Artificial Intelligence in Medicine (2005) **33**, 61—88

ARTIFICIAL
INTELLIGENCE
IN MEDICINE

ELSEVIER

http://www.intl.elsevierhealth.com/journals/aiim

GenSo-FDSS: a neural-fuzzy decision support system for pediatric ALL cancer subtype identification using gene expression data

W.L. Tung, C. Quek*

Centre for Computational Intelligence, School of Computer Engineering, Nanyang Technological University, Blk N4 #2A-32, Nanyang Avenue, Singapore 639798, Singapore

Received 8 April 2003; received in revised form 19 November 2003; accepted 11 March 2004

KEYWORDS
Neural fuzzy system;
Fuzzy decision support system;
GenSoFNN;
Truth-value restriction;
Discrete incremental clustering;
Fuzzy inference and pediatric acute lymphoblastic leukaemia (ALL)

Summary

Objective: Acute lymphoblastic leukemia (ALL) is the most common malignancy of childhood, representing nearly one third of all pediatric cancers. Currently, the treatment of pediatric ALL is centered on tailoring the intensity of the therapy applied to a patient's risk of relapse, which is linked to the type of leukemia the patient has. Hence, accurate and correct diagnosis of the various leukemia subtypes becomes an important first step in the treatment process. Recently, *gene expression* profiling using DNA microarrays has been shown to be a viable and accurate diagnostic tool to identify the known prognostically important ALL subtypes. Thus, there is currently a huge interest in developing autonomous classification systems for cancer diagnosis using gene expression data. This is to achieve an unbiased analysis of the data and also partly to handle the large amount of genetic information extracted from the DNA microarrays.

Methodology: Generally, existing medical decision support systems (DSS) for cancer classification and diagnosis are based on traditional statistical methods such as Bayesian decision theory and machine learning models such as neural networks (NN) and support vector machine (SVM). Though high accuracies have been reported for these systems, they fall short on certain critical areas. These included (a) being able to present the extracted knowledge and explain the computed solutions to the users; (b) having a logical deduction process that is similar and intuitive to the human reasoning process; and (c) flexible enough to incorporate new knowledge without running the risk of eroding old but valid information. On the other hand, a neural fuzzy system, which is synthesized to emulate the human ability to learn and reason in the presence of imprecise and incomplete information, has the ability to overcome the above-mentioned shortcomings. However, existing neural fuzzy systems have their

[*] Corresponding author. Tel.: +65 6790 4926; fax: +65 6792 6559.
E-mail addresses: wltung@pmail.ntu.edu.sg (W.L. Tung), ashcquek@ntu.edu.sg (C. Quek).

0933-3657/$ — see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.artmed.2004.03.009

own limitations when used in the design and implementation of DSS. Hence, this paper proposed the use of a novel neural fuzzy system: the *generic self-organising fuzzy neural network* (GenSoFNN) with *truth-value restriction* (TVR) fuzzy inference, as a fuzzy DSS (denoted as GenSo-FDSS) for the classification of ALL subtypes using gene expression data.

Results and conclusion: The performance of the GenSo-FDSS system is encouraging when benchmarked against those of NN, SVM and the K-nearest neighbor (K-NN) classifier. On average, a classification rate of above 90% has been achieved using the GenSo-FDSS system.

1. Introduction

Acute lymphoblastic leukemia (ALL) is the most common malignancy of childhood, representing nearly one third of all pediatric cancers. Annual incidence of ALL is about 30 cases per million populations, with a peak incidence in patients 2—5 years of age. In the United States, 2000—2500 new cases of childhood ALL are diagnosed each year [1]. Although a small percentage of cases are associated with inherited genetic syndromes, the cause of ALL remains largely unknown. ALL is a heterogeneous disease consisting of various leukemia subtypes that markedly differ in their response to chemotherapy treatment [2]. Currently, the treatment of pediatric ALL is centered on tailoring the intensity of the therapy applied to a patient's risk of relapse [3]. Thus, it is important to group patients into specific risk groups according to the leukemia they are diagnosed with in order to achieve an accurate assessment of a patient's risk of relapse. Subsequently, this determines the course of treatment for the patients. Through the integration of risk assessment with contemporary treatment protocols, overall long-term event-free survival rates approaching 80% have been achieved [4,5]. In [3], it has been comprehensively shown that *gene expression* profiling using DNA microarrays is a viable and accurate diagnostic tool to identify the known prognostically important acute lymphoblastic leukemia subtypes.

Generally, most reported work in the literature that uses gene expression data for clinical diagnosis of cancer subtypes or cancer classifications are based on traditional classifiers developed from statistical approaches such as discriminant analysis [6], Bayesian decision theory [7] or conventional machining learning techniques such as neural network [8], decision trees [9] and support vector machines (SVM) [3,10]. These classifiers are being implemented as a *decision support system* (DSS) that attempts to provide a *systematic* and *consistent* way to analyze the huge amount of scientific data generated from the microarrays and to assist doctors and clinicians in their decision-making process. Although high accuracies have been reported by using these techniques, the classification and decision processes of these classifiers are not apparent or transparent to the user. That is, these classifiers functioned as black boxes and/or the decision-making process is not intuitive to the human cognitive processes. This contributes to a lack of interpretation of the computed decisions. More importantly, the knowledge extracted by these classifiers from the numerical training data cannot be easily assessed. Although some work on knowledge solicitation (extraction) and cancer classification using fuzzified decision trees [11] and manually constructed rule-based systems [12] have been reported in the literature, they have not attempted to address the fundamental issues of a DSS in the context of medical diagnosis. That is, a DSS for medical diagnosis should support a comprehensible reasoning schema that corresponds to the human reasoning process where having a logical deduction is vital and such a system needs to be self-evolving to overcome the *stability—plasticity dilemma* [13]. Thus, the needs to (1) interpret the knowledge derived from the numerical training data; (2) support a logical reasoning schema; and (3) be sufficiently flexible to learn new knowledge without the risk of eroding old but valid information (*catastrophic forgetting*) rendered most existing medical decision support systems inadequate and thus provide the motivations for this piece of work.

On the other hand *soft computing* [14], which emulates the human style of reasoning and decision-making when solving complex problems, can overcome the above-mentioned deficiencies of the conventional medical decision support systems that are based on statistical models and traditional artificial intelligence (AI) techniques. The objective of the various soft computing approaches is to synthesize the human ability to tolerate and process *uncertain*, *imprecise* and *incomplete* information during the decision-making process. A topical approach is the integration of neural network and fuzzy system to create a hybrid structure known as a *neural fuzzy* *network*. Neural fuzzy (or neuro-fuzzy) networks [13,15] such as POPFNN [16—18], ANFIS [19], Falcon-ART [20], Falcon-MART [21], SOFIN [22], EFuNN [23] and DENFIS [24] are the realizations of the functionality of fuzzy systems using neural techniques. The main advantage of a neural fuzzy network is its ability to model the characteristics of a given problem using a high-level linguistic model instead of low-level complex mathematical expressions. The linguistic model is essentially a fuzzy rule-base consisting of a set of IF-THEN fuzzy rules. The fuzzy rules are highly intuitive and can easily be comprehended by the human users. In addition, a neural fuzzy network can self-adjust the parameters of the fuzzy rules using learning algorithms derived from the neural paradigm.

However, existing neural fuzzy systems have their own limitations when used in the design and implementation of medical DS systems. Most neural fuzzy systems proposed in the literature suffers from one or more of the following deficiencies: (1) an *inconsistent* rule-base [15]; (2) operations of the nodes are of heuristic nature or opaque; (3) a poor or weak noise tolerance capability due to the way the clusters (or fuzzy sets) are computed; (4) the *stability—plasticity* dilemma [13] where the ability to incorporate new clusters of data after training is compromised by the risk of eroding away old (but still valid) knowledge. Thus, the system may require retraining to incorporate new information; and (5) required *prior* knowledge such as the number of clusters *C* or the number of fuzzy rules to be computed given a set of numerical training data.

In view of the shortcomings of existing neural fuzzy networks, this paper proposed the use of a novel neural fuzzy system, the GenSoFNN-TVR(S) network, for the implementation of a decision support system for the diagnosis of pediatric ALL cancer subtypes based on gene expression data. The GenSoFNN-TVR(S) network is created by mapping the *truth-value restriction* (TVR) [25] fuzzy inference scheme onto the *generic self-organising fuzzy neural network* (GenSoFNN) [26] architecture, which has been designed to overcome the problems faced by existing neural fuzzy systems. The GenSoFNN-TVR(S) network employs a lean and efficient training cycle that maintains a consistent fuzzy rule-base that can be easily accessed and interpreted by a human user. Moreover, the mapping of the TVR inference scheme to define the computations of the GenSoFNN-TVR(S) network provides it with a strong and intuitive fuzzy reasoning framework that corresponds to the human cognitive process (and thus decision-making). In addition, the GenSoFNN-TVR(S) network can self-evolve and accommodate *incremental learning*, thus avoiding the *stability—plasticity* dilemma.

This paper is organized as follows. Section 2 briefly introduces the concepts of the TVR fuzzy inference scheme employed by the proposed GenSoFNN-TVR(S) network and Section 3 describes the generic structure of the GenSoFNN [26] architecture from which the GenSoFNN-TVR(S) network is developed. In addition, the one-pass training cycle of the GenSoFNN network, which consists of the *self-organizing*, *rule mapping* and *parameter learning* phases and is adopted by the GenSoFNN-TVR(S) network, is addressed. Section 4 presents the detailed node functions of the GenSoFNN-TVR(S) network that are derived from the operations of the TVR inference scheme. In Section 5, the methodology of employing the GenSoFNN-TVR(S) network as a fuzzy DSS for ALL cancer subtype identification using gene expression data is presented. Section 6 discusses the significance of the results obtained and Section 7 concludes this paper.

2. Truth-value restriction fuzzy inference scheme

A fuzzy *proposition* such as "*x* is A" that appeared in the linguistic IF-THEN fuzzy rules of a neural fuzzy system is an extension of the classical binary (crisp or two-valued) logic proposition. Under the fuzzy logic framework [27], A is defined as a fuzzy set or a possibility distribution that binds the possible values of *x* with varying degrees of membership to the fuzzy concept induced by A. For instance, for the fuzzy proposition "*x* is OLD", *x* denotes the range of possible ages while OLD is a fuzzy concept represented by a fuzzy set or possibility distribution. There are two levels of representation for a fuzzy logic proposition: (1) the description of the proposition itself using a fuzzy set or (2) the *degree of possibility* that the proposition is true (or false), otherwise known as the *truth-value* of the proposition, which serves as a *confidence measurement* [28] for the proposition based on observed (or supporting) evidence. Consequently, a fuzzy proposition can be analysed and manipulated in two different ways: by means of the fuzzy set denoting the proposition or through the numeric or *linguistic truth-value* (LTV) [29] of the proposition. However, the truth-value of a fuzzy proposition is *not* the actual degree of truth of a proposition. This is because a proposition (fuzzy or non-fuzzy) has to be either *TRUE* or *FALSE*. A fuzzy proposition is induced by *vague*, *imprecise* and/or *incomplete* information. Thus, if one has absolute knowledge of the information, then a fuzzy proposition reduces to a binary one that evaluates as *TRUE* or *FALSE*.

Hence, *approximate reasoning* [13,30] based on fuzzy inference using fuzzy rules that contain fuzzy propositions can be performed in two ways: using fuzzy sets or fuzzy truths. In the former, the inference is based on the relation from the fuzzy sets in the antecedents to the fuzzy sets in the consequent of an IF-THEN fuzzy rule. In the latter, the inference is based on the relation from the truth-value of the antecedents to the truth-value of the consequent of the fuzzy rule. Accordingly, there are two general approaches to the inference process in fuzzy logic: the *compositional rule of inference* (CRI) [27] and the *truth-value restriction* (TVR) [25] inference schemes. Details on the TVR inference scheme are presented as Appendix A. Many existing neural fuzzy networks proposed in the literature employed the truth-value restriction (TVR) method to perform the fuzzy inference process in order to compute the network outputs [17,31—33]. This is because the computed truth-values of the antecedents can be effectively propagated through the hybrid structure of a neural fuzzy system. This makes the truth-value restriction method a viable and attractive alternative to the CRI inference scheme for approximate reasoning in a neural fuzzy system such as the GenSoFNN [26] architecture.

3. The generic self-organising fuzzy neural network

The main problems [26] dogging most existing neural fuzzy systems are: (1) susceptibility towards noisy/spurious training data due to the choice of clustering technique; (2) the *stability—plasticity* dilemma [13] in which the neural fuzzy system is not flexible enough to incorporate new clusters of data (knowledge) after training has completed; (3) require *prior* knowledge of the number of clusters to be computed or has to predefine the number of fuzzy rules to be formulated; (4) inconsistent rule-base or inconsistent representation of fuzzy labels [15]; and (5) operations of the nodes are of heuristic nature or not clearly defined, which leads to a poor interpretation of the reasoning/decision-making process of the system. These deficiencies are essentially due to the architectural design and training techniques employed to construct the neural fuzzy systems. Thus, the *generic self-organising fuzzy neural network* (GenSoFNN) [26] is developed with a *lean* and *structured* training cycle that consists of three distinct phases: (a) *self-organising*: clustering of the numerical training data into fuzzy sets; (b) *rule formulation*: constructing a set of IF-THEN fuzzy rules using the computed fuzzy sets to adequately represent the underlying knowledge of the training data; and (c) *parameter learning*: supervised tuning of the fuzzy sets to achieve the desired output response(s). The well-ordered training cycle of the GenSoFNN network serves as a basis for the crafting of a consistent rule-base and as a primary solution for the deficiencies listed above.

3.1. Structure of GenSoFNN

The GenSoFNN network (Fig. 1) [26] consists of five layers of nodes.

Each input node IV_i, $i \in \{1, \ldots, n1\}$, has a single input denoted as x_i. The vector $\mathbf{X} = [x_1, \ldots, x_i, \ldots, x_{n1}]^T$ represents all the inputs to the GenSoFNN network. Each output node OV_m, where $m \in \{1, \ldots, n5\}$, computes a single output denoted by y_m. The vector $\mathbf{Y} = [y_1, \ldots, y_m, \ldots, y_{n5}]^T$ denotes the outputs of the GenSoFNN network with respect to the input stimulus $\mathbf{X}$. In addition, the vector $\mathbf{D} = [d_1, \ldots, d_m, \ldots, d_{n5}]^T$ represents the desired network outputs required for the *back propagation* (BP) [34] based parameter-learning phase of the training cycle. The trainable weights of the GenSoFNN network are found in layers 2 and 5 (enclosed in rectangular boxes in Fig. 1). Layer 2 (5) links contain the parameters of the input (output) fuzzy sets. The weights of the remaining connections are unity. The trainable weights (parameters) are interpreted as the corners of the *normal* trapezoidal-shaped fuzzy sets (Fig. 2) computed by the GenSoFNN network, where the maximum membership is unity. They are denoted as l and r (left and right support points), and u and v (left and right kernel points). The subscripts denote the pre-synaptic and post-synaptic nodes respectively. For clarity in subsequent discussions, the variables i, j, k, l, m are used to refer to arbitrary nodes in layers 1, 2, 3, 4 and 5, respectively. The output of a node is denoted as Z and the subscripts specify its origin.

Each input node IV_i may have different number of input fuzzy terms J_i. The input terms are denoted as $\mathrm{IL}_{i,j}$, where $i = \{1, \ldots, n1\}$ and $j = \{1, \ldots, J_i\}$. Hence, the number of layer 2 nodes is $n2 = \sum_{i=1}^{n1} J_i$. Layer 3 consists of the rule nodes R_k, where $k = \{1, \ldots, n3\}$. At layer 4, an output term node $\mathrm{OL}_{l,m}$ may have more than one fuzzy rule attached to it. Each output node OV_m in layer 5 can have different number of output fuzzy terms L_m. Hence, the number of layer 4 nodes is $n4 = \sum_{m=1}^{n5} L_m$. The GenSoFNN network adopts

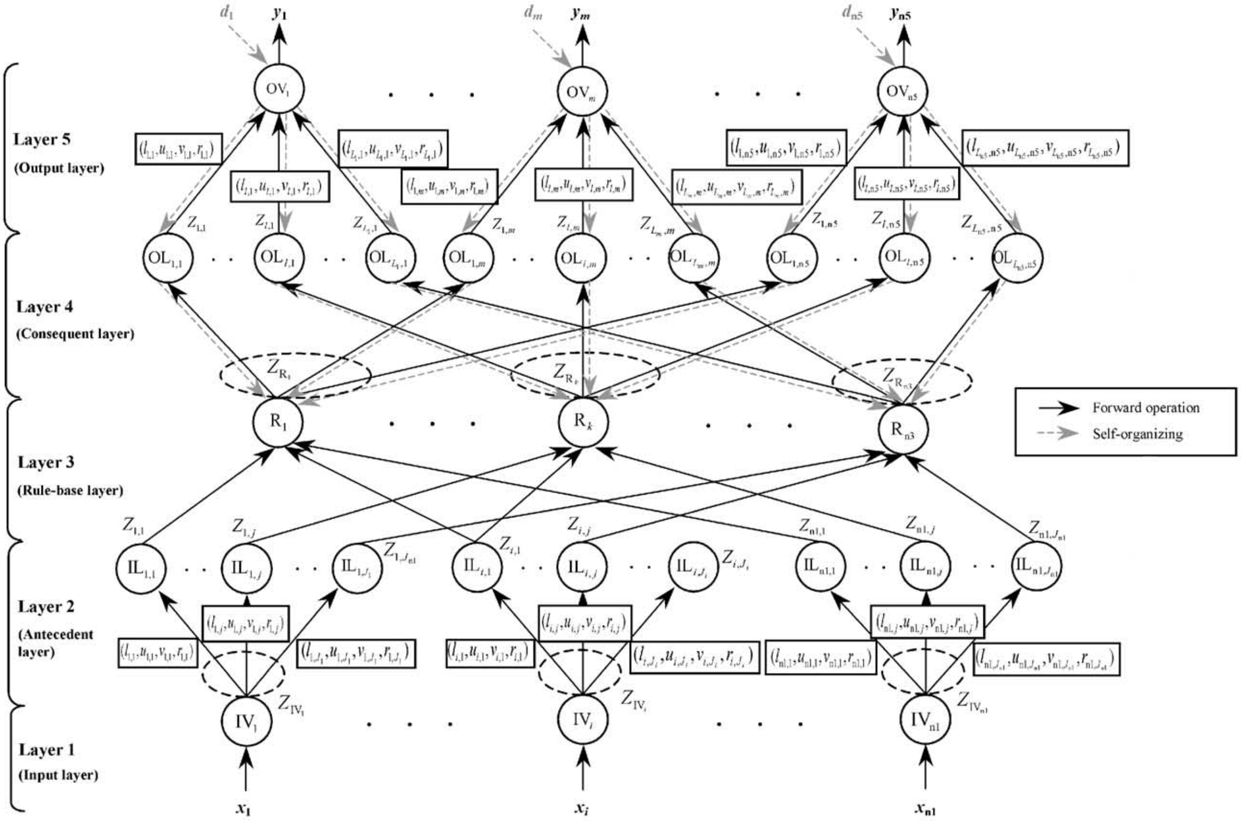
Layer 5
(Output layer)
Layer 4
(Consequent layer)
Layer 3
(Rule-base layer)
Layer 2
(Antecedent layer)
Layer 1
(Input layer)
Forward operation
Self-organizing
d_1 y_1
d_m y_m
d_{n5} y_{n5}
OV_1
OV_m
OV_{n5}
$OL_{1,1}$ $OL_{l,1}$ $OL_{L_1,1}$ $OL_{1,m}$ $OL_{l,m}$ $OL_{L_m,m}$ $OL_{1,n5}$ $OL_{l,n5}$ $OL_{L_{n5},n5}$
Z_{R_1} Z_{R_k} $Z_{R_{n3}}$
R_1 R_k R_{n3}
$IL_{1,1}$ $IL_{1,j}$ $IL_{1,J_{n1}}$ $IL_{i,1}$ $IL_{i,j}$ IL_{i,J_i} $IL_{n1,1}$ $IL_{n1,j}$ $IL_{n1,J_{n1}}$
Z_{IV_1} Z_{IV_i} $Z_{IV_{n1}}$
IV_1 IV_i IV_{n1}
x_1 x_i x_{n1}

Figure 1 Structure of the GenSoFNN network.

the *Mamdani* fuzzy model [15]. The *k*th fuzzy rule has the form as shown in Eq. (1).

$$\begin{aligned} R_k: \quad & \textbf{If } x_1 \text{ is } \mathrm{IL}_{(1,j)_k} \ldots \text{ and } x_i \text{ is } \mathrm{IL}_{(i,j)_k} \ldots \text{ and } \\ & x_{n1} \text{ is } \mathrm{IL}_{(n1,j)_k} \\ & \textbf{Then } y_1 \text{ is } \mathrm{OL}_{(l,1)_k} \ldots \text{ and } y_m \text{ is } \mathrm{OL}_{(l,m)_k} \ldots \text{ and } \\ & y_{n5} \text{ is } \mathrm{OL}_{(l,n5)_k} \end{aligned} \tag{1}$$

where $\mathrm{IL}_{(i,j)_k}$ is the *j*th fuzzy label of the *i*th input that is connected to rule R_k; and $\mathrm{OL}_{(l,m)_k}$is the *l*th fuzzy label of the *m*th output to which rule R_k is connected to.

