

Universidade Federal de Juiz de Fora
Engenharia Elétrica
Programa de Pós-Graduação em Engenharia Elétrica

Renan Piazzaroli Finotti Amaral

**Type-1 and Singleton Fuzzy Logic System Trained by a Fast Scaled
Conjugate Gradient Methods for Dealing With Classification Problems**

Juiz de Fora

2017

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Amaral, Renan P. F..

Type-1 and Singleton Fuzzy Logic System Trained by a Fast Scaled
Conjugate Gradient Methods for Dealing With Classification Problems /
Renan Piazzaroli Finotti Amaral. – 2017.

55 f. : il.

Orientador: Moisés Vidal Ribeiro

Dissertação de Mestrado – Universidade Federal de Juiz de Fora, Enge-
nharia Elétrica. Programa de Pós-Graduação em Engenharia Elétrica, 2017.

1. Fuzzy Logic System. 2. Multiclass classification. 3. Scaled Conjugate
Gradient. 4. Hessian-free. I. Vidal Ribeiro, Moisés, orient. II. Título.

Renan Piazzaroli Finotti Amaral

**Type-1 and Singleton Fuzzy Logic System Trained by a Fast Scaled
Conjugate Gradient Methods for Dealing With Classification Problems**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora, na área de concentração em sistemas eletrônicos, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Moisés Vidal Ribeiro

Juiz de Fora

2017

Renan Piazzaroli Finotti Amaral

Type-1 and Singleton Fuzzy Logic System Trained by a Fast Scaled
Conjugate Gradient Methods for Dealing With Classification Problems

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora, na área de concentração em sistemas eletrônicos, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Aprovada em: 01 de setembro de 2017

BANCA EXAMINADORA



Prof. Dr. Moisés Vidal Ribeiro - Orientador
Universidade Federal de Juiz de Fora



Prof. Dr. Eduardo Pestana de Aguiar
Universidade Federal de Juiz de Fora



Prof. Dr. Ivo Chaves da Silva Junior
Universidade Federal de Juiz de Fora



Prof. Dr. Frederico Gadelha Guimarães
Universidade Federal de Minas Gerais

ACKNOWLEDGEMENTS

To God, for giving me health, family, friends, and for guiding me to the right choices in my life.

To my parents, Cleide and Devailton, for love, affection, principles and for my character formation.

To my brothers, Rafaele and Rafael for brotherhood, companionship, and friendship. To all my relatives from Extrema and Juiz de Fora for all support during the master course.

To my girlfriend, for companionship, affection, comprehension and for love. Thanks for always making my life light and happy.

To Prof. Eduardo, for invited me to work in scientific researches in graduating course, which encouraged me to achieve the master's degree. Thanks for all assistance, advices and guidance, helping me to conclude this work.

To Prof. Moisés, for the opportunity to work with him, for guidance and for all assistance to finish this work.

To my colleagues from LCOM, for all support in the disciplines.

To all my friends for friendship and brotherhood.

“Our world, our life, our destiny, are dominated by uncertainty; this is perhaps the only statement we may assert without uncertainty.”

(Bruno De Finetti)

ABSTRACT

This thesis presents and discusses improvements in the type-1 and singleton fuzzy logic system for dealing with classification problems. Two training methods are addressed, the scaled conjugate gradient, which uses the second order information approximating the multiplication of the Hessian matrix \mathbf{H} by the directional vector \mathbf{v} (i.e. $\mathbf{H}\mathbf{v}$), and the same method using the differential operator $\mathcal{R}\{\cdot\}$ to compute the exact value of $\mathbf{H}\mathbf{v}$. Also, in order to adapt the fuzzy model to handle multiclass classification problems, it is developed a novel fuzzy model with a vector as output. All proposals are tested through the performance metrics analysis based on data sets provided by UCI Machine Learning Repository. The reported results show the high convergence speed and better classification rates of the proposed training methods than others presented in the literature. Additionally, the novel fuzzy model has a significant reduction in computational and classifier complexity, especially when the number of classes in classification problem increases.

Key-words: fuzzy logic system, multiclass classification, scaled conjugate gradient, hessian-free.

