

Computer Aided Development of Fuzzy, Neural and Neuro-Fuzzy Systems

Priti Srinivas Sajja

Department of Computer Science and Technology

Sardar Patel University

Vallabh Vidyanagar, India

Abstract— Development of an expert system is difficult because of two challenges involve in it. The first one is the expert system itself is high level system and deals with knowledge, which make is difficult to handle. Second, the systems development is more art and less science; hence there are little guidelines available about the development. This paper describes computer aided development of intelligent systems using modem artificial intelligence technology. The paper illustrates a design of a reusable generic framework to support friendly development of fuzzy, neural network and hybrid systems such as neuro-fuzzy system. The reusable component libraries for fuzzy logic based systems, neural network based system and hybrid system such as neuro-fuzzy system are developed and accommodated in this framework. The paper demonstrates code snippets, interface screens and class libraries overview with necessary technical details.

Keywords: *Fuzzy logic, Neural network, Neuro-fuzzy systems, Soft computing, Automatic development.*

I. INTRODUCTION

Advancement of science and technology has been increasingly utilized as a major tool for uplift of mankind. Innovations of modern information and communication technologies (ICT) made human life smoother and problem solving has become easier. It is observed that usage of modern ICT (like Internet) has become ambient and ubiquitous. In spite of availability of lots of tools and technologies, expectations from mighty machines are continuously increasing and demand for more and more human like intelligent systems in various fields has evolved.

Development of an intelligent system is a challenging job due to several reasons. The prime among them are abstract nature of knowledge, volume of knowledge, lack of knowledge acquisition and representation techniques and lack of models/quality standards for the development of an intelligent system. Typical Artificial Intelligence (AI) techniques facilitate development of intelligent system with the aforementioned problems. However, they lack self learning, human like interaction, processing and require lot of efforts as well as cost. There are some new AI techniques such as bio-inspired techniques that offer some advantages above the typical AI techniques. These techniques include artificial neural network,

swarm intelligence, fuzzy logic etc. and their hybridization (such as neuro-fuzzy approach). These techniques are sometimes also known as soft computing techniques. Many tools are available to develop such soft AI based intelligent systems. However, these tools are costly, less user friendly and application specific. Further, these tools need training and practice to certify their efficient utilization. There is a requirement of a generic tool which interacts through native/natural language of non-computer professionals/users, reduces effort of development of an intelligent system and saves time of development. No such generic tool is available as per the literature found at present. This paper presents a design and implementation of generic tool that facilitates automatic development of soft computing intelligent systems such as neural network based systems, fuzzy logic based systems and hybrid neuro-fuzzy systems in a given domain.

The paper organization is as follows. Section 2 of the paper presents national and international scenario related to the aforementioned problem and discusses limitations of the existing solutions. The aims and objectives of the proposed research work are enlisted in this section.

Section 3 describes architecture of the proposed framework which can be utilized as generic and user friendly tool that facilitates automatic development of soft computing intelligent systems. An important component of the framework is library of reusable components to develop intelligent systems. Design and implementation issues of the centralized library along with other necessary components are described here. At the end, the paper concludes with advantages and application of the framework generated in different other areas. The section also presents future scope of the research work.

II. LITERATURE SURVEY

To develop soft computing system techniques such as artificial neural network (ANN), type 1 and type 2 fuzzy logic (T1FL and T2FL) and hybrid neuro-fuzzy systems are vastly used. This section presents in brief about each techniques and work done in different areas using the respective techniques.

A. Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) is an interconnected group of artificial neurons that uses a computational model for processing information based on a connectionist approach to computation to impart self learning and other characteristics associated with intelligence. ANN is powerful tool for modeling intelligent systems when self learning is most desired feature and generalized knowledge is not available or can not be documented easily. ANN has ability to identify and learn the correlation between the input patterns, data or cases provided to it. Due to this feature of ANN, it is increasingly being used in intelligent systems where one has little or incomplete understanding of the problem under research but experience or training data of domain expert is readily available. The popular models for implementing ANN are Hopfield model, multilayer perceptron with supervised learning, Kohonen model (self organizing map), and recurrent network. Programming environment like java, c# and other third generation programming languages help in implementing ANN based systems. There are some packages like matlab (www.mathworks.com/products/matlab) are also available which facilitates development of such system.