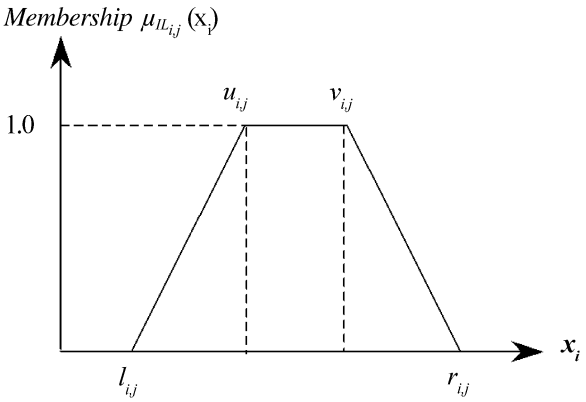

Figure 2 *Normal* trapezoidal fuzzy set representing the *j*th fuzzy term of the *i*th input (denoted as $\mathrm{IL}_{i,j}$).

3.2. Self-organisation of GenSoFNN

The *discrete incremental clustering* (DIC) [26,35] technique is developed and integrated into the GenSoFNN network to automatically compute the input—output clusters from the numerical training data. The DIC technique maintains a consistent representation of the fuzzy sets (fuzzy labels) by performing clustering on a local basis. This is similar to the ART concept [36]. However, unlike ART, the number of fuzzy sets for each input/output dimension may be different and if the fuzzy label (fuzzy set) for a particular input/output dimension already exists, then it is not "re-created". Hence, DIC ensures that a fuzzy label is uniquely defined by a fuzzy set and this serves as a basis to formulate a consistent rule-base using the GenSoFNN network. The DIC technique has five parameters: a plasticity parameter β, a tendency parameter TD, an expansion parameter STEP, a membership threshold MT and a fuzzy set support parameter SLOPE. The plasticity parameter β and the parameter STEP control the appropriate expansion of a fuzzy set to include a new training data point. The tendency parameter TD maintains the integrity of a fuzzy set so that only similar data points are clustered together while SLOPE defines the gradients of the left and right-sided slopes of the trapezoidal fuzzy sets. The membership threshold MT determines the minimum membership value a data point should have before it is considered for inclusion into an existing cluster. Else a new cluster is created to hold the dissimilar data point. During the self-organisation phase, the plasticity parameter β and the tendency parameter TD for an expanding cluster (fuzzy set) is gradually reduced to constrain its future expansion. For the interested reader, more details on the DIC technique and the use of its various parameters are reported in [26,35]. The algorithmic form of the DIC technique is presented as follows. DIC is applied to cluster both the input and output data points.

Algorithm DIC

Assume a dataset $\tilde{X} = \{X^{(1)}, \ldots, X^{(p)}, \ldots, X^{(P)}\}$, where P is the number of training vectors and $X^{(p)} = \{x_1^{(p)}, \ldots, x_i^{(p)}, \ldots, x_n^{(p)}\}$ denotes the pth training vector in the space $\Re^n$.... Initialize STEP, SLOPE and set β = TD = 0.5. The threshold MT ∈ (0,1] is user-defined.

```
For all training vector $X^{(p)}$, where $p \in \{1, \ldots, P\}$ do {
  For all dimensions $i \in \{1, \ldots, n\}$ do {
    If there are no clusters (fuzzy sets) in the $i$th dimension (i.e. $J_i = 0$)
      Create a new cluster using the point $x_i^{(p)}$
    Otherwise
      Find the best-fit cluster Winner for $x_i^{(p)}$ using Eq. (2).
```

$$\text{Winner} = \arg \max_{j \in \{1 \ldots J_i\}} \{\mu_{i,j}(x_i^{(p)})\} \tag{2}$$

where $\mu_{i,j}$ is the membership function of the jth fuzzy set in dimension i.

```
      If $\mu_{i,\text{Winner}}(x_i^{(p)}) > \text{MT}$   /* Membership value greater than threshold */
        Update the kernel of Winner   /* grows the cluster Winner */
        Update $\beta$ and TD
      Otherwise
        Create a new cluster using the point $x_i^{(p)}$
  } End For all dimensions $i \in \{1, \ldots, n\}$
} End For all training vector $X^{(p)}$, $p \in \{1, \ldots, P\}$
End DIC
```

3.3. Rule formulation of GenSoFNN

The rule-base of the GenSoFNN network is formulated from the numerical training data pairs (**X**, **D**) using a *rule mapping* process named RuleMAP. Under the GenSoFNN framework, "input space partition of rule k" (ISP$_k$) is the collective term for all the input fuzzy labels (layer 2 nodes) that contribute to the antecedent of rule node R_k. Similarly, "output space partition of rule k" (OSP$_k$) refers to all the output fuzzy labels (layer 4 nodes) that form the consequent of rule R_k. During the rule mapping process, each rule R_k, $k \in \{1, \ldots, n3\}$, activates its ISP (OSP) with a firing of layers 1 and 2 (layers 4 and 5) of the GenSoFNN network with the input stimulus **X** (desired outputs **D**) feeding into layer 1 (layer 5). The backward links depicted by the dashed, gray arrows in Fig. 1 are used for the activation of the OSPs. Fig. 3 presents the flowchart of the RuleMAP process with the embedded self-organising and parameter learning phases. The function *EstLink* identifies the proper connections between the input fuzzy labels (layer 2 nodes), the fuzzy rules (layer 3 nodes) and the output fuzzy labels (layer 4 nodes). Overlapping input/output labels are annexed and their respective rules are combined if necessary to

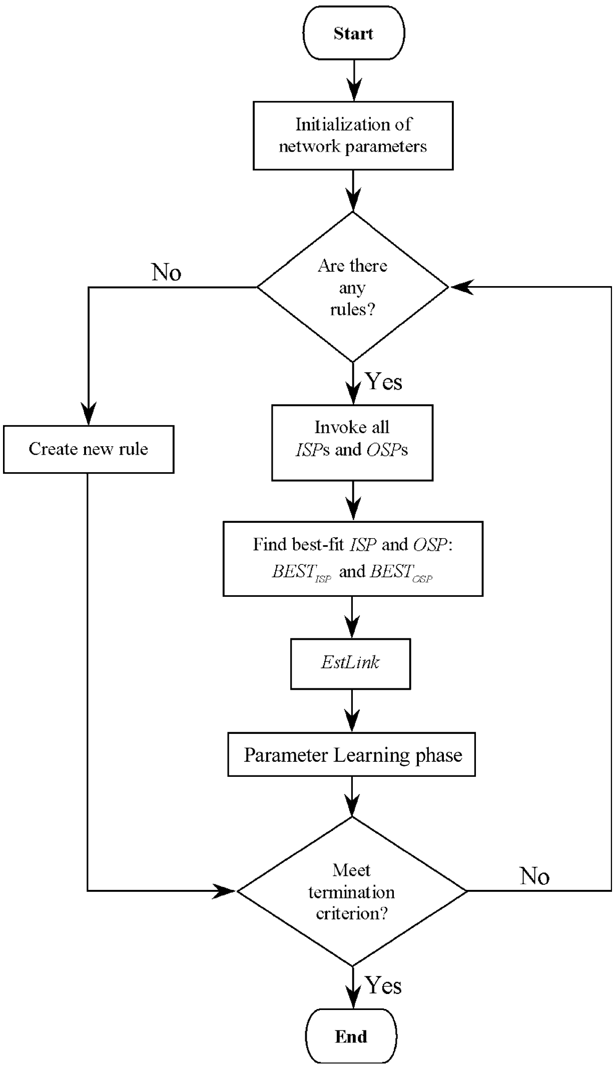
Start
Initialization of network parameters
Are there any rules?
No
Yes
Create new rule
Invoke all ISPs and OSPs
Find best-fit ISP and OSP: BEST_ISP and BEST_OSP
EstLink
Parameter Learning phase
Meet termination criterion?
No
Yes
End

Figure 3 Flowchart of RuleMap process.

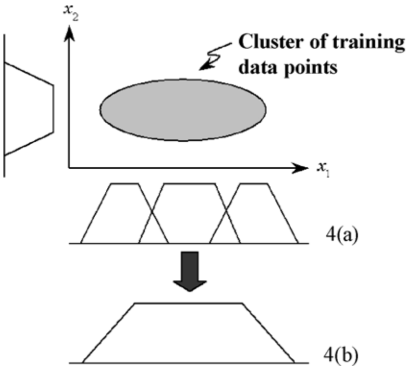

Figure 4 Defragmenting small clusters.

maintain a consistent rule-base. Details on the RuleMAP process are described in [37]. The RuleMAP process is responsible for the *structural learning* of the GenSoFNN network. The crafted rule-base is consistent but not compact, as there may be numerous redundant and/or obsolete rules. Redundant and obsolete rules are the results of the dynamic training of the GenSoFNN where the fuzzy sets of the fuzzy rules are constantly tuned by the back-propagation algorithm. To maintain the *integrity*, *accuracy* as well as the *compactness* of the rule-base, these redundant rules are deleted at the end of each training epoch.

3.4. Cluster defragmentation

Although the DIC clustering technique employed by the GenSoFNN network does not require a prior knowledge of the number of clusters to be computed, it suffers from a condition known as "cluster fragmentation". That is, numerous small clusters may be created when there should only be one large cluster. This problem is illustrated in Fig. 4.

In GenSoFNN, the first scenario as depicted by Fig. 4(a) may occur. Therefore, to transform the first scenario to the second scenario (Fig. 4(b)), a merging algorithm is developed for the self-organizing phase of the GenSoFNN network. This is to better reflect the clustering nature of the training data points and to reduce the number of fuzzy rules formulated. This improves the intuitiveness of the linguistic model and enhances the comprehensibility of the fuzzy rule-base. Defragmentation is only performed on the input fuzzy labels (fuzzy sets) and only fragmented clusters belonging to the same class of data points are merged to form a larger cluster.

3.5. Pruning of weak/insignificant rules

The rule formulation phase of the training cycle of the GenSoFNN network is responsible for the derivation of the fuzzy rules based on the computed clusters from the self-organizing phase. The appropriate input space partitions are mapped or linked to the appropriate output space partitions through the layer 3 rule nodes to derive the fuzzy rules. The GenSoFNN network adopts the incremental rule-learning approach. That is, no fuzzy rules initially existed and they are constructed only if there are training data points that justified the existence of the fuzzy rules. However, some rules may be more important than others in the modeling of the problem domain, especially if their input—output space partitions covered a significant portion of the input—output space. There are also insignificant/weak rules created due to the existence of noisy/spurious training data points. These insignificant or weak rules may interfere and contribute errors to the network outputs during the output inference process. Hence, such insignificant rules are identified and removed. In the GenSoFNN network, the strengths of the fuzzy rules are computed during the training cycle and rules with strengths that fall below a predefined threshold Thres_{Pr} are pruned away. The strength of a fuzzy rule R_k, denoted as S_k, is computed using Eq. (3).

$$S_k(T+1) = S_k(T) + (F_{\text{ISP}_k}(T) \times F_{\text{OSP}_k}(T)),\quad S_k(0) = 0 \tag{3}$$

where $F_{\text{ISP}_k}(T)$ is the forward aggregated input to rule R_k due to the activation of its ISP; and $F_{\text{OSP}_k}(T)$ is backward aggregated input to R_k due to the activation of its OSP at time T; and $S_k(0)$ is the initial strength of a newly created rule R_k in the GenSoFNN network. Fig. 5 illustrates the concept of $F_{\text{ISP}_k}(T)$ and $F_{\text{OSP}_k}(T)$ of a fuzzy rule R_k.

Hence, at the end of the training cycle, the aggregated sum of all the rule strengths in the GenSoFNN network is computed and a rule R_k is pruned if Eq. (4) equates as true.

$$n3\left(\frac{S_k}{\sum_{k=1}^{n3} S_k}\right) < \text{Thres}_{\text{Pr}} \tag{4}$$

where Thres_{Pr} is a user pre-defined parameter for pruning of weak/insignificant rules.

3.6. Parameter learning of GenSoFNN

The back-propagation learning equations for the parameter-learning phase depended on the fuzzy inference scheme adopted by the GenSoFNN network. In this paper, the operations of the TVR inference with Larsen implication are mapped onto the GenSoFNN network in Section 4 to produce the GenSoFNN-TVR(S) network. For the interested reader, please refer to [38] on the full derivation

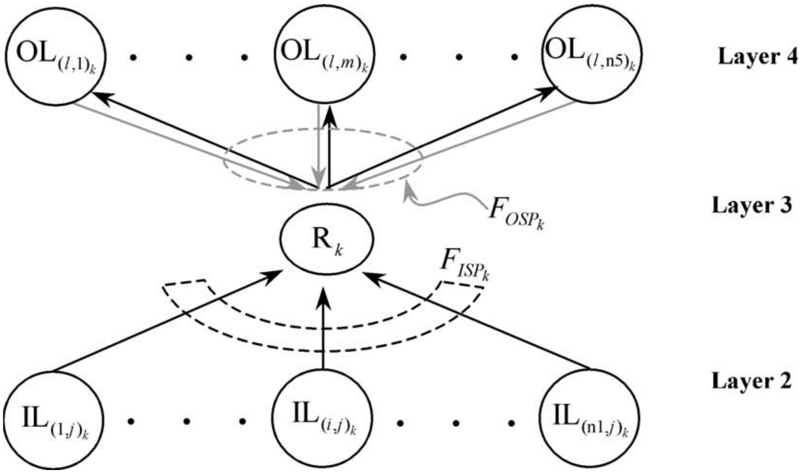

Figure 5 An arbitrary rule R_k in the GenSoFNN network and its $F_{\mathrm{ISP}_k}(T)$ and $F_{\mathrm{OSP}_k}(T)$.

of the back-propagation learning equations for the GenSoFNN-TVR(S) network.

Two motivations drive the development of the GenSoFNN network. The first is to define a systematic way of crafting the linguistic model required in neural fuzzy systems to describe the dynamics and characteristics of a problem domain. The second motivation is to develop a generalised network architecture whereby different fuzzy inference schemes such as the *compositional rule of inference* (CRI) [27], *truth-value restriction* (TVR) [25] and *approximate analogical reasoning schema* (AARS) [39] can be readily mapped onto with ease. This correlates to the definition of a neural fuzzy network (system). That is, a neural fuzzy network is the integration of fuzzy system and neural network whereby the operations of the hybrid system should be functionally equivalent to a similar standalone fuzzy system based on the adopted fuzzy inference scheme. Hence, the operations and outputs of the various nodes in the GenSoFNN network are defined by the fuzzy inference scheme adopted by the network. In this paper, the TVR inference scheme is mapped to the GenSoFNN network to define the node functions. The resultant network is henceforth referred to as GenSoFNN-TVR(S), where "(S)" denotes the *singleton fuzzifiers* [40] implemented at the input layer of the network. The next section presents the detailed operations of the proposed GenSoFNN-TVR(S) network.

4. The GenSoFNN-TVR(S) network

The proposed GenSoFNN-TVR(S) network is created by mapping the operations of the TVR inference scheme onto the generic structure of the GenSoFNN network. As highlighted in Appendix A, the TVR inference process can be intuitively partitioned into five steps. These steps are logically mapped to each of the five layers of the GenSoFNN-TVR(S) network. The *aggregation* and *activation* functions of an arbitrary node in layer l are denoted as $f^{(l)}$ and $a^{(l)}$ respectively, where $l \in \{1, \ldots, 5\}$.

4.1. Layer 1—singleton fuzzifier

Layer 1 nodes are the input nodes of the GenSoFNN-TVR(S) network. They acted as singleton fuzzifiers that performed fuzzification of crisp-valued inputs presented to the GenSoFNN-TVR(S) network. Fuzzification of the inputs is necessary in order to map the operations of the GenSoFNN-TVR(S) network to the TVR inference scheme. With respect to input node IV_i,

$$\text{net synaptic input of node } \mathrm{IV}_i,\ \mathrm{Net}_{\mathrm{IV}_i} = f^{(1)}(x_i) = x_i \tag{5}$$

$$\text{net synaptic output of node } \mathrm{IV}_i,\ Z_{\mathrm{IV}_i} = a^{(1)}(\mathrm{Net}_{\mathrm{IV}_i}) = a^{(1)}(x_i) = \bar{x}_i \tag{6}$$

where x_i is the ith input to the GenSoFNN-TVR(S) network; and $\bar{x}_i$ is the fuzzified equivalent of the input x_i. The value $\bar{x}_i$ is the only element in the fuzzy set $\bar{X}_i$ whose membership function $\mu_{\bar{X}_i}$ is defined as:

$$\mu_{\bar{X}_i}(\bar{x}_i) = \begin{cases} 1, & \text{if } \bar{x}_i = x_i \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

The graphical interpretation of the singleton fuzzifier is depicted in Fig. 6.

