Modular Rule Base Fuzzy Networks for Linguistic Composition Based Modelling

Alexander Gegov*, Nedyalko Petrov*, David Sanders+, Boriana Vatchova^

*University of Portsmouth, School of Computing

Buckingham Building, Portsmouth PO1 3HE, United Kingdom

+University of Portsmouth, School of Engineering

Anglesea Building, Portsmouth PO1 3DJ, United Kingdom

^Bulgarian Academy of Sciences, Institute of Information and Communication Technologies

Block 2, 1113 Sofia, Bulgaria

Abstract: This paper proposes a linguistic composition based modelling approach by networked fuzzy systems that are known as fuzzy networks. The nodes in these networks are modules of fuzzy rule bases and the connections between these modules are the outputs from some rule bases that are fed as inputs to other rule bases. The proposed approach represents a fuzzy network as an equivalent fuzzy system by linguistic composition of the network nodes. In comparison to the known multiple rule base approaches, this networked rule base approach reflects adequately the structure of the modelled process in terms of interacting sub-processes and leads to more accurate solutions. The approach improves significantly the transparency of the associated model while ensuring a high level of accuracy. Another advantage of this fuzzy network approach is that it fits well within the existing approaches with single rule base and multiple rule bases.

Keywords: fuzzy system models, decision analysis, large-scale systems, linguistic modelling.

1. Introduction

Complexity is a versatile feature of existing systems that cannot be described by a single definition. In this context, complexity is usually associated with a number of attributes such as uncertainty, dimensionality and structure, which make the modelling of systems with these attributes more difficult. Therefore, the complexity of a given system can be accounted for by identifying the complexity related attributes that are to be found in this system.

Fuzzy logic has proved itself as a powerful tool for dealing with uncertainty as an attribute of systemic complexity. In this context, fuzziness is quite suitable for reflecting non-probabilistic uncertainty such as imprecision, incompleteness and ambiguity [1-3].

More recently, fuzzy logic has also been made more effective in dealing with dimensionality as a systemic complexity attribute by means of rule base reduction and compression. Dimensionality in rule base reduction is associated with the number of rules, which is an exponential function of the number of system inputs and the number of linguistic terms per input [4-7]. In rule base compression, dimensionally is associated with the amount of on-line operations required during fuzzification, inference and defuzzification [8].

However, as far as structure is concerned, fuzzy logic is still unable to reflect adequately any interacting modules within a modelled process. This is due to the black-box nature of fuzzy models that cannot take into account explicitly any interactions among sub-processes [9-12]. In this respect, the following paragraphs discuss some of the main approaches in fuzzy modelling and their ability to deal with structure as a systemic complexity attribute.

The most common type of fuzzy system is with a single rule base [13-15]. This type of system is usually referred to as Standard Fuzzy System (SFS). The latter is characterised by a black-box nature whereby the inputs are mapped directly to the outputs without the consideration of any internal connections. The operation of SFS is based on a single Fuzzification-Inference-Defuzzification (FID) sequence and it is usually quite accurate for output modelling as it reflects the simultaneous influence of all inputs on the output. However, the efficiency and transparency of SFS deteriorate with the increase of the number of rules. Therefore, as the number of rules increases, it not only takes longer to simulate the model output but it is also less clear how this output is affected by the model inputs.

Another type of fuzzy system is with multiple rule bases [16-19]. This type of system is often described by cascaded rule bases and it is referred to as Chained Fuzzy System (CFS) or Hierarchical Fuzzy System (HFS). Both CFS and HFS are characterised by a white-box nature whereby the inputs are mapped to the outputs by means of some internal variables in the form of connections. The operation of CFS and HFS is based on multiple FID sequences whereby each connection links the FID sequences for two adjacent rule bases.

CFS has an arbitrary structure in terms of subsystems and the connections among them [20-22]. In this case, each subsystem represents an individual rule base whereas each interaction is represented by a connection linking a pair of adjacent rule bases. This connection is identical with an output from the first rule base and an input to the second rule base in the pair. CFS is usually used as a detailed presentation of SFS for the purpose of improving transparency by explicitly taking into account all subsystems and the interactions among them. Also, efficiency is improved because of the smaller number of inputs to the individual rule bases. However, accuracy may be lost due to the accumulation of errors as a result of the multiple FID sequences.

HFS is a special type of CFS that has a specific structure [23-27]. Each subsystem in HFS has two inputs and one output. Some connections represent identical mappings, which may propagate across parts of the system. HFS is often used as an alternative presentation of SFS for the purpose of improving transparency by explicitly taking into account all subsystems and the interactions among them. Efficiency is also improved by the reduction of the overall number of rules, which is a linear function of the number of inputs to the subsystems and the number of linguistic terms per input. However, these improvements are at the expense of accuracy due to the accumulation of errors as a result of the multiple FID sequences.

A third type of fuzzy system is with networked rule bases. This type of system is referred to as Networked Fuzzy System (NFS) and it has been introduced recently in [28]. NFS is characterised by a white-box nature whereby the inputs are mapped to the outputs by means of connections. Subsystems in NFS are represented by nodes and the interactions among subsystems are the connections among these nodes. NFS is a hybrid between SFS and CFS/HFS. On one hand, the structure of NFS is similar to the structure of CFS/HFS due to the explicit presentation of subsystems and the interactions among them. On the other hand, the operation of NFS resembles the operation of SFS as the multiple rule bases are simplified to a linguistically equivalent single rule base. This simplification is based on the linguistic composition approach that is described further below. As a hybrid concept, NFS has the potential of combining the advantages of SFS and CFS/HFS.

Properties of fuzzy systems such as accuracy, efficiency and transparency are directly related to attributes of systemic complexity such as uncertainty, dimensionality and structure. In this respect, uncertainty is an obstacle to accuracy as it is harder to build an accurate model from uncertain data [29-32]. Furthermore, dimensionality represents an obstacle to efficiency because it is more difficult to reduce the amount of computations in a FID sequence for a large number of rules [33-36]. Finally, structure is an obstacle to transparency as it is harder to understand the behaviour of a black-box model that doesn't reflect the interactions among subsystems [37-40].

This paper introduces a theoretical framework for NFS as a novel type of fuzzy system and validates NFS as a modelling tool with respect to SFS and CFS/HFS. For clarity and simplicity, NFS is referred to as Fuzzy Network (FN). Besides this, the paper addresses several attributes of systemic complexity including uncertainty, dimensionality and structure and the associated properties of the above fuzzy systems such as accuracy, efficiency and transparency. This research methodology is more balanced than the one used in many current studies as they usually focus on only one attribute of systemic complexity and the associated property of the fuzzy systems used.

The proposed theoretical framework is used in this paper for fuzzy network analysis whereby all networked rule bases and the connections among them are given in some form. In this case, the task is to simulate this networked model in order to evaluate the performance of the fuzzy network. However, the framework can also be used for fuzzy network design whereby some networked rule bases are unknown and have to be identified in a way that guarantees some pre-specified performance of the fuzzy network.

The remaining part of this paper is structured as follows. Section 2 provides some theoretical preliminaries for fuzzy networks. Section 3 introduces the linguistic composition approach. Section 4 illustrates the application of this approach for an ore flotation case study. Section 5 evaluates the performance of the approach in a quantitative and comparative context. Section 6 summarises the main advantages of the approach and highlights future research directions.

2. Theoretical Preliminaries

A fuzzy system with r rules, m inputs $x_1 \ldots x_m$ taking linguistic terms from the input sets $\{A_{11}, \ldots, A_{1r}\}, \ldots, \{A_{m1}, \ldots, A_{mr}\}$ and n outputs $y_1 \ldots y_n$ taking linguistic terms from the output sets $\{B_{11}, \ldots, B_{1r}\}, \ldots, \{B_{n1}, \ldots, B_{nr}\}$ can be represented by the following rule base

$$\begin{array}{c} \textit{Rule 1: If } x_1 \textit{ is } A_{11} \textit{ and } \ldots \textit{ and } x_m \textit{ is } A_{m1}\textit{, then } y_1 \textit{ is } B_{11} \textit{ and } \ldots \textit{ and } y_n \textit{ is } B_{n1} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \textit{Rule r: If } x_1 \textit{ is } A_{1r} \textit{ and } \ldots \textit{ and } x_m \textit{ is } A_{mr}\textit{, then } y_1 \textit{ is } B_{1r} \textit{ and } \ldots \textit{ and } y_n \textit{ is } B_{nr} \end{array} \quad (1)$$

A fuzzy network with $p{\times}q$ nodes $\{N_{11} \ldots N_{p1}\}, \ldots, \{N_{1q} \ldots N_{pq}\}$, $p{\times}q$ node inputs $\{x_{11} \ldots x_{p1}\}, \ldots, \{x_{1q} \ldots x_{pq}\}$ taking linguistic terms from any admissible input sets, $p{\times}q$ node outputs $\{y_{11} \ldots y_{p1}\}, \ldots, \{y_{1q} \ldots y_{pq}\}$ taking linguistic terms from any admissible output sets, p

horizontal levels and q vertical layers in the general grid structure for this network can be described by Equation (2)

$$\begin{array}{lc} & Layer\ 1 \ldots\ldots\ldots\ldots\ldots Layer\ q \\ Level\ 1 & N_{11}(x_{11}, y_{11}) \ldots\ldots\ldots N_{1q}(x_{1q}, y_{1q}) \\ \ldots\ldots\ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ Level\ p & N_{p1}(x_{p1}, y_{p1}) \ldots\ldots\ldots N_{pq}(x_{pq}, y_{pq}) \end{array} \quad (2)$$

where the subscripts for the nodes specify their location in the grid structure and the subscripts for the associated inputs and outputs are identical with the ones for their nodes.

