolhossein, M.M. Husein, K. Karan, M. Sharifi, Application of adaptive neuro fuzzy interface system optimized with evolutionary algorithms for modeling CO 2 -crude oil minimum miscibility pressure, Fuel 205 (2017) 34–45.
[66] S.V.R. Termeh, A. Kornejady, H.R. Pourghasemi, S. Keesstra, Science of the Total Environment Flood susceptibility mapping using novel ensembles of adaptive neuro fuzzy inference system and metaheuristic algorithms, Sci. Total Environ. 615 (2018) 438–451.
[67] J. Kennedy, R. Eberhart, Particle swarm optimization, in: 1995 IEEE International Conference on Neural Networks (ICNN 95), Vol. 4, 1995, pp. 1942–1948.
[68] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39–43.
[69] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8 (3) (2004) 240–255.
[70] R. Roy, S. Dehuri, S.B. Cho, A novel particle swarm optimization algorithm for multi-objective combinatorial optimization problem, Int. J. Appl. Metaheuristic Comput. 2 (4) (2011) 41–57.
[71] A. Cheung, X.-M. Ding, H.-B. Shen, OptiFel: a convergent heterogeneous particle swarm optimization algorithm for Takagi-Sugeno fuzzy modeling, IEEE Trans. Fuzzy Syst. 22 (4) (2014) 919–933.
[72] A. Chatterjee, K. Pulasinghe, K. Watanabe, K. Izumi, A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems, IEEE Trans. Ind. Electron. 52 (6) (2005) 1478–1489.
[73] M.V Oliveira, R. Schirru, Applying particle swarm optimization algorithm for tuning a neuro-fuzzy inference system for sensor monitoring, Prog. Nuclear Energy 51 (2009) 177–183.
[74] M.A. Shoorehdeli, M. Teshnehlab, A.K. Sedigh, M.A. Khanesar, Identification using ANFIS with intelligent hybrid stable learning algorithm approaches and stability analysis of training methods, Appl. Soft Comput. 9 (2009) 833–850.
[75] D.P. Rini, S.M. Shamsuddin, S.S. Yuhaniz, Particle swarm optimization for ANFIS interpretability and accuracy, Soft Comput. 20 (2016) 251–262.
[76] M.Y. Chen, A hybrid ANFIS model for business failure prediction utilizing particle swarm optimization and subtractive clustering, Inf. Sci. 220 (2013) 180–195.
[77] R.L. Squares, F. Link, A. Neural, A. Neural, R.B. Function, PSO tuned ANFIS equalizer based on fuzzy C-means clustering algorithm, Int. J. Electron. Commun. (AEÜ) 70 (2016) 799–807.
[78] M.A. Shoorehdeli, M. Teshnehlab, A.K. Sedigh, Training ANFIS as an identifier with intelligent hybrid stable learning algorithm based on particle swarm optimization and extended Kalman filter, Fuzzy Sets Syst. 160 (7) (2009) 922–948.
[79] J.P.S. Catalao, H.M.I. Pousinho, V.M.F. Mendes, Hybrid wavelet-PSO-ANFIS approach for short-term electricity prices forecasting, IEEE Trans. Power Syst. 26 (1) (2011) 137–144.
[80] J.P.S. Catalao, H.M.I. Pousinho, V.M.F. Mendes, Hybrid wavelet-PSO-ANFIS approach for short-term electricity wind power forecasting in Portugal, IEEE Trans. Sustainable Energy 2 (1) (2011) 50–59.
[81] A. Bagheri, H. Mohammadi, M. Akbari, Expert systems with applications financial forecasting using ANFIS networks with quantum-behaved particle swarm optimization, Expert Syst. Appl. 41 (14) (2014) 6235–6250.
[82] A. Chatterjee, P. Siarry, A PSO-aided neuro-fuzzy classifier employing linguistic hedge concepts, Expert Syst. Appl. 33 (2007) 1097–1109.
[83] C. Lin, An efficient immune-based symbiotic particle swarm optimization learning algorithm for TSK-type neuro-fuzzy networks design, Fuzzy Sets Syst. 159 (21) (2008) 2890–2909.
[84] A. Mozaffari, N.L. Azad, An ensemble neuro-fuzzy radial basis network with self-adaptive swarm based supervisor and negative correlation for modeling automotive engine coldstart hydrocarbon emissions: a soft solution to a crucial automotive problem, Appl. Soft Comput. 32 (2015) 449–467.
[85] P. Hájek, V. Olej, Intuitionistic neuro-fuzzy network with evolutionary adaptation, Evolving Syst. 8 (1) (2017) 35–47.
[86] S. Chakravarty, P.K. Dash, A PSO based integrated functional link net and interval type-2 fuzzy logic system for predicting stock market indices, Appl. Soft Comput. 12 (2) (2012) 931–941.
[87] C. Lin, S. Hong, The design of neuro-fuzzy networks using particle swarm optimization and recursive singular value decomposition, Neurocomputing 71 (2007) 297–310.
[88] C. Lin, C. Chen, C. Lin, Efficient self-evolving evolutionary learning for neuro-fuzzy inference systems, IEEE Trans. Fuzzy Syst. 16 (6) (2008) 1476–1490.
[89] C. Lin, C. Chen, C. Lin, A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications, IEEE Trans. Syst. Man Cybern.—Part C: Appl. Rev. 39 (1) (2009) 55–68.
[90] M. Elloumi, M. Krid, D.S. Masmoudi, A self-constructing neuro-fuzzy classifier for breast cancer diagnosis using swarm intelligence, in: IEEE International conference on Image Processing, Applications and Systems (IPAS), 2016, pp. 1–6.
[91] P. Lučić, D. Teodorović, Computing with bees: attacking complex transportation engineering problems, Int. J. Artif. Intell. Tools 12 (3) (2003) 375–394.
[92] D. Karaboga, E. Kaya, Training ANFIS using artificial bee colony algorithm, in:

IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA), 2013, pp. 1–5.