4.2. Layer 2—antecedent matching

Layer 2 nodes are known as the input fuzzy label (or term) nodes and they encapsulate the input fuzzy

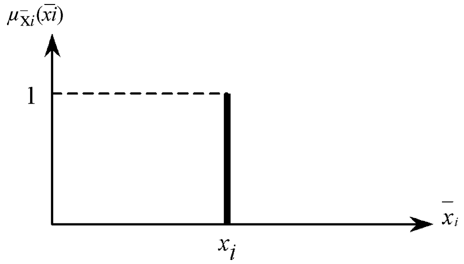

Figure 6 Singleton fuzzifier employed by the GenSoFNN-TVR(S) network.

sets of the GenSoFNN-TVR(S) network. The function of the layer 2 nodes is to perform antecedent matching of the inputs against the fuzzy sets they represent and compute a similarity measure known as the membership value (MV). This membership value is presented as output of the layer 2 nodes. With respect to term node $\mathrm{IL}_{i,j}$ (the jth label of the ith input node IV_i),

$$\text{net synaptic input of node } \mathrm{IL}_{i,j},\ \mathrm{Net}_{i,j} = f^{(2)}(\mathrm{Z}_{\mathrm{IV}_i}) = \bar{x}_i \tag{8}$$

$$\text{net synaptic output of node } \mathrm{IL}_{i,j},\ \mathrm{Z}_{i,j} = a^{(2)}(\mathrm{Net}_{i,j}) = \mu_{\mathrm{IL}_{i,j}}(\bar{x}_i) = \mu_{\mathrm{IL}_{i,j}}(x_i) \tag{9}$$

where $\mu_{\mathrm{IL}_{i,j}}$ is the membership function of term node $\mathrm{IL}_{i,j}$. Since trapezoidal-shaped fuzzy sets are used in the proposed GenSoFNN-TVR(S) network, the membership function $\mu_{\mathrm{IL}_{i,j}}$ of node $\mathrm{IL}_{i,j}$ is defined as in Eq. (10). The fuzzy set of node $\mathrm{IL}_{i,j}$ is shown as Fig. 2.

$$\mu_{\mathrm{IL}_{i,j}}(x_i) = \begin{cases} 0, & \text{if } x_i \le l_{i,j} \\ \dfrac{x_i - l_{i,j}}{u_{i,j} - l_{i,j}}, & \text{if } l_{i,j} < x_i < u_{i,j} \\ 1, & \text{if } u_{i,j} \le x_i \le v_{i,j} \\ \dfrac{r_{i,j} - x_i}{r_{i,j} - v_{i,j}}, & \text{if } v_{i,j} < x_i < r_{i,j} \\ 0, & \text{if } x_i \ge r_{i,j} \end{cases} \tag{10}$$

4.3. Layer 3—rule fulfillment

Layer 3 of the GenSoFNN-TVR(S) network contains the fuzzy rule-base of the network. Each layer 3 node is a fuzzy rule in the rule-base of the proposed GenSoFNN-TVR(S) network. There are a total of $n3$ rule nodes in layer 3. Each rule node R_k computes the degree of fulfillment of the current inputs with respect to the antecedents of the fuzzy rule it represents. The higher the degree of fulfillment, the greater is the compatibility of the inputs to the *input space partition* (ISP) of rule R_k. Each layer 3 node presents the computed degree of rule fulfillment as its output. The net synaptic input of node R_k, Net_{R_k}, and the net synaptic output Z_{R_k} are defined as:

$$\begin{aligned}\text{net synaptic input of node } R_k,\ \mathrm{Net}_{R_k} &= f^{(3)}(Z_{(i,j)_k}, \ldots, Z_{(i,j)_k}, \ldots, Z_{(n1,j)_k}) \\ &= \{Z_{(1,j)_k}, \ldots, Z_{(i,j)_k}, \ldots, Z_{(n1,j)_k}\}\end{aligned} \tag{11}$$

$$\begin{aligned}\text{net synaptic output of node } R_k,\ Z_{R_k} &= a^{(3)}(\mathrm{Net}_{R_k}) \\ &= \min_{i \in \{1,\ldots,n1\}} \{Z_{(1,j)_k}, \ldots, Z_{(i,j)_k}, \ldots, Z_{(n1,j)_k}\}\end{aligned} \tag{12}$$

where $Z_{(i,j)_k}$ is the output of the jth fuzzy label of the ith input ($\mathrm{IL}_{i,j}$) that is connected to rule R_k; and $n1$ is the number of inputs to the GenSoFNN-TVR(S) network.

4.4. Layer 4—consequent derivation

The layer 4 nodes in the GenSoFNN-TVR(S) network are the output fuzzy term nodes that hold the consequent of the fuzzy rules in layer 3. Each layer 4 node represents an output fuzzy set. The parameters of the output fuzzy sets are implemented as weights on layer 5 links (Fig. 1). Each $\mathrm{OL}_{l,m}$ node may have more than one fuzzy rule feeding into it because different rules can have the same consequent. However, each $\mathrm{OL}_{l,m}$ node is connected to only one output node OV_m in layer 5. The net synaptic input $\mathrm{Net}_{l,m}$ and net synaptic output $\mathrm{Z}_{l,m}$ of node $\mathrm{OL}_{l,m}$ are defined as:

$$\begin{aligned}\text{net synaptic input of node } \mathrm{OL}_{l,m}, \mathrm{Net}_{l,m} &= f^{(4)}(Z_{R_1}^{(l,m)}, \ldots, Z_{R_k}^{(l,m)}, \ldots, Z_{R_{N_{l,m}}}^{(l,m)}) \\ &= \{Z_{R_1}^{(l,m)}, \ldots, Z_{R_k}^{(l,m)}, \ldots, Z_{R_{N_{l,m}}}^{(l,m)}\}\end{aligned} \tag{13}$$

$$\begin{aligned}\text{net synaptic output of node } \mathrm{OL}_{l,m}, \mathrm{Z}_{l,m} &= a^{(4)}(\mathrm{Net}_{l,m}) \\ &= \max_{k \in \{1,\ldots,N_{l,m}\}} \{Z_{R_1}^{(l,m)}, \ldots, Z_{R_k}^{(l,m)}, \ldots, Z_{R_{N_{l,m}}}^{(l,m)}\}\end{aligned} \tag{14}$$

where $Z_{R_k}^{(l,m)}$ is the output of the kth rule in GenSoFNN-TVR(S) with $\mathrm{OL}_{l,m}$ as its consequent; and $\mathrm{N}_{l,m}$ is the total number of rules in GenSoFNN-TVR(S) with $\mathrm{OL}_{l,m}$ as their consequent.

The feed-forward operation of node $\mathrm{OL}_{l,m}$ is to filter the largest input from layer 3 that is feeding into it. This largest input from layer 3 corresponds to the output of the rule that has the highest degree of rule fulfillment due to the current inputs and has term $\mathrm{OL}_{l,m}$ as part of its consequent. Node $\mathrm{OL}_{l,m}$ presents this largest input as its output to the corresponding output node in layer 5.

4.5. Layer 5—output defuzzification

Layer 5 nodes are the output nodes of the GenSoFNN-TVR(S) network. There are a total of $n5$ output nodes. During the feed-forward operation of the GenSoFNN-TVR(S) network, the output nodes are responsible for the defuzzification of the derived fuzzy outputs from the TVR inference scheme before presenting them as crisp network outputs. A modified weighted *center of averaging* (COA) [13] defuzzification technique is used to compute the crisp network outputs. The net synaptic input $\mathrm{Net}_{\mathrm{OV}_m}$ and net synaptic output y_m for output node OV_m are defined as:

net synaptic input for node $\mathrm{OV}_m, \mathrm{Net}_{\mathrm{OV}_m}$

$$f^{(5)}(Z_{1,m}, \ldots, Z_{l,m}, \ldots, Z_{L_m,m}) = \{Z_{1,m}, \ldots, Z_{l,m}, \ldots, Z_{L_m,m}\} \tag{15}$$

net synaptic output for node OV_m, y_m

$$= a^{(5)}(\mathrm{Net}_{\mathrm{OV}_m}) = \frac{\overbrace{\sum_{l=1}^{L_m}[(Z_{l,m} \times \tilde{m}_{l,m})/\mathrm{Ker}(\mathrm{OL}_{l,m})]}^{\mathrm{MZSum}_m}}{\underbrace{\sum_{l=1}^{L_m}[Z_{l,m}/\mathrm{Ker}(\mathrm{OL}_{l,m})]}_{\mathrm{ZSum}_m}} = \frac{\mathrm{MZSum}_m}{\mathrm{ZSum}_m} \tag{16}$$

where $Z_{l,m}$ is the output of node $\mathrm{OL}_{l,m}$ in layer 4; L_m is the number of output fuzzy term nodes OV_m has; and $\tilde{m}_{l,m}$ is the mean point of the kernel of the fuzzy set represented by output term $\mathrm{OL}_{l,m}$. The parameter $\tilde{m}_{l,m}$ is defined by Eq. (17):

$$\tilde{m}_{l,m} = \frac{1}{2}(u_{l,m} + v_{l,m}) \tag{17}$$

where $u_{l,m}$ and $v_{l,m}$ are the left and right kernel points of the fuzzy set of $\mathrm{OL}_{l,m}$. The term $\mathrm{Ker}(\mathrm{OL}_{l,m})$ in Eq. (16) is a measure of the precision of the output fuzzy set $\mathrm{OL}_{l,m}$. Thus, $1/\mathrm{Ker}(\mathrm{OL}_{l,m})$ is a weighting factor that assigns greater weights towards output fuzzy terms $\mathrm{OL}_{l,m}$ with a small kernel range (i.e. $v_{l,m} - u_{l,m}$). This is equivalent to associating larger weights to better-defined output fuzzy terms that are more precise. The term $\mathrm{Ker}(\mathrm{OL}_{l,m})$ is defined as

$$\mathrm{Ker}(\mathrm{OL}_{l,m}) = \begin{cases} v_{l,m} - u_{l,m}, & \text{if } v_{l,m} > u_{l,m} \\ 1, & \text{otherwise} \end{cases} \tag{18}$$

The nodal operations of the proposed GenSoFNN-TVR(S) network are as presented above. A layer-by-layer mapping to the inference steps of the TVR inference scheme is presented as Appendix B to demonstrate that the GenSoFNN-TVR(S) network indeed realizes the operations of a standalone TVR-based fuzzy system.

5. ALL cancer subtype classification using gene expression data

In this paper, the GenSoFNN-TVR(S) network is employed as a *fuzzy decision support system* (GenSo-FDSS) for ALL cancer subtype identification and classification. The proposed GenSo-FDSS system is evaluated using a pediatric *acute lymphoblastic leukemia* (ALL) cancer dataset [3]. The purpose of the experiment is to correctly classify the leukemia cancers into their proper subtypes using *gene expressions* identified with DNA *microarrays* [41,42].

5.1. DNA microarray technology

Expression genomics is an approach that examines gene expression in a comprehensive and massively parallel fashion [43]. The core technology in realizing expression genomics is DNA microarrays [44], whereby thousands of DNA probes are hybridized against fluorophore-labeled cDNA or cRNA targets from template RNA sources. This allows for a global and simultaneous view on the transcriptional levels of thousands of genes (of sample tissue cell) and has been proven valuable in the classification of human cancers [45,46]. For a detailed discussion on DNA microarrays, please refer to [44] and [47].

The major challenges in using the gene expression data from DNA microarrays for cancer classification is the analysis and interpretation of the massive data output [43,48] due to (1) the large number of genes observed; and (2) the extremely skewed ratio between the number of genes monitored and the actual number of tissue samples available. In most microarray experiments, the number of clinical tissue samples obtained from patients is limited to hundreds or less while the number of genes monitored is in the order of thousands. Hence, gene expression data contains a lot of genetic noise and redundant features. Therefore, *feature selection* [49] has to be performed prior to the development of a classifier as it has been well studied in the machine-learning community that no classifier is able to perform satisfactory when the number of features far-exceeded the number of training samples available.

5.2. Gene expression data of pediatric ALL

In this paper, the newly crafted GenSoFNN-TVR(S) network is used in the design and implementation of a FDSS for pediatric ALL cancer classification using the gene expression data extracted from the bone marrow tissue samples of a group of 327 child patients. Compared to traditional classifiers such as the SVM and K-NN classifiers, a neural fuzzy based classifier such as the proposed GenSoFNN-TVR(S) network has the ability to solicit knowledge from the numerical training data and express this knowledge in an easily accessible form through its trained structure. The extracted knowledge is retrieved by means of highly comprehensive linguistic IF-THEN fuzzy rules. The pediatric ALL cancer gene expression dataset used in this simulation is obtained using Affymetrix oligonucleotide microarrays [50] with 12,600 probe sets. After preprocessing using a variation filter to remove genes with little or no activity in $<$1% of the samples, a final dataset of 12,558 genes is obtained. This dataset is available online at [51]. There are a total of seven subclasses of leukemia tissue samples in this dataset. They are denoted as: Class 1, BCR-ABL; Class 2, E2A-PBX1; Class 3, Hyper $>$ 50; Class 4, MLL; Class 5, T-ALL; Class 6, TEL-AML1; and Class 7, Others. For the definitions and detailed differences of each subclass of ALL leukemia, the reader is referred to the supplementary data available at [51]. Due to the large number of input features, feature selection has to be performed prior to training.

In this paper, a novel feature selection algorithm named *Monte Carlo evaluative selection* (MCES) [52] is employed to identify the relevant genes (features) that separate the various subclasses of ALL leukemia using the gene expression data. The MCES method has the advantages of (1) low computational cost; (2) has the ability to identify both correlated and irrelevant features based on weight ranking; (3) is applicable to both classification and regression tasks; and (4) is independent of the underlying induction algorithm used to derive the performance measures when performing feature selection.

5.3. Monte Carlo evaluative selection (MCES)

Monte Carlo simulation dates back to the 1940s when physicists at Los Alamos devised games of chance that they could study to help understand complex physical phenomena relating to the atom bomb [53]. Monte Carlo methods are online simulation methods that learn from experience based on randomly generated simulations, without the need for complete knowledge of the environment. Given a random set of experiences (or trials), the simulation results would eventually converge according to the *central limit theorem* (CLT) [54], when each state is encountered for an infinite number of trials.

The MCES feature selection method is described as follows: given a random set of features, evaluative feedback is collected when one of the feature states is changing from *enabled* (on or present) to *disabled* (off or not present), and vice-versa. This is achieved by using two feature masks. The evaluative feedback is collected for each instance of training examples being presented to the induction algorithm based on the state of the feature sets. On the basis of averaging the evaluative feedbacks collected for a number of trials for that particular feature, the mean value eventually converges. The converged mean value represents the degree of relevancy of that particular feature over other features. The MCES algorithm is attached as Appendix C.

5.4. Methodology

For this simulation, the MCES algorithm is employed to select a set of relevant genes (features) to construct the GenSoFNN-TVR(S) based classifier. A group of such classifiers subsequently functioned as a fuzzy decision support system (GenSo-FDSS) for the classification of ALL cancer subtypes using gene expression data. However, a limitation of the MCES algorithm is that it is essentially a *wrapper-approach* [49] based feature selection technique. That is, it assumes that the underlying induction algorithm has a good representation of the inherent characteristics of the classification task. However, for applications where the number of features is too large (for example, $>$100 features), the induction algorithm may become badly trained, and thus the feature selection process suffers from degradation in accuracy. Therefore, *filter* approaches based on statistical measures such as the *T-statistics* or *Chi-square* (χ^2) metrics [54] could first be used to refine the raw feature set prior to the application of the MCES algorithm. In this way, the search space for the relevant feature genes is greatly reduced. The methodology employed in this application to build the GenSoFNN-TVR(S) based GenSo-FDSS system is depicted as Fig. 7.

The original pediatric ALL cancer dataset from [51] consists of 327 data samples and six ALL cancer subtypes ("BCR-ABL", "E2A-PBX1", "Hyper $>$ 50", "MLL", "T-ALL" and "TEL-AML1"). Samples that cannot be classified into any of the subtypes are collectively grouped into a class labeled as "Others". There are 215 training and 112 testing data samples with 12,558 genes (numeric) characterizing each sample. The training and testing sets and the number of training and testing samples in each cancer subtype had been pre-defined by the contributors of the dataset and is tabulated as Table 1.

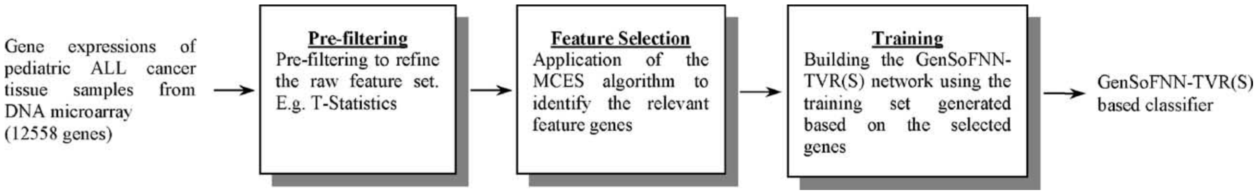
Gene expressions of pediatric ALL cancer tissue samples from DNA microarray (12558 genes)
Pre-filtering
Pre-filtering to refine the raw feature set. E.g. T-Statistics
Feature Selection
Application of the MCES algorithm to identify the relevant feature genes
Training
Building the GenSoFNN-TVR(S) network using the training set generated based on the selected genes
GenSoFNN-TVR(S) based classifier

Figure 7 Methodology to construct the GenSoFNN-TVR(S) based GenSo-FDSS to identify the various acute lymphoblastic leukemia cancer subtypes.

In the experiment, a divide-and-conquer approach is adopted for the classification of the ALL cancers. The proposed GenSo-FDSS cancer classification system consists of six GenSoFNN-TVR(S) based classifiers built to perform the classification, with each classifier entrusted with the task of differentiating the samples from a specific cancer subtype against the rest of the samples. For example, the first classifier is trained to identify the cancer subclass "BCR-ABL" from the rest while the second classifier focuses on the subclass "E2A-PBX1" and so on. Consequently, the GenSoFNN-TVR(S) based GenSo-FDSS cancer classification system implements a diagnostic decision tree (adapted from [3]) as shown in Fig. 8.