Each node in Equation (2) is a separate fuzzy system as the one described by Equation (1) whereby each node input and output is either of scalar or vector type. The levels in this grid structure represent a spatial hierarchy of the nodes in terms of subordination in space and the layers represent a temporal hierarchy in terms of consecutiveness in time. For completeness, the fuzzy network described by Equation (2) has a node in each cell of the grid structure but in general a grid structure may have empty cells.

Equation (2) does not give any information about the connections among the nodes in the fuzzy network. However, such information is contained by the sample connection structure in Equation (3) whereby the $p \times (q-1)$ node connections $\{z_{11,12} \ldots z_{p1,p2}\}, \ldots, \{z_{1q-1,1q} \ldots z_{pq-1,pq}\}$ take linguistic terms from the admissible sets for the associated node outputs and inputs

$$\begin{array}{lc} & Layer\ 1 \ldots\ldots\ldots\ldots\ldots Layer\ q\text{-}1 \\ Level\ 1 & z_{11,12} = y_{11} = x_{12} \ldots\ldots\ldots z_{1q-1,1q} = y_{1q-1} = x_{1q} \\ \ldots\ldots\ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ Level\ p & z_{p1,p2} = y_{p1} = x_{p2} \ldots\ldots\ldots z_{pq-1,pq} = y_{pq-1} = x_{pq} \end{array} \quad (3)$$

where for each connection the first subscript is identical with the subscript for its departure node and the second subscript is identical with the subscript for its arrival node.

Like each node input and output from the general grid structure in Equation (2), each node connection from the sample connection structure in Equation (3) can be either of scalar or vector type. For simplicity, this interconnection structure describes only connections that are of feedforward type and among adjacent nodes in the same level but it can be easily extended for connections that are of feedback type or among non-adjacent nodes in any levels.

As a fuzzy network represents an extension of a fuzzy system, i.e. it can be viewed as a system of fuzzy systems or a network whose nodes are fuzzy systems, some of the general presentation techniques for fuzzy systems can be used also for fuzzy networks. However, other presentation techniques that are specific to fuzzy networks are required for the simplification of a fuzzy network to a linguistically equivalent fuzzy system. These techniques use compressed information about nodes in fuzzy networks and they are discussed further below.

3. Linguistic Composition Approach

The proposed linguistic composition approach uses Boolean matrices for the presentation of individual rule bases in fuzzy networks and operations on these matrices for manipulating the rule bases. A Boolean matrix compresses the information from a rule base that is represented by a node. In this case, the row and column labels of the Boolean matrix are all possible permutations of linguistic terms of the inputs and the outputs for this rule base. The elements of the Boolean matrix are either '0's or '1's whereby each '1' reflects a present rule. The Boolean matrix presentation of the rule base from Equation (1) is given by Equation (4).

$$\begin{array}{cccc} & B_{11}\ldots B_{n1} & \ldots & B_{1r}\ldots B_{nr} \\ A_{11}\ldots A_{m1} & 1 & \ldots & 0 \\ \ldots & & \ldots & \\ A_{1r}\ldots A_{mr} & 0 & \ldots & 1 \end{array} \quad (4)$$

The proposed approach uses also topological expressions for the overall presentation of fuzzy networks and the connections among the individual rule bases. Like grid and interconnection structures, topological expressions describe the location of nodes and the connections among them. In this case, the subscripts of each node specify its location in the network whereby the first subscript gives the level number and the second subscript gives the layer number. Besides this, topological expressions specify all inputs, outputs and connections for the nodes. The topological expression presentation of the fuzzy network from Equations (2)-(3) is given by Equation (5).

$$\{[N_{11}]\ (x_{11} |\ z_{11,12}=y_{11}=x_{12})\ *\ \ldots\ *\ [N_{1q}]\ (z_{1q-1,1q}=y_{1q-1}=x_{1q}\ |\ y_{1q})\}\ + \qquad (5)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$+\ \{[N_{p1}]\ (x_{p1} |\ z_{p1,p2}=y_{p1}=x_{p2})\ *\ \ldots\ *\ [N_{pq}]\ (z_{pq-1,pq}=y_{pq-1}=x_{pq}\ |\ y_{pq})\}$$

As shown in Equation (5), each node in a topological expression is placed within a pair of square brackets '*[ ]*'. The inputs and the outputs for each node are placed within a pair of simple brackets '*( )*' right after the node. In this case, the inputs are separated from the outputs by a vertical slash '|'. Nodes in sequence are designated by the symbol '*' for horizontal relative location whereas nodes in parallel are designated by the symbol '+' for vertical relative location. Curly brackets *'{ }'* are used to specify the priority of linguistic composition operations in the fuzzy network, i.e. whether nodes with horizontal or vertical relative location have to be manipulated first.

Boolean matrices and topological expressions are very suitable for formal representation of fuzzy networks. While Boolean matrices describe fuzzy networks at a lower level of abstraction with respect to individual nodes, topological expressions describe these networks at a higher level of abstraction with respect to the whole network. In this context, Boolean matrices and topological expressions lend themselves easily to manipulation for the purpose of simplifying fuzzy networks to linguistically equivalent fuzzy systems using the linguistic composition approach. More details on this approach are presented below.

The linguistic composition approach is based on horizontal and vertical merging operations for nodes in fuzzy networks. These operations are binary in that can be applied to a pair of nodes. For simplicity, the operations are illustrated for nodes with scalar inputs, outputs and connections but their extension to the vector case is straightforward. The operations make use of Boolean matrices at the node level and topological expressions at the network level.

Horizontal merging can be applied to a pair of sequential nodes, i.e. nodes located in the same level of the fuzzy network. This operation merges the operand nodes from the pair into a single product node. The operation can be applied when the output from the first node is fed forward as an input to the second node in the form of a connection. In this case, the product node has the same input as the one to the first operand node and the same output as the one from the second operand node whereas the connection does not appear in the product node.

The horizontal merging operation is identical with Boolean matrix multiplication. The latter is similar to conventional matrix multiplication whereby each arithmetic multiplication is replaced by a '*min*' operation and each arithmetic addition is replaced by a '*max*' operation. In this case, the row labels of the product matrix are the same as the row labels of the first operand matrix whereas the column labels of the product matrix are the same as the column labels of the second operand matrix.

Therefore, if the first operand node is the rule base from Equation (1) that is presented by the Boolean matrix from Equation (4) and the second operand node is the rule base in Equation (6) that is presented by the Boolean matrix in Equation (7),

$$\begin{array}{c} \textit{Rule 1: If } y_1 \textit{ is } B_{11} \textit{ and ... and } y_n \textit{ is } B_{n1}\textit{, then } v_1 \textit{ is } C_{11} \textit{ and ... and } v_g \textit{ is } C_{g1} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \textit{Rule r: If } y_1 \textit{ is } B_{1r} \textit{ and ... and } y_n \textit{ is } B_{nr}\textit{, then } v_1 \textit{ is } C_{1r} \textit{ and ... and } v_k \textit{ is } C_{gr} \end{array} \quad (6)$$

$$\begin{array}{cccc} & C_{11}\ldots C_{g1} & \ldots & C_{1r}\ldots C_{gr} \\ B_{11}\ldots B_{n1} & 1 & \ldots & 0 \\ \ldots & & \ldots & \\ B_{1r}\ldots B_{nr} & 0 & \ldots & 1 \end{array} \quad (7)$$

the product node is the rule base in Equation (8) that is presented by the Boolean matrix in Equation (9)

$$\textit{Rule 1: If } x_1 \textit{ is } A_{11} \textit{ and ... and } x_m \textit{ is } A_{m1}\textit{, then } v_1 \textit{ is } C_{11} \textit{ and ... and } v_g \textit{ is } C_{g1} \quad (8)$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\textit{Rule r: If } x_1 \textit{ is } A_{1r} \textit{ and ... and } x_m \textit{ is } A_{mr}\textit{, then } v_1 \textit{ is } C_{1r} \textit{ and ... and } v_g \textit{ is } C_{gr}$$

$$\begin{array}{c|ccc} & C_{11}\ldots C_{g1} & \ldots & C_{1r}\ldots C_{gr} \\ A_{11}\ldots A_{m1} & 1 & \ldots & 0 \\ \ldots & & \ldots & \\ A_{1r}\ldots A_{mr} & 0 & \ldots & 1 \end{array} \quad (9)$$

In this case, the fuzzy system described by the rule base in Equation (6) is with *r* rules, *n* inputs $y_1 \ldots y_n$ taking linguistic terms from the input sets $\{B_{11},\ldots,B_{1r}\},\ldots,\{B_{n1},\ldots,B_{nr}\}$ and *g* outputs $v_1 \ldots v_g$ taking linguistic terms from the output sets $\{C_{11},\ldots,C_{1r}\},\ldots,\{C_{g1},\ldots,C_{gr}\}$. Similarly, the fuzzy system described by the rule base in Equation (8) is with *r* rules, *m* inputs $x_1 \ldots x_m$ taking linguistic terms from the input sets $\{A_{11},\ldots,A_{1r}\},\ldots,\{A_{m1},\ldots,A_{mr}\}$ and *g* outputs $v_1 \ldots v_g$ taking linguistic terms from the output sets $\{C_{11},\ldots,C_{1r}\},\ldots,\{C_{g1},\ldots,C_{gr}\}$. In general, the operand rule bases may have a different number of rules but the number of rules in the product rule base is equal to the number of rules in the first operand rule base.