LIST OF FIGURES

Figure 1 – Type-1 fuzzy logic system with a single output.	16
Figure 2 – Type-1 fuzzy logic system multiple output.	17
Figure 3 – The algorithm of the proposed SCG method for training T1-FLS and T1-FLSMO.	24
Figure 4 – Sonar data set: performance of T1-FLS on 4th fold. (a) Accuracy. (b) MSE.	28
Figure 5 – Pima data set: performance of T1-FLS on 1st fold. (a) Accuracy. (b) MSE.	28
Figure 6 – Ionosphere data set: performance of T1-FLS on 3rd fold. (a) Accuracy. (b) MSE.	28
Figure 7 – Contraceptive data set: performance of T1-FLSMO on 4th fold. (a) Accuracy. (b) MSE.	30
Figure 8 – Ecoli data set: performance of T1-FLSMO on 3rd fold. (a) Accuracy. (b) MSE.	31
Figure 9 – Iris data set: performance of T1-FLSMO on 4th fold. (a) Accuracy. (b) MSE.	31
Figure 10 – Wine data set: performance of T1-FLSMO on 5th fold. (a) Accuracy. (b) MSE.	31
Figure 11 – Comparison of the quantity of update parameters for T1-FLS with OvA and T1-FLSMO for different quantities of outputs for a data set with 10 input features.	33

LIST OF TABLES

Table 1 – Details of datasets to perform the training method analysis.	27
Table 2 – Performance in terms of the mean and deviation for each metric with 95% t-student confidence interval.	29
Table 3 – Details of data sets used to perform the classification proposal analysis.	30
Table 4 – Performance in terms of the mean and deviation for each metric with 95% t-student confidence interval.	32

ACRONYMS

ANFIS artificial neuro-fuzzy inference system

CG conjugate gradient

FBF fuzzy basis function

FL fuzzy logic

FLS fuzzy logic system

GFS genetic fuzzy system

MCP multiclass classification problem

MSE mean squared error

OvA one-vs-all

OvO one-vs-one

SCG scaled conjugate gradient

SD steepest descent

T1-FLS type-1 and singleton fuzzy logic system

T1-FLSMO type-1 and singleton fuzzy logic system multiple output

CONTENTS

1	INTRODUCTION	11
1.1	OBJECTIVES	12
1.2	WORK ORGANIZATION	13
2	TYPE-1 AND SINGLETON FUZZY LOGIC SYSTEM FOR CLASSIFICATION PROBLEMS	14
2.1	PROBLEM FORMULATION	15
2.2	THE PROPOSED TYPE-1 AND SINGLETON FUZZY LOGIC SYS- TEM MULTIPLE OUTPUT	17
2.2.1	Relationship with the type-1 and singleton fuzzy logic system single output	19
3	T1-FLSMO AND T1-FLS TRAINED BY SCALED CONJU- GATE GRADIENT AND SCALED CONJUGATE GRADI- ENT WITH DIFFERENTIAL OPERATOR $\mathcal{R}\{\cdot\}$ METHODS	20
3.1	PROBLEM FORMULATION	22
3.2	T1-FLSMO TRAINED BY SCALED CONJUGATE GRADIENT	22
3.2.1	The SCG method with the differential operator $\mathcal{R}\{\cdot\}$	24
4	NUMERICAL RESULTS	26
4.1	TRAINING METHOD COMPARISON	27
4.1.1	Convergence speed analysis	27
4.1.2	Classification rate analysis	29
4.2	CLASSIFICATION MODEL ANALYSIS COMPARISON	29
4.2.1	Convergence speed analysis	30
4.2.2	Classification rate analysis	32
4.2.3	Computational complexity analysis	33
5	CONCLUSION	34
5.1	FUTURE WORKS	35
	REFERENCES	36
	Appendix A – Deduction of the $H(\mathbf{w}^{(q)})\mathbf{v}^{(q)}$ from $\mathcal{R}\{\cdot\}$ opera- tor for T1-FLSMO	41

Appendix B – Deduction of the $H(w^{(q)}v^{(q)})$ from $\mathcal{R}\{\cdot\}$ operator for T1-FLS 48

Appendix C – Publications 55

1 INTRODUCTION

The fuzzy logic (FL) was introduced by Lotfi A. Zadeh in 1965 inspired by the processes of human perception and cognition. While in the classical logic exist only truth and false, in the FL was developed the mathematical concept of partial truth and partial false. Based on this concept, the fuzzy logic system (FLS) emerged by promising to deal with problems owning uncertainties that may not be modeled by well-established theories, such as statistical theory. Over the years, the applications that use FLS has spread across different fields of knowledge. In these fields, the most common applications of FLS are control and automation [1], supervised and unsupervised (clustering) pattern recognition [2, 3], regression [4] and prediction [5].