B. Fuzzy Logic

Fuzzy logic is a multi-valued logic conceived by Zadeh [1] and used to achieve human like loose categorization of objects into classes without boundaries. The fuzzy logic is based on fuzzy sets. Every given object has the partial (fuzzy) belongingness of the concerned fuzzy sets, which is determined by its membership function. For example, if the temperature is 18 degree centigrade, then it belongs fully (say 1) to the set of comfortable temperature. If the temperature is 12 degree centigrade, then it belongs partially (say 0.4) to the set of comfortable temperature. Young lady, rich person, luxury car and comfortable temperature are a few examples of linguistic vague words which can be interpreted by machine with the help of fuzzy membership functions, hence can be easily used in 'if...then...else' rules. System with mainly such fuzzy 'if...then...else' rules are capable of representing knowledge and are known as fuzzy rule based system. Mamdani [2] and Takagi & Sugeno [3] are vastly utilized basic modeling techniques for fuzzy logic based systems. Further, Jerry Mendel has proposed an idea of type 2 fuzzy logic which was an extension of the original fuzzy logic proposed by Prof. Zadeh [1]. The basis of type 2 fuzzy logic is type 2 fuzzy sets. Type 2 fuzzy sets incorporate uncertainty as extra third dimension which gives much clear and logical information about the problem under research. Further, type 2 fuzzy systems need to accompany with type reducer component that converts type 2 membership functions into simple fuzzy membership functions. The popular tools to develop a fuzzy logic based system are third generation programming languages such as java and c# of .Net framework (www.microsoft.com/net) and packages like Matlab (www.mathworks.in). Many specific packages also have been developed such as Fuzzy Attitude (www.fuzzytech.com),

JFuzzyLogic (sourceforge.net/projects/javafuzzyyed/), Fuzzy Editor (<http://jeux.windows.simplenet.com/>), etc.

C. Hybrid Systems

Every technique has its advantages and limitations. The idea behind hybridization of two or more techniques is to get advantages of all candidate technique in a common application. For example, consider hybridization of ANN and FL. It is observed that both ANN and FL have their own pros and cons. ANN based systems are good where there is availability of data but lack of generalized knowledge behind it. ANN systems are good in self learning however lacks in documentation of knowledge. This is the prime reason why such system cannot provide reasoning and detail explanation of decision made. Fuzzy logic is good in handling uncertainty and handling natural linguistic values, but lacks self learning and enforces documentation of knowledge in generalized form. Hybridizing these two technologies provide dual advantages of FL and ANN both in one common application. Specifically Neuro-Fuzzy hybridization (NF) achieves advantages of self learning, explanation and reasoning and user friendly interface along with the advantages associated with documentation of knowledge. As stated earlier, third generation programming languages and packages like Matlab can be used to develop such hybrid systems.

Fuzzy Adaptive Learning Control Network (FALCON) [4] Adaptive Neuro Fuzzy Inference System (ANFIS) [5], Generalized Approximated Reasoning based Intelligent Control (GARIC) [6], Neuro-Fuzzy Control (NEFCON) [7], Fuzzy Inference and Neural Network in Fuzzy Inference Software (FINEST) [8], Fuzzy Net (FUN) [9], Evolving Fuzzy Neural Network (EFuNN) [10], Self Constructing Neural Fuzzy Inference Network (SONFIN) [11], etc are popular tools that help in development of such hybrid system.

Not only NF system, but also different techniques such as genetic algorithm and fuzzy logic, neuro-fuzzy-genetic and neuro-genetic hybridization are also popular. At present our framework was designed and implemented to support development of FL, ANN and NF type of systems. However, it can be enhanced to encompass other soft computing techniques and tools. Hence, the literature survey is restricted to these approaches only.

The above tools are either costly or application specific such as Cihan H et al. [12]; or not web-enabled. Many of them are not user friendly and to use such tools would be challenge for the non-computer professionals. Tools like Adaptive Network Fuzzy Inference System (ANFIS) and Dynamic Evolving Neuro-Fuzzy Inference System (DENFIS) need platform of MatLab which is costly. Many researchers have experimented development of dedicated applications in the field. Pioneer of them can be given as Ajit Abraham [13] Jang et al. [14], Mendel [15] and Wu & Mendel [16], Emilio Soria-Olivas et al. [17], Ching Long Su et al. [18], John & Coupland [19], Oscar Castillo & Patricia Melin [20]. Much application specific work is done by various researchers including Malkawi & Murad [21], Nie et al. [22], Bouzaidaa et al. [23] and Azriyenni & Mustafa [24].