The horizontal merging operation above can be described by the block-scheme in Figure 1 and the topological expression in Equation (10)

$$[N_{11}]\ (x_1,\ldots,x_m \mid y_1,\ldots,y_n) * [N_{12}]\ (y_1,\ldots,y_n \mid v_1,\ldots,v_g) = [N_{11*12}]\ (x_1,\ldots,x_m \mid v_1,\ldots,v_g) \quad (10)$$

where N_{11} and N_{12} are the two operand nodes from the fuzzy network and N_{11*12} is the product node for the fuzzy system. For simplicity, the notations used in Figure 1 are in a vector form.

Vertical merging can be applied to a pair of parallel nodes, i.e. nodes located in the same layer of the fuzzy network. This operation merges the operand nodes from the pair into a single product node. The operation can be applied when the outputs from the operand nodes are not fed as inputs to these nodes. In this case, the inputs to the product node represent the union of the inputs to the operand nodes whereas the outputs from the product node represent the union of the outputs from the operand nodes.

The vertical merging operation is identical with Boolean matrix Kroneker product that represents an expansion of the first operand matrix along its rows and columns. In particular, the product matrix is obtained by expanding each non-zero element from the first operand matrix to a block that is the same as the second operand matrix and by expanding each zero element from the first operand matrix to a zero block of the same dimension as the second operand matrix. In this case, the row labels of the product matrix are all possible permutations of row labels of the operand matrices whereas the column labels of the product matrix are all permutations of column labels of the operand matrices.

Therefore, if the first operand node is the rule base from Equation (1) that is presented by the Boolean matrix from Equation (4) and the second operand node is the rule base in Equation (11) that is presented by the Boolean matrix in Equation (12)

Rule 1: If v_1 *is* C_{11} *and … and* v_g *is* C_{g1}*, then* w_1 *is* D_{11} *and … and* w_h *is* D_{h1} (11)

…………………………………………………………………………

Rule s: If v_1 *is* C_{1s} *and … and* v_g *is* C_{gs}*, then* w_1 *is* D_{1s} *and … and* w_h *is* D_{hs}

	$D_{11}…D_{h1}$	…	$D_{1s}…D_{hs}$
$C_{11}…C_{g1}$	1	…	0
…		…	
$C_{1s}…C_{gs}$	0	…	1

(12)

the product node is the rule base in Equation (13) that is presented by the Boolean matrix in Equation (14)

Rule 1: If x_1 *is* A_{11} *and … and* x_m *is* A_{m1} *and* v_1 *is* C_{11} *and … and* v_g *is* C_{g1}*,* (13)
then y_1 *is* B_{11} *and … and* y_n *is* B_{n1} *and* w_1 *is* D_{11} *and … and* w_h *is* D_{h1}

…………………………………………………………………………

Rule r. s: If x_1 *is* A_{1r} *and … and* x_m *is* A_{mr} *and* v_1 *is* C_{1s} *and … and* v_g *is* C_{gs}*,*
then y_1 *is* B_{1r} *and … and* y_n *is* B_{nr} *and* w_1 *is* D_{1s} *and … and* w_h *is* D_{hs}

$$
\begin{array}{lccc}
 & B_{11}\ldots B_{n1}D_{11}\ldots D_{h1} & \ldots & B_{1r}..B_{nr}D_{1s}..D_{hs} \\
A_{11}\ldots A_{m1}C_{11}\ldots C_{g1} & 1 & \ldots & 0 \\
\ldots & & \ldots & \\
A_{1r}\ldots A_{mr}C_{1s}\ldots C_{gs} & 0 & \ldots & 1
\end{array} \tag{14}
$$

In this case, the fuzzy system described by the rule base in Equation (11) is with s rules, g inputs $v_1 \ldots v_g$ taking linguistic terms from the input sets $\{C_{11},\ldots,C_{1s}\},\ldots,\{C_{g1},\ldots,C_{gs}\}$ and h outputs $w_1 \ldots w_h$ taking linguistic terms from the output sets $\{D_{11},\ldots,D_{1s}\},\ldots,\{D_{h1},\ldots,D_{hs}\}$. However, the fuzzy system described by the rule base in Equation (13) is with $r.s$ rules, $m+g$ inputs $x_1 \ldots x_m$, $v_1 \ldots v_g$ taking linguistic terms from the input sets $\{A_{11},\ldots,A_{1r}\},\ldots,\{A_{m1},\ldots,A_{mr}\}$, $\{C_{11},\ldots,C_{1s}\},\ldots,\{C_{g1},\ldots,C_{gs}\}$ and $n+h$ outputs $y_1 \ldots y_g$, $w_1 \ldots w_h$ taking linguistic terms from the output sets $\{B_{11},\ldots,B_{1r}\},\ldots,\{B_{n1},\ldots,B_{nr}\}$, $\{D_{11},\ldots,D_{1s}\},\ldots,\{D_{h1},\ldots,D_{hs}\}$. The number of rules in the product rule base is equal to the product of the number of rules in the operand rule bases.

The vertical merging operation above can be described by the block-scheme in Figure 2 and the topological expression in Equation (15)

$$
\begin{gathered}
[N_{11}]\ (x_1,\ldots,x_m \mid y_1,\ldots,y_n) + [N_{21}]\ (v_1,\ldots,v_g \mid w_1,\ldots,w_h) = \\
[N_{11+21}]\ (x_1,\ldots,x_m, v_1,\ldots,v_g \mid y_1,\ldots,y_n, w_1,\ldots,w_h)
\end{gathered} \tag{15}
$$

where N_{11} and N_{21} are the two operand nodes from the fuzzy network and N_{11+21} is the product node for the fuzzy system. For simplicity, the notations used in Figure 2 are in a vector form.

The horizontal and vertical merging operations on nodes introduced above are quite basic in that they can be applied only to fairly simple fuzzy networks with a pair of nodes. However, a more complex fuzzy network may be with a large number of sequential and parallel nodes that have to be merged horizontally and vertically using the linguistic composition approach. This is possible due to the associativity property of the horizontal and vertical merging operations. These properties are proved below by theorems for scalar inputs, outputs and connections but the extension of the proofs to the vector case is straightforward.

The proofs presented below are based on binary relational presentation of Boolean matrices. A binary relation compresses further the information from a Boolean matrix representation of a rule base. In this case, the pairs in the binary relation are the permutations of linguistic terms of the inputs and the outputs from the row and column labels for the Boolean matrix. Therefore, each pair in the binary relation reflects a rule from the rule base. In this case, the Boolean matrices from Equations (4), (7), (9), (12) and (14) can be presented by the binary relations in Equations (16)-(20).

$$\{(A_{11}\ldots A_{m1}, B_{11}\ldots B_{n1}),\ \ldots\ ,(A_{1r}\ldots A_{mr}, B_{1r}\ldots B_{nr})\} \quad (16)$$

$$\{(B_{11}\ldots B_{n1}, C_{11}\ldots C_{g1}),\ \ldots\ ,(B_{1r}\ldots B_{nr}, C_{1r}\ldots C_{gr})\} \quad (17)$$

$$\{(A_{11}\ldots A_{m1}, C_{11}\ldots C_{g1}),\ \ldots\ ,(A_{1r}\ldots A_{mr}, C_{1r}\ldots C_{gr})\} \quad (18)$$

$$\{(C_{11}\ldots C_{g1}, D_{11}\ldots D_{h1}),\ \ldots\ ,(C_{1s}\ldots C_{gs}, D_{1s}\ldots D_{hs})\} \quad (19)$$

$$\{(A_{11}\ldots A_{m1}\, C_{11}\ldots C_{g1}, B_{11}\ldots B_{n1}\, D_{11}\ldots D_{h1}),\ \ldots\ ,(A_{1r}\ldots A_{mr}\, C_{1s}\ldots C_{gs}, B_{1r}\ldots B_{nr}\, D_{1s}\ldots D_{hs})\} \quad (20)$$

As binary relations are an alternative to Boolean matrices for representing nodes in fuzzy networks, they can also be used for horizontal and vertical merging operations on these nodes. In this case, horizontal merging is identical with standard relational composition whereas vertical merging is identical with a modified type of Cartesian product that is applied separately to the first and second elements from the pairs of the operand relations. These details of binary relations are used in Theorems 1-2 below.

When the property of associativity is related to the operation of horizontal merging, the latter is applied to three sequential nodes for the purpose of merging them into a single node. In particular, this property allows the merging of three operand nodes *A*, *B* and *C* into a product node *A*B*C* to take place as a sequence of two binary merging operations that can be applied either from left to right or from right to left. The property can be applied when the output from the first node A is fed forward as an input to the second node B in the form of a connection and the output from the second node B is fed forward as an input to the third node C in the form of another connection. In this case, the product node $A*B*C$ has the same input as the input to the first operand node A and the same output as the output from the third operand node C whereas the two connections do not appear in the product node.

Theorem 1: The operation of horizontal merging denoted by the symbol '*' is associative in accordance with Equation (21)

$$(A*B)*C = A*(B*C) = A*B*C \tag{21}$$

whereby the horizontal merging of any three operand nodes A, B and C from left to right is equivalent to their horizontal merging from right to left.