The output of FLS has been widely discussed in the literature, owing the FLS has only one output. This characteristic imply in difficulties when dealing with multiclass classification problems (MCP). Usually the classifier which has this limitation needs a decomposition strategy applied to solve it, reducing the initial MCP in several binary classification problems. In other words, many two-classes classifiers must be designed, and as consequence, the computational complexity increases. In [6] was used a FLS to deal with MCP by employing pairwise learning decomposition strategy, resulting on improvement of the FLS performance. The authors in [7] proposed a fuzzy support vector machine to obtain the fuzzy rules by one-vs-all decomposition strategy. Also, in [8] was used a fuzzy support vector machine in fault diagnosis of wind turbine classification; however, the decomposition strategy adopted in this work was one-vs-one decomposition strategy. All works aforementioned use a decomposition strategy, even FLS combined or not with other models, evidencing the high limitation of FLS model to deal with MCP.

Another challenging issue about FL, is how to determine the number and the type of rules to use. Based on inference of IF-THEN rules, the existing set of fuzzifications, t-norms, t-conorms, memberships functions and defuzzifications result in an explosive number of distinct FLS. Furthermore, two main approaches are widely investigated to design FLS, the first is a merged model associating the fuzzy rule based method with the neural network, in which the neural network is a non-linear function used to identify the extremely non-linear system parameters. This merged model is well-know called by artificial neuro-fuzzy inference system (ANFIS) [9]. This approach is used to solve several problems, such as, in [10] is used an ANFIS to control the continuously variable transmission ratio to extract the maximal wind energy through the wind turbine. The researchers in [11], identify the dialect from Assamese speech using ANFIS through of prosodic features, the comparison performed in this work showed that the ANFIS provides around 23% improvement than feed forward neural network. The authors in [12] proposed a new ANFIS classification technique for data mining, using the multilayer perceptron backpropagation network and fuzzy set theory. The work concluded that this

new technique offer improvements over previous ANFIS classification techniques. Also, a hybrid classification technique integrating the fuzzy c-means clustering-piloting Particle Swarm Optimization with ANFIS was proposed in [13], aiming to perform a non-intrusive load monitoring system to identify the power-intensive household.

In another direction to design the FLS, the second main approach is to use the evolutionary optimization method associated with FLS, such as genetic fuzzy system (GFS) [14]. The GFS is a robust method applied in many kind of problems, in [15] the authors use it within a pairwise learning framework for increase the detection rate of a intrusion detection system. The method consists in use the higher interpretability of fuzzy rules with the divide-and-conquer learning scheme. Moreover, [16] proposes an adaptive GFS to optimize rules and membership functions for medical data classification process. The quantitative, qualitative and comparative analysis performed show that the adaptive GFS obtained better accuracy when compared with the existing GFS. In [17] was proposed a methodology to obtain GFS under the iterative rule learning approach to forecast energy consumption of an office building. Even having good results, the evolutionary algorithms are computationally expensive by themselves due to the large number of iterations needed to reach convergence [18]. On the other hand, the design of FLS based on first and second information [19], which updates a fixed number of rules created initially by user, results in lower computational cost and has the tendency to use fewer rules than evolutionary fuzzy approaches.

Based on the aforementioned discussions, the following problems emerged: the limitation of FLS to handle MCP and the design of FLS. The former problem is well established in the literature, in fact, the FLS was designed to have only one output. As a result, it cannot be directly applied to handle problems demanding two or more outputs. For instance, MCP must be broken into several two-classes problems to allow the use of FLS to handle it. In another perspective, the latter problem focuses on approach to design the FLS without merged it with any model. The approach should reduce the computational complexity and the dependence of user's defined parameters. Both problems are very interesting to be investigated, especially the latter when does not exists the knowledge from the specialist.

1.1 OBJECTIVES

Aiming to deal with the well posed problems, the objective of this thesis are summarized as follows:

- To propose the extension of type-1 and singleton FLS single output (T1-FLS) to multiple outputs (T1-FLSMO), enabling to deal with MCP without the need of binary decomposition strategies, and consequently, to avoid its computational

complexity. In addition, to present performance analyses in terms of accuracy and convergence speed based on well-known data sets provided by UCI Machine Learning Repository [20].

- To introduce the use of scaled conjugate gradient (SCG) into training procedures for T1-FLS and T1-FLSMO, which can reduce the dependence of user's defined parameters, increase the classification performance during the training phase through the approximation of the multiplication of the Hessian matrix \mathbf{H} by the directional vector \mathbf{v} (i.e., $\mathbf{H}\mathbf{v}$). Additionally, to show how the differential operator $\mathcal{R}\{\cdot\}$ can be successfully used to compute the exact value of $\mathbf{H}\mathbf{v}$ and, as consequence, to come up with additional increase in the convergence speed during the training phase and computational complexity reduction when compared with the computational of \mathbf{H} and \mathbf{v} .