Considering the aforementioned work; following observations can be made.

- Majority of the existing solutions are application specific;
- The solutions may not be web based;
- The solution, which supports computer aided development may not support modern artificial intelligent techniques;
- The existing solutions may not be flexible and extendible to accommodate users new requirements and other technology in future;
- The exiting solutions/tools may not be reusable; and
- The exiting solutions are generally meant for programmers hence they may not be user-friendly to non-computer professionals; etc.

This leads to a development of generic, web-enabled and user friendly architecture that supports interactive development of all type of soft computing intelligent system. As a prototype, the framework is designed and implemented for automatic development of FL, ANN and NF hybridized systems. However, it is designed in a flexible way to support many more latest technologies. The prime objectives were decided as follows.

- Development of fuzzy logic based editor to facilitate working with linguistic variable and vague input;
- Development of popular fuzzy membership functions with fuzzification and defuzzification techniques;
- Development of type reducer component for converting type 2 fuzzy systems into typical fuzzy systems so as the above fuzzy function can be used;

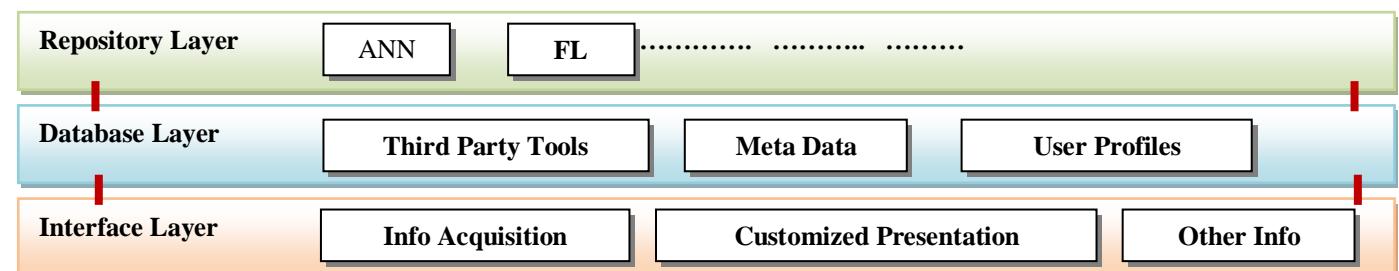


Figure 1. Multi layer framework.

- Development of reusable library containing activation functions and learning algorithms for artificial neural network;
- Mechanism for Neuro-fuzzy hybridization; and
- Development of miscellaneous features such as back up, security feature, web templates and theme library of variety of presentation of generated code, etc;

Section 3 describes the design of the framework.

III. DESIGN OF THE FRAMEWORK

Design of the framework which meets objectives finalized and mentioned in earlier section is presented here. The framework is divided into three layers. In its first layer repositories for reusable codes are stored as generic independent objects. The reusable codes included here are neural network, fuzzy logic and neuro-fuzzy systems. Second layer is a database layer. The database accommodates third party tool, user profiles, meta data repositories and protocols, if any. Third layer is an interface layer. The interface layer accommodates information acquisition and interaction facilities for users of the system. This layer also provides facility of customized representation of the output to the users of the system. The interface layer also accommodates other information such as local databases, log of the system and frequently asked queries. Fig. 1 illustrates these layers in a generic framework.

Using the framework one can generate artificial neural network based systems, fuzzy logic based systems and neuro-fuzzy systems at this stage. The framework is flexible enough to add reusable component of other paradigms such as Genetic Algorithms (GA). In this case, GA based systems can be developed or neuro-genetic systems can be developed using the framework.