Proof 1: The proof is based on the use of binary relations for representing the operand nodes A, B and C. In this case, the elements of the relational pairs are denoted by the letter a in A, the letters a and c in B, and the letter c in C, as shown in Equations (22)-(24). For clarity, all pairs in the middle relation B are assumed to be composable with pairs from the left relation A and the right relation C. This is why the first and the second element of each pair in B are denoted by a and c, respectively, and not by b.

$$A = \{(a_1^1, a_2^1), \ldots, (a_1^p, a_2^p)\} \tag{22}$$

$$B = \{(a_2^1, c_1^1), \ldots, (a_2^1, c_1^q), \ldots, (a_2^p, c_1^1), \ldots, (a_2^p, c_1^q)\} \tag{23}$$

$$C = \{(c_1^1, c_2^1), \ldots, (c_1^q, c_2^q)\} \tag{24}$$

The first and the second element of any relational pair in A and C are denoted by the subscripts '*1*' and '*2*', respectively. However, the superscripts for the first and the second element of any relational pair in A and C are identical as they indicate the corresponding number for each pair. In particular, the relation A has p pairs and the relation C has q pairs. The subscripts for the first and the second element of any relational pair in B are '*2*' and '*1*', respectively. This is due to the requirement for left and right composability of *B*, i.e. the first element of each pair in *B* must be identical with a second element of a pair in *A* whereas the second element of each pair in *B* must be identical with a first element of a pair in *C*. In this case, the superscripts for the elements of the relational pairs in *B* don't have to be identical and therefore the relation *B* is assumed to have *p.q* pairs.

The horizontal composition of the operand relations *A* and *B* gives the temporary relation *A*B*, as shown in Equation (25)

$$A*B = \{(a_1^1, c_1^1), \ldots, (a_1^1, c_1^q), \ldots, (a_1^p, c_1^1), \ldots, (a_1^p, c_1^q)\} \quad (25)$$

Further on, the horizontal composition of the temporary relation *A*B* and the operand relation *C* gives the product relation *(A*B)*C*, as shown in Equation (26)

$$(A*B)*C = \{(a_1^1, c_2^1), \ldots, (a_1^1, c_2^q), \ldots, (a_1^p, c_2^1), \ldots, (a_1^p, c_2^q)\} \quad (26)$$

On the other hand, the horizontal composition of the operand relations *B* and *C* gives the temporary relation *B*C*, as shown in Equation (27)

$$B*C = \{(a_2^1, c_2^1), \ldots, (a_2^1, c_2^q), \ldots, (a_2^p, c_2^1), \ldots, (a_2^p, c_2^q)\} \quad (27)$$

In this case, the horizontal composition of the operand relation *A* and the temporary relation *B*C* gives the product relation *A*(B*C)*. As the latter is identical with the product relation *(A*B)*C* from Equation (26), this implies Equation (21) and concludes the proof.

When the property of associativity is related to the operation of vertical merging, the latter is applied to three parallel nodes for the purpose of merging them into a single node. In particular, this property allows the merging of three operand nodes *A*, *B* and *C* into a product node *A+B+C* to take place as a sequence of two binary merging operations that can be applied either from top to bottom or from bottom to top. The property can be applied when none of the outputs from any of the three nodes *A*, *B* and *C* are fed as any of the three inputs to these nodes. In this case, the input set to the product node *A+B+C* is the union of the inputs to the operand nodes *A*, *B* and *C* whereas the output set from the product node is the union of the outputs from the operand nodes.

Theorem 2: The operation of vertical merging denoted by the symbol '+' is associative in accordance with Equation (28)

$$(A+B)+C = A+(B+C) = A+B+C \tag{28}$$

whereby the vertical merging of any three operand nodes *A*, *B* and *C* from top to bottom is equivalent to their vertical merging from bottom to top.

Proof 2: The proof is based on the use of binary relations for representing the operand nodes *A*, *B* and *C*. In this case, the elements of the relational pairs are denoted by the letter *a* in *A*, the letter *b* in *B* and the letter *c* in *C*, as shown in Equations (29)-(31)

$$A = \{(a_1^1, a_2^1), \ldots, (a_1^p, a_2^p)\} \tag{29}$$

$$B = \{(b_1^1, b_2^1), \ldots, (b_1^q, b_2^q)\} \tag{30}$$

$$C = \{(c_1^1, c_2^1), \ldots, (c_1^r, c_2^r)\} \tag{31}$$

The first and the second element of any relational pair in *A*, *B* and *C* are denoted by the subscripts '*1*' and '*2*', respectively. However, the superscripts for the first and the second element of any relational pair in *A*, *B* and *C* are identical as they indicate the corresponding number for each pair. In particular, the relation *A* has *p* pairs, the relation *B* has *q* pairs and the relation *C* has *r* pairs.

The vertical composition of the operand relations *A* and *B* gives the temporary relation *A*+*B*, as shown in Equation (32)

$$A+B = \{(a_1^1\, b_1^1, a_2^1\, b_2^1), \ldots, (a_1^1\, b_1^q, a_2^1\, b_2^q), \ldots, (a_1^p\, b_1^1, a_2^p\, b_2^1), \ldots, (a_1^p\, b_1^q, a_2^p\, b_2^q)\} \tag{32}$$

Further on, the vertical composition of the temporary relation *A*+*B* and the operand relation *C* gives the product relation *(A*+*B)*+*C*, as shown in Equation (33)

$$(A+B)+C = \{(a_1^1\, b_1^q\, c_1^1,\, a_2^1\, b_2^q\, c_2^1), \ldots, (a_1^1\, b_1^q\, c_1^r,\, a_2^1\, b_2^q\, c_2^r), \ldots, \\ (a_1^p\, b_1^1\, c_1^1,\, a_2^p\, b_2^1\, c_2^1), \ldots, (a_1^p\, b_1^1\, c_1^r,\, a_2^p\, b_2^1\, c_2^r), \ldots, \\ (a_1^p\, b_1^q\, c_1^1,\, a_2^p\, b_2^q\, c_2^1), \ldots, (a_1^p\, b_1^q\, c_1^r,\, a_2^p\, b_2^q\, c_2^r)\} \quad (33)$$

On the other hand, the vertical composition of the operand relations *B* and *C* gives the temporary relation *B*+*C*, as shown in Equation (34)

$$B+C = \{(b_1^1\, c_1^1,\, b_2^1\, c_2^1), \ldots, (b_1^1\, c_1^r,\, b_2^1\, c_2^r), \ldots, (b_1^q\, c_1^1,\, b_2^q\, c_2^1), \ldots, (b_1^q\, c_1^r,\, b_2^q\, c_2^r)\} \quad (34)$$

In this case, the vertical composition of the operand relation *A* and the temporary relation *B*+*C* gives the product relation *A+(B+C)*. As the latter is identical with the product relation *(A+B)+C* from Equation (33), this implies Equation (28) and concludes the proof.

Although Theorems 1-2 prove the associativity property only for fuzzy networks with three sequential and parallel nodes, respectively, this property can be trivially extended for fuzzy networks with an arbitrary number of nodes. Therefore, this property can be viewed in the context of the linguistic composition approach as the glue that makes the building blocks for simplification of a fuzzy network to a fuzzy system, i.e. the horizontal and merging operations on nodes, stick together. In this case, the generalisation of the associativity property for horizontal and vertical merging can be presented by Equations (35)-(36)

$$(((\ldots((A*B)*C*)\ldots*X)*Y)*Z) = (A*(B*(C*\ldots*(X*(Y*Z))\ldots))) = \\ A*B*C*\ldots*X*Y*Z \quad (35)$$

$$(((\ldots((A+B)+C+)\ldots+X)+Y)+Z) = (A+(B+(C+\ldots+(X+(Y+Z))\ldots))) = \\ A+B+C+\ldots+X+Y+Z \quad (36)$$

where *A, B, C, … , X, Y, Z* are operand nodes from a fuzzy network with a single level and layer, respectively.

The associativity property of horizontal and merging operations from Theorems 1-2 provides the basis for the application of the linguistic composition approach to complex fuzzy networks with an arbitrary number of nodes. In particular, the nodes can be merged quite flexibly, i.e. from left to right or right to left within the same level and from top to bottom or from bottom to top within the same layer. In this case, the resulting single equivalent system is the same irrespective of the order of application of the binary merging operations.

The linguistic composition approach can be applied in the context of the three types of fuzzy systems discussed earlier – with single rule base, multiple rule bases and networked rule bases. This process consists of two stages whereby a multiple rule base system such as HFS is first converted into a networked fuzzy system such as FN and then the latter is composed into a single rule base system such as SFS. The theoretical validity of the above two-stage process is proved by means of topological expressions in Theorem 3.

Theorem 3: A HFS with set of m inputs $\{x_1, x_2, \dots, x_m\}$, a set of $m-1$ network nodes $\{N_{11}, N_{12}, \dots, N_{1,m-1}\}$, a set of $m-2$ connections $\{z_1, z_2, \dots, z_{m-2}\}$ and a single output y, as described by the block-scheme in Figure 3 and the topological expression in Equation (37)

$$[N_{11}]\ (x_1, x_2 \mid z_1) * [N_{12}]\ (z_1, x_3 \mid z_2) * \dots * [N_{1,m-1}]\ (z_{m-2}, x_m \mid y) \qquad (37)$$

can be represented as a SFS with the same set of m inputs, a single network node N, no connections and the same single output, as described by the block-scheme in Figure 4 and the topological expression in Equation (38)

$$[*_{p=1}^{m-1}\ (N_{1p} + +_{q=p+1}^{m-1} I_{qp})]\ (x_1, x_2, \dots, x_m \mid y) \qquad (38)$$

where $N = *_{p=1}^{m-1}\ (N_{1p} + +_{q=p+1}^{m-1} I_{qp})$.