[93] K. Dervis, E. Kaya, Training ANFIS by using the artificial bee colony algorithm, Turkish J. Electr. Eng. Comput. Sci. 25 (3) (2017) 1669–1679.
[94] F.H. Ismail, M.A. Aziz, A.E. Hassanien, Optimizing the parameters of Sugeno based adaptive neuro fuzzy using artificial bee colony: a case study on predicting the wind speed, in: Proceedings of the Federated Conference on Computer Science and Information Systems, 2016, pp. 645–651.
[95] D. Karaboga, E. Kaya, An adaptive and hybrid artificial bee colony algorithm (aABC) for ANFIS training, Appl. Soft Comput. 49 (2016) 423–436.
[96] G. Liao, T. Tsao, Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting, IEEE Trans. Evol. Comput. 10 (3) (2006) 330–340.
[97] S. Ho, L. Shu, S. Ho, Optimizing fuzzy neural networks for tuning PID controllers using an orthogonal simulated annealing algorithm OSA, IEEE Trans. Fuzzy Syst. 14 (3) (2006) 421–434.
[98] X. Yang, S. Deb, Cuckoo search via levy flights, in: Proceedings of IEEE World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), 2009, pp. 210–214.
[99] J.-F. Chen, Q.H. Do, A cooperative cuckoo search – hierarchical adaptive neuro-fuzzy inference system approach for predicting student academic performance, J. Intell. Fuzzy Syst. 27 (5) (2014) 2551–2561.
[100] S. Araghi, A. Khosravi, D. Creighton, Design of an optimal ANFIS traffic signal controller by using cuckoo search for an isolated intersection, in: IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2015, pp. 2078–2083.
[101] X. Ding, Z. Xu, N.J. Cheung, X. Liu, Parameter estimation of Takagi–Sugeno fuzzy system using heterogeneous cuckoo search algorithm, Neurocomputing 151 (2015) 1332–1342.
[102] F. Glover, Tabu search–part I, ORSA J. Comput. 1 (3) (1989) 190–206.
[103] F. Glover, Tabu search–part II, ORSA J. Comput. 2 (I) (2001) 4–32.
[104] G. Liu, Y. Fang, X. Zheng, Y. Qiu, Tuning neuro-fuzzy function approximator by Tabu search, in: International Symposium on Neural Networks (ISNN), 2004, pp. 276–281.
[105] H.M. Khalid, S.Z. Rizvi, R. Doraiswami, L. Cheded, A. Khoukhi, A Tabu-search based neuro-fuzzy inference system for fault diagnosis, in: IET UKACC International Conference on Control, 2010, pp. 1–6.
[106] P.K. Mohanty, D.R. Parhi, A new hybrid optimization algorithm for multiple mobile robots navigation based on the CS-ANFIS approach, Memetic Comput. 7 (4) (2015) 255–273.
[107] C. Cortes, V. Vapnik, Support vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[108] J.A.K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (3) (1999) 293–300.
[109] O.L.M. Fung, Glenn, Proximal support vector machine, in: in Proc. Int. Conf. Knowl. Discov. Data Mining, 2001, pp. 77–86.
[110] G.M. Fung, O.L. Mangasarian, Multicategory proximal support vector machine classifiers, Mach. Learn. 59 (1–2) (2005) 77–97.
[111] H. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, V. Vapnik, Support vector regression machines, in: Proceedings of the 9th International Conference on Neural Information Processing Systems (NIPS 1996), 1996, pp. 155–161.
[112] A.J. Smola, B. Scholkopf, A tutorial on support vector regression, Stat. Comput. 14 (3) (2004) 199–222.
[113] B. Scholkopf, et al., Comparing support vector machines with Gaussian kernels to radial basis function classifiers, IEEE Trans. Signal Process. 45 (11) (1997) 2758–2765.
[114] B. Scholkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, London, England, 2002.
[115] C.-F. Lin, S.-D. Wang, Fuzzy support vector machines, IEEE Trans. Neural Networks 13 (2) (2002) 464–471.
[116] E.C.C. Tsang, D.S. Yeung, P.P.K. Chan, Fuzzy support vector machines for solving two-class problems, in: Proceedings of the Second International Conference on Machine Learning and Cybernetics, 2003, pp. 1080–1083.
[117] D. Tsujinishi, S. Abe, Fuzzy least squares support vector machines for multiclass problems, Neural Networks 16 (2003) 785–792.
[118] H.-P. Huang, Y.-H. Liu, Fuzzy support vector machines for pattern recognition and data mining, Int. J. Fuzzy Syst. 4 (3) (2002) 826–835.
[119] C. fu Lin, S. de Wang, Training algorithms for fuzzy support vector machines with noisy data, Pattern Recognit. Lett. 25 (14) (2004) 1647–1656.
[120] D.H. Hong, C. Hwang, Support vector fuzzy regression machines, Fuzzy Sets Syst. 138 (2) (2003) 271–281.
[121] Z.S.Z. Sun, Y.S.Y. Sun, Fuzzy support vector machine for regression estimation, in: IEEE International Conference on Systems, Man and Cybernetics, 4, 2003, pp. 3336–3341.
[122] Y. Wang, S. Wang, K.K. Lai, A new fuzzy support vector machine to evaluate credit risk, IEEE Trans. Fuzzy Syst. 13 (6) (2005) 820–831.
[123] S. Xiong, H. Liu, X. Niu, Fuzzy support vector machines based on FCM Clustering, in: Proceedings of Fourth International Conference on Machine Learning and Cybernetics, 2005, pp. 18–21.
[124] A. Çelikyilmaz, I. Burhan Türkşen, Fuzzy functions with support vector machines, Inf. Sci. 177 (23) (2007) 5163–5177.
[125] Y. Chen, J.Z. Wang, Support vector learning for fuzzy rule-based classification systems, IEEE Trans. Fuzzy Syst. 11 (6) (2003) 716–728.
[126] X. Yang, G. Zhang, J. Lu, J. Ma, A kernel fuzzy c-Means clustering-based fuzzy support vector machine algorithm for classification problems with outliers or noises, IEEE Trans. Fuzzy Syst. 19 (1) (2011) 105–115.