A. Generation of ANN

The artificial neural network repository include code segments for feed forward, radial basis function, Kohonen self organized maps, learning vector quantization, recurrent networks, etc. Some of the codes follow existing methodology and some use novel mechanisms such as modified sigmoid and tangent activation functions. When parameters of the ANN such as number of layers and neurons, learning rate, learning algorithm, etc; the ANN is generated. To generate any feed forward, fully connected, back propagation type of multilayer ANN with supervised learning following code snippet is used.

```

public class NeuralNetwork
{
    protected Layer[ ] layers;
    protected int ni;
    protected LearningAlgorithm la;
    public int N_Inputs
    { get { return ni; } }
    public int N_Outputs
    { { return layers[N_Layers - 1].N_Neurons; }
    }

    public int N_Layers
    { get { return layers.Length; } }

    public LearningAlgorithm LearningAlg
    { { return la; }
        set { la = (value != null) ? value : la; } }

    public Layer this[int n]
    { { return layers[n]; } }

    public NeuralNetwork(int inputs, int[] layers_desc,
        ActivationFunction n_act, LearningAlgorithm learn)
    {
        if (layers_desc.Length < 1)
            throw new Exception("PERCEPTRON : cannot build
perceptron, it must have at least 1 layer of neurone");
        if (inputs < 1)
            throw new Exception("PERCEPTRON : cannot
build perceptron, it must have at least 1 input");
        la = learn;
        ni = inputs;
        layers = new Layer[layers_desc.Length];
        layers[0] = new Layer(layers_desc[0], ni);
        for (int i = 1; i < layers_desc.Length; i++)
            layers[i] = new Layer(layers_desc[i], layers_desc[i
- 1], n_act);
    }
}

public void randomizeWeight()
{ foreach (Layer l in layers)
    l.randomizeWeight(); }

public void randomizeThreshold()
{ foreach (Layer l in layers)
    l.randomizeThreshold(); }

public void randomizeAll()
{ foreach (Layer l in layers)
    l.randomizeAll(); }

public void
setActivationFunction(ActivationFunction f)
{ foreach (Layer l in layers)
    l.setActivationFunction(f); }

{ foreach (Layer l in layers)
    l.setRandomizationInterval(min, max); }

public float[] Output(float[] input)
{ (input.Length != ni)
throw new Exception("PERCEPTRON : Wrong input
vector size, unable to compute output value");
float[] result;
result = layers[0].Output(input);
}

```

```

        for (int i = 1; i < N_Layers; i++)
            result = layers[i].Output(result);
        return result;
    }
}

```

However, users are not aware of the background code; they can interact through the framework by a friendly interface as shown in Fig. 2.

B. Generation of Fuzzy Logic

To generate fuzzy logic based system the components such as fuzzification method, defuzzification method, type reduction codes (for type 2 FL systems), etc are developed and kept in the repository layer. Many of these methods are innovative. Fig. 3 provides an overview of the components of the FL repositories.

Codes of each component shown in Fig. 3 are developed and kept ready. To generate fuzzy logic based system, an interface is generated as shown in Fig. 4.

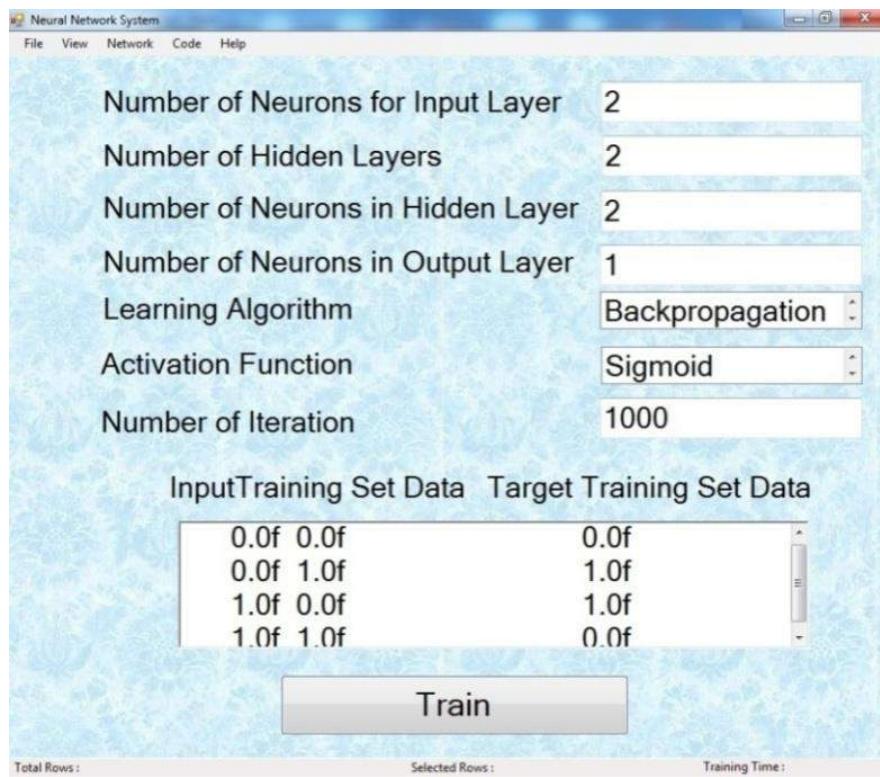


Figure 2. Interface to create ANN.