Proof 3: The HFS from Equation (37) can first be converted into a FN by representing all identity mappings propagating through any layers in the grid structure with the set of identity nodes $\{I_{21}\}, \dots, \{I_{m-1,1}, I_{m-1,2}, \dots\}$. This FN can be described by the block-scheme in Figure 5 and the topological expression in Equation (39)

$$\{[N_{11}]\ (x_1, x_2 \mid z_1) + [I_{21}]\ (x_3 \mid x_3) + \dots + [I_{m-1,1}]\ (x_m \mid x_m)\} * \qquad (39)$$

$$\{[N_{12}]\ (z_1, x_3 \mid z_2) + \dots + [I_{m-1,2}]\ (x_m \mid x_m)\} *$$

$$\dots\dots\dots\dots\dots\dots *$$

$$[N_{1,m-1}]\ (z_{m-2}, x_m \mid y)$$

where each network node has two inputs and one output as opposed to each identity node that has one input and one output. In this case, the input to each identical node is identical with the output from the same node.

The FN can then be composed into a SFS by merging first vertically and then horizontally all network and identity nodes into a single network node $N = *_{p=1}^{m-1} (N_{1p} + +_{q=p+1}^{m-1} I_{qp})$. In this case, the SFS is like a single node FN with the same set of m inputs $\{x_1, x_2, \ldots, x_m\}$ and the same single output y as the HFS. This SFS can de described by the topological expression from Equation (38) that uses prefix notation for the horizontal merging operation and a mixture of infix/prefix notation for the vertical merging operation. This concludes the proof.

Theorem 3 is applicable only to single-output systems but it can be extended trivially for multiple-output systems. In this case, the HFS would have a set of n outputs $\{y_1, y_2, \ldots, y_n\}$ and it could be presented as a set of n independent systems. Therefore, the two-step process from the theorem above would be repeated for each independent system and its output.

As opposed to most existing approaches where the focus is to improve efficiency by representing a SFS as a HFS with rule bases of smaller size, the focus of the linguistic composition approach is to improve accuracy by representing a HFS as a SFS with a single FID sequence. Apart from accuracy, transparency is also improved by means of the modular rule bases in the FN that reflect the subsystems of the modelled system. This is not the case in most existing approaches where the HFS is a mathematical representation of the SFS that does not reflect the subsystems of the modelled system.

When SFS, HFS and FN are used for modelling, the quality of the associated models can be quantified using performance indicators. In particular, three model performance indicators are introduced further below. They are called Accuracy Index (AI), Efficiency Index (EI) and

Transparency Index (TI). These performance indicators represent modifications of performance indicators used for fuzzy systems that can also be used for fuzzy networks.

The first performance indicator AI reflects the accuracy of the model by means of the absolute difference between the model and the data, as shown by Equation (40)

$$AI = sum_{i=1}^{nl}\ sum_{j=1}^{qil}\ sum_{k=1}^{vji}\ (|y_{ji}^{k} - d_{ji}^{k}| / vij) \quad (40)$$

The notations in Equation (40) are as follows: *nl* is the number of nodes in the last layer, *qil* is the number of outputs from the *i*-th node in the last layer, *vji* is the number of discrete values for the *j*-th output from the *i*-th node in the last layer, y_{ji}^{k} is the simulated *k*-th discrete value for the *j*-th output from the *i*-th node in the last layer and d_{ji}^{k} is the measured *k*-th discrete value for the *j*-th output from the *i*-th node in the last layer, '*sum*' is a symbol for arithmetic summation and '| |' is a symbol for absolute value. Identity nodes are included in this indicator alongside any other nodes in the last layer because their outputs also have to be compared with the data. As a model is more accurate when the absolute difference between the model and the data given by Equation (40) is smaller, a lower AI implies better accuracy.

The second performance indicator EI reflects the efficiency of the model by means of the overall number of rules, as shown by Equation (41)

$$EI = sum_{i=1}^{n}\ (q_i^{FID} \,.\, r_i^{FID}) \quad (41)$$

The notations in Equation (41) are as follows: *n* is the number of non-identity network nodes, q_i^{FID} is the number of outputs from the *i*-th non-identity node with an associated FID sequence, r_i is the number of rules for the *i*-th non-identity node with an associated FID sequence and '*sum*' is a symbol for arithmetic summation. Identity nodes are excluded from this indicator because they are virtual nodes for converting a HFS into a FN that do not affect the efficiency. As a model is more efficient when the overall number of rules given by Equation (41) is smaller, a lower EI implies better efficiency.

The third performance indicator TI reflects the transparency of the model by means of the extent of its opaqueness from the inside, as shown by Equation (42)

$$TI = (p + q) / (n + m) \quad (42)$$

The notations in Equation (42) are as follows: p is the overall number of inputs, q is the overall number of outputs, n is the number of non-identity nodes, m is the number of non-identity connections and '*sum*' is a symbol for arithmetic summation. Identity nodes are excluded from this indicator as they are virtual nodes for converting a HFS into a FN that do not affect the transparency. As a model is more transparent when the extent of its opaqueness from the inside given by Equation (42) is smaller, i.e. the overall number of inputs and outputs is bigger while at the same time the number of sub-models and connections is smaller, a lower TI implies better transparency.

The use of the three model performance indicators discussed above is quite important for the theoretical evaluation of the proposed approach. Recent research trends show that using multiple performance indicators is more informative for model evaluation than just a single performance indicator. This allows models to be evaluated with regard to different performance related requirements.

4. Simulation Results

The linguistic composition approach is applied to a case study on the first stage of an ore flotation process by using available data from the mining industry. This application is similar to the one described in [41] in that it uses the same approach but the case study here is quite different from the ones presented there. The process deals with the enrichment of raw ore and it is implemented by processing a mixture of ore, water and reagents called pulp. For simplicity, the first stage of the flotation process is referred to as 'flotation process' further in the paper whereby the second stage is taken into account only implicitly in the considerations. Also, the relevant rule bases for the flotation process are given only partially due to space limitations.

The inputs to the flotation process x_1, x_2, x_3 are the concentration of copper in the pulp in [%], the concentration of iron in the pulp in [%] and the pulp debit in [l/min]. The output y from the flotation process is the intermediate concentration of copper in the pulp in [%]. In this context, the output y has the same physical meaning as the first input x_1 but it usually takes higher values than x_1 due to the increased concentration of the copper.

The flotation process can be modelled by a SFS, as shown by the topological expression in Equation (43). The notations used are as follows: N is the rule base for the SFS, the first input x_1 is the concentration of copper in the pulp, the second input x_2 is the concentration of iron in the pulp, the third input x_3 is the pulp debit and the output y is the intermediate concentration of copper in the pulp.

$$[N]\ (x_1, x_2, x_3 \mid y) \tag{43}$$

The flotation process can also be modelled by a HFS, as shown by the topological expression in Equation (44). The notations used are as follows: N_{11} is the first rule base for the HFS, N_{12} is the second rule base for the HFS, the inputs x_1, x_2, x_3 and the output y are the same as the ones for the SFS, whereas the connection z has the same meaning as the output y for the SFS but it represents the provisional intermediate concentration of copper in the pulp.

$$[N_{11}]\ (x_1, x_2 \mid z) * [N_{12}]\ (z, x_3 \mid y) \tag{44}$$

The flotation process can be modelled by a FN as well, as shown by the topological expression in Equation (45). Most notations used are the same as the ones for the HFS. The only new notation is the identity rule base I_{21} representing the propagation of the identity mapping x_3 through the first layer of the grid structure. In this context, N_{11} and N_{12} are the network rules bases and they are usually of non-identity type.

$$\{[N_{11}]\ (x_1, x_2 \mid z) + I_{21}\,(x_3 \mid x_3)\} * [N_{12}]\ (z, x_3 \mid y) \tag{45}$$

Using the proposed linguistic composition approach, the HFS with multiple rule bases can be converted first to a FN with networked rule bases. The latter can then be simplified to a SFS with a single rule base, as shown by the topological expression in Equation (46). In this equation, the composite rule base $(N_{11} + I_{21}) * N_{12}$ for the SFS is derived in accordance with the topological expression in Equation (38) and the associated merging operations for rule bases by means of Boolean matrices.

$$[(N_{11} + I_{21}) * N_{12}]\ (x_1, x_2, x_3 \mid y) \tag{46}$$

The inputs x_1, x_2, x_3, the output y and the connection z are presented by eleven linguistic terms each, as shown in Figures 6-10. These terms belong to the set *{low5, low4, low3, low2, low1, average, high1, high2, high3, high4, high5}* and they are represented by triangular fuzzy membership functions that cover uniformly the whole variation range for the inputs, the output and the connection.

The linguistic terms in the rule bases for the SFS, the HFS and the FN are represented by positive integers. In this case, the substitutions are in accordance with Equation (47)

$$\begin{gathered} low5 = 1,\ low4 = 2,\ low3 = 3,\ low2 = 4,\ low1 = 5,\ average = 6, \\ high1 = 7,\ high2 = 8,\ high3 = 9,\ high4 = 10,\ high5 = 11 \end{gathered} \tag{47}$$

The initial part of the rule base for the SFS is shown in Table 1. This rule base is derived from data about the product pricing process and in accordance with Equation (43). The derivation is done using a clustering approach whereby the rules represent an approximation of the input-output data points from the data set for the process.

The initial parts of the two rule bases for the HFS are shown in Tables 2-3. These rule bases are derived from data about the two sub-processes within the flotation process and in accordance with Equation (44). The derivation is done using a clustering approach whereby the rules represent an approximation of the input-output data points from the data sets for the sub-processes.

The initial parts of the identical rule base and the single equivalent rule base for the FN are shown in Tables 4-5. In this case, the single equivalent rule base is derived in accordance with Equation (46).