Classification is performed using a decision tree format (Fig. 8), in which the first decision is T-ALL versus B lineage (non-T-ALL) samples. Then within the B-lineage subset, samples are sequentially classified into the known subtypes characterized by the presence of E2A-PBX1, TEL-AML1, BCR-ABL, and MLL chimeric genes, and lastly Hyper>50 chromosomes. Samples not identified as belonging to any of the known subtypes are classified as "Others". In the classification process, a classifier is responsible for separating samples of its target class from samples belonging to other classes further down the decision tree.

As previously mentioned, the MCES algorithm employed in this application requires a pre-filtering stage to refine the raw feature set of 12,558 genes prior to the identification of the relevant feature genes. In the experiment, two methods based on *T-statistics* (*T*-Stats) and the *self-organising map*-cum-*discriminant analysis with variance* (SOM/DAV) are adopted for the pre-filtering stage. The genes selected by these two methods are listed as Table S-13 and Table S-15 respectively in the *supplementary data* of the pediatric ALL cancer dataset [51]. Within each set of selected genes, the genes are further categorized into the respective six ALL cancer subtype classes, namely T-ALL, E2A-PBX1, TEL-AML1, BCR-ABL, MLL and Hyper > 50. The member genes within each class are deemed as promising genes with discerning properties that are able to identify samples of their respective target class from the rest of the data samples. Subsequently, the MCES algorithm uses these two sets of genes to identify the relevant feature genes for the ALL cancer classification task. For each subtype class in the two sets of selected genes, the MCES performs 20 iterations using the predefined training samples listed in Table 1. At the end of the 20 iterations, all the genes are ranked according to their computed average evaluative feedback values. After each cycle of 20 iterations, the presentation order of the training samples is reshuffled and the feature selection process repeated. This is performed a total of 10 cycles and the ranking results from the 10 cycles are

Table 1 Number of training and testing samples for each cancer subtype

Group (class)	Number of training samples	Number of testing samples
BCR-ABL	9	6
E2A-PBX1	18	9
Hyper > 50	42	22
MLL	14	6
T-ALL	28	15
TEL-AML1	52	27
Others	52	27
Total	215	112

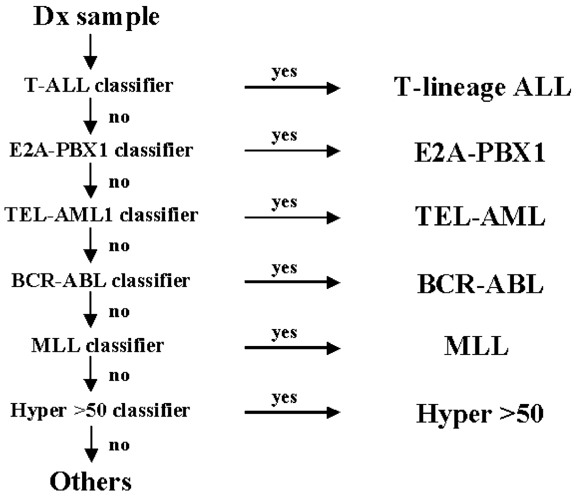

Figure 8 Diagnostic decision tree used to perform classification for the ALL cancer subtypes.

aggregated. In the current implementation, the underlying induction algorithm in MCES is a three-layered neural network (NN) with a structure of $k-2k-1$, where k is the number of promising genes in the feature selection process.

After the feature selection process, the relevant feature genes for each ALL cancer subtype are identified based on their ranking and associated weights (evaluative feedback values). The respective sets of relevant genes for all the six cancer subtypes identified with MCES using the T-statistic and SOM/DAV pre-filtering approach are listed as Appendix D. Each of the six GenSoFNN-TVR(S) based classifiers is subsequently trained with the relevant feature genes identified for that target ALL cancer subtype. The classification results are presented as follows.

6. Results and discussions

For each of the six GenSoFNN-TVR(S) classifiers, the assigned class is identified with an output '1' while the data samples belonging to the classes beneath the target class in the decision tree of Fig. 8 are identified by an output of '0'. All the classifiers are trained with the training sets and classification performances evaluated using the predefined testing sets of Table 1. The presentation order of the data samples in the training sets is shuffled to create ten cross-validation groups (CV1—CV10). The mean classification results of the GenSoFNN-TVR(S) based classifiers across CV1 to CV10 with T-statistics and SOM/DAV as pre-filtering are summarized in the form of *receiver-operating-characteristics* (ROC) curves depicted in Fig. 9. Type I error is defined as the error of falsely rejecting a data sample as not belonging to the assigned class while Type II error is the error of falsely accepting a data sample as belonging to the assigned class. EER denotes the *equal error rates* where Type I equal Type II errors. The ROC plots are obtained by varying the classification thresholds of the classifiers to evaluate the *robustness* and *differentiating capability* of the GenSoFNN-TVR(S) based GenSo-FDSS classification system. From Fig. 9,

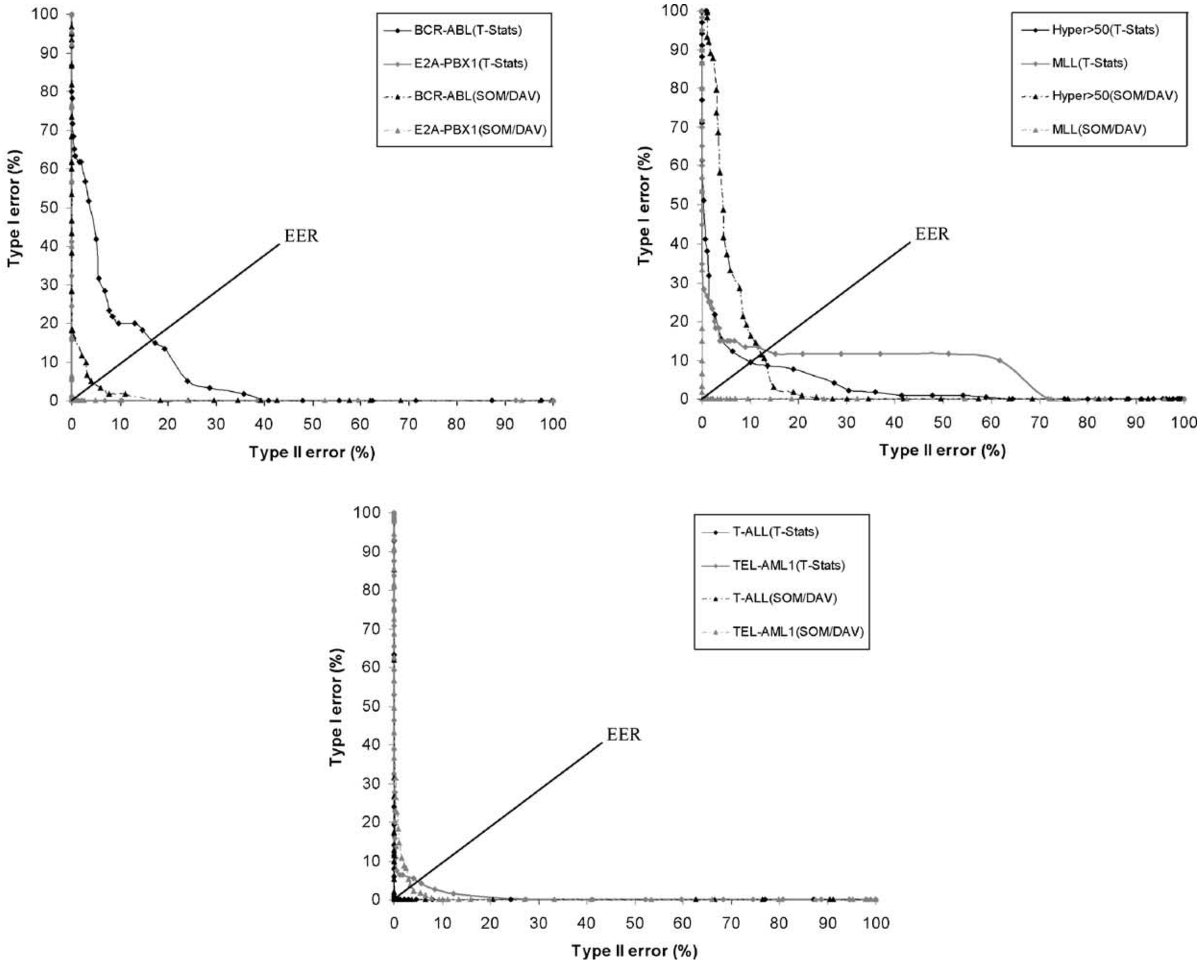

Figure 9 (a) Class 1: BCR-ABL and Class 2: E2A-PBX1; (b) Class 3: Hyper > 50 and Class 4: MLL; and (c) Class 5: T-ALL and Class 6: TEL-AML1 with T-statistics and SOM/DAV as pre-filtering.

one can observed that the classification system built with SOM/DAV as the pre-filtering technique generally outperforms its counterpart built using the *T*-statistics metric. Hence, in this application, SOM/DAV is more efficient than *T*-statistics for pre-filtering prior to the use of MCES to identify the relevant feature genes necessary for the classification task.

The classification task is subsequently repeated using two outputs to benchmark against the performances of the support vector machine (SVM), K-nearest neighbour (K-NN) classifier and neural network as reported in [51]. That is, for each of the six GenSoFNN-TVR(S) classifiers, the assigned class is identified with outputs '10' while the data samples belonging to the classes beneath the target class in the decision tree of Fig. 8 are identified by outputs of '01'.

Hence, output 1 denotes that a data sample belongs to the assigned class while output 2 denotes otherwise. A data sample is classified by determining which of the two outputs has a larger value. The mean classification rates of the GenSoFNN-TVR(S) classifiers across CV1—CV10 are compared against the classification performances of the various benchmarking systems in Table D.3 (Appendix D). In this table, the number of positive (belonging to the assigned class) and negative (not belonging to the assigned class) samples presented to the respective classifier is listed in the squared brackets. *Sensitivity* is defined as 100%-type I error while *specificity* is 100%-type II error. The average number of positive and negative samples misclassified by each of the classifiers across CV1—CV10 is provided (in brackets) next to these measures.

An analysis of Table D.3 in Appendix D shows that the two GenSoFNN-TVR(S) based GenSo-FDSS cancer classification systems built using *T*-statistics and SOM/DAV as pre-filtering techniques have comparable (and in some subtypes better) performances to the SVM, K-NN and ANN for the cancer subtype T-ALL, E2A-PBX1, BCR-ABL and MLL. However, slightly poorer classification results are observed in subtype TEL-AML1 and Hyper $>$ 50. This could be attributed to the number of relevant genes used to perform the classification for these two cancer subtypes. The GenSoFNN-TVR(S) classifiers for Hyper $>$ 50 and TEL-AML1 with *T*-statistics for pre-filtering are each trained using five relevant genes while their counterparts with SOM/DAV are trained with 10 and 11 relevant genes respectively. On the other hand, the ANN classifiers used 50 genes identified with the Wilkins' selection metric [51] to perform the classification task while the number of genes used by the SVM and K-NN classifiers were not reported. Nevertheless, these three classification systems functioned as black boxes. There is no knowledge on how the computed decisions are derived. The GenSoFNN-TVR(S) based GenSo-FDSS systems, on the other hand, has the capability to solicit inherent knowledge from the numerical gene expression dataset and express the characteristics (of relevant genes) of each cancer subtypes in a form easily accessed and comprehended by the human users (i.e. the IF-THEN fuzzy rules).

6.1. Fuzzy rules analysis

As mentioned earlier, a key advantage of using the GenSo-FDSS cancer classification system to identify the pediatric ALL cancer subtypes (instead of other techniques such as SVM and ANN) is the ability to extract intuitive fuzzy rules from the trained structures of its component GenSoFNN-TVR(S) networks. To demonstrate the comprehensiveness of the formulated fuzzy rules, the rule-base of the GenSoFNN-TVR(S) classifier for the cancer subtype BCR-ABL trained using the first cross-validation group (CV1) with *T*-statistics for pre-filtering is analyzed. The computed fuzzy sets for each of the five genes identified for BCR-ABL (see Table D.1) are depicted as Fig. 10. Using the fuzzy labels and the associated semantic meanings depicted in Fig. 10, some of the formulated fuzzy rules are extracted from the structure of the GenSoFNN-TVR(S) network assigned to the classification of BCR-ABL and trained with the training set of CV1. These rules are listed as Table 2. *Positive rules* referred to the rules used to classify a data sample as belonging to subtype BCR-ABL while *negative rules* denote the fuzzy rules that classify a data sample as not belonging to BCR-ABL. In addition, the aggregated firing strengths of the respective fuzzy rules during the training process are provided for analysis.

From Table 2, one can conclude that a sample belonging to the ALL cancer subtype BCR-ABL generally has *medium* expression values for the five identified relevant genes. On the other hand, a data sample not belong to BCR-ABL generally has *low* to *minute* expression values for these genes. Thus, such fuzzy rules aid in the data interpretation of the pediatric ALL cancer dataset and also explain the classification decisions of the GenSoFNN-TVR(S) classifiers. As knowledge is the basis for making decisions, the ability to formulate and extract knowledge from a given dataset is a key advantage over other types of classifiers such as the SVM, ANN and K-NN.

6.2. Stress and hypothetical scenario reasoning

Another prominent feature of the GenSoFNN-TVR(S) based GenSo-FDSS pediatric ALL cancer

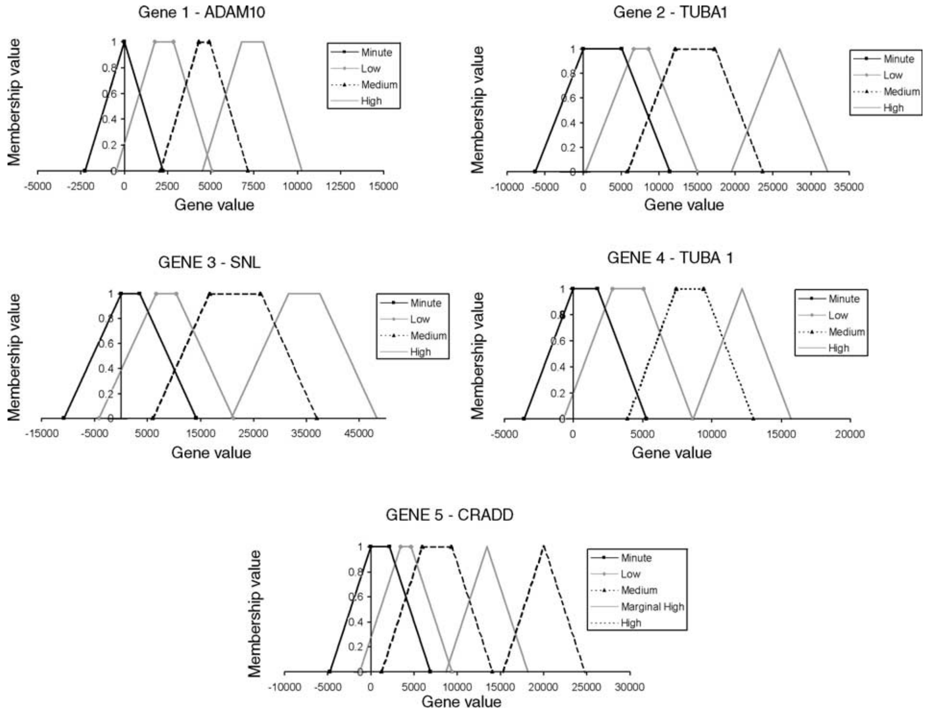

Figure 10 Computed fuzzy sets of the GenSoFNN-TVR(S) classifier for the subclass "BCR-ABL" of the pediatric ALL cancer dataset with *T*-statistics as the pre-filtering technique.

classification system is its ability to perform *stress* and *hypothetical scenario* reasoning by modifying the fuzzy rules using *linguistic hedges* [13]. In this experiment for example, applying the linguistic hedges "*very*" and "*rather*" uniformly to the fuzzy labels of the fuzzy rules simulates stress scenarios. The hedge "*very*" creates *positive stress* by strongly enforcing the conditions leading to the firing of the rules while the hedge "*rather*" gives rise to *negative stress* as it relaxes the conditions for rule firing. On the other hand, numerous hypothetical scenarios can be constructed/simulated by applying a mixture of the "*very*" and "*rather*" linguistic hedges to the labels of the fuzzy rules. More details are reported in [55]. The effects of the linguistic hedges "*very*" and "*rather*" on an input fuzzy label $\mathrm{IL}_{i,j}$ are defined as in Eq. (19) and depicted as Fig. 11. The effects on the output fuzzy labels are similar.

$$\begin{aligned} \textit{very}(\mathrm{IL}_{i,j}) &= (\mathrm{IL}_{i,j})^2 = [\mu_{\mathrm{IL}_{i,j}}(x_i)]^2 \\ \textit{rather}(\mathrm{IL}_{i,j}) &= (\mathrm{IL}_{i,j})^{1/2} = [\mu_{\mathrm{IL}_{i,j}}(x_i)]^{1/2} \end{aligned} \tag{19}$$

where $\mu_{\mathrm{IL}_{i,j}}$ is the membership function of the input fuzzy label $\mathrm{IL}_{i,j}$.

Hence, the linguistic hedges "*very*" and "*rather*" are shown to have similar effects as

Table 2 Fuzzy rules extracted from the structure of the GenSoFNN-TVR(S) network assigned to the classification of BCR-ABL data samples using the training set of CV1

Positive rules

Rule 1: IF gene1 value is *low* and gene2 value is *medium* and gene3 value is *medium* and gene4 value is *medium* and gene5 value is *medium*
THEN data sample is *BCR-ABL* (firing strength = 69.2067)

Rule 2: IF gene1 value is *medium* and gene2 value is *medium* and gene3 value is *medium* and gene4 value is *medium* and gene5 value is *low*
THEN data sample is *BCR-ABL* (firing strength = 65.4364)

Negative rules

Rule 1: IF gene2 value is *minute*
THEN data sample is *not BCR-ABL* (firing strength = 63.4209)

Rule 2: IF gene1 value is *low* and gene2 value is *low* and gene3 value is *low* and gene4 value is *low* and gene5 value is *minute*
THEN data sample is *not BCR-ABL* (firing strength = 63.0388)

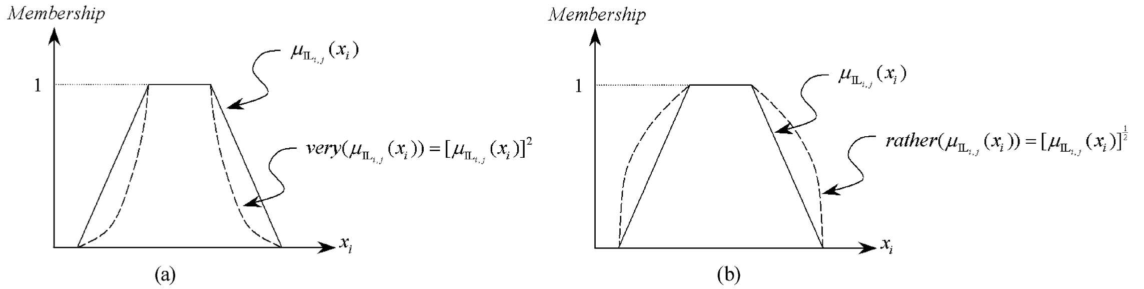
Membership
1
$\mu_{\mathrm{IL}_{i,j}}(x_i)$
$very(\mu_{\mathrm{IL}_{i,j}}(x_i)) = [\mu_{\mathrm{IL}_{i,j}}(x_i)]^2$
x_i
(a)
Membership
1
$\mu_{\mathrm{IL}_{i,j}}(x_i)$
$rather(\mu_{\mathrm{IL}_{i,j}}(x_i)) = [\mu_{\mathrm{IL}_{i,j}}(x_i)]^{\frac{1}{2}}$
x_i
(b)

Figure 11 Applying linguistic hedges on an input fuzzy label $\mathrm{IL}_{i,j}$: (a) *very* $\mathrm{IL}_{i,j}$ and (b) *rather* $\mathrm{IL}_{i,j}$.