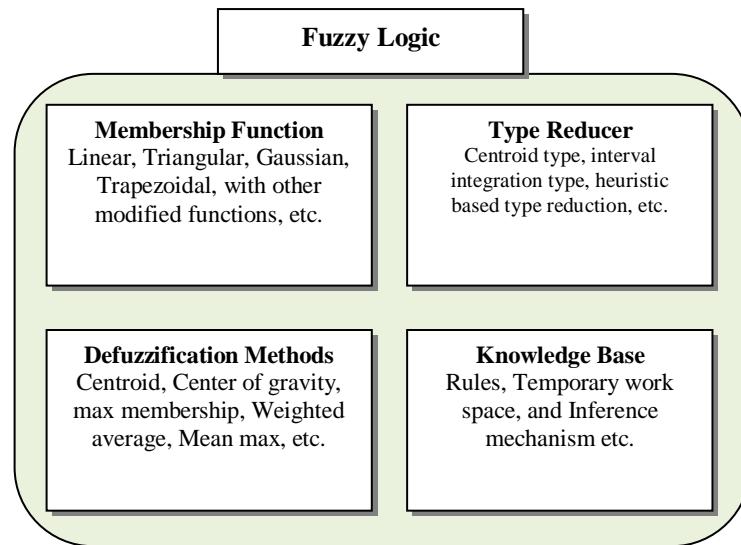


Figure 3. FL repository components.

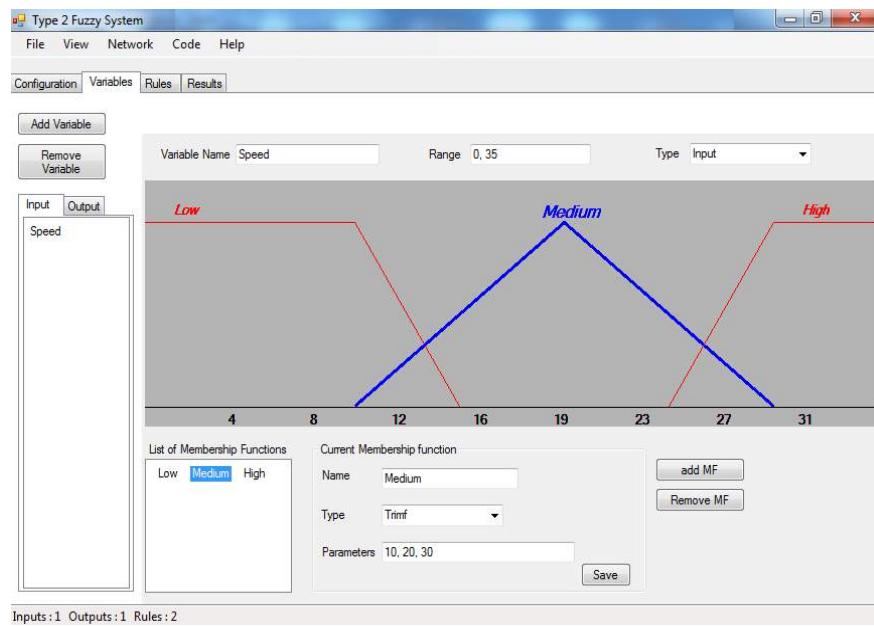


Figure 4. Generation of FL based system.