The block-schemes for the SFS, the HFS and the FN are given in Figures 11-13. The simulation results for the SFS, the HFS and the FN are shown in Figures 14-16 where the data and the model output are presented together by the ´o´ and the ´x´ marker, respectively. In this case, each of the three models is simulated for all available 76 input data points. With the exception of a few data points around data point 40 and 70 where the model output has a more significant deviation from the data output, for the remaining about data points the model output is very close to the data output.

5. Performance Evaluation

The proposed linguistic composition approach is evaluated comparatively in terms of accuracy, efficiency and transparency. In particular, a FN that uses the linguistic composition approach and a single FID sequence is compared to a SFS that uses a single FID sequence and a HFS that uses a multiple FID sequence. The main purpose of the evaluation is to demonstrate the transparency of the FN in relation to the SFS and the accuracy of the FN in relation to the HFS while also observing the efficiency of the FN in relation to both the SFS and the HFS.

The comparative evaluation of the SFS, the HFS and the FN is shown in Table 6. This evaluation uses the performance indicators from Equations (40)-(42). Table 6 shows that in terms of feasibility the FN is superior to the SFS and equivalent to the HFS. With regard to accuracy, the FN is inferior to the SFS but superior the HFS. As far as efficiency is concerned, the FN is equivalent to the SFS but inferior to the HFS. And finally, in terms of transparency, the FN is superior to the SFS and equivalent to the HFS.

In summary, Table 6 shows that the proposed FN approach compares very well with the established SFS and HFS approaches. In particular, FN is slightly superior, inferior or equivalent to SFS and HFS depending on the model performance indicator used. Although the results obtained and the related conclusions are case study dependent and may not be valid for other data sets, the advantages of the proposed method are expected to be comparable in other similar circumstances, i.e. for modelling processes with structure in terms of interacting sub-processes.

6. Conclusion

The proposed linguistic composition approach provides a novel theoretical framework for fuzzy systems with networked rule bases called fuzzy networks. These networks compare well with established fuzzy systems such as standard fuzzy systems with a single rule base and hierarchical fuzzy systems with multiple rule bases in terms of accuracy, efficiency and transparency. The approach is suitable for modelling processes characterised by uncertainty, dimensionality and structure.

The proposed approach has been applied successfully to a case study from the mining industry. The case study describes an ore flotation process process that is characterised by uncertainty in terms of noisy data from sensors, dimensionality in terms of large number of process variables and structure in terms of cascaded interconnected sub-processes. The approach has been validated quantitatively using established metrics for accuracy, efficiency and transparency in a comparative fashion against two established approaches.

The theoretical framework shows a novel application of discrete mathematics and systems theory. It uses Boolean matrices and binary relations for presenting fuzzy network nodes as well as topological expressions and connectionism ideas for presenting fuzzy networks as a whole.

In this framework, a fuzzy network represents an extension of a standard fuzzy system and a hierarchical fuzzy system. In particular, a fuzzy network is a compact way of representing a hierarchical fuzzy system by means of a standard fuzzy system whereby structure is dealt with during the linguistic composition process. The main purpose in this case is to achieve transparency while improving accuracy and maintaining efficiency.

Apart from being an extension, a fuzzy network is also like a bridge between a standard fuzzy system and a hierarchical fuzzy system. This is done by means of the linguistic composition process whereby a hierarchical fuzzy system is first converted into a fuzzy network and the latter is then composed into a standard fuzzy system. During this process some performance indicators may be improved without deteriorating other indicators. Therefore, this bridging capability of fuzzy networks improves the flexibility of fuzzy systems in terms of modelling depending on the specific requirements to these models.

The linguistic composition approach can be used in a wide range of application areas where the knowledge or data about the modelled process can be provided in a modular fashion, i.e. for each interacting sub-process by means of individual rule bases. Such modular processes are quite common in many areas such as decision making, manufacturing, communications and transport. In this case, the interacting modules can be decision units, manufacturing cells, communication nodes or traffic junctions. Also, the approach can be easily extended for non-fuzzy rule based systems that use deterministic or probabilistic logic.

Acknowledgement

The first author would like to thank the Faculty of Technology at the University of Portsmouth for the granted research sabbatical that made possible the writing of this paper.

References

[1] Fernandez-Caballero A (2009) Contribution of fuzziness and uncertainty to modern artificial intelligence. Fuzzy Sets and Systems 160/2:129-129

[2] Garibaldi J M, Ozen T (2007) Uncertain fuzzy reasoning: a case study in modelling expert decision making. IEEE Transactions on Fuzzy Systems 15/1:16-30

[3] Gil-Aluja (2004) Fuzzy sets in the management of uncertainty. Springer, Berlin

[4] Chen B, Liu X (2005) Fuzzy approximate disturbance decoupling of MIMO nonlinear systems by backstepping and application to chemical processes. IEEE Transactions on Fuzzy Systems 13/6:832-847

[5] Chen B, Tong S, Liu X (2007) Fuzzy approximate disturbance decoupling of MIMO nonlinear systems by back stepping approach. Fuzzy sets and systems 158/10:1097-1125

[6] Wang W J, Lin W W (2005) Decentralized PDC for large-scale fuzzy systems. IEEE Transactions on Fuzzy Systems 13/6:779-786

[7] Wang W J, Luoh L (2004) Stability and stabilization of fuzzy large-scale systems. IEEE Transactions on Fuzzy Systems 12/3:309-315

[8] Gegov A (2007) Complexity management in fuzzy systems: a rule base compression approach. Springer, Berlin

[9] Kim K J, Cho S B (2007) Personalized mining of web documents link structures and fuzzy concept networks. Applied Soft Computing 7/1:398-410

[10] Xu C, Shin Y C (2007) Interaction analysis of MIMO nonlinear systems based on fuzzy basis function network model. Fuzzy sets and systems 158/18:2013-2015

[11] Zhang H, Dang C, Li C (2009) Decentralized H-infinity filter design for discrete-time interconnected fuzzy systems. IEEE Transactions on Fuzzy Systems 17/6:1428-1440

[12] Zhang T P (2001) Stable adaptive fuzzy sliding mode control of interconnected systems. Fuzzy Sets and Systems 122/1:5-19

[13] Buckley, J J (2005) Simulating fuzzy systems. Springer, Berlin

[14] Piegat A (2001) Fuzzy modelling and control. Springer, Berlin

[15] Ross T (2004) Fuzzy logic with engineering applications. Wiley, Chichester

[16] Ji-Cheng Duan, Fu-Lai Chung (2001) Cascaded fuzzy network model based on syllogistic fuzzy reasoning. IEEE Transactions on Fuzzy Systems 9/2:293-306

[17] Lendek Z, Babuska R, De Schutter B (2009) Stability of cascaded fuzzy systems and observers. IEEE Transactions on Fuzzy Systems 17/3:641-653

[18] Mar J, Lin H T (2005) A car-following collision prevention control device based on the cascaded fuzzy inference system. Fuzzy Sets and Systems 150/3:457-473

[19] Yeh Z M, Li K H (2004) A systematic approach for designing multistage fuzzy control systems. Fuzzy Sets and Systems 143/2:251-273

[20] Bucolo M, Fortuna L, La Rosa M (2004) Complex dynamics through fuzzy chains. IEEE Transactions on Fuzzy Systems 12/3:289-295

[21] Hall L O (2001) Rule chaining in fuzzy expert systems. IEEE Transactions on Fuzzy Systems 9/6:822-828

[22] Igel C, Temme K H (2004) The chaining syllogism in fuzzy logic. IEEE Transactions on Fuzzy Systems 12/6:849-853

[23] Aja-Fernandez S, Alberola-Lopez C (2008) Matrix modelling of hierarchical fuzzy systems. IEEE Transactions on Fuzzy Systems 16/3:585-599

[24] Chen Y, Yang B, Abraham A, Peng L (2007) Automatic design of hierarchical Takagi-Sugeno type fuzzy systems using evolutionary algorithms. IEEE Transactions on Fuzzy Systems 15/3:385-397

[25] Cheong F, Lai R (2007) Designing a hierarchical fuzzy logic controller using the differential evolution approach. Applied Soft Computing 7/2:481-491

[26] Joo M G, Lee J S (2005) A class of hierarchical fuzzy systems with constraints on the fuzzy rules. IEEE Transactions on Fuzzy Systems 13/2:194-203

[27] Lee M L, Chung H Y, Yu F M (2003) Modelling of hierarchical fuzzy systems. Fuzzy Sets and Systems 135/2:343-361

[28] Gegov A (2010) Fuzzy networks for complex systems: a modular rule base approach. Springer, Berlin

[29] Landago M, Rio M J, Perez A (2001) A note on smooth approximation capabilities of fuzzy systems. IEEE Transactions on Fuzzy Systems 9/2:229-237

[30] Li Y M, Shi Z K, Li Z H (2002) Approximation theory of fuzzy systems based upon genuine many-valued implications –MIMO cases. Fuzzy Sets and Systems 130/2:159-174

[31] Moreno-Velo F J, Baturone I, Barriga A, Sanchez-Solano S (2007) Automatic tuning of complex fuzzy systems with X-fuzzy. Fuzzy sets and systems 158/18:2026-2038

[32] Wan F, Shang H, Wang L X, Sun Y X (2005) How to determine the minimum number of fuzzy rules to achieve given accuracy: a computational geometric approach to SISO case. Fuzzy Sets and Systems 150/2:199-209

[33] Gil-Aluja (2004) Fuzzy sets in the management of uncertainty. Springer, Berlin

[34] Joo M G, Sudkamp T (2009) A method of converting a fuzzy system to a two-layered hierarchical fuzzy system and its run-time efficiency. IEEE Transactions on Fuzzy Systems 17/1:93-103