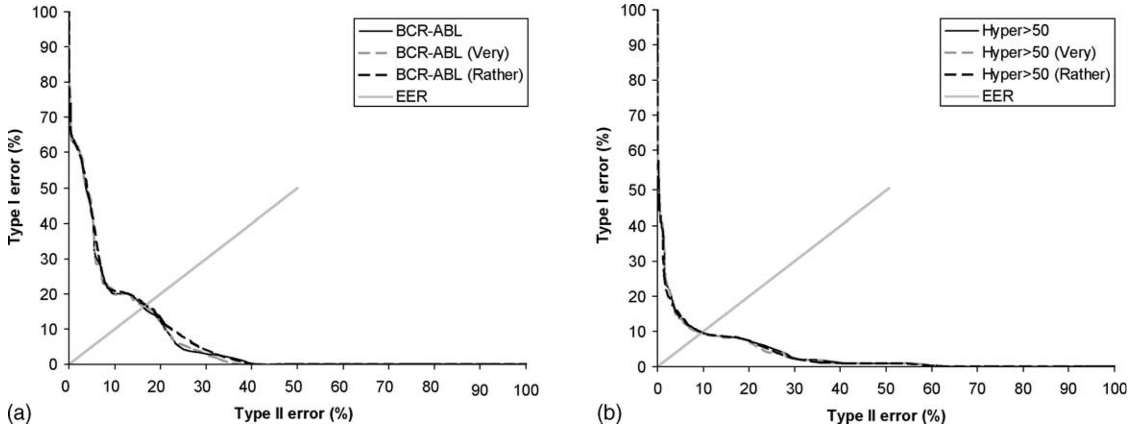

Figure 12 Error bounds of the ROC classification curves for cancer subtype: (a) "BCR-ABL" and (b) "Hyper $>$ 50" computed using linguistic hedges "*very*" and "*rather*" with CV1 and *T*-statistics.

concentration and *dilation* on the fuzzy labels respectively. In this application, positive stress as induced by applying the hedge "*very*" to the fuzzy labels forms the upper (lower) bound of the classification (error) rates of the GenSoFNN-TVR(S) classifiers. This is because enforcing the conditions leading to the firing of the fuzzy rules reduces the classification errors between closely similar classes. Similar, negative stress as induced by applying the linguistic hedge "*rather*" is perceived as defining the lower (upper) bound for the classification (error) rates as it introduces a greater overlap of the fuzzy regions between the fuzzy sets, thus resulting in more errors. Due to space constraint, only the error bounds for the classification of cancer subtypes "BCR-ABL" and "Hyper $>$ 50" are presented in this paper. The classification error bounds for cancer subtype "BCR-ABL" and "Hyper $>$ 50" computed using the linguistic hedges "*very*" and "*rather*" are depicted as Fig. 12. The GenSoFNN-TVR(S) classifiers are trained with the training set of CV1 and employed *T*-statistics as the pre-filtering technique. The error bounds for the other cancer subtypes can be evaluated using the same approach and numerous hypothetical scenarios can be simulated by applying a mixture of the hedges "*very*" and "*rather*" to the fuzzy labels of the formulated fuzzy rules. For the interested reader, please refer to [55].

7. Conclusions

This paper proposed the use of a novel neural fuzzy system: the GenSoFNN-TVR(S) network, as a fuzzy decision support system (denoted as GenSo-FDSS) for the classification of ALL cancer subtypes using gene expression data extracted from DNA microarrays. The GenSoFNN-TVR(S) network is synthesized by mapping the *truth-value restriction* (TVR) [25] fuzzy inference scheme onto the generic structure of the *generic self-organizing fuzzy neural network* (GenSoFNN) [26] architecture. The TVR inference scheme operates in the truth-space and uses numeric or linguistic truth-value functions such as *TRUE*, More or less *TRUE* and *FALSE* to denote the *degree of possibility of truth* of the fuzzy propositions in the IF-THEN fuzzy rules during the fuzzy inference process. The TVR inference scheme corresponds closely to the human reasoning process.

Hence, the use of TVR to define the node operations of the GenSoFNN-TVR(S) network provides it with a firm fuzzy logic foundation and a logical deduction process.

In addition, being a neural fuzzy system, the GenSoFNN-TVR(S) network has the ability to solicit knowledge from the numerical training data and express this knowledge in an easily accessible form by means of the IF-THEN fuzzy rules. This enables the human users of the GenSo-FDSS system to understand the classification decisions computed and the traits characterizing each of the ALL cancer subtypes. Such knowledge discovery/data mining capability of the GenSo-FDSS system is important as (1) it aids the *data interpretation* and analysis of the numerical dataset; (2) it may uncover novel knowledge related to the characteristics of the ALL cancer subtypes previously unknown to the doctors/clinicians (*knowledge bridging*); (3) it could identify new areas/characteristics of the various cancers against which novel therapeutics may be developed/designed to enhance the survival/recovery rates of cancer patients (*new treatment protocol*). Due to the large number of features encountered in the gene expression dataset, a newly developed feature selection algorithm named *Monte Carlo evaluative selection* (MCES) [52] is employed to identify the relevant genes used to classify data samples into their corresponding ALL subtypes. Extensive experimental results clearly showed that the performance of the GenSo-FDSS system is comparable to those of traditional classifiers such as *neural network* (NN), K-*nearest-neighbor* (K-NN) and the *support vector machine* (SVM) albeit using a much smaller set of features (genes).

Currently, extensive efforts have been invested at the Centre for Computational Intelligence (C2i) [56], School of Computer Engineering, Nanyang Technological University (Singapore), to further improve the classification rates and reduce the errors of the GenSoFNN-TVR(S) based pediatric ALL cancer classification decision support system. A *non-singleton fuzzifier* based GenSoFNN-TVR(S) network for the same purpose is also currently under investigation. The C2i undertakes intense research in the study and development of advanced hybrid neural fuzzy architectures [16—18,26,57] for the modeling of *complex*, *dynamic* and *non-linear* systems. These techniques have been successfully applied to numerous novel applications such as automated driving [58], signature forgery detection [59], gear control for *continuous variable transmission* (CVT) in car system [60], fingerprint verification [61] and bank failure classification and early-warning system (EWS) [62].

Appendix A

The truth-value restriction (TVR) fuzzy inference scheme

To illustrate the concepts of the TVR inference scheme, assume the following simple *single-input-single-output* (SISO) fuzzy system:

$$\text{fuzzy rule}: \quad \text{If} \overbrace{\underbrace{x \text{ is } A}_{\text{fuzzy proposition } p}}^{\text{antecedent}} \text{Then} \overbrace{\underbrace{y \text{ is } B}_{\text{fuzzy proposition } q}}^{\text{consequent}}$$

$$\text{observed input}: \quad \underbrace{x \text{ is } \tilde{A}}_{\text{fuzzy proposition } p'}$$

The TVR inference scheme uses the *inverse truth function modification* (ITFM) process to compute a function $\tau_{A\tilde{A}}$ that would transform the fuzzy proposition p to p'. In order words, $\tau_{A\tilde{A}}$ would modify the fuzzy set **A** to $\tilde{\mathbf{A}}$. Thus, $\tau_{A\tilde{A}}$ denotes the truth-value of the proposition p given p'. That is, the degree of possibility that the proposition p is true given the observed proposition p'. The function $\tau_{A\tilde{A}}$ is in fact a fuzzy set in the *truth space* and is defined as

$$\tau_{A\tilde{A}}(a) = \begin{cases} \sup\limits_{x}\{\mu_{\tilde{A}}(x) | x \in \mu_{A}^{-1}(a)\}, & \mu_{A}^{-1}(a) \neq \phi \\ 0, & \text{otherwise} \end{cases} \tag{A.1}$$

where $\mu_{\mathbf{A}}$ and $\mu_{\tilde{A}}$ are the respective membership functions of the fuzzy set **A** and $\tilde{A}$ defined on the universe of discourse **X**; x denotes a value in **X**; a is the membership value of x in fuzzy set **A**; $\mu_{A}^{-1}(a)$ is the set of values of x in **X** that take membership value a in the fuzzy set **A**; and ϕ denotes the empty or null set. Fig. A.1 shows some of the commonly defined truth-value functions in the truth space.

Hence, based on the computed truth-function $\tau_{A\tilde{A}}$, the SISO fuzzy system can be re-expressed as:

$$\begin{array}{lll} \text{Fuzzy rule:} & \text{If} \overbrace{\underbrace{x \text{ is } \mathbf{A}}_{\text{fuzzy proposition } p}}^{\text{Antecedent}} \text{Then} \overbrace{\underbrace{y \text{ is } \mathbf{B}}_{\text{fuzzy proposition } q}}^{\text{Consequent}} \\ \text{Observed input:} & \left[\underbrace{x \text{ is } \mathbf{A}}_{\text{fuzzy proposition } p}\right] \text{is } \tau_{\mathrm{A\tilde{A}}} \triangleq \underbrace{x \text{ is } \tilde{\mathbf{A}}}_{\text{fuzzy proposition p'}} \\ \hline \text{Conclusion:} & \underbrace{y \text{ is } \tilde{\mathbf{B}}}_{\text{inferred fuzzy proposition q'}} \triangleq \left[\underbrace{y \text{ is } \mathbf{B}}_{\text{fuzzy proposition q}}\right] \text{is } \tau_{\mathrm{B\tilde{B}}} \end{array}$$

where $\tau_{B\tilde{B}}$ is the truth-value of the consequent "*y* is **B**" based on the inferred output "*y* is $\tilde{\boldsymbol{B}}$".

The *generalized modus ponens* (GMP) [13] reasoning rule is subsequently used to infer the corresponding conclusion from the SISO fuzzy system by means of the TVR inference scheme. The truth-value $\tau_{A\tilde{A}}$ of the antecedent propagates through the fuzzy deduction process of the TVR inference scheme to determine the corresponding truth-value $\tau_{B\tilde{B}}$ of the consequent "y is **B**". Thus, in terms of mathematical expressions, the TVR based GMP inference process is defined as:

$$\begin{array}{ll} \text{Fuzzy rule:} & (p \Rightarrow q) \text{ is } \tau_{\text{Implication}} \\ \text{Observed input:} & p \text{ is } \tau_{\mathrm{A\tilde{A}}} \\ \hline \text{Conclusion:} & q \text{ is } \tau_{\mathrm{B\tilde{B}}} = \tau_{\mathrm{A\tilde{A}}}(a) \circ \tau_{\text{Implication}}(I(a,b)) \end{array}$$

where $\tau_{\text{Implication}}$ is the truth-value of the fuzzy rule induced by the implication function adopted; and a and b are the truth-value of the propositions p and q, respectively. In a neural fuzzy implementation, propositions p and q are induced by the data clusters they represented. Thus, they are supported by strong evidence and hence must be *TRUE* (Fig. A.1). Therefore, a and b are implicitly the membership values of fuzzy sets **A** and **B**. That is, $a = \mu_A(x)$ and $b = \mu_B(y)$. When the sup-T operation is used to resolve the composition of the observed input and the fuzzy rule [63], the truth-value $\tau_{B\tilde{B}}$ of the consequent "y is **B**" can be computed using Eq. (A.2):

$$\tau_{B\tilde{B}}(b) = \sup_a \{m_I[\tau_{A\tilde{A}}(a), \tau_{\text{Implication}}(I(a,b))]\} \qquad \text{(A.2)}$$

where m_I is the forward reasoning function [64] (usually a T-norm operation); and I is the implication rule [65]. The truth-value $\tau_{B\tilde{B}}$ is subsequently "inverted" to determine the inferred conclusion. That is, a fuzzy proposition "y is $\tilde{\mathbf{B}}$" is computed using the *truth function modification* (TFM) process such that the degree of possibility that the proposition "y is **B**" is true given "y is $\tilde{\mathbf{B}}$" is described by the truth-value $\tau_{B\tilde{B}}$. That is, the derivation of the fuzzy set $\tilde{\mathbf{B}}$ from the truth-value $\tau_{B\tilde{B}}$ is performed using Eq. (A.3):

$$\begin{aligned} &\tilde{\mathbf{B}} = \tau_{B\tilde{B}} \circ \mathbf{B} \\ &\Rightarrow \underbrace{b'}_{\mu_{\tilde{B}}(y)} = \tau_{B\tilde{B}}(\underbrace{b}_{\mu_B(y)}) \\ &\Rightarrow \mu_{\tilde{B}}(y) = \tau_{B\tilde{B}}(\mu_B(y)) \end{aligned} \qquad \text{(A.3)}$$

where b' and b are the respective truth-values of the propositions "y is $\tilde{\mathbf{B}}$" and "y is **B**"; and μ_B and $\mu_{\tilde{B}}$ are the membership functions of the fuzzy sets **B** and $\tilde{\mathbf{B}}$, respectively.

One of the main advantages of using the TVR inference scheme is its ability to handle *uncertainty* in the inference process in an intuitive manner. Uncertainty may be caused by *incomplete* or *imprecise* information. TVR uses fuzzy truth-values such as *TRUE* and *Very TRUE* to describe the degree of possibility that a proposition is true (or false). This makes the reasoning process human-like and can be easily comprehended by the human users. As compared to the CRI inference scheme, TVR performs in a more logical manner when implemented in a *multiple-input-single-output* (MISO) fuzzy system. This form of fuzzy system is the most commonly encountered, as a SISO system is simple and has limited usage while a *multiple-input-multiple-output* (MIMO) fuzzy system is essentially an aggregation of multiple MISO systems. To illustrate the intuitiveness of the TVR over the CRI inference scheme in a MISO system, consider the following simple 2-input-1-output fuzzy system with a single fuzzy rule:

$$\text{fuzzy rule}: \text{ If } \overbrace{\underbrace{x \text{ is } A}_{\text{fuzzy proposition } p}}^{\text{antecedent 1}} \text{ and } \overbrace{\underbrace{y \text{ is } B}_{\text{fuzzy proposition } q}}^{\text{antecedent 2}}$$

$$\text{Then } \overbrace{\underbrace{z \text{ is } C}_{\text{fuzzy proposition } r}}^{\text{consequent}}$$

Under the TVR framework and the knowledge that the above fuzzy rule is supported by the existence of data clusters representing the problem domain, one can easily deduced that the truth-values τ_A, τ_B and τ_C of the propositions p, q and r are all *TRUE*. That is,

$$\tau_A(a) = a = \mu_A(x); \qquad \tau_B(b) = b = \mu_B(y); \quad \text{and} \quad \tau_C(c) = c = \mu_C(z) \qquad \text{(A.4)}$$

where μ_A, μ_B and μ_C are the membership functions of the fuzzy sets **A**, **B** and **C**, respectively.

During the inference process based on the GMP reasoning rule, one needs to compute the *conjunction* of the multiple antecedents (two in this case). With TVR, this can be easily performed using Eq. (A.5):

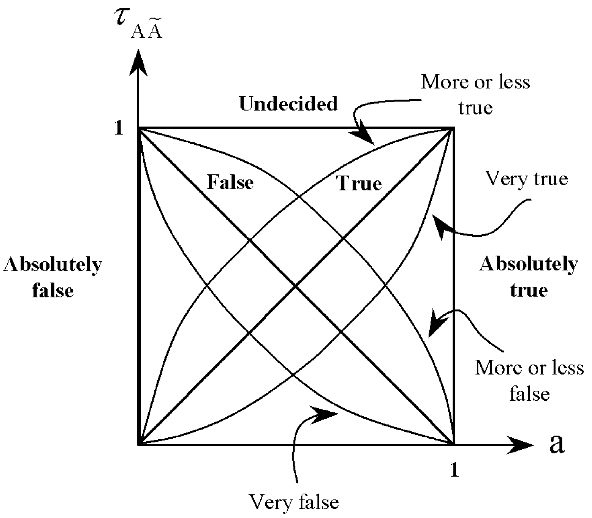

Figure A.1 Commonly defined truth-value functions.

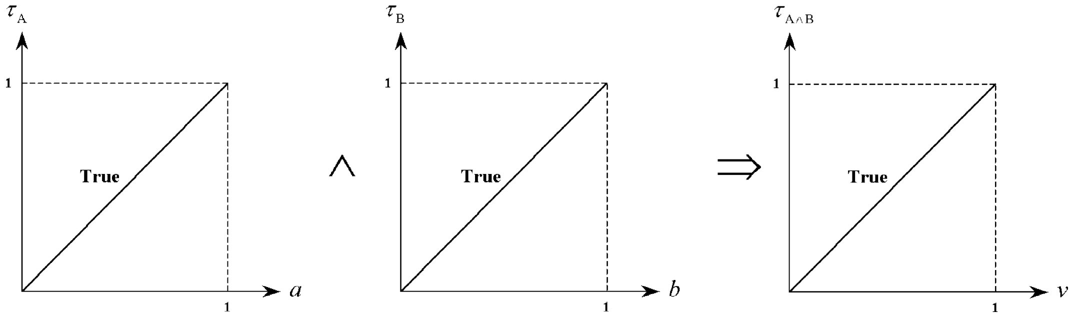

Figure A.2 Deriving the truth-value $\tau_{A\wedge B}$ of the conjunction of two propositions with truth-values as *true*.

$$\begin{aligned}&(p \text{ is } \tau_A) \wedge (q \text{ is } \tau_B)\\ &\Rightarrow (p \wedge q) \text{ is } \tau_{A\wedge B}\end{aligned} \tag{A.5}$$

where $\tau_{A\wedge B}$ is the truth-value associated with the conjunction of the propositions p and q. This truth-value is defined in [66] as:

$$\begin{aligned}\tau_{A\wedge B}(v) = \max[&\min[\max_{a\in[v,1]} \tau_A(a), \tau_B(v)], \min[\tau_A(v),\\ &\max_{b\in[v,1]} \tau_B(b)]], \quad v \in [0,1]\end{aligned} \tag{A.6}$$

Thus, the truth-value of the conjunction of the propositions p and q, $\tau_{A\wedge B}$, is deduced to be *TRUE*. This is illustrated as Fig. A.2.