1.2 WORK ORGANIZATION

This thesis is organized as follows:

- Chapter 2 formulates the problem and introduces T1-FLSMO for dealing with MCP. In this regard, will be described a special attention to T1-FLS is given in the formulation problem to emphasize the need for introducing the T1-FLSMO.
- Chapter 3 introduces the SCG and SCGR training methods for T1-FLS and T1-FLSMO. The aspects of SCG and others gradient training methods are discussed and its adaptation for T1-FLS and T1-FLSMO are presented. Also, details of the differential operator $\mathcal{R}\{\cdot\}$ to compute the exact value of $\mathbf{H}\mathbf{v}$ for T1-FLS and T1-FLSMO are provided.
- Chapter 4 discussed performance results based on two approaches: The prime analysis the influence of the training methods, while the latter focuses on the performance in terms of the FLS models discussed for dealing with MCP.
- Chapter 5 presents the main aspects and conclusions of the proposed model and training methods in this thesis, summarizing the main contributions and the directions for future works.

Additionally, the Appendix A and B presents the detailed deduction of equations to compute the $\mathbf{H}\mathbf{v}$ for T1-FLSMO and T1-FLS respectively using $\mathcal{R}\{\cdot\}$.

2 TYPE-1 AND SINGLETON FUZZY LOGIC SYSTEM FOR CLASSIFICATION PROBLEMS

The multiclass classification problem is one of the principal branch in machine learning, formulated by the discrimination of patterns in more than two classes (discriminating between two classes is denominated binary classification). This problem exists in distinct research fields, such as biomedical [21], surveillance and security [22], computer vision [23], [24], [25], industrial [26], [27], and many others.

In general, the technique used to solve MCP may be organized into two categories: having a single output (binary classifier) and having multiple outputs (multiclass classifier). The binary classifier handles MCP using different strategies, the simplest and most applied are One-vs-One (OvO) [28] and One-vs-All (OvA) [29]. The OvO strategy groups each pair of classes by decomposing the original set of classes in several binary subsets of classes, requiring to use $\Upsilon(\Upsilon - 1)/2$ classifiers, where $\Upsilon \in \mathbb{N}^*$ is the number of classes. At final decision stage of the OvO strategy, the chosen class is obtained by a voting scheme performed by all classifiers and the class with the highest votes win. In the OvA strategy, the classification problem is decomposed in Υ binary problems, and each problem is formed to distinguish one class from all remaining classes. Several contributions show that the OvO strategy is better than OvA [30,31]; however, this advantage is somehow attenuated because an increasing and relevant computational cost is associated with the number of the classes increases.

Regarding MCP solved by FLS, it is necessary a decomposition strategy due to the fact that FLS has only one output. Based on this strategy, the researchers improve the performance of FLS by focusing on improvements on the training methods. In this regard, the authors in [32] proposed a adaptation of the inference in fuzzy association rule-based classification model, aiming to produce more accurate aggregations and, as consequence, to improve the classification through the OvO and OvA strategies. Moreover, in [33] was used a FLS-based classification technique with binary decomposition strategy (i.e. OvO and OvA) to deal with MCP, attempting to study the best method in decision stage of classification. Furthermore, the authors in [2] improved the classification performance in MCP introducing the CG training method, which uses the second order information for training FLS. In addition, [27] improved the training method of FLS in terms of computational complexity and convergence speed, by using a cost function derived from the Adaptive Filter Theory [34], which was named Set-Membership adaptive training technique. This technique can reduces remarkably the computational complexity and increase the convergence speed during the training phase, in comparison to the steepest descent (SD) method. The aforementioned proposal improve indirectly the solution of MCP using FLS and, as a consequence, exists a rising of fuzzy rules demanded to design FLS for several binary classification problems. The solution of this problem based on FLS

demands the design of FLS with multiple outputs.

In an attempt to overcome the limitation of FLS for handling MCP, this chapter proposes the extension of T1-FLS to the so-called T1-FLS multiple outputs (T1-FLSMO). The most important aspect of T1-FLSMO is that to deal with MCP with only one classifier, consequently, it can remarkably reduce the computational complexity of the FLS-based classifier when OvO and OvA strategies are taken into account.

This chapter is organized as follows: Section 2.1 presents the T1-FLS with single output and formulates the problem for introducing T1-FLSMO, and in Section 2.2 details the idea behind T1-FLSMO and shows that T1-FLS is a particular case of T1-FLSMO.