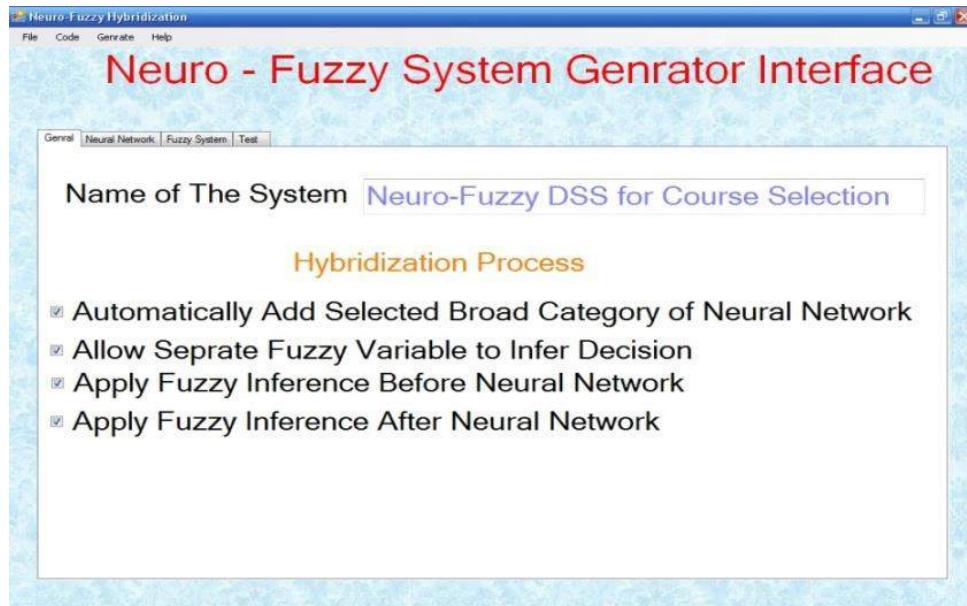


Figure 5. Developing a hybrid neuro-fuzzy systems.

C. Generation of Neuro Fuzzy Component

Since the components of the ANN and FL are ready they can be reused to develop hybrid neuro-fuzzy systems in different way. An interface is developed to select an appropriate type of the hybridization as shown in Fig. 5.

IV. CONCLUSION

The generic neuro-fuzzy framework presented here facilitates automatic development of intelligent system in friendly way. Using the framework many successful systems can be developed. To name a few: course selection and career advice system, student's aptitude evaluation system, neuro-fuzzy recruitment system, and portfolio management system.

It provides advantages such as reusability, modularity, and flexibility. The system developed using this framework can be saved for future use and served as good knowledge management tool. As user need not have to write codes for the components (just on demand attachment of component can be done), user may concentrate on analysis and design of the system in order to increase quality of the system. Obviously, the framework reduces the total man power required to build intelligent system, especially minimizes the role and need of computer professionals. It will also save time and save manual programming work. This generic library and framework are developed using Microsoft's Dot Net technology (Visual Studio 2010) and this can be updated easily to future release of versions. In future more components /codes can be added in to the framework along with visual interface.