[127] S. Ding, Y. Han, J. Yu, Y. Gu, A fast fuzzy support vector machine based on information granulation, Neural Comput. Appl. 23 (6) (2013) 139–144.
[128] W. An, M. Liang, Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noises, Neurocomputing 110 (2013) 101–110.
[129] J. Hang, J. Zhang, M. Cheng, Application of multi-class fuzzy support vector machine classifier for fault diagnosis of wind turbine, Fuzzy Sets Syst. 297 (2016) 128–140.
[130] Q. Wu, et al., Hybrid BF-PSO and fuzzy support vector machine for diagnosis of fatigue status using EMG signal features, Neurocomputing 173 (2016) 483–500.
[131] C.-T. Lin, C.-M. Yeh, S.-F. Liang, J.-F. Chung, N. Kumar, Support-vector-based fuzzy neural network for pattern classification, IEEE Trans. Fuzzy Syst. 14 (1) (2006) 31–41.
[132] J.-H. Chiang, P.-Y. Hao, Support vector learning mechanism for fuzzy rule-based modeling: a new approach, IEEE Trans. Fuzzy Syst. 12 (1) (2004) 1–12.
[133] C.F. Juang, S.H. Chiu, S.W. Chang, A self-organizing TS-type fuzzy network with support vector learning and its application to classification problems, IEEE Trans. Fuzzy Syst. 15 (5) (2007) 998–1008.
[134] C.-F. Juang, S.-J. Shiu, Using self-organizing fuzzy network with support vector learning for face detection in color images, Neurocomputing 71 (16) (2008) 3409–3420.
[135] C.C. Chuang, Fuzzy weighted support vector regression with a fuzzy partition, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 37 (3) (2007) 630–640.
[136] C.F. Juang, C.D. Hsieh, A locally recurrent fuzzy neural network with support vector regression for dynamic-system modeling, IEEE Trans. Fuzzy Syst. 18 (2) (2010) 261–273.
[137] C.F. Juang, C. Da Hsieh, A fuzzy system constructed by rule generation and iterative linear svr for antecedent and consequent parameter optimization, IEEE Trans. Fuzzy Syst. 20 (2) (2012) 372–384.
[138] R. Khemchandani Jayadeva, S. Chandra, Fast and robust learning through fuzzy linear proximal support vector machines, Neurocomputing 61 (2004) 401–411.
[139] Q. Wu, Regression application based on fuzzy ν-support vector machine in symmetric triangular fuzzy space, Expert Syst. Appl. 37 (4) (2010) 2808–2814.
[140] B. Scholkopf, A.J. Smola, R.C. Williamson, P.L. Bartlett, New support vector algorithms, Neural Comput. 12 (5) (2000) 1207–1245.
[141] Q. Wu, Hybrid fuzzy support vector classifier machine and modified genetic algorithm for automatic car assembly fault diagnosis, Expert Syst. Appl. 38 (3) (2011) 1457–1463.
[142] Q. Wu, R. Law, Fuzzy support vector regression machine with penalizing Gaussian noises on triangular fuzzy number space, Expert Syst. Appl. 37 (12) (2010) 7788–7795.
[143] A. Miranian, M. Abdollahzade, Developing a local least-squares support vector machines-based neuro-fuzzy model for nonlinear and chaotic time series prediction, IEEE Trans. Neural Networks Learn. Syst. 24 (2) (2013) 207–218.
[144] L.I.U. Han, D. Yi, Z. Lifan, L.I.U. Ding, Neuro-fuzzy control based on on-line least square support vector machines, in: Proceedings of the 34th Chinese Control Conference, 2015, pp. 3411–3416.
[145] S. Iplikci, Support vector machines based neuro-fuzzy control of nonlinear systems, Neurocomputing 73 (10–12) (2010) 2097–2107.
[146] M. Zhang, X. Liu, A soft sensor based on adaptive fuzzy neural network and support vector regression for industrial melt index prediction, Chemom. Intell. Lab. Syst. 126 (2013) 83–90.
[147] G. Huang, C. Siew, Extreme learning machine with randomly assigned RBF kernels, Int. J. Inf. Technol. 11 (1) (2005) 16–24.
[148] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.
[149] G.-B. Huang, X. Ding, H. Zhou, Optimization method based extreme learning machine for classification, Neurocomputing 74 (2010) 155–163.
[150] G.B. Huang, D.H. Wang, Y. Lan, Extreme learning machines: a survey, Int. J. Mach. Learn. Cybern. 2 (2) (2011) 107–122.
[151] N. Wang, M.J. Er, M. Han, Generalized single-hidden layer feedforward networks for regression problems, IEEE Trans. Neural Networks Learn. Syst. 26 (6) (2015) 1161–1176.
[152] X. Liu, S. Lin, J. Fang, Z. Xu, Is extreme learning machine feasible? A theoretical assessment (Part II), IEEE Trans. Neural Networks Learn. Syst. 26 (1) (2015) 21–34.
[153] G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Networks 17 (4) (2006) 879–892.
[154] G. Bin Huang, M. Bin Li, L. Chen, C.K. Siew, Incremental extreme learning machine with fully complex hidden nodes, Neurocomputing 71 (4–6) (2008) 576–583.
[155] N.-Y. Liang, G.-B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Trans. Neural Networks 17 (6) (2006) 1411–1423.
[156] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, A. Lendasse, OP-ELM: optimally pruned extreme learning machine, IEEE Trans. Neural Networks 21 (1) (2010) 158–162.
[157] W.B. Zhang, H.B. Ji, Fuzzy extreme learning machine for classification, Electron. Lett. 49 (7) (2013) 448–450.
[158] Z.-L. Sun, K.-F. Au, T.-M. Choi, A neuro-fuzzy inference system through integration of fuzzy logic and extreme learning machines, IEEE Trans. Syst. Man Cybern. Part B, Cybern. 37 (5) (2007) 1321–1331.