2.1 PROBLEM FORMULATION

The classification problem may be stated as mapping between a vector $\mathbf{x} \in \mathbb{R}^{P \times 1}$ and a label. The input space is thereby divided into *decision regions* whose boundaries are called *decision boundaries* or *decision surfaces* [35]. In this context, each decision region is assigned to one class. The FLS classifier maps the input vector to the decision regions by using IF-THEN rules. T1-FLS was proposed in [19], for handling classification problem, by adopting singleton fuzzification, max-product composition, product implication¹ and height defuzzification. As result, the output of T1-FLS is expressed as [19]

$$f_s(\mathbf{x}) = \frac{\sum_{l=1}^M \theta_l \prod_{k=1}^P \mu_{F_k^l}(x_k)}{\sum_{l=1}^M \prod_{k=1}^P \mu_{F_k^l}(x_k)}, \quad (2.1)$$

where x_k is the k -th element of the vector \mathbf{x} , which is constituted by P features, \prod denotes the product operator, $\mu_{F_k^l}(x_k)$ is the membership function associated with the k -th input of the l -th rule and θ_l is the weight associated with the l -th rule, $l = 1, \dots, M$. Figure 1 shows a diagram block for T1-FLS. The subscript 's' in $f_s(\mathbf{x})$ makes it clear that this is a type-1 and singleton FLS with single output. T1-FLS adopts the gaussian membership function, which is given by

$$\mu_{F_k^l}(x_k) = \exp \left\{ -\frac{1}{2} \left(\frac{x_k - m_{F_k^l}}{\sigma_{F_k^l}} \right)^2 \right\}, \quad (2.2)$$

where $m_{F_k^l}$ and $\sigma_{F_k^l}^2$ are the mean and the variance respectively, for a given set of input-output pairs $(\mathbf{x}^{(q)} : y^{(q)})$, where 'q' denote the q -th iteration.

By using T1-FLS, the solution of binary classification problem may be understood as the minimization of a given cost function. Usually, the cost function is expressed as [19]

$$\mathbf{J}(\mathbf{w}^{(q)}) = \frac{1}{2} [f_s(\mathbf{x}^{(q)}) - y^{(q)}]^2, \quad (2.3)$$

¹ t-norm

where $q = 1, \dots, Q$, and Q is the number of patterns used for training the T1-FLS, $\mathbf{w}^{(q)}$ denotes the vector of parameters for T1-FLS at the q -th iteration. Note that the vector $\mathbf{w}^{(q)}$ is expressed as

$$\mathbf{w}^{(q)} = \left[m_{F_1^1}(q), \dots, m_{F_P^1}(q), \dots, m_{F_1^M}(q), \dots, m_{F_P^M}(q), \dots, \right. \\ \left. \sigma_{F_1^1}(q), \dots, \sigma_{F_P^1}(q), \dots, \sigma_{F_1^M}(q), \dots, \sigma_{F_P^M}(q), \dots, \right. \\ \left. \theta_1(q), \dots, \theta_M(q) \right]^T. \quad (2.4)$$

It is important to emphasize that T1-FLS has only a single output, this characteristic imply in binary decomposition strategy dependence when MCP applies. This dependence turn this model less attractive if the number of classes are large. At this moment, it is important to highlight that the defuzzification is responsible for T1-FLS having only one output. Therefore, it seems to be the starting point for introducing modification in FLSs that will allow them to have multiple outputs. Focusing on defuzzification of T1-FLS, Section 2.2 introduces a simple and effective modification that allow a T1-FLS has several outputs.