This form of deduction is very logical and intuitive to the human reasoning process, as one would expect to get a deduction of *TRUE* if two propositions of truth-values *TRUE* are combined. Similarly, using the truth-value deduction process of the TVR inference scheme, one would get *FALSE* from the conjunction of a *TRUE* proposition and a *FALSE* proposition and *More or less TRUE* from the conjunction of a *TRUE* proposition and a *More or less TRUE* proposition. Fig. A.3 illustrates this.

In comparison, if one uses the CRI inference scheme and computes the conjunction of the antecedents or input propositions (p and q in this case) of a fuzzy rule using the respective fuzzy sets, then the result would be a fuzzy set binding the input values. This is,

$$\begin{aligned}&p \wedge q\\ &\Rightarrow v = T(a, b)\\ &\Rightarrow \mu_{A\wedge B}(x, y) = T(\mu_A(x), \mu_B(y))\end{aligned} \tag{A.7}$$

where v is $\mu_{A\wedge B}(x, y)$, a is $\mu_A(x)$ and b is $\mu_B(y)$ under the CRI inference framework; and $\mu_A(x)$ and $\mu_B(y)$

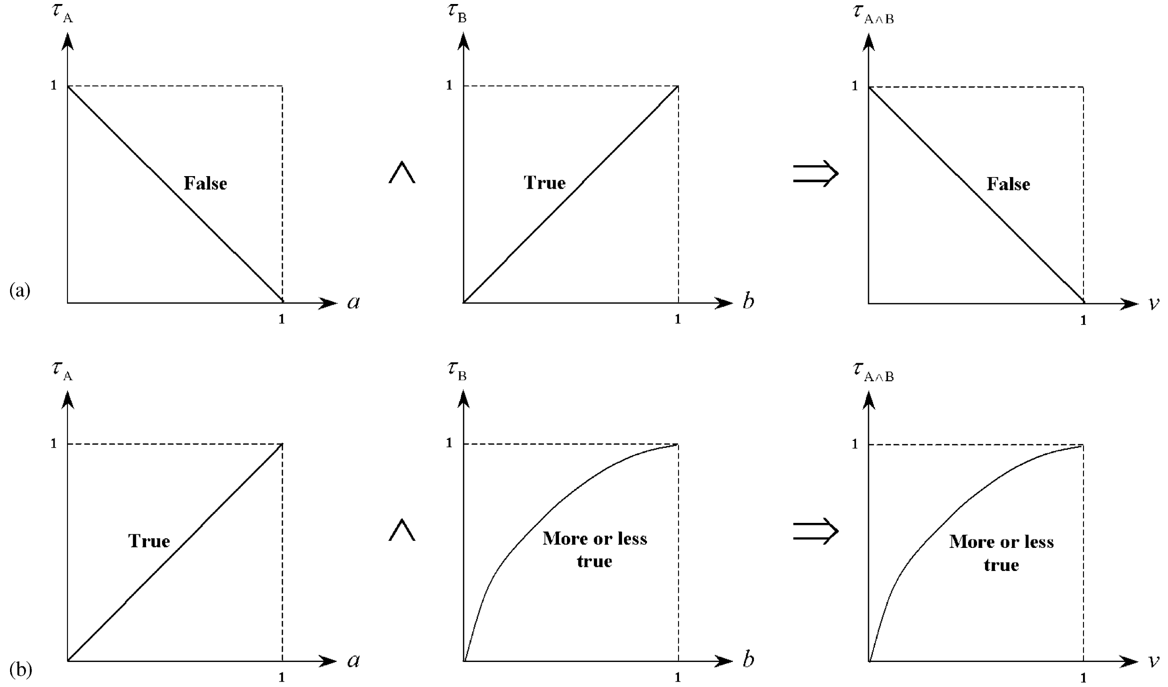

Figure A.3 Truth-value computation based on conjunction of propositions: (a) *false* and *true* gives *false*; (b) *true* and *more or less true* gives *more or less true*.

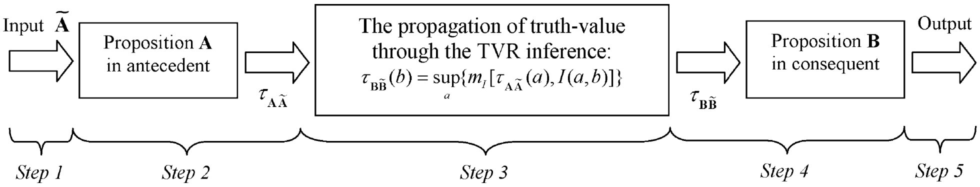
Input $\tilde{\mathbf{A}}$
Proposition **A** in antecedent
$\tau_{\mathbf{A}\tilde{\mathbf{A}}}$
The propagation of truth-value through the TVR inference:
$\tau_{\mathbf{B}\tilde{\mathbf{B}}}(b) = \sup_a\{m_I[\tau_{\mathbf{A}\tilde{\mathbf{A}}}(a), I(a,b)]\}$
$\tau_{\mathbf{B}\tilde{\mathbf{B}}}$
Proposition **B** in consequent
Output $\tilde{\mathbf{B}}$
Step 1
Step 2
Step 3
Step 4
Step 5

Figure A.4 The TVR inference process.

are the membership functions of the fuzzy sets **A** and **B**, respectively.

Thus, the conjunction of fuzzy sets **A** and **B** could result in an un-normalized and irregularly shaped fuzzy set defined on the Cartesian space of **X** × **Y**. Comparing the two inference schemes in a MISO system, the TVR inference scheme appeared more logical and is more intuitive to the human reasoning process than the CRI inference scheme, albeit it being more complex due to the computations performed in the truth-space. Fig. A.4 illustrates the TVR inference process using the GMP reasoning rule.

The TVR inference process can be intuitively partitioned into five steps as shown in Fig. A.4. These five steps can be conveniently mapped onto each of the five layers of the proposed GenSoFNN-TVR(S) network: layer 1, singleton fuzzifier; layer 2, antecedent matching; layer 3, rule fulfillment; layer 4, consequent derivation; and layer 5, defuzzification. The process of mapping the TVR inference scheme onto the GenSoFNN-TVR(S) network to define its nodal operations is presented as Appendix B.

Appendix B

Mapping of the TVR inference scheme onto the GenSoFNN-TVR(S) network

As a neural realization of a *TVR-based fuzzy system* with singleton fuzzifiers (henceforth denoted as TVR-FS), the operations and outputs of the nodes in the proposed GenSoFNN-TVR(S) network are derived from the inference steps of such a fuzzy system. Each layer of the GenSoFNN-TVR(S) network is mapped to a corresponding inference step in the TVR-FS. This section presents the respective inference steps of the TVR-FS and demonstrates the functional equivalence between the operations of the GenSoFNN-TVR(S) network and that of the TVR-FS. With reference to Fig. A.4, the operations of the TVR inference scheme are partitioned into five steps. Assume that the general-purposed TVR-based fuzzy system (TVR-FS) has n1 inputs and n5 outputs. Fig. B.1 illustrates the various functional blocks of such a system.

B.1. Step 1—input fuzzification

Since the inputs to the TVR-FS are crisp-valued, fuzzification of the inputs has to be performed before the inference engine (Fig. B.1) can make use of the fuzzified inputs to compute the appropriate fuzzified outputs. The vector **X** = $[x_1, \ldots, x_i, \ldots, x_{n1}]^{\mathrm{T}}$ denotes the inputs to the TVR-FS. For input x_i, it is fuzzified into its corresponding fuzzy set $\bar{\mathrm{X}}_i$ using the singleton fuzzifier defined in Eq. (7). The operation of the singleton fuzzifier is subsequently mapped onto layer 1 of the GenSoFNN-TVR(S) network using Eqs. (5) and (6). Hence, the input vector **X** = $[x_1, \ldots, x_i, \ldots, x_{n1}]^{\mathrm{T}}$ becomes $\bar{\mathbf{X}} = [\bar{\mathrm{X}}_1, \ldots, \bar{\mathrm{X}}_i, \ldots, \bar{\mathrm{X}}_{n1}]^{\mathrm{T}}$.

B.2. Step 2—antecedent matching

The fuzzified inputs from Step 1 are then compared against their corresponding input labels that form the antecedent section of the fuzzy rules in the TVR-FS. For fuzzified input $\bar{\mathrm{X}}_i$, its corresponding *j*th label is denoted as $\mathrm{IL}_{i,j}$. Both $\bar{\mathrm{X}}_i$ and $\mathrm{IL}_{i,j}$ are fuzzy sets defined on the space of the *i*th input dimension. Under the TVR inference framework, the antecedent matching between $\bar{\mathrm{X}}_i$ and $\mathrm{IL}_{i,j}$ computes the truth-value of $\mathrm{IL}_{i,j}$ given the input $\bar{\mathrm{X}}_i$ (denoted as $\tau_{\mathrm{IL}_{i,j}\bar{\mathrm{X}}_i}$):

$$\tau_{\mathrm{IL}_{i,j}\bar{\mathrm{X}}_i}(a) = \begin{cases} \sup\limits_{\bar{x}_i}\left\{\mu_{\bar{\mathrm{X}}_i}(\bar{x}_i)\middle|\bar{x}_i = x_i \in \mu^{-1}_{\mathrm{IL}_{i,j}}(a)\right\}, & \mu^{-1}_{\mathrm{IL}_{i,j}}(a) \neq \phi \\ 0, & \text{otherwise} \end{cases} \tag{B.1}$$

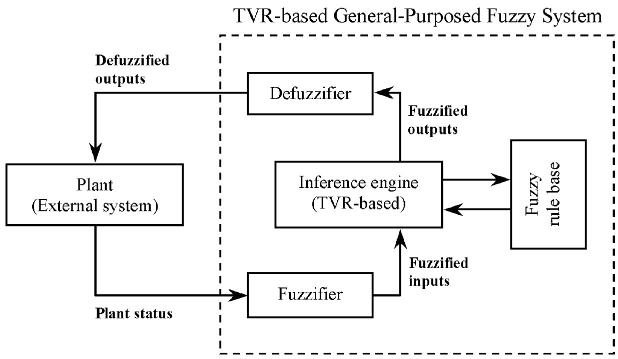

Figure B.1 A general-purposed fuzzy system with TVR-based inference.

where $\mu_{\bar{\mathrm{X}}_i}$ and $\mu_{\mathrm{IL}_{i,j}}$ are the membership functions of $\bar{\mathrm{X}}_i$ and $\mathrm{IL}_{i,j}$ respectively. Notice that Eqs. (A.1) and (B.1) are similar. Since $\bar{\mathrm{X}}_i$ is a singleton fuzzy set, $\tau_{\mathrm{IL}_{i,j}\bar{\mathrm{X}}_i}$can be simplified:

$$\tau_{\mathrm{IL}_{i,j}\bar{\mathrm{X}}_i}(a) = \begin{cases} 1, & a = \mu_{\mathrm{IL}_{i,j}}(\bar{x}_i) = \mu_{\mathrm{IL}_{i,j}}(x_i) \in [0,1] \\ 0, & \text{otherwise} \end{cases} \tag{B.2}$$

The computation of a in Eq. (B.2) is realized by the operations of layer 2 nodes defined by Eqs. (8)—(10). Thus, the computation of the truth-value $\tau_{\mathrm{IL}_{i,j}\bar{\mathrm{X}}_i}$ is implicitly mapped onto layer 2 of the proposed GenSoFNN-TVR(S) network.

B.3. Step 3—rule fulfillment

This step is to compute the *overall* truth-value of the fuzzy propositions in the antecedent of a fuzzy rule R_k given the fuzzified input $\bar{\mathbf{X}} = [\bar{\mathrm{X}}_1, \ldots, \bar{\mathrm{X}}_i, \ldots, \bar{\mathrm{X}}_{n1}]^{\mathrm{T}}$ using the concept of *conjunction* (of two or more fuzzy propositions) as introduced in Appendix A. Hence, from Eqs. (A.5) and (A.6), one can easily deduced that

$$\tau_{\{\mathrm{IL}_{(1,j)_k}\wedge\cdots\wedge\mathrm{IL}_{(i,j)_k}\wedge\cdots\wedge\mathrm{IL}_{(n1,j)_k}\}\bar{\mathrm{X}}}(a) = \begin{cases} 1, & a = T\Big(\mu_{\mathrm{IL}_{(1,j)_k}}(x_1), \ldots, \mu_{\mathrm{IL}_{(i,j)_k}}(x_i), \ldots, \mu_{\mathrm{IL}_{(n1,j)_k}}(x_{n1})\Big) \in [0,1] \\ 0, & \text{otherwise} \end{cases} \tag{B.3}$$

where $\tau_{\{\mathrm{IL}_{(1,j)_k}\wedge\cdots\wedge\mathrm{IL}_{(i,j)_k}\wedge\cdots\wedge\mathrm{IL}_{(n1,j)_k}\}\bar{\mathrm{X}}}$ is the overall truth-value of the aggregated fuzzy propositions in the rule R_k given the fuzzified input $\bar{\mathbf{X}} = [\bar{\mathrm{X}}_1, \ldots, \bar{\mathrm{X}}_i, \ldots, \bar{\mathrm{X}}_{n1}]^{\mathrm{T}}$; and $T(\cdot)$ denotes any T-norm operation. Subsequently, Eqs. (11) and (12) are derived if the computationally simple *min* operator is used to define the T-norm operation of Eq. (B.3). Therefore, the computation of the overall truth-value $\tau_{\{\mathrm{IL}_{(1,j)_k}\wedge\cdots\wedge\mathrm{IL}_{(i,j)_k}\wedge\cdots\wedge\mathrm{IL}_{(n1,j)_k}\}\bar{\mathrm{X}}}$ is again implicitly mapped to the operations of the layer 3 nodes in the proposed GenSoFNN-TVR(S) network. Notice that $\tau_{\{\mathrm{IL}_{(1,j)_k}\wedge\cdots\wedge\mathrm{IL}_{(i,j)_k}\wedge\cdots\wedge\mathrm{IL}_{(n1,j)_k}\}\bar{\mathrm{X}}}$ is actually a singleton when the inputs to the GenSoFNN-TVR(S) network are crisp-valued.

B.4. Step 4—consequent derivation

In Step 4, the consequences of firing the fuzzy rules in the TVR-FS are determined. The proof of the functional equivalence between layer 4 of the proposed GenSoFNN-TVR(S) network and the consequent derivation phase of the TVR-FS is more complex than the previous three steps. The label $\mathrm{OL}_{(l,m)_k}$ denotes the lth fuzzy label of the mth output ($\mathrm{OL}_{l,m}$) that forms part of the consequent of rule R_k. Hence, assuming that the inferred fuzzified output $\tilde{\mathrm{OL}}_{(l,m)_k}$ due to the firing of rule R_k is already known, the truth-value for $\mathrm{OL}_{(l,m)_k}$ (given the deduced output $\tilde{\mathrm{OL}}_{(l,m)_k}$) based on the GMP reasoning rule in Appendix A and the reasoning process as defined by Eq. (A.2) is computed as:

$$\tau_{\mathrm{OL}_{(l,m)_k}\tilde{\mathrm{OL}}_{(l,m)_k}}(b) = \sup_a \Big\{ m_I \Big[\tau_{\{\mathrm{IL}_{(1,j)_k}\wedge\ldots\wedge\mathrm{IL}_{(i,j)_k}\wedge\ldots\wedge\mathrm{IL}_{(n1,j)_k}\}\bar{\mathrm{X}}}(a), \tau_{\mathrm{Implication}}(I_k(a,b)) \Big] \Big\} \tag{B.4}$$

where $\tau_{\mathrm{OL}_{(l,m)_k}\tilde{\mathrm{OL}}_{l,m}}$ is the truth-value of $\mathrm{OL}_{(l,m)_k}$ given the inferred output $\tilde{\mathrm{OL}}_{(l,m)_k}$; $\tau_{\{\mathrm{IL}_{(1,j)_k}\wedge\cdots\wedge\mathrm{IL}_{(i,j)_k}\wedge\cdots\wedge\mathrm{IL}_{(n1,j)_k}\}\bar{\mathrm{X}}}$ is the overall truth-value of the conjunction of the input fuzzy propositions given the fuzzified input $\bar{\mathbf{X}} = [\bar{\mathrm{X}}_1, \ldots, \bar{\mathrm{X}}_i, \ldots, \bar{\mathrm{X}}_{n1}]^{\mathrm{T}}$; $\tau_{\mathrm{Implication}}$ is the truth-value of the fuzzy rule R_k; and $I_k(\cdot)$ is the implication rule that defines how R_k maps the inputs to the outputs.