[159] H.J. Rong, G. Bin Huang, N. Sundararajan, P. Saratchandran, H.J. Rong, Online sequential fuzzy extreme learning machine for function approximation and classification problems, IEEE Trans. Syst. Man Cybern., Part B: Cybern. 39 (4) (2009) 1067–1072.
[160] F.M. Pouzols, A. Lendasse, Evolving fuzzy optimally pruned extreme learning machine for regression problems, Evolving Syst. 1 (1) (2010) 43–58.
[161] Y. Qu, C. Shang, W. Wu, Q. Shen, Evolutionary fuzzy extreme learning machine for mammographic risk analysis, Int. J. Fuzzy Syst. 13 (4) (2011) 282–291.
[162] S.Y. Wong, K.S. Yap, H.J. Yap, S.C. Tan, S.W. Chang, On equivalence of FIS and ELM for interpretable rule-based knowledge representation, IEEE Trans. Neural Networks Learn. Syst. 26 (7) (2015) 1417–1430.
[163] S. Thomas, G.N. Pillai, K. Pal, P. Jagtap, Prediction of ground motion parameters using randomized ANFIS (RANFIS), Appl. Soft Comput. 40 (2016) 624–634.
[164] S. KV, G.N. Pillai, Regularized extreme learning adaptive neuro-fuzzy algorithm for regression and classification, Knowledge-Based Syst. 127 (2017) 100–113.
[165] F.M. Pouzols, A. Lendasse, Evolving fuzzy optimally pruned extreme learning machine: comparative analysis, in: IEEE International Conference on Fuzzy Systems (FUZZ), 2010, pp. 1–8.
[166] W.S. Liew, C.K. Loo, T. Obo, Genetic optimized fuzzy extreme learning machine ensembles for affect classification, in: 17th International Symposium on Advanced Intelligent Systems, 2016, pp. 305–310.
[167] K.V Shihabudheen, G.N. Pillai, Evolutionary fuzzy extreme learning machine for inverse kinematic modeling of robotic arms, in: 9th National Systems Conference (NSC), 2015, pp. 1–6.
[168] O. Yazdanbakhsh, S. Dick, ANCFIS-ELM: a machine learning algorithm based on complex fuzzy sets, in: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2016, pp. 2007–2013.
[169] K.V Shihabudheen, M. Mahesh, G.N. Pillai, Particle swarm optimization based extreme learning neuro-fuzzy system for regression and classification, Expert Syst. Appl. 92 (2018) 474–484.
[170] K.B. Nahato, K.H. Nehemiah, A. Kannan, Hybrid approach using fuzzy sets and extreme learning machine for classifying clinical datasets, Inf. Med. Unlocked 2 (2016) 1–11.
[171] A. Sungheetha, R. Rajesh Sharma, Extreme learning machine and fuzzy K-nearest neighbour based hybrid gene selection technique for cancer classification, J. Med. Imag. Health Inf. 6 (7) (2016) 1652–1656.
[172] G.N. Pillai, J. Pushpak, M.G. Nisha, Extreme learning ANFIS for control applications, in: IEEE Symposium on Computational Intelligence in Control and Automation (CICA), 2015, pp. 1–8.
[173] K.V Shihabudheen, G.N. Pillai, B. Peethambaran, Prediction of landslide displacement with controlling factors using extreme learning adaptive neuro–fuzzy inference system (ELANFIS), Appl. Soft Comput. 61 (2017) 892–904.
[174] K.V Shihabudheen, B. Peethambaran, Landslide displacement prediction technique using improved neuro-fuzzy system, Arabian J. Geosci. 10 (2017) 502.
[175] S.K. Meher, Efficient pattern classification model with neuro-fuzzy networks, Soft Comput. 21 (12) (2017) 3317–3334.
[176] T. Goel, V. Nehra, V.P. Vishwakarma, An adaptive non-symmetric fuzzy activation function-based extreme learning machines for face recognition, Arabian J. Sci. Eng. 42 (2) (2017) 805–816.
[177] H. Wang, L. Li, W. Fan, Time series prediction based on ensemble fuzzy extreme learning machine, in: Proceedings of the IEEE International Conference on Information and Automation, 2016, pp. 2001–2005.
[178] S.O. Olatunji, A. Selamat, A. Abdulraheem, A hybrid model through the fusion of type-2 fuzzy logic systems and extreme learning machines for modelling permeability prediction, Inf. Fusion 16 (1) (2014) 29–45.
[179] S. Hassan, A. Khosravi, J. Jaafar, M.A. Khanesar, A systematic design of interval type-2 fuzzy logic system using extreme learning machine for electricity load demand forecasting, Int. J. Electr. Power Energy Syst. 82 (2016) 1–10.
[180] Z. Yong, E.M. Joo, S. Sundaram, Meta-cognitive fuzzy extreme learning machine, in: 13th International Conference on Control Automation Robotics and Vision, ICARCV 2014, 2014, pp. 613–618.
[181] D. Wu, On the fundamental differences between type-1 and interval type-2 fuzzy logic controllers, IEEE Trans. Fuzzy Syst. 20 (5) (2012) 832–848.
[182] J.M. Mendel, Type-2 fuzzy sets and systems: an overview, IEEE Comput. Intell. Mag. 2 (1) (2007) 20–29.
[183] R. Scherer, J.T. Starczewski, Relational type-2 interval fuzzy systems, in: R. Wyrzykowski, J. Dongarra, K. Karczewski, J. Wasniewski (Eds.), Parallel Processing and Applied Mathematics. PPAM 2009, Springer, Berlin, Heidelberg, 2010, pp. 360–368.
[184] J. Starczewski, R. Scherer, M. Korytkowski, R. Nowicki, Modular type-2 neuro–fuzzy systems, in: R. Wyrzykowski, J. Dongarra, K. Karczewski, J. Wasniewski (Eds.), Parallel Processing and Applied Mathematics. PPAM 2007, Springer, Berlin, Heidelberg, 2007, pp. 570–578.