[35] Pal N R, Eluri V K, Mandal G K (2002) Fuzzy logic approaches to structure preserving dimensionality reduction. IEEE Transactions on Fuzzy Systems 10/3:277-286

[36] Xiong N, Litz L (2002) Reduction of fuzzy control rules by means of premise learning – method and case study. Fuzzy Sets and Systems 132/2:217-231

[37] Fiordaliso A (2001) A constrained Takagi-Sugeno fuzzy system that allows for better interpretation and analysis. Fuzzy Sets and Systems 118/2:307-318

[38] Guillaume S (2001) Designing fuzzy inference systems from data: an interpretability-oriented review. IEEE Transactions on Fuzzy Systems 9/3:426-443

[39] Kumar M, Stoll R, Stoll N (2006) A robust design criterion for interpretable fuzzy models with uncertain data. IEEE Transactions on Fuzzy Systems 14/2:314-328

[40] Zhou S M, Gan J Q (2008) Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling. Fuzzy sets and systems 159/23:3091-3131

[41] Gegov A, Arabikhan F, Petrov N (2015) Linguistic composition based modelling by fuzzy networks with modular rule bases. Fuzzy sets and systems 269:1-29

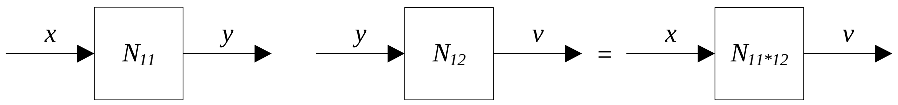

Figure 1: Horizontal merging of rule bases

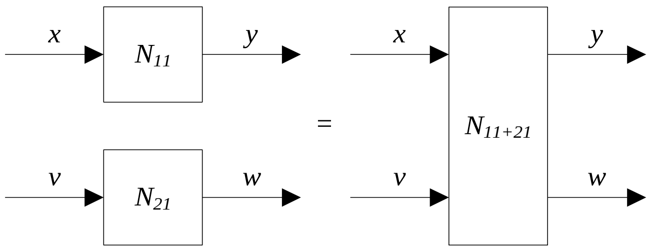

Figure 2: Vertical merging of rule bases

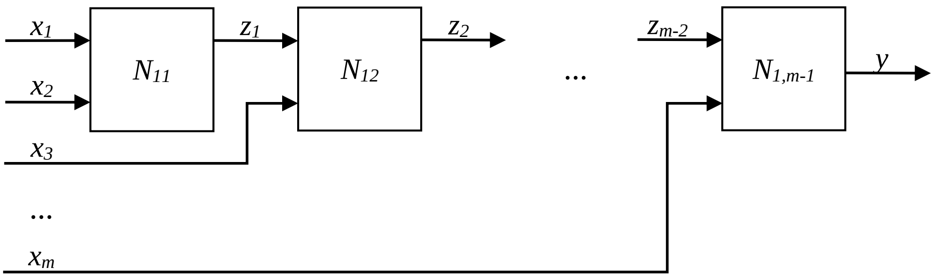

Figure 3: Hierarchical fuzzy system

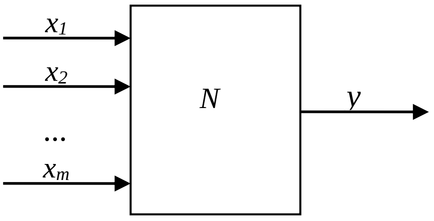

Figure 4: Standard fuzzy system

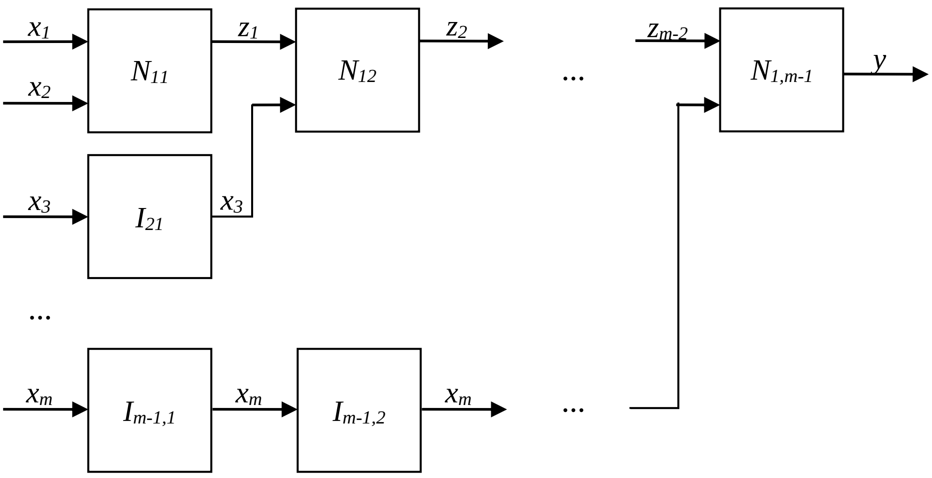

Figure 5: Fuzzy network

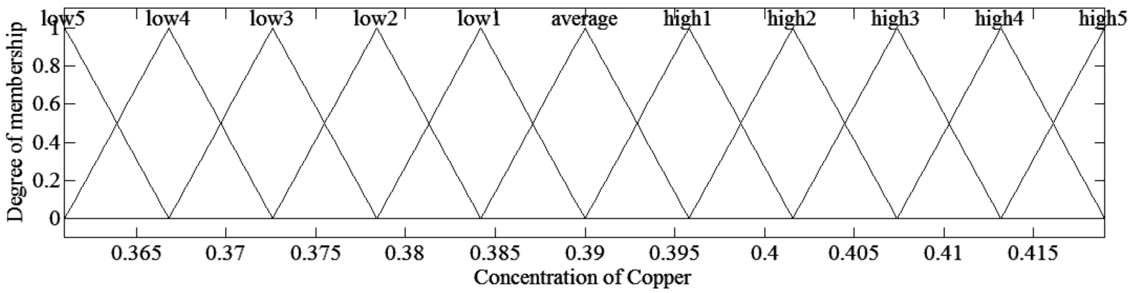

Figure 6: Linguistic terms for first input

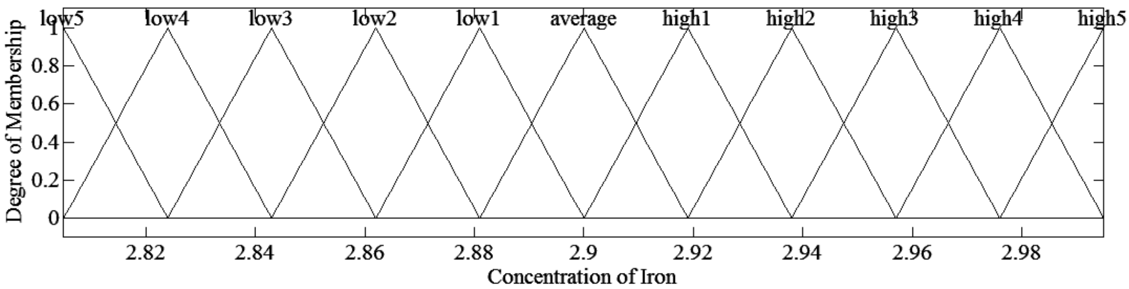

Figure 7: Linguistic terms for second input