REFERENCES

- [1] Zadeh, A.L. (1965). Fuzzy sets. *Journal of Information and Control*, 8(3), 338-353.
- [2] Mamdani, E.H. (1993). Twenty years of fuzzy control: Experiences gained and lessons learnt. In Proceedings of Second IEEE International Conference on Fuzzy Systems, San Francisco, CA, 339-344.
- [3] Takagi, T., & Sugeno, M. (1985). Fuzzy identification on systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1), 116-132.
- [4] Lin, C.T., & Lee, C.S.G. (1991). Neural network based fuzzy logic control and decision system. *IEEE Transactions on Computer*, 40(12), 1320-1336.
- [5] Jang, R. (1992). Neuro-fuzzy modeling: Architectures, analyses and applications. PhD Thesis, University of California, Berkeley.
- [6] Bherenji, H.R., & Khedkar, P. (1992). Learning and tuning fuzzy logic controllers through reinforcements. *IEEE Transactions on Neural Networks*, 3(5), 724-740.
- [7] Nauck, D., & Kruse, R. (1997). Neuro-fuzzy systems for function approximation. In Proceedings of 4th International Workshop on Fuzzy-Neuro Systems, Germany.
- [8] Tano, S., Oyama, T., & Arnould, T. (1996). Deep combination of fuzzy inference and neural network in fuzzy inference. *Fuzzy Sets and Systems*, 82(2), 151-160.
- [9] Sulzberger, S.M., Gurman, N.N., Vestli, S.J. (1993). Fun: Optimization of fuzzy rule based systems using neural networks. In Proceedings of IEEE Conference on Neural Networks, San Francisco, 312-316.
- [10] Kasabov, N., & Qun, S. (1999). Dynamic evolving fuzzy neural networks with 'm-out-of-n' activation nodes for on-line adaptive systems. Technical Report TR99/04, Department of Information Science, University of Otago.
- [11] Feng, J.C., Teng, L.C. (1998). An online self constructing neural fuzzy inference network and its applications. *IEEE Transactions on Fuzzy Systems*, 6(1), 12-32.
- [12] Cihan H. Dagli, Victor Pulla, Xavier Serrano, Modeling of a Neuro Fuzzy System to Develop an Efficient Method to Get a Specific Color Paint from the Color Model Cyan, Magenta and Yellow (CMY) under Terms of Open Source, Procedia Computer Science, Volume 61, 2015, Pages 486-491, ISSN 1877-0509 , Complex Adaptive Systems San Jose, CA November 2-4, 2015, <http://dx.doi.org/10.1016/j.procs.2015.09.196>.
- [13] Abraham, A. (2005). Adaptation of fuzzy inference system using neural learning. *StudFuzz* 181, 53-83, http://www.softcomputing.net/nf_chapter.pdf
- [14] Jang, S.R., Sun, C.T., & Mizutani, E. (1997). Neuro fuzzy and soft computing. Prentice Hall of India Ltd, India.
- [15] Mendel, J.M. (2007). Type-2 fuzzy sets and systems: An Overview. *IEEE Computational Intelligence Magazine*, 2(2), 20-29.
- [16] Wu, H. & Mendel, J.M. (2002). Uncertainty bounds and their use in the design of interval type-2 fuzzy logic system. *IEEE Transactions on Fuzzy Systems*, 10(5), 622-639.
- [17] Olivas, E.S., Guerrero, J.D., Valls, G.C., López, A.J., Maravilla, J.C. & Chova, L.G. (2003). A low-complexity fuzzy activation function for artificial neural networks. *IEEE Transactions on Neural Networks*, 14(6), 1576-1579.
- [18] Su, C.L., Chen, C.J., & Yang, S.M. (2010). A self-organized neuro-fuzzy system for stock market dynamics modeling and forecasting. *WSEAS Transactions on Information Science and Applications*, 7(9), 1137-1149.
- [19] John, R.I., & Coupland, S. (2007). Type-2 fuzzy logic: A historical view. *IEEE Computational Intelligence Magazine*, 2(1), 57-62.
- [20] Castillo, O., & Melin, P. (2008). Type 2 fuzzy logic: Theory and applications. Springer.
- [21] Malkawi, M., & Murad, O. (2013). Artificial neuro fuzzy logic system for detecting human emotions. *Human-Centric Computing and Information Sciences*, 3(3).
- [22] Nie, D., Wang, X., Shi, L., & Lu, B. (2011). EEG-based emotion recognition during watching movies. In Proceedings of International IEEE EMBS Conference on Neural Engineering, Cancun, Mexico, 667-670.
- [23] Bouzaidaa, S., Saklya, A., & Sahlia, F.M. (2014). Extracting TSK-type neuro-fuzzy model using the hunting search algorithm. *International Journal of General Systems*, 43(1), 32-43.
- [24] Azriyenni, & Mustafa, M.W. (2013). Performance neuro-fuzzy for power system fault location. *International Journal of Engineering and Technology*, 3(4), 497-501.

AUTHOR PROFILE

Priti Srinivas Sajja is a Professor at P. G. Department of Computer Science, Sardar Patel University, India since 1994. She specializes in Artificial Intelligence, soft computing and multiagent systems. She is co-author of Intelligent Techniques for Data Science (2016); Intelligent Technologies for Web Applications (2012) and Knowledge-Based Systems (2009) published at Switzerland and USA apart from four books published in India. She is supervising work of a few doctoral research scholars while six candidates have completed their Ph.D. research under her guidance. She was Principal Investigator of a major research project funded by UGC, India. She has 171 publications in books, book chapters, journals, and in the proceedings of national and international conferences out of which five publications have won best research paper awards.



© 2017 by the author(s); licensee Empirical Research Press Ltd. United Kingdom. This is an open access article distributed under the terms and conditions of the Creative Commons by Attribution (CC-BY) license. (<http://creativecommons.org/licenses/by/4.0/>).