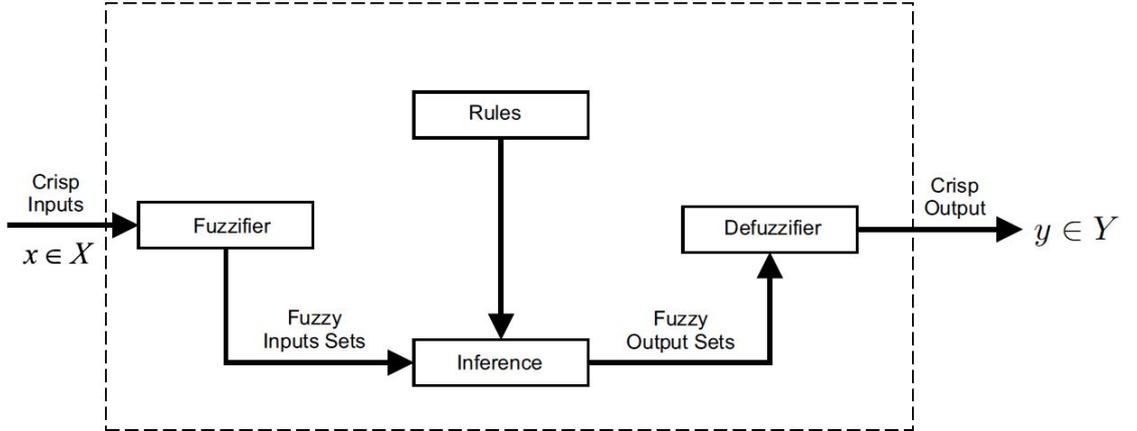


Figure 1: Type-1 fuzzy logic system with a single output.

2.2 THE PROPOSED TYPE-1 AND SINGLETON FUZZY LOGIC SYSTEM MULTIPLE OUTPUT

The extension of the T1-FLS multiple output will be called type-1 and singleton fuzzy logic system multiple output (T1-FLSMO). The inference block of T1-FLSMO is shown in Figure 2. In other words, T1-FLSMO is the T1-FLS with multiple outputs.

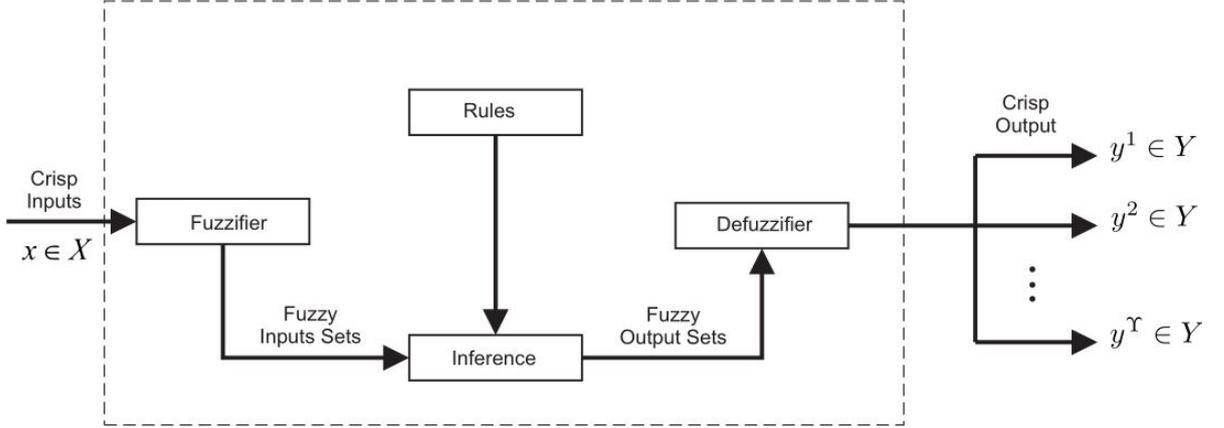


Figure 2: Type-1 fuzzy logic system multiple output.

In this context, let us assume that T1-FLS adopt singleton fuzzification, max product composition, product implication² and height defuzzification. Then, it is well-established that in height defuzzification, each rule output fuzzy set is replaced by a singleton (θ_i) at the point having maximum membership in that output set, and afterward, calculates the centroid of the type-1 set comprised of these singletons. Instead of have a (θ_i) only for each rule, T1-FLSMO assumes the existence of a singleton (θ_i) for each output (t) too, forming the matrix of singletons for height defuzzification, which is expressed as

$$\Theta = \begin{bmatrix} \theta_1^1 & \theta_2^1 & \cdots & \theta_M^1 \\ \theta_1^2 & \theta_2^2 & \cdots & \theta_M^2 \\ \vdots & \vdots & \ddots & \vdots \\ \theta_1^\Upsilon & \theta_2^\Upsilon & \cdots & \theta_M^\Upsilon \end{bmatrix}, \quad (2.5)$$

where $t = 1, \dots, \Upsilon$. Therefore, the output of T1-FLSMO using the height defuzzification is given by

$$\mathbf{y}^{(q)} = \mathbf{f}_{s_{mo}}(\mathbf{x}^{(q)}) = \Theta \Phi(\mathbf{x}^{(q)}), \quad (2.6)$$

where

$$\mathbf{f}_{s_{mo}}(\mathbf{x}^{(q)}) = [f_{s_{mo}}^1(\mathbf{x}^{(q)}), f_{s_{mo}}^2(\mathbf{x}^{(q)}), \dots, f_{s_{mo}}^\Upsilon(\mathbf{x}^{(q)})]^T, \quad (2.7)$$

$$\Phi(\mathbf{x}^{(q)}) = [\phi_1(\mathbf{x}^{(q)}) \quad \phi_2(\mathbf{x}^{(q)}) \quad \cdots \quad \phi_M(\mathbf{x}^{(q)})]^T \quad (2.8)$$