[185] J.M. Mendel, R.I. John, F. Liu, Interval type-2 fuzzy logic systems made simple, IEEE Trans. Fuzzy Syst. 14 (6) (2006) 808–821.
[186] J.M. Mendel, Computing derivatives in interval type-2 fuzzy logic systems, IEEE Trans. Fuzzy Syst. 12 (1) (2004) 84–98.
[187] Q. Liang, J.M. Mendel, Interval type-2 fuzzy logic systems: theory and design, IEEE Trans. Fuzzy Syst. 8 (5) (2000) 535–550.
[188] J.M. Mendel, F. Liu, Super-exponential convergence of the Karnik–Mendel algorithms for computing the centroid of an interval type-2 fuzzy set, IEEE Trans. Fuzzy Syst. 15 (2) (2007) 309–320.
[189] D. Wu, J.M. Mendel, Enhanced Karnik–Mendel algorithms, IEEE Trans. Fuzzy Syst. 17 (4) (2009) 923–934.
[190] S. Hassan, M.A. Khanesar, E. Kayacan, J. Jaafar, A. Khosravi, Optimal design of adaptive type-2 neuro-fuzzy systems: a review, Appl. Soft Comput. J. 44 (2016) 134–143.
[191] C.-H. Wang, C.-S. Cheng, T.-T. Lee, Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN), IEEE Trans. Syst. Man Cybern. Part B: Cybern. 36 (3) (2004) 1462–1477.
[192] G.M. Mendez, O. Castillo, Interval type-2 TSK fuzzy logic systems using hybrid learning algorithm, in: The 14th IEEE International Conference on Fuzzy Systems, FUZZ'05, 2005, pp. 230–235.
[193] C. Juang, S. Member, Y. Tsao, A self-evolving interval type-2 fuzzy neural network with online structure, IEEE Trans. Fuzzy Syst. 16 (6) (2008) 1411–1424.
[194] S. Panahian Fard, Z. Zainuddin, Interval type-2 fuzzy neural networks version of the Stone-Weierstrass theorem, Neurocomputing 74 (14–15) (2011) 2336–2343.
[195] R.A. Aliev, et al., Type-2 fuzzy neural networks with fuzzy clustering and differential evolution optimization, Inf. Sci. 181 (9) (2011) 1591–1608.
[196] C. Li, J. Yi, M. Wang, G. Zhang, Monotonic type-2 fuzzy neural network and its application to thermal comfort prediction, Neural Comput. Appl. 23 (7–8) (2013) 1987–1998.
[197] S.F. Tolue, M.-R. Akbarzadeh-T., Dynamic fuzzy learning rate in a self-evolving interval type-2 tsk fuzzy neural network, in: 13th Iranian Conference on Fuzzy Systems (IFSC), 2013, pp. 1–6.
[198] G.D. Wu, P.H. Huang, A vectorization-optimization-method-based type-2 fuzzy neural network for noisy data classification, IEEE Trans. Fuzzy Syst. 21 (1) (2013) 1–15.
[199] Y.Y. Lin, J.Y. Chang, C.T. Lin, A TSK-type-based self-evolving compensatory interval type-2 fuzzy neural network (TSCIT2FNN) and its applications, IEEE Trans. Ind. Electron. 61 (1) (2014) 447–459.
[200] Y.-Y. Lin, S.-H. Liao, J.-Y. Chang, C.-T. Lin, Simplified interval type-2 fuzzy neural networks, IEEE Trans. Neural Networks Learn. Syst. 25 (5) (2014) 959–969.
[201] C.F. Juang, C.Y. Chen, Data-driven interval type-2 neural fuzzy system with high learning accuracy and improved model interpretability, IEEE Trans. Cybern. 43 (6) (2013) 1781–1795.
[202] S.W. Tung, C. Quek, C. Guan, ET2FIS: an evolving type-2 neural fuzzy inference system, Inf. Sci. 220 (2013) 124–148.
[203] C.T. Lin, N.R. Pal, S.L. Wu, Y.T. Liu, Y.Y. Lin, An interval type-2 neural fuzzy system for online system identification and feature elimination, IEEE Trans. Neural Networks Learn. Syst. 26 (7) (2015) 1442–1455.
[204] K. Subramanian, A.K. Das, S. Sundaram, S. Ramasamy, A meta-cognitive interval type-2 fuzzy inference system and its projection based learning algorithm, Evolving Syst. 5 (4) (2014) 219–230.
[205] A.K. Das, S. Sundaram, S. Member, N. Sundararajan, L. Fellow, A Self-Regulated interval type-2 neuro-fuzzy inference system for handling nonstationarities in EEG signals for BCI, IEEE Trans. Fuzzy Syst. 24 (6) (2016) 1565–1577.
[206] V. Sumati, P. Chellapilla, S. Paul, L. Singh, Parallel interval type-2 subsethood neural fuzzy inference system, Expert Syst. Appl. 60 (2016) 156–168.
[207] C.F. Juang, R.B. Huang, Y.Y. Lin, A recurrent self-evolving interval type-2 fuzzy neural network for dynamic system processing, IEEE Trans. Fuzzy Syst. 17 (5) (2009) 1092–1105.
[208] Y.Y. Lin, J.Y. Chang, N.R. Pal, C.T. Lin, A mutually recurrent interval type-2 neural fuzzy system (MRIT2NFS) with self-evolving structure and parameters, IEEE Trans. Fuzzy Syst. 21 (3) (2013) 492–509.
[209] M. Pratama, E. Lughofer, M.J. Er, W. Rahayu, T. Dillon, Evolving type-2 recurrent fuzzy neural network, in: International Joint Conference on Neural Networks (IJCNN), 2016, pp. 1841–1848.
[210] M. Pratama, J. Lu, E. Lughofer, G. Zhang, M.J. Er, Incremental learning of concept drift using evolving type-2 recurrent fuzzy neural network, IEEE Trans. Fuzzy Syst. 25 (5) (2017) 1175–1192.
[211] A. Mohammadzadeh, S. Ghaemi, Synchronization of chaotic systems and identification of nonlinear systems by using recurrent hierarchical type-2 fuzzy neural networks, ISA Trans. 58 (2015) 318–329.