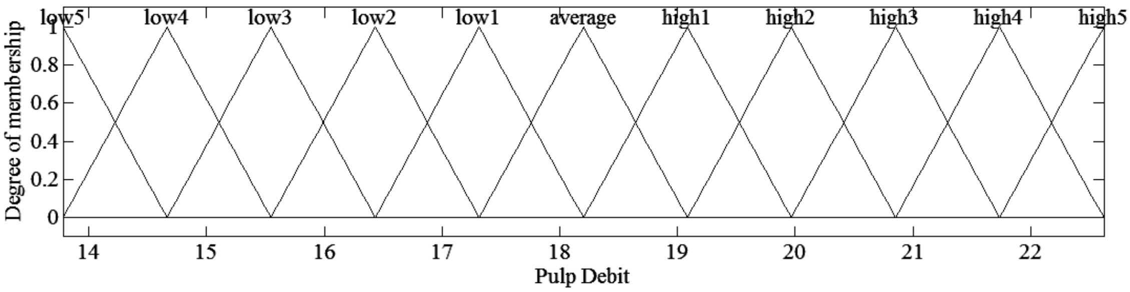

Figure 8: Linguistic terms for third input

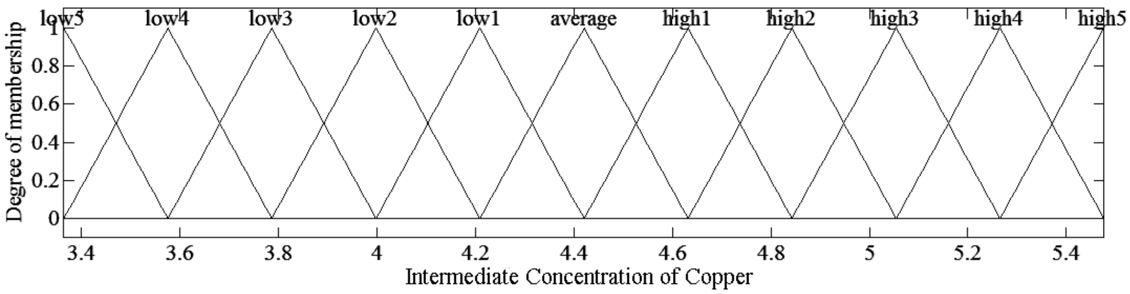

Figure 9: Linguistic terms for output

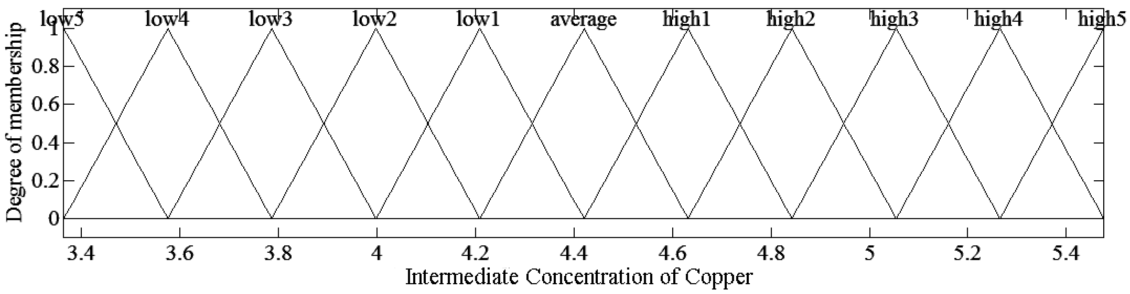

Figure 10: Linguistic terms for connection

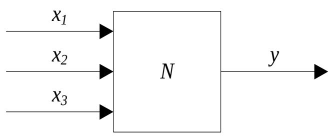

Figure 11: Standard fuzzy system for case study

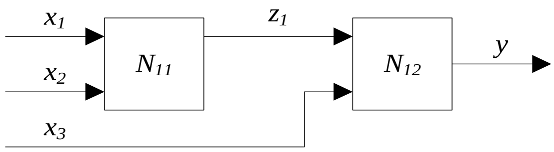

Figure 12: Hierarchical fuzzy system for case study

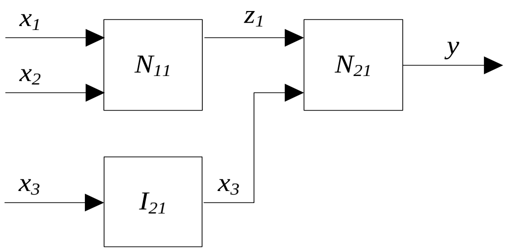

Figure 13: Fuzzy network for case study

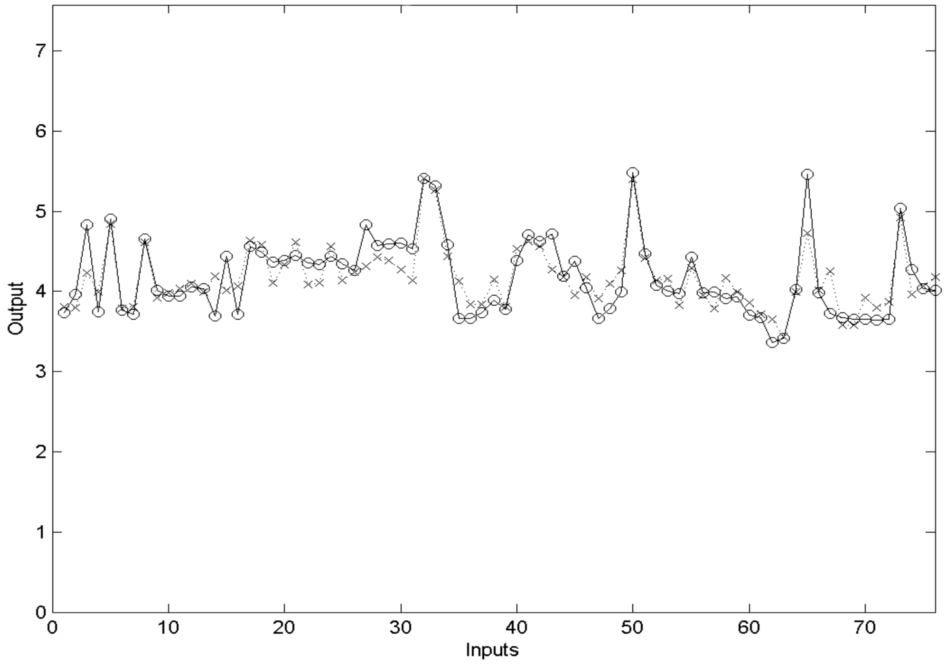

Figure 14: Simulation results for standard fuzzy system

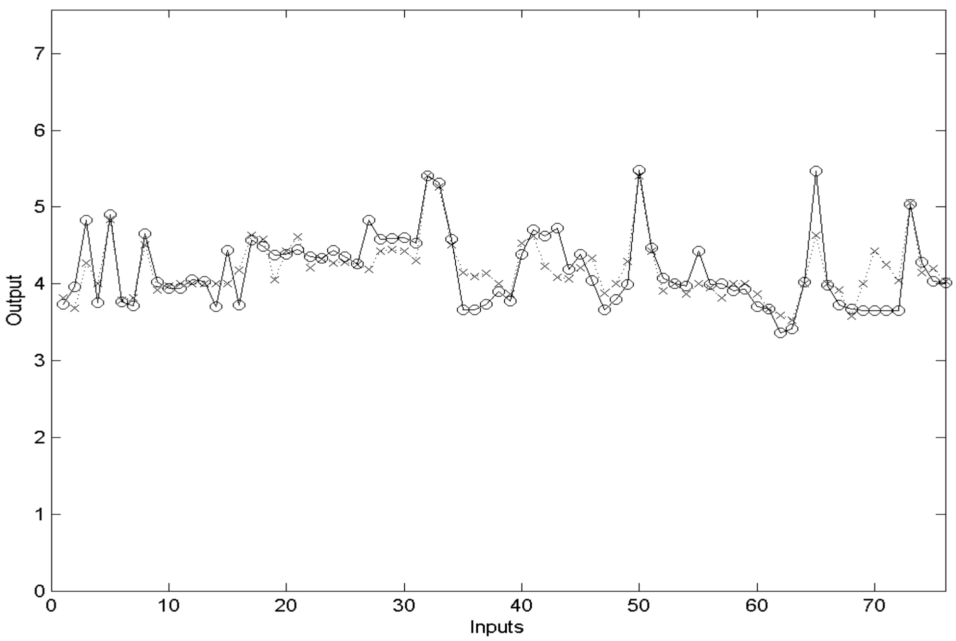

Figure 15: Simulation results for hierarchical fuzzy system

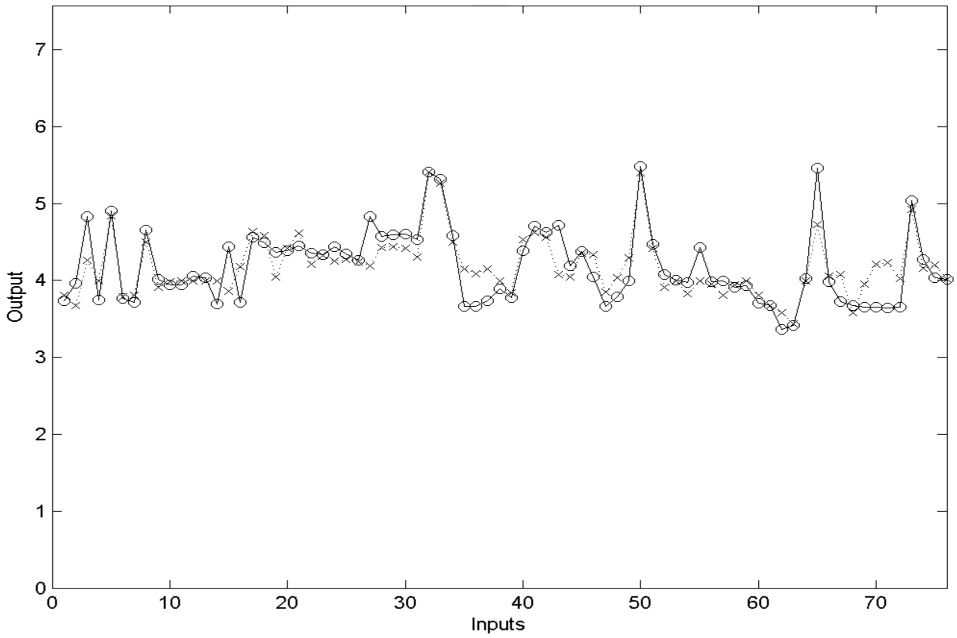

Figure 16: Simulation results for fuzzy network

Table 1: Initial part of rule base for standard fuzzy system

Rule number	x_1, x_2, x_3	y
1	[1 2 10]	7
2	[1 4 10]	6
3	[1 5 11]	10
…	…	…

Table 2: Initial part of first rule base for hierarchical fuzzy system

Rule number	x_1, x_2	z
1	[1 2]	7
2	[1 4]	6
3	[1 5]	10
…	…	…

Table 3: Initial part of second rule base for hierarchical fuzzy system

Rule number	z, x_3	y
1	[1 3]	1
2	[1 4]	1
3	[2 3]	2
…	…	…

Table 4: Initial part of identity rule base for fuzzy network

Rule number	x_3	x_3
1	[1]	1
2	[2]	2
3	[3]	3
…	…	…

Table 5: Initial part of single equivalent rule base for fuzzy network

Rule number	x_1, x_2, x_3	y
1	[1 2 4]	7
2	[1 2 5]	7
3	[1 2 6]	7
…	…	…

Table 6: Comparative evaluation of three fuzzy models

Performance indicator	Standard fuzzy system	Hierarchical fuzzy system	Fuzzy network
Accuracy	4.35	4.76	4.60
Efficiency	1331	242	1331
Transparency	4	1.33	1.33