In the proposed GenSoFNN-TVR(S) network, a fuzzy rule R_k is created and maintained only when supported by the existence of data clusters in the input and output space. Hence, the truth-value of the fuzzy rule R_k is *TRUE*. That is, from Eq. (B.4), $\tau_{\mathrm{Implication}}(I_k(a,b)) = I_k(a,b)$. In addition, Step 3 clearly showed that $\tau_{\{\mathrm{IL}_{(1,j)_k}\wedge\cdots\wedge\mathrm{IL}_{(i,j)_k}\wedge\cdots\wedge\mathrm{IL}_{(n1,j)_k}\}\bar{\mathrm{X}}}$ is a singleton defined at $a = T(\mu_{\mathrm{IL}_{(1,j)_k}}(x_1), \ldots, \mu_{\mathrm{IL}_{(i,j)_k}}(x_i), \ldots, \mu_{\mathrm{IL}_{(n1,j)_k}}(x_{n1})) \in [0,1]$. Hence, $\sup_a \tau_{\{\mathrm{IL}_{(1,j)_k}\wedge\cdots\wedge\mathrm{IL}_{(i,j)_k}\wedge\cdots\wedge\mathrm{IL}_{(n1,j)_k}\}\bar{\mathrm{X}}}(a) = 1$ is deduced given the condition of a as defined in Eq. (B.3). Using the above knowledge, Eq. (B.4) can be reduced as follows.

$$\begin{aligned}
\tau_{\mathrm{OL}_{(l,m)_k}\tilde{\mathrm{OL}}_{(l,m)_k}}(b) &= \sup_a \Big\{ m_I \Big[\tau_{\{\mathrm{IL}_{(1,j)_k}\wedge\cdots\wedge\mathrm{IL}_{(i,j)_k}\wedge\cdots\wedge\mathrm{IL}_{(n1,j)_k}\}\bar{\mathrm{X}}}(a), \tau_{\mathrm{Implication}}(I_k(a,b)) \Big] \Big\} \\
&= m_I \left[1, \underbrace{\tau_{\mathrm{Implication}}(I_k(a,b))}_{=I_k(a,b)} \right] \quad \text{where } a = T(\mu_{\mathrm{IL}_{(1,j)_k}}(x_1), \ldots, \mu_{\mathrm{IL}_{(i,j)_k}}(x_i), \ldots, \mu_{\mathrm{IL}_{(n1,j)_k}}(x_{n1})) \in [0,1] \\
&= m_I[1, I_k(a,b)] \quad \text{where } m_I(\cdot) \text{ is a } T\text{-norm} \\
&= T[1, I_k(a,b)] \\
&= I_k(a,b) \quad (\text{because of boundary condition of } T\text{-norm})
\end{aligned} \tag{B.5}$$

where $m_l(\cdot)$ is the forward reasoning function [64]. When the value of a is computed by taking the minimum value of $\mu_{\mathrm{IL}_{(i,j)_k}}(x_i), \forall i \in \{1,\ldots,n1\}$ and the *Larsen implication rule* [63] is used to define how a rule R_k maps the inputs to the outputs, the truth-value $\tau_{\mathrm{OL}_{(l,m)_k}\tilde{\mathrm{OL}}_{(l,m)_k}}$ is defined as

$$\begin{aligned}\tau_{\mathrm{OL}_{(l,m)_k}\tilde{\mathrm{OL}}_{(l,m)_k}}(b) &= I_k(a,b)\\ &= a\times b \quad \text{where } I_k(\cdot) \text{ is resolved using Larsen implication}\\ &= \underbrace{[\min_{i\in\{1\ldots n1\}}\{\mu_{\mathrm{IL}_{(1,j)_k}}(x_1),\ldots,\mu_{\mathrm{IL}_{(i,j)_k}}(x_i),\ldots,\mu_{\mathrm{IL}_{(n1,j)_k}}(x_{n1})\}]}_{=Z_{R_k}}\times b\\ &= Z_{R_k}\times b = Z_{R_k}^{(l,m)}\times b\end{aligned} \tag{B.6}$$

where $\mu_{\mathrm{IL}_{(i,j)_k}}$ is the membership function of the input fuzzy label $\mathrm{IL}_{i,j}$ that is connected to rule R_k; and $Z_{R_k}^{(l,m)}$ is the computed output of rule R_k that connects to output label $\mathrm{OL}_{l,m}$.

Hence, using the computed truth-value $\tau_{\mathrm{OL}_{(l,m)_k}\tilde{\mathrm{OL}}_{(l,m)_k}}$ and the *truth function modification* (TFM) process defined in the TVR inference scheme (Appendix A), the inferred fuzzified output $\tilde{\mathrm{OL}}_{(l,m)_k}$ due to the firing of rule R_k can be deduced as:

$$\begin{aligned}\tilde{\mathrm{OL}}_{(l,m)_k} &= \mu_{\tilde{\mathrm{OL}}_{(l,m)_k}}(y_m) = \tau_{\mathrm{OL}_{(l,m)_k}\tilde{\mathrm{OL}}_{(l,m)_k}}(b)\\ &= Z_{R_k}^{(l,m)}\times\underbrace{b}_{=\mu_{\mathrm{OL}_{(l,m)_k}}(y_m)}\end{aligned}\tag{B.7}$$

b is the truth-value of the consequent supported by data cluster (TRUE ⇒ $\tau_{\mathrm{OL}_{(l,m)_k}}(b) = b = \mu_{\mathrm{OL}_{(l,m)_k}}(y_m)$) $= Z_{R_k}^{(l,m)}\times\underbrace{\mu_{\mathrm{OL}_{(l,m)_k}}(y_m)}_{\mu_{\mathrm{OL}_{l,m}}(y_m)} = Z_{R_k}^{(l,m)}\times\mu_{\mathrm{OL}_{l,m}}(y_m)$

where $\mu_{\tilde{\mathrm{OL}}_{(l,m)_k}}$ is the membership function of the inferred fuzzified output for label $\mathrm{OL}_{l,m}$ that rule R_k connects to (denoted as $\mathrm{OL}_{(l,m)_k}$); $\tau_{\mathrm{OL}_{(l,m)_k}}$ is the truth-value of the output fuzzy proposition "y_m is $\mathrm{OL}_{l,m}$" due to rule R_k given the training data; $\mu_{\mathrm{OL}_{l,m}}$ is the membership function of the output label $\mathrm{OL}_{l,m}$; $Z_{R_k}^{(l,m)}$ is the computed output of rule R_k that connects to $\mathrm{OL}_{l,m}$; and y_m is the output of node OV_m.

However, the computed $\tilde{\mathrm{OL}}_{(l,m)_k}$ defined in Eq. (B.7) is due only to the firing of rule R_k alone. Since more than one fuzzy rule may share the same output label $\mathrm{OL}_{l,m}$ as consequent, the effects due to the firing of the various rules must be aggregated. In the literature, two approaches are proposed: the *infer-first-then-aggregate* (IFTA) and *aggregate-first-then-infer* (AFTI) [63] methods. In the proposed GenSoFNN-TVR(S) network, the first approach is adopted as it best suits the computing nature of the hybrid structure. That is, deduce the inferred fuzzified outputs due to the firing of the individual rules and subsequently obtain the overall inferred output by aggregating the effects of the individual rules. Hence, the overall inferred fuzzified output $\tilde{\mathrm{OL}}_{l,m}$ for the output label $\mathrm{OL}_{l,m}$ is defined as:

$$\begin{aligned}\tilde{\mathrm{OL}}_{l,m} &= \bigcup_{k\in\{1,\ldots,N_{l,m}\}}\tilde{\mathrm{OL}}_{(l,m)_k}\\ &= \max_{k\in\{1,\ldots,N_{l,m}\}}(\mu_{\tilde{\mathrm{OL}}_{(l,m)_k}}(y_m)) \quad \text{because } \tilde{\mathrm{OL}}_{(l,m)_k}, \forall k\in\{1,\ldots,N_{l,m}\}, \text{ are fuzzy sets}\\ &= \max_{k\in\{1,\ldots,N_{l,m}\}}(Z_{R_k}^{(l,m)}\times\mu_{\mathrm{OL}_{l,m}}(y_m))\\ &= \underline{\max_{k\in\{1,\ldots,N_{l,m}\}}(Z_{R_k}^{(l,m)})}\times\mu_{\mathrm{OL}_{l,m}}(y_m)\end{aligned}\tag{B.8}$$

Note from Section 4.4 that
$Z_{l,m} = \max_{k\in\{1,\ldots,N_{l,m}\}}\left(Z_{R_k}^{(l,m)}\right) = \max_{k\in\{1,\ldots,N_{l,m}\}}\left\{Z_{R_1}^{(l,m)},\ldots,Z_{R_k}^{(l,m)},\ldots,Z_{R_{N_{l,m}}}^{(l,m)}\right\}$

where $\mathrm{N}_{l,m}$ is the number of rules that have the output label $\mathrm{OL}_{l,m}$ as their consequent.

Notice that the underlined term in Eq. (B.8) directly maps to the operations of the layer 4 nodes through Eqs. (13) and (14). Hence, it has been clearly shown that the consequent derivation phase of the TVR-FS corresponds to the functionality of the layer 4 nodes in the GenSoFNN-TVR network. Fig. B.2 depicts the computation of the overall inferred fuzzified output for an arbitrary

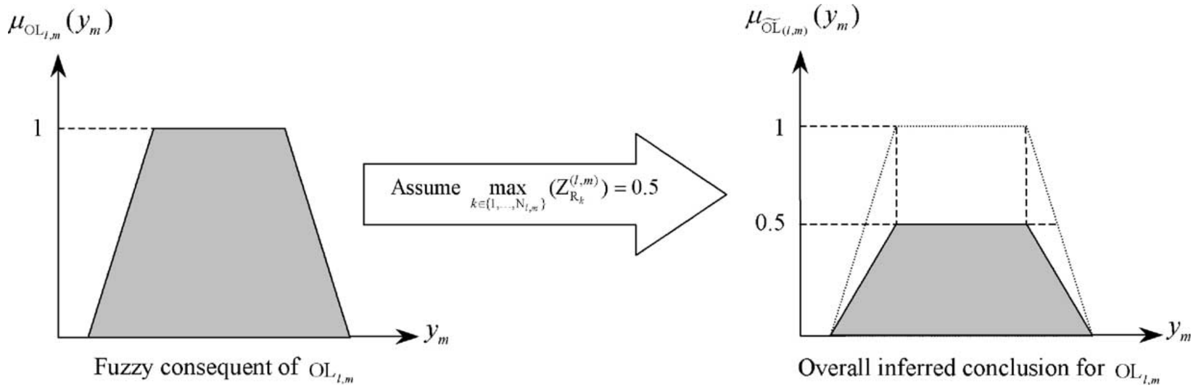
$\mu_{OL_{l,m}}(y_m)$
1
y_m
Fuzzy consequent of $OL_{l,m}$
Assume $\max_{k\in\{1,\ldots,N_{l,m}\}}(Z_{R_k}^{(l,m)}) = 0.5$
$\mu_{\widetilde{OL}(l,m)}(y_m)$
1
0.5
y_m
Overall inferred conclusion for $OL_{l,m}$

Figure B.2 Derivation of the overall inferred output for label $OL_{l,m}$ using TVR and Larsen implication.

output label $OL_{l,m}$ in the proposed GenSoFNN-TVR(S) network.

B.5. Step 5—output defuzzification

The last step in the inference process of the TVR-FS system is to defuzzify the derived fuzzy conclusions and present them as crisp system outputs. For each output node OV_m, the derived conclusions of all its output fuzzy labels are aggregated using a modified weighted *center of averaging* (COA) technique to compute the final crisp output y_m. This technique is implemented in the proposed GenSoFNN-TVR(S) network using Eqs. (15)—(17).

Appendix C

Monte Carlo evaluative selection (MCES)

The algorithmic form of the *Monte Carlo evaluative selection* (MCES) technique used to select the relevant features in the cancer application is as listed below.

Algorithm MCES

```
a. Generate a random mask of the feature set, denoted as mask 1 (i.e. xxx1xxxxx ...).
b. Select a random position P in the feature set.
c. Invert position P of the generated mask 1 to form mask 2 (i.e. xxx0xxxxx ...).
d. Obtain an instance of the training examples.
e. Define Example 1 = instance with mask 1.
f. Define Example 2 = instance with mask 2.
   For mask with an enabled (1) position, preserve the value of the input feature.
   For mask with a disabled (0) position, use the average value for the input feature.
g. Compute output $y_1$ with Example 1 using the induction algorithm.
h. Compute output $y_2$ with Example 2 using the induction algorithm.
i. If computed output $y_1$ is closer than computed output $y_2$ to the target output $y$,
      Assign a positive feedback $r_{jP}$ to mask 1 over mask 2 for that selected position $P$.
   Else
      Assign a negative feedback $r_{jP}$ to mask 1 over mask 2 for that selected position $P$.
j. Repeat step (a) to (i) for all training examples for a predefined number of iterations.
k. Compute the average feedback value for each bit position (input feature).
l. Rank the features in the feature set according to the computed weights.

End MCES
```

To illustrate the workings of the MCES algorithm, assume a training dataset where there are I input features and J training exemplars. In addition,

$\mathbf{X}$ denotes a matrix of size $J \times I$ consisting of J training exemplars and I input variables;

$\mathbf{x}_i$ is a column vector that denotes the values of feature i for all the J training exemplars, such that $x_i = [x_{1i}, x_{2i}, \ldots, x_{ji}, \ldots, x_{Ji}]^T$;

$\bar{x}_i$ denotes the average value of all elements in x_i such that

$$\bar{\mathrm{x}}_i = \frac{1}{J}\sum_{j=1}^{J} x_{ji} \tag{C.1}$$

$\mathbf{x}_j$ is a row vector that denotes the values of all the input features for exemplar j such that $x_j = [x_{j1}, x_{j2}, \ldots, x_{ji}, \ldots, x_{jI}]$;

x_{ji} denotes the value of feature i in exemplar j, where $x_{ji} \in \mathbf{X}$;

m_j denotes the mask vector for exemplar j, where $m_j = [m_{j1}, \ldots, m_{ji}, \ldots, m_{jI}]$;

m_{ji} denotes the masking bit for feature i in exemplar j, where $m_{ji} \in \{0,1\}$;

$\bar{r}_i$ denotes the average feedback value for feature i;

r_{jP} denotes the positive/negative feedback of feature *P* for exemplar *j* due to the switching of feature *P* from "on" to "off" or vice-versa based on bit *P* in *mask 1* and *mask 2* (note that *mask 1* and *mask 2* differs only by the single bit *P*);
W denotes the parameter set of the underlying induction algorithm; and
$E(\cdot)$ denotes the error function based on the distance between the predicted and target outputs.

According to steps (a)—(i) of the MCES algorithm, the feedback r_{jP} is computed as:

$$r_{jP} = \begin{cases} |E(\bar{x}_P, \underset{-P}{\text{Mask}}(x_j, m_j), W) - E(x_{jP}, \underset{-P}{\text{Mask}}(x_j, m_j), W)|, & \text{if } |E(\bar{x}_P, \ldots)| \geq |E(x_{jP}, \ldots)| \\ -|E(\bar{x}_P, \underset{-P}{\text{Mask}}(x_j, m_j), W) - E(x_{jP}, \underset{-P}{\text{Mask}}(x_j, m_j), W)|, & \text{otherwise} \end{cases} \tag{C.2}$$

where $\text{Mask}_{-P}(\cdot)$ denotes the masking operation based on the similar parts in *mask 1* and *mask 2* without the differing bit *P*. The masking operation is defined by Eq. (C.3) as follows.

$$\text{Mask}(x_j, \boldsymbol{m}_j) = \underset{\forall i \in \{1,\ldots,I\} \wedge i \neq P}{\text{Mask}}([x_{ji}], [m_{ji}])$$
$$\text{s.t.}\, \text{Mask}(x_{ji}, m_{ji}) = \begin{cases} x_{ji}, & \text{if } m_{ji} = 1 \\ \bar{x}_i, & \text{if } m_{ji} = 0 \end{cases} \tag{C.3}$$

Subsequently, the average feedback value $\bar{r}_i$ for feature *i* is computed as

$$\bar{r}_i = \frac{1}{J}\sum_{j=1}^{J} r_{ji} \tag{C.4}$$

The computed average feedback value or weight for each feature *i* reflects the relative importance of that feature over the rest of the input features. Negative or near-zero weight indicates that the corresponding feature is irrelevant to the output(s). The degree of relevance increases with a larger value of the weight. In the current implementation, the error function *E* is defined as the difference between the actual and the predicted outputs. That is, $E = (y_m - y)$, $m \in \{1, 2\}$. Thus, Eq. (C.2) can be simplified as:

$$\begin{aligned} r_{jP} &= \begin{cases} |(y_1 - y) - (y_2 - y)|, & \text{if } |E(\bar{x}_P, \ldots)| \geq |E(x_{jP}, \ldots)| \\ -|(y_1 - y) - (y_2 - y)|, & \text{otherwise} \end{cases} \\ &= \begin{cases} |y_1 - y_2|, & \text{if } |E(\bar{x}_P, \ldots)| \geq |E(x_{jP}, \ldots)| \\ -|y_1 - y_2|, & \text{otherwise} \end{cases} \end{aligned} \tag{C.5}$$

where *y* denotes the actual (target) output; y_1 denotes the predicted output by the underlying induction algorithm with *mask 1* being applied to the inputs, and y_2 denotes the predicted output with the *mask 2* applied to the inputs.

The working of the novel MCES algorithm is graphically summarized as Fig. C.1.

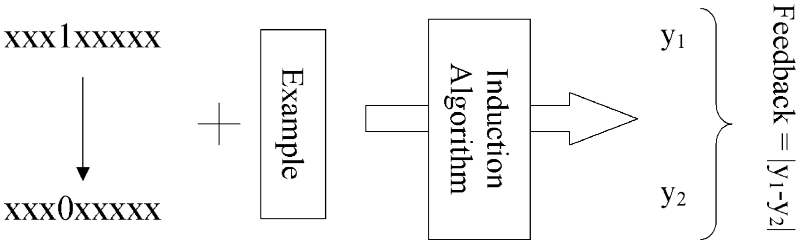

Figure C.1 Feature selection process of the MCES algorithm.

Appendix D.