[212] C.F. Juang, R.B. Huang, W.Y. Cheng, An interval type-2 Fuzzy-neural network with support-vector regression for noisy regression problems, IEEE Trans. Fuzzy Syst. 18 (4) (2010) 686–699.
[213] C.F. Juang, P.H. Wang, An interval type-2 neural fuzzy classifier learned through soft margin minimization and its human posture classification application, IEEE Trans. Fuzzy Syst. 23 (5) (2015) 1474–1487.
[214] Z. Deng, K.S. Choi, L. Cao, S. Wang, T2FEA: type-2 fuzzy extreme learning algorithm for fast training of interval type-2 TSK fuzzy logic system, IEEE Trans. Neural Networks Learn. Syst. 25 (4) (2014) 664–676.
[215] M. Pratama, G. Zhang, M.J. Er, S. Anavatti, An incremental type-2 meta-cognitive extreme learning machine, IEEE Trans. Cybern. 47 (2) (2017) 339–353.
[216] M. Pratama, J. Lu, G. Zhang, A novel meta-cognitive extreme learning machine to learning from large data streams an incremental interval type-2 neural fuzzy classifier, in: IEEE International Conference on Systems, Man, and Cybernetics, 2015, pp. 2792–2797.
[217] S. Hassan, M.A. Khanesar, J. Jaafar, A. Khosravi, A multi-objective genetic type-2 fuzzy extreme learning system for the identification of nonlinear, in: IEEE International Conference on Systems, Man, and Cybernetics (SMC 2016), 2016, pp. 155–160.
[218] S. Hassan, J. Jaafar, M.A. Khanesar, A. Khosravi, Artificial bee colony optimization of interval type-2 fuzzy extreme learning system for chaotic data, in: 3rd International Conference on Computer and Information Sciences (ICCOINS) Artificial, 2016, pp. 334–339.

[219] F. Lin, P. Chou, P. Shieh, S. Chen, Robust control of an LUSM-based X–Y–θ motion control stage using an adaptive interval type-2 fuzzy neural network, IEEE Trans. Fuzzy Syst. 17 (1) (2009) 24–38.
[220] F. Lin, S. Member, P. Chou, Adaptive control of two-axis motion control system using interval type-2 fuzzy neural network, IEEE Trans. Ind. Electron. 56 (1) (2009) 178–193.
[221] E. Kayacan, O. Cigdem, O. Kaynak, Sliding mode control approach for online learning as applied to type-2 fuzzy neural networks and its experimental evaluation, IEEE Trans. Ind. Electron. 59 (9) (2012) 3510–3520.
[222] Y.H. Chang, W.S. Chan, Adaptive dynamic surface control for uncertain nonlinear systems with interval type-2 fuzzy neural networks, IEEE Trans. Cybern. 44 (2) (2014) 293–304.
[223] A. Mohammadzadeh, O. Kaynak, M. Teshnehlab, Two-mode indirect adaptive control approach for the synchronization of uncertain chaotic systems by the use of a hierarchical interval type-2 fuzzy neural network, IEEE Trans. Fuzzy Syst. 22 (5) (2014) 1301–1312.
[224] F. Liu, An efficient centroid type-reduction strategy for general type-2 fuzzy logic system, Inf. Sci. 178 (2008) 2224–2236.
[225] D. Zhai, J.M. Mendel, Computing the centroid of a general type-2 fuzzy set by means of the centroid-flow algorithm, IEEE Trans. Fuzzy Syst. 19 (3) (2011) 401–422.
[226] B. Xie, S. Lee, An extended type-reduction method for general type-2 fuzzy sets, IEEE Trans. Fuzzy Syst. 25 (3) (2017) 715–724.
[227] W.-H.R. Jeng, C.-Y. Yeh, S.-J. Lee, General type-2 fuzzy neural network with hybrid learning for function approximation, in: IEEE International Conference on Fuzzy Systems, 2009, pp. 1534–1539.
[228] C.Y. Yeh, W.R. Jeng, S.J. Lee, Data-based system modeling using a type-2 fuzzy neural network with a hybrid learning algorithm, IEEE Trans. Neural Networks 22 (12) (2011) 2296–2309.
[229] L. Rutkowski, K. Cpalka, Flexible neuro-fuzzy systems, IEEE Trans. Neural Networks 14 (3) (2003) 554–574.
[230] L. Rutkowski, Flexible neuro-fuzzy systems, in: Computational Intelligence, Springer, Berlin, Heidelberg, 2008, pp. 449–493.
[231] L. Rutkowski, Flexible Neuro-Fuzzy Systems Structures, Learning and Performance Evaluation, Kluwer, Boston, MA, 2004.
[232] K. Cpalka, A method for designing flexible neuro-fuzzy systems, in: L. Rutkowski, R. Tadeusiewicz, L.A. Zadeh, J. Żurada (Eds.), Artificial Intelligence and Soft Computing – ICAISC 2006, Springer, Berlin, Heidelberg, 2006, pp. 212–219.
[233] K. Cpałka, A new method for design and reduction of neuro-fuzzy classification systems, IEEE Trans. Neural Networks 20 (4) (2009) 701–714.
[234] K. Cpalka, L. Rutkowski, Evolutionary learning of flexible neuro-fuzzy systems, in: IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence), 2008, pp. 969–975.
[235] K. Cpałka, On evolutionary designing and learning of flexible neuro–fuzzy structures for nonlinear classification, Nonlinear Anal. 71 (12) (2009) e1659–e1672.
[236] K. Cpałka, O. Rebrova, R. Nowicki, L. Rutkowski, On design of flexible neuro-fuzzy systems for nonlinear modelling, Int. J. Gen. Syst. 42 (6) (2013) 706–720.
[237] K. Cpałka, O. Rebrova, R. Nowicki, L. Rutkowski, On design of flexible neuro-fuzzy systems for nonlinear modelling, Int. J. Gen. Syst. 42 (6) (2013) 706–720.