² t-norm

and $\phi_l(\mathbf{x})$ is a fuzzy basis function (FBF) [19], which is expressed as

$$\phi_l(\mathbf{x}) = \frac{\prod_{k=1}^P \mu_{F_k^l}(x_k)}{\sum_{l=1}^M \prod_{k=1}^P \mu_{F_k^l}(x_k)}. \quad (2.9)$$

The subscript s_{mo} in equation (2.6) means that it refers to a type-1 and singleton FLS with multiple outputs. In other words, the T1-FLS uses the rules to compute the single output, and the T1-FLSMO uses all the same rules to compute several outputs, which is similar with a neural network with several outputs. Given a set of input-output pairs $(\mathbf{x}^{(q)} : \mathbf{y}^{(q)})$, where 'q' denote the q -th iteration, the MCP can be understood as a minimization of the following cost function:

$$J(\mathbf{w}^{(q)}) = \frac{1}{2} \sum_{t=1}^{\Upsilon} (f_{s_{mo}}^t(\mathbf{x}^{(q)}) - y^t(q))^2, \quad (2.10)$$

where $\mathbf{w}^{(q)}$ denotes the parameter vector for the T1-FLSMO, which is expressed as

$$\begin{aligned} \mathbf{w}^{(q)} = & \left[m_{F_1^1}(q), \dots, m_{F_P^1}(q), \dots, m_{F_1^M}(q), \dots, m_{F_P^M}(q), \dots, \right. \\ & \left. \sigma_{F_1^1}(q), \dots, \sigma_{F_P^1}(q), \dots, \sigma_{F_1^M}(q), \dots, \sigma_{F_P^M}(q), \dots, \right. \\ & \left. \theta_1^1(q), \dots, \theta_M^1(q), \dots, \theta_M^{\Upsilon}(q), \dots, \theta_M^{\Upsilon}(q) \right]^T. \end{aligned} \quad (2.11)$$

Also the gradient vector $\nabla \mathbf{J}(\mathbf{w}^{(q)})$ is given by

$$\begin{aligned} \nabla \mathbf{J}(\mathbf{w}^{(q)}) = & \left[\frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial m_{F_1^1}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial m_{F_P^1}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial m_{F_1^M}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial m_{F_P^M}(q)}, \dots, \right. \\ & \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \sigma_{F_1^1}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \sigma_{F_P^1}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \sigma_{F_1^M}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \sigma_{F_P^M}(q)}, \dots, \\ & \left. \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \theta_1^1(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \theta_M^1(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \theta_1^{\Upsilon}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \theta_M^{\Upsilon}(q)} \right]^T, \end{aligned} \quad (2.12)$$

in which the derivatives of $\mathbf{J}(\mathbf{w}^{(q)})$ with regard to parameters $m_{F_k^l}(q)$, $\sigma_{F_k^l}(q)$ and $\theta_l^t(q)$, are as follows:

$$\frac{\partial J(\mathbf{w}^{(q)})}{\partial m_{F_k^l}(q)} = \sum_{t=1}^{\Upsilon} ([f_{s_{mo}}^t(\mathbf{x}^{(q)}) - y^t(q)] [\theta_l^t(q) - f_{s_{mo}}^t(\mathbf{x}^{(q)})]) a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}), \quad (2.13)$$

$$\frac{\partial J(\mathbf{w}^{(q)})}{\partial \sigma_{F_k^l}(q)} = \sum_{t=1}^{\Upsilon} ([f_{s_{mo}}^t(\mathbf{x}^{(q)}) - y^t(q)] [\theta_l^t(q) - f_{s_{mo}}^t(\mathbf{x}^{(q)})]) b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \quad (2.14)$$

and

$$\frac{\partial J(\mathbf{w}^{(q)})}{\partial \theta_l^t(q)} = [f_{s_{mo}}^t(\mathbf{x}^{(q)}) - y^t(q)] \phi_l(\mathbf{x}^{(q)}), \quad (2.15)$$

where $a_{F_k^l}(q)$ and $b_{F_k^l}(q)$ are given by:

$$a_{F_k^l}(q) = \frac{x_k^{(q)} - m_{F_k^l}(q)}{\sigma_{F_k^l}^2(q)} \quad (2.16)$$

and

$$b_{F_k^l}(q) = \frac{\left(x_k^{(q)} - m_{F_k^l}(q)\right)^2}{\sigma_{F_k^l}^3(q)}, \quad (2.17)$$

respectively. Finally, it is important to notice that the concepts of T1-FLSMO related to the extended version of the height defuzzification, can be introduced in other kind of defuzzification types of LFS.