Table D.1 Relevant genes selected for the ALL cancer subtypes (i.e. BCR-ABL, E2A-PBX1, Hyper > 50, MLL, T-ALL, and TEL-AML1) by MCES based on *T*-statistics pre-filtering

Subtype	Affymetrix no.	Gene name	Gene symbol	MCES weight
BCR-ABL	40798_s_at	Disintegrin and metalloproteinase domain 10	ADAM10	0.27760
	36591_at	Tubulin alpha 1 testis specific	TUBA1	0.22779
	39070_at	Singed Drosophila like sea urchin fascin homolog like	SNL	0.17792
	330_s_at	Tubulin, alpha 1, isoform 44	TUBA1	0.15191
	1211_s_at	CASP2 and RIPK1 domain containing adaptor with death domain	CRADD	0.14231
E2A-PBX1	33355_at	Homo sapiens cDNA FLJ12900 fis clone NT2RP2004321 (by CELERA search of target sequence = PBX1)	PBX1	0.75617
Hyper > 50	1447_at	Proteasome prosome macropain subunit beta type 1	PSMB1	0.19138
	41724_at	Accessory proteins BAP31/BAP29	DXS1357E	0.17898
	39867_at	Tu translation elongation factor mitochondrial	TUFM	0.17610
	40875_s_at	Small nuclear ribonucleoprotein 70kD polypeptide RNP antigen	SNRP70	0.16819
	39878_at	Protocadherin 9	PCDH9	0.16432
MLL	1389_at	Membrane metallo-endopeptidase neutral endopeptidase enkephalinase CALLA CD10	MME	0.57293
	33412_at	LGALS1 Lectin, galactoside-binding, soluble, 1 (galectin 1)	LGALS1	0.42637
	40520_g_at	Protein tyrosine phosphatase receptor type C	PTPRC	0.34451
	40519_at	Protein tyrosine phosphatase receptor type C	PTPRC	0.32823
	1520_s_at	Interleukin 1 beta	IL1B	0.22234
	794_at	Protein tyrosine phosphatase non-receptor type 6	PTPN6	0.20051
	307_at	Arachidonate 5-lipoxygenase	ALOX5	0.17356
	37398_at	Platelet/endothelial cell adhesion molecule CD31 antigen	PECAM1	0.15865
	2062_at	Insulin-like growth factor binding protein 7	IGFBP7	0.15317
	174_s_at	Intersectin 2	ITSN2	0.12112
	39705_at	KIAA0700 protein	KIAA0700	0.10828
T-ALL	38319_at	CD3D antigen delta polypeptide TiT3 complex	CD3D	1.30900
TEL-AML1	35614_at	Transcription factor-like 5 basic helix-loop-helix	TCFL5	0.32187
	1325_at	MAD mothers against decapentaplegic Drosophila homolog 1	MADH1	0.19613
	32163_f_at	EST		0.16443
	37780_at	Piccolo presynaptic cytomatrix protein	PCLO	0.15817
	39329_at	Actinin alpha 1	ACTN1	0.14742

Table D.2 Relevant genes selected for the ALL cancer subtypes (i.e. BCR-ABL, E2A-PBX1, Hyper > 50, MLL, T-ALL, and TEL-AML1) by MCES based on SOM/DAV pre-filtering

Subtype	Affymetrix no.	Gene name	Gene symbol	MCES weight
BCR-ABL	37600_at	Extracellular matrix protein 1	ECM1	0.51796
	1636_g_at	Human proto-oncogene tyrosine-protein kinase (ABL) gene, exon 1a & exons 2—10, complete cds	ABL	0.42095
	40196_at	HYA22 protein	HYA22	0.39430
	330_s_at	Tubulin, alpha 1, isoform 44	TUBA1	0.30305
	36591_at	Tubulin alpha 1 testis specific	TUBA1	0.29618
	39250_at	Nephroblastoma overexpressed gene	NOV	0.29250
	1635_at	Human proto-oncogene tyrosine-protein kinase (ABL) gene, exon 1a and exons 2—10, complete cds	ABL	0.28424
	33362_at	Cdc42 effector protein 3	CEP3	0.18516
	39730_at	V-abl Abelson murine leukemia viral oncogene homolog 1	ABL1	0.17464
	38312_at	DKFZp564O222 from clone DKFZp564O222	ECM1	0.09037

Table D.2 (*Continued*)

Subtype	Affymetrix no.	Gene name	Gene symbol	MCES weight
E2A-PBX1	33355_at	Homo sapiens cDNA FLJ12900 fis clone NT2RP2004321	PBX1	0.60712
	39929_at	KIAA0922 protein	KIAA0922	0.36537
	40454_at	FAT tumor suppressor Drosophila homolog	FAT	0.29005
	1287_at	ADP-ribosyltransferase NAD poly ADP-ribose polymerase	ADPRT	0.27955
	37225_at	KIAA0172 protein	KIAA0172	0.26415
	41146_at	ADP-ribosyltransferase NAD poly ADP-ribose polymerase	ADPRT	0.21687
	430_at	nucleoside phosphorylase	NP	0.21072
	41139_at	melanoma antigen family D 1	MAGED1	0.17533
	39402_at	interleukin 1 beta	IL1B	0.15771
	41017_at	Myosin-binding protein H	MYBPH	0.15019
Hyper > 50	38518_at	Sex comb on midleg Drosophila like 2	SCML2	0.46520
	36620_at	Superoxide dismutase 1 soluble amyotrophic lateral sclerosis 1 adult	SOD1	0.32442
	32207_at	Membrane protein palmitoylated 155kD	MPP1	0.28770
	38968_at	SH3-domain binding protein 5 BTK-associated	SH3BP5	0.26155
	36795_at	Prosaposin variant Gaucher disease and variant metachromatic leukodystrophy	PSAP	0.19196
	37326_at	Proteolipid protein 2 colonic epithelium-enriched	PLP2	0.18148
	38738_at	SMT3 suppressor of mif two 3 yeast homolog 1	SMT3H1	0.16157
	37543_at	Rac/Cdc42 guanine exchange factor GEF 6	ARHGEF6	0.12527
	39039_s_at	CGI-76 protein	LOC51632	0.11812
	39168_at	Ac-like transposable element	ALTE	0.11328
MLL	33412_at	LGALS1 Lectin, galactoside-binding, soluble, 1 (galectin 1)	LGALS1	0.38662
	36777_at	DNA segment on chromosome 12 unique 2489 expressed sequence	D12S2489E	0.38576
	34306_at	muscleblind Drosophila like	MBNL	0.33317
	32193_at	Plexin C1	PLXNC1	0.25448
	2062_at	Insulin-like growth factor binding protein 7	IGFBP7	0.18186
	40763_at	Meis1 mouse homolog	MEIS1	0.17544
	657_at	protocadherin gamma subfamily C 3	PCDHGC3	0.11709
	38391_at	capping protein actin filament gelsolin-like	CAPG	0.11704
	1126_s_at	Human cell surface glycoprotein CD44 (CD44) gene, 3’ end of long tailed isoform.	CD44	0.11504
	1102_s_at	Nuclear receptor subfamily 3 group C member 1	NR3C1	0.10890
T-ALL	38319_at	CD3D antigen delta polypeptide TiT3 complex	CD3D	1.10900
TEL-AML1	1488_at	Protein tyrosine phosphatase receptor type K	PTPRK	0.26832
	577_at	Midkine neurite growth-promoting factor 2	MDK	0.24706
	41442_at	Core-binding factor runt domain alpha subunit 2 translocated to 3	CBFA2T3	0.24170
	35614_at	Transcription factor-like 5 basic helix-loop-helix	TCFL5	0.21197
	33162_at	Insulin receptor	INSR	0.21112
	36239_at	POU domain class 2 associating factor 1	POU2AF1	0.19316
	33690_at	cDNA DKFZp434A202 from clone DKFZp434A202		0.19218
	36985_at	Isopentenyl-diphosphate delta isomerase	IDI1	0.17230
	38652_at	Hypothetical protein FLJ20154	FLJ20154	0.17104
	40745_at	Adaptor-related protein complex 1 beta 1 subunit	AP1B1	0.14523
	41200_at	CD36 antigen collagen type I receptor thrombospondin receptor like 1	CD36L1	0.10462

Table D.3 Classification results on the test set using support vector machine (SVM) with polynomial kernel, K-Nearest Neighbor (K-NN) classifier, artificial neural network (ANN) and the GenSoFNN-TVR(S) network

Classifier		SVM		K-NN		ANN	GenSoFNN-TVR(S)	
		T-statistics	SOM/DAV	*T*-statistics	Wilkins'	Wilkins'	*T*-statistics	SOM/DAV
T-ALL	Sensitivity (%)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)
[15+; 97−]	Specificity (%)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)
E2A-PBX1	Sensitivity (%)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)
[9+; 88−]	Specificity (%)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)
TEL-AML1	Sensitivity (%)	100 (0)	100 (0)	96 (1.08)	100 (0)	100 (0)	90 (2.7)	94 (1.62)
[27+; 61−]	Specificity (%)	97 (1.83)	97 (1.83)	100 (0)	100 (0)	100 (0)	98 (1.22)	99 (0.61)
BCR-ABL	Sensitivity (%)	33 (4.02)	83 (1.02)	50 (3)	67 (1.98)	83 (1.02)	67 (1.98)	65 (2.1)
[6+; 55−]	Specificity (%)	100 (0)	98 (1.1)	100 (0)	96 (2.2)	98 (1.1)	99 (0.55)	100 (0)
MLL	Sensitivity (%)	100 (0)	86 (0.84)	100 (0)	100 (0)	100 (0)	70 (1.8)	100 (0)
[6+; 49−]	Specificity (%)	100 (0)	100 (0)	94 (2.94)	100 (0)	100 (0)	99 (0.49)	100 (0)
Hyper $>$ 50	Sensitivity (%)	100 (0)	95 (1.1)	95 (1.1)	100 (0)	100 (0)	99 (0.22)	98 (0.44)
[22+; 27−]	Specificity (%)	93 (1.89)	93 (1.89)	93 (1.89)	96 (1.08)	89 (2.97)	85 (4.05)	85 (4.05)

References

[1] Rubnitz JE. Acute lymphoblastic leukaemia, e-Medicine: instant access to the minds of medicine, 2002 [online]. http://www.emedicine.com/ped/topic2587.htm.

[2] Pui CH, Evans WE. Acute lymphoblastic leukemia. N Engl J Med 1998;339:605—15.

[3] Yeoh EJ, Ross ME, Shurtleff SA, et al. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. Cancer Cell 2002;1(2):133—43.

[4] Schrappe M, Reiter A, Ludwig WD, et al. Improved outcome in childhood acute lymphoblastic leukemia despite reduced use of anthracyclines and cranial radiotherapy: results of trial ALL-BFM 90. Blood 2000;95:3310—22.

[5] Silverman LB, Gelber RD, Dalton VK, et al. Improved outcome for children with acute lymphoblastic leukemia: results of Dana-Farber Consortium Protocol 91-01. Blood 2001;97:1211—8.

[6] Nguyen DV, Rocke DM. Tumor classification by partial least squares using microarray gene expression data. Bioinformatics 2002;18(1):39—50.

[7] Li W, Yang Y. How many genes are needed for a discriminant microarray data analysis. In: Lin SM, Johnson KF, editors. Methods of Microarray Data Analysis. Hingham, MA: Kluwer Academic Publishers; 2003.

[8] Ben-Dor A, Shamir R, Yakhini Z. Clustering gene expression patterns. J Computat Biol 1999;6:281—97.

[9] Dudoit S, Fridlyand J, Speed TP. Comparison of discrimination methods for the classification of tumors using gene expression data. Technical Report 576, Department of Statistics, University of California, Berkeley; 2000.

[10] Furey TS, Cristianini N, Duffy N, et al. Support vector machine classification and verification of cancer tissue samples using microarray expression data. Bioinformatics 2000;16(10):906—14.

[11] Wang X, Hong J. On the handling of fuzziness for continuous-valued attributes in decision tree generation. Fuzzy Sets Syst 1998;99:283—90.

[12] Belacel N, Vincke Ph., Scheiff JM, Boulassel MR. Acute leukemia diagnosis aid using multicriteria fuzzy assignment methodology. Comput Methods Programs Biomed 2001; 64(2):145—51.

[13] Lin CT, Lee CSG. Neural fuzzy systems—a neuro-fuzzy synergism to intelligent systems. Englewood Cliffs, NJ: Prentice Hall; 1996.

[14] Zadeh LA. Fuzzy logic, neural networks and soft computing. Commun ACM 1994;37(3):77—84.

[15] Nauck D, Klawonn F, Kruse R. Foundations of neuro-fuzzy systems. Chichester, England, New York: Wiley; 1997.

[16] Ang KK, Quek C, Pasquier M. POPFNN-CRI(S): pseudo outer product based fuzzy neural network using the compositional rule of inference and singleton fuzzifier. IEEE Trans Syst Man Cybernet B, in press.

[17] Quek C, Zhou RW. POPFNN: a pseudo outer-product based fuzzy neural network. Neural Netw 1996;9(9):1569—81.

[18] Quek C, Zhou RW. POPFNN-AARS(S): a pseudo outer-product based fuzzy neural network. IEEE Trans Syst Man Cybern 1999;29(6):859—70.

[19] Jang JSR. ANFIS: adaptive-network-based fuzzy inference systems. IEEE Trans Syst Man Cybernet 1993;23:665—85.

[20] Lin CJ, Lin CT. An ART-based fuzzy adaptive learning control network. IEEE Trans Fuzzy Syst 1997;5(4):477—96.

[21] Quek C, Tung WL. A novel approach to the derivation of fuzzy membership functions using the Falcon-MART architecture. Pattern Recogn Lett 2001;22(9):941—58.

[22] Juang CF, Lin CT. An on-line self-constructing neural fuzzy inference network and its applications. IEEE Trans Fuzzy Syst 1998;6(1):12—32.

[23] Kasabov N. On-line learning, reasoning, rule extraction and aggregation in locally optimized evolving fuzzy neural networks. Neurocomputing 2001;41:25—45.

[24] Kasabov N, Song Q. DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. IEEE Trans Fuzzy Syst 2002;10(2):144—54.

[25] Mantaras RL. Approximate reasoning models. Chichester, West Essex, England: Ellis Horwood Limited; 1990.

[26] Tung WL, Quek C. GenSoFNN: a generic self-organizing fuzzy neural network. IEEE Trans Neural Netw 2002;13(5):1075—86.

[27] Zadeh LA. Calculus of fuzzy restrictions. In: Fuzzy sets and their applications to cognitive and decision processes. New York: Academic Press; 1975. p. 1—39.

[28] Shen ZL, Lui HC, Ding LY. Approximate case-based reasoning on neural networks. Int J Approx Reason 1994;10:75—98.

[29] Bugarin AJ, Barro S. Reasoning with truth values on compact fuzzy chained rules. IEEE Trans Syst Man Cybernet B 1998;28(1):34—46.

[30] Klir GJ, Yuan B. Fuzzy sets and fuzzy logic: theory and applications. Upper Saddle River, NJ: Prentice Hall; 1995.

[31] Berenji HR, Khedkar P. Learning and tuning fuzzy logic controllers through reinforcements. IEEE Trans Neural Netw 1992;3:724—40.

[32] Keller JM, Yager RR, Tahani H. Neural network implementation of fuzzy logic. Fuzzy Sets Syst 1992;45(1):1—12.

[33] Yager RR. Modeling and formulating fuzzy knowledge bases using neural networks. Neural Netw 1994;7(8):1273—83.

[34] Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL, editors. et al. Parallel distributed processing, vol. 1. Cambridge, MA: MIT Press; 1986 [Chapter 8].

[35] Tung WL, Quek C. DIC: a novel discrete incremental clustering technique for the derivation of fuzzy membership functions. PRICAI trends in artificial intelligence In: Ishizuka M, Sattar A, editors. Proceedings of the 7th Pacific Rim International Conference on Artificial Intelligence. 2002. p. 178—87.

[36] Grossberg S. Adaptive pattern classification and universal recoding. II. Feedback, expectation, olfaction, illusions. Biol Cybernet 1976;23:187—202.

[37] Tung WL. A generalized framework for fuzzy neural architecture. Technical Report, ISL-TR-01/01, School of Computer Engineering, Nanyang Technological University, Singapore, 2001.

[38] Tung W.L. Back-propagation learning for the GenSoFNN-TVR(S) network. Technical Report, ISL-TR-01/02, School of Computer Engineering, Nanyang Technological University, Singapore, 2002.

[39] Turksen IB, Zhong Z. An approximate analogical reasoning scheme based on similarity measures and interval valued fuzzy sets. Fuzzy Sets Syst 1990;34:323—46.

[40] Mendel JM. Uncertain rule-based fuzzy logic systems: introduction and new directions. Upper Saddle River, NJ: Prentice Hall; 2001.

[41] Hayashibara KC, Ray E, Elashoff D, et al. Mapping single and multigenetic traits in *S. cerevisiae* by genomic mismatch scanning and DNA microarrays. Nat Genet 1999;23:49—50.

[42] Talaat AM, Hunter P, Johnston SA. Genome-directed primers for selective labeling of bacterial transcripts for DNA microarray analysis. Nat Biotechnol 2000;18:679—82.

[43] Miller LD, Long MP, Wong L, et al. Optimal gene expression analysis by microarrays. Cancer Cell 2002;2(5):353—61.

[44] Harrington CA, Rosenow C, Retief J. Monitoring gene expression using DNA microarrays. Curr Opin Microbiol 2000; 3:285—91.

[45] Alizadeh AA, Eisen MB, Davis RE, et al. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 2000;403:503—11.

[46] Sorlie T, Perou CM, Tibshirani R, et al. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. Proc Natl Acad Sci USA 2001; 98(19):10869—74.

[47] Shamir R, Sharan R. Algorithmic approaches to clustering gene expression data. In: Current topics in computational molecular biology. Cambridge, MA: MIT Press; 2002. p. 269—300.

[48] Futschik ME, Reeve A, Kasabov N. Evolving connectionist systems for knowledge discovery from gene expression data of cancer tissue. Artif Intell Med 2003;28(2):165—89.

[49] Liu H, Motoda H. Feature selection for knowledge discovery and data mining. Boston: Kluwer Academic Publishers; 1998.

[50] Affymetrix, Santa Clara, CA, US [online]. http://www.affymetrix.com/index.affx.

[51] St. Jude Children's Research Hospital [online]. http://www.stjuderesearch.org/data/ALL1/.

[52] Quah KH, Quek C. MCES: a novel Monte Carlo evaluative selection approach for objective feature selections in data modelling. Submitted to IEEE Trans Neural Netw for review, 2004.

[53] Sutton RS, Barto AG. Reinforcement learning: an introduction. Cambridge, MA: MIT Press; 1998.

[54] Goldman RN, Weinberg JS. Statistics: an introduction. Englewood Cliffs, NJ: Prentice Hall; 1985.

[55] Tung WL. Stresses and hypothetical reasoning with generic self-organising neural fuzzy systems. Technical Report, ISL-TR-01/03, School of Computer Engineering, Nanyang Technological University, Singapore, 2003.

[56] Centre for Computational Intelligence (C2i) [online]. http://www.c2i.ntu.edu.sg.

[57] Tung WL, Quek C. PACL-FNNS: a novel class of falcon-like fuzzy neural networks based on positive and negative exemplars.#Intelligent Systems: Technology and Applications In: Leondes CT, editor. Fuzzy Systems Neural Networks and Expert Systems, vol. II. CRC Press; 2002. p. 257—320 [Chapter 10].

[58] Pasquier M, Quek C, Toh M. Fuzzylot: a self-organising fuzzy-neural rule-based pilot system for automated vehicles. Neural Networ 2001;14(8):1099—112.

[59] Quek C, Zhou RW. Antiforgery: a novel pseudo-outer product based fuzzy neural network driven signature verification system. Pattern Recogn Lett 2002;23(14):1795—816.

[60] Ang KK, Quek C. Improved MCMAC with momentum, neighborhood, and averaged trapezoidal output. IEEE Trans Syst Man Cybernet B 2000;30(3):491—500.

[61] Quek C, Tan B, Sagar V. POPFNN-based fingerprint verification system. Neural Netw 2001;14:305—23.

[62] Tung WL, Quek C, Cheng PYK. GenSo-EWS: a novel neural-fuzzy based early warning system for predicting bank failures. Neural Netw 2004;17(4):567—87.

[63] Fuller R. On fuzzy reasoning schemes. In: Carlsson C, editor. The state of the art of information systems in 2007. TUCS General Publications, No. 16, Turku Centre for Computer Science; 1999. p. 85—112.

[64] Trilla E, Valverde L. On inference in fuzzy logic. In: Proceedings of 2nd IFSA Congress; 1987. p. 294—7.

[65] Dubois D, Prade H. Fuzzy sets and systems: theory and application. Academic Press; 1980.

[66] Zadeh LA. In: Hayes JE, Michie D, Kulich LI, editors. A theory of approximate reasoning, machine intelligence. New York: Academic Press; 1979. p. 149—94.