[238] L. Rutkowski, K. Cpałka, Designing and learning of adjustable quasi-triangular norms with applications to neuro-fuzzy systems, IEEE Trans. Fuzzy Syst. 13 (1) (2005) 140–151.
[239] R. Nowicki, Rough neuro-fuzzy structures for classification with missing data, IEEE Trans. Syst. Man Cybern., Part B: Cybern. 39 (6) (2009) 1334–1347.
[240] K. Siminski, Rough subspace neuro-fuzzy system, Fuzzy Sets Syst. 269 (2015) 30–46.
[241] M. Korytkowski, R. Nowicki, L. Rutkowski, R. Scherer, AdaBoost ensemble of DCOG rough–neuro–fuzzy systems, in: P. Jędrzejowicz, N.T. Nguyen, K. Hoang (Eds.), Computational Collective Intelligence. Technologies and Applications, Springer, Berlin, Heidelberg, 2011, pp. 62–71.
[242] M. Korytkowski, R. Nowicki, R. Scherer, Neuro-fuzzy rough classifier ensemble, in: C. Lippi, M. Polycarpou, C. Panayiotou, G. Ellinas (Eds.), Artificial Neural Networks – ICANN 2009, Springer, Berlin, Heidelberg, 2009, pp. 817–823.
[243] R.K. Nowicki, M. Korytkowski, B.A. Nowak, R. Scherer, Design methodology for rough-neuro-fuzzy classification with missing data, in: IEEE Symposium Series on Computational Intelligence, 2015, pp. 1650–1657.
[244] J.R. Castro, O. Castillo, P. Melin, A. Rodríguez-Díaz, A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks, Inf. Sci. 179 (13) (2009) 2175–2193.
[245] A.K. Das, K. Subramanian, S. Suresh, A evolving interval type-2 neuro-fuzzy inference system and its meta-cognitive sequential learning algorithm, IEEE Trans. Fuzzy Syst. 23 (6) (2015) 2080–2093.
[246] C. Juang, C. Lin, An on-line self-constructing neural fuzzy inference network and its applications, IEEE Trans. Fuzzy Syst. 6 (1) (1998) 12–32.
[247] J.S. Wang, C.S.G. Lee, Self-adaptive neuro-fuzzy inference systems for classification applications, IEEE Trans. Fuzzy Syst. 10 (6) (2002) 790–802.
[248] S. Wu, M.J. Er, Dynamic fuzzy neural networks—a novel approach to function approximation, IEEE Trans. Syst. Man Cybern.—Part B: Cybern. 30 (2) (2000) 358–364.
[249] G. Leng, T.M. Mcginnity, G. Prasad, An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network, Fuzzy Sets Syst. 150 (2005) 211–243.
[250] N.K. Kasabov, Q. Song, DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction, IEEE Trans. Fuzzy Syst. 10 (2) (2002) 144–154.
[251] H.-G. Han, L.-M. Ge, J.-F. Qiao, An adaptive second order fuzzy neural network for nonlinear system modeling, Neurocomputing 214 (2016) 1–11.
[252] S. Lee, C. Ouyang, A neuro-fuzzy system modeling with self-constructing rule generation and hybrid SVD-based learning, IEEE Trans. Fuzzy Syst. 11 (3) (2003) 341–353.
[253] H. Rong, N. Sundararajan, G. Huang, P. Saratchandran, Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction, Fuzzy Sets Syst. 157 (2006) 1260–1275.
[254] M. Hamzehnejad, K. Salahshoor, A new adaptive neuro-fuzzy approach for on-line nonlinear system identification, in: IEEE Conference on Cybernetics and Intelligent Systems, 2008, pp. 142–146.
[255] H.R.N.S.G. Huang, Extended sequential adaptive fuzzy inference system for classification problems, Evolving Syst. 2 (2) (2011) 71–82.
[256] H. Li, Z. Liu, A probabilistic neural-fuzzy learning system for stochastic modeling, IEEE Trans. Fuzzy Syst. 16 (4) (2008) 898–908.
[257] Z. Liu, H.X. Li, A probabilistic fuzzy logic system for modeling and control, IEEE Trans. Fuzzy Syst. 13 (6) (2005) 848–859.
[258] N. Wang, M. Joo, X. Meng, A fast and accurate online self-organizing scheme for parsimonious fuzzy neural networks, Neurocomputing 72 (16–18) (2009) 3818–3829.
[259] L. Network, SOFMLS: online self-organizing fuzzy modified, IEEE Trans. Fuzzy Syst. 17 (6) (2009) 1296–1309.
[260] K. Subramanian, S. Suresh, A meta-cognitive sequential learning algorithm for neuro-fuzzy inference system, Appl. Soft Comput. 12 (2012) 3603–3614.
[261] K. Subramanian, S. Member, S. Suresh, S. Member, A metacognitive neuro–fuzzy inference system (McFIS) for sequential classification problems, IEEE Trans. Fuzzy Syst. 21 (6) (2013) 1080–1095.
[262] M. Pratama, J. Lu, S. Anavatti, E. Lughofer, C.P. Lim, An incremental meta-cognitive-based scaffolding fuzzy neural network, Neurocomputing 171 (2016) 89–105.
[263] J. Zhang, A.J. Morris, Recurrent neuro-fuzzy networks for nonlinear process modeling, IEEE Trans. Neural Networks 10 (2) (1999) 313–326.
[264] Y.Y. Lin, J.Y. Chang, C.T. Lin, Identification and prediction of dynamic systems using an interactively recurrent self-evolving fuzzy neural network, IEEE Trans. Neural Networks Learn. Syst. 24 (2) (2013) 310–321.
[265] F.J. Lin, I.F. Sun, K.J. Yang, J.K. Chang, Recurrent fuzzy neural cerebellar model articulation network fault-tolerant control of six-phase permanent magnet synchronous motor position servo drive, IEEE Trans. Fuzzy Syst. 24 (1) (2016) 153–167.
[266] C. Juang, Y. Lin, C. Tu, A recurrent self-evolving fuzzy neural network with local feedbacks and its application to dynamic system processing, Fuzzy Sets Syst. 161 (19) (2010) 2552–2568.
[267] M. Pratama, S.G. Anavatti, P.P. Angelov, E. Lughofer, PANFIS: a novel incremental learning machine, IEEE Trans. Neural Networks Learn. Syst. 25 (1) (2014) 55–68.
[268] M. Pratama, S.G. Anavatti, E. Lughofer, Genefis: toward an effective localist network, IEEE Trans. Fuzzy Syst. 22 (3) (2014) 547–562.
[269] A.M. Silva, W. Caminhas, A. Lemos, F. Gomide, A fast learning algorithm for evolving neo-fuzzy neuron, Appl. Soft Comput. J. 14 (PART B) (2014) 194–209.
[270] T. Yamakawa, high speed fuzzy learning machine with guarantee of global minimum and its application to chaotic system identification and medical image processing, Int. J. Artif. Intell. Tools 5 (1) (1996) 23–39.
[271] M. Lee, S. Lee, C.H. Park, A new neuro-fuzzy identification model of nonlinear dynamic systems, Int. J. Appr. Reason. 10 (1) (1994) 29–44.
[272] A.M. Silva, W.M. Caminhas, A.P. Lemos, F. Gomide, Evolving neo-fuzzy neural network with adaptive feature selection, in: 2013 BRICS Congress on Computational Intelligence & 11th Brazilian Congress on Computational Intelligence, 2013, pp. 341–349.
[273] S. Yilmaz, Y. Oysal, Fuzzy wavelet neural network models for prediction and identification of dynamical systems, IEEE Trans. Neural Networks 21 (10) (2010) 1599–1609.
[274] Y. Bodyanskiy, A. Dolotov, O. Vynokurova, Evolving spiking wavelet-neuro–fuzzy self-learning system, Appl. Soft Comput. Journal 14 (PART B) (2014) 252–258.
[275] S.M. Bohte, H. La Poutré, J.N. Kok, Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer RBF networks, IEEE Trans. Neural Networks 13 (2) (2002) 426–435.
[276] J.P.S. Catalaõ, H.M.I. Pousinho, V.M.F. Mendes, Hybrid intelligent approach for short-term wind power forecasting in Portugal, IET Renew. Power Gener. 5 (3) (2011) 251–257.
[277] M. Mehdi, A. Salimi-badr, CFNN: correlated fuzzy neural network, Neurocomputing 148 (2015) 430–444.
[278] M. Prasad, Y.Y. Lin, C.T. Lin, M.J. Er, O.K. Prasad, A new data-driven neural fuzzy system with collaborative fuzzy clustering mechanism, Neurocomputing 167 (2015) 558–568.
[279] G. Leng, G. Prasad, T.M. Mcginnity, An on-line algorithm for creating self-organizing fuzzy neural networks, Neural Networks 17 (2004) 1477–1493.
[280] B. Pizzileo, K. Li, S. Member, G.W. Irwin, W. Zhao, Improved structure optimization for fuzzy-neural networks, IEEE Trans. Fuzzy Syst. 20 (6) (2012) 1076–1089.
[281] H. Han, X.L. Wu, J.F. Qiao, Nonlinear systems modeling based on self-organizing fuzzy-neural-network with adaptive computation algorithm, IEEE Trans. Cybern. 44 (4) (2014) 554–564.

[282] H.A. Cid, R. Salas, A. Veloz, C. Moraga, H. Allende, SONFIS: structure identification and modeling with a self-organizing neuro-fuzzy inference system, Int. J. Comput. Intell. Syst. 9 (3) (2016) 416–432.
[283] A.H. Sonbol, M. Sami Fadali, S. Jafarzadeh, TSK fuzzy function approximators: Design and accuracy analysis, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 42 (3) (2012) 702–712.
[284] H. Takagi, I. Hayashi, NN-driven fuzzy reasoning, Int. J. Approximate Reason. 5 (1991) 191–212.
[285] T. Similä, J. Tikka, Multiresponse sparse regression with application to multidimensional scaling, in: 15th International Conference on Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005, 2005, pp. 97–102.
[286] G. Bontempi, M. Birattari, H. Bersini, Recursive lazy learning for modeling and control, in: Proceedings of 10th European Conference on Machine Learning, 1998, pp. 292–303.
[287] P. Hokstad, A method for diagnostic checking of time series models, J. Time Ser. Anal. 4 (3) (1983) 177–183.
[288] C.L. Blake, C.J. Merz, UCI Repository of Machine Learning Databases, Dept. Inf. Comput. Sci., Univ. California, Irvine, CA, 1998 [Online]. Available: http://archive.ics.uci.edu/ml/datasets.html.
[289] J. Casilias, O. Cordon, F. Herrera, L. Magdalena, Interpretability improvements to find the balance interpretability-accuracy in fuzzy modeling: an overview, in: Interpretability Issues in Fuzzy Modeling, Springer, Berlin, Heidelberg, 2003, pp. 3–22.
[290] S.Y. Wong, K.S. Yap, H.J. Yap, S.C. Tan, S.W. Chang, On equivalence of FIS and ELM for interpretable rule-based knowledge representation, IEEE Trans. Neural Networks Learn. Syst. 26 (7) (2015) 1417–1430.
[291] K. Cplka, K. Lapa, A. Przybyl, M. Zalasinski, A new method for designing neuro-fuzzy systems for nonlinear modelling with interpretability aspects, Neurocomputing 135 (2014) 203–217.
[292] K. Cpałka, Design of Interpretable Fuzzy Systems, 1st ed., Springer International Publishing, 2017.
[293] X. Huang, D. Ho, J. Ren, L.F. Capretz, Improving the COCOMO model using a neuro-fuzzy approach, Appl. Soft Comput. 7 (2007) 29–40.
[294] A. Azar, Modeling and Control of Dialysis Systems: Biofeedback Systems and Soft Computing Techniques of Dialysis, vol. 2, Springer-Verlag, Berlin Heidelberg, 2013.