2.2.1 Relationship with the type-1 and singleton fuzzy logic system single output

A particular case of the T1-FLSMO is obtained when it has a unique output $\Upsilon = 1$, which reduces T1-FLSMO to T1-FLS [19]. Using the singleton fuzzification, max-product composition, product implication³ and height defuzzification, the T1-FLS output has the form given by equation (2.1) which also can be expressed as

$$f_s(\mathbf{x}) = \sum_{l=1}^M \theta_l \phi_l(\mathbf{x}). \quad (2.18)$$

Given a set of input-output pairs $(\mathbf{x}^{(q)} : y^{(q)})$, (2.10) reduces to (2.3) and the gradient given by (2.13), (2.14) and (2.15) result in the following equation:

$$\frac{\partial J(\mathbf{w}^{(q)})}{\partial m_{F_k^l}(q)} = [f_s(\mathbf{x}^{(q)}) - y(q)] [\theta_l(q) - f_s(\mathbf{x}^{(q)})] a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}), \quad (2.19)$$

$$\frac{\partial J(\mathbf{w}^{(q)})}{\partial \sigma_{F_k^l}(q)} = [f_s(\mathbf{x}^{(q)}) - y(q)] [\theta_l(q) - f_s(\mathbf{x}^{(q)})] b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \quad (2.20)$$

and

$$\frac{\partial J(\mathbf{w}^{(q)})}{\partial \theta_l(q)} = [f_s(\mathbf{x}^{(q)}) - y(q)] \phi_l(\mathbf{x}^{(q)}), \quad (2.21)$$

respectively. A comparison between the deduction of T1-FLS in [19] and (2.19)–(2.21) confirms that T1-FLS is a particular case of T1-FLSMO.

³ t-norm

3 T1-FLSMO AND T1-FLS TRAINED BY SCALED CONJUGATE GRADIENT AND SCALED CONJUGATE GRADIENT WITH DIFFERENTIAL OPERATOR $\mathcal{R}\{\cdot\}$ METHODS

In general, supervised training methods mainly make use of first-order information of cost function. A popular gradient based method is the steepest descent (SD), which uses the first-order information. Its form and its calculations are simple; however, it offers slow convergence rate [36]. Therefore, the first-order information based methods are not suitable for handling cost functions with complex problems, usually stopping in a poor local minimal. Some alternative methods have been developed by exploiting second-order information and, as a consequence, increased convergence rate can be reached [37], [9].

The convergence rates of the methods based on the second-order information are higher than those yielded by the first-order information methods [38], [39]. Nevertheless, using second-order information for large scale problems adds some difficulties, such as: i) the evaluation of the Hessian matrix, ii) the calculation of its inverse matrix, and iii) the large memory usage [40]. To overcome these difficulties and to speed up the training process, one approach is to estimate the Hessian matrix or its inverse [9], [41], [42], [43]. Another approach is to calculate the Hessian matrix directly from eigenvectors corresponding to its largest eigenvalues [44]. A third approach is the calculation of the Hessian matrix by using the first-order information as in the conjugate gradient (CG) method [9]. The CG method uses the line-search technique to determine the current step size [45], [46].

A well-know variation of the CG method is the scaled conjugate gradient (SCG) method. The SCG method is based on the second-order information supervised learning procedure [36], [47], [48]. In the SCG method two gradients are calculated per iteration: the first gradient is calculated with a small step size, and the second gradient is calculated with a large step size. Also the SCG method executes a trust-region technique rather than the line-search technique to scale the step size. The line search technique uses a model function to generate a search direction, focusing on to determine a suitable step length along this direction. With the use of the trust-region technique [49], [50], a region is defined around the current search point, ‘trusting’ the model function to be an appropriate representation of the cost function to be minimized, choosing the step size to be the approximate minimizer of the model in this region. A trust-region technique is more robust than a line-search techniques [36], [51], and line search technique requires more user’s dependence parameters to determine the step size, something that increases the time interval and for hardware resource usage demanded by a training procedure. This disadvantage is eliminated in the SCG method by using the trust-region technique as it is well-addressed in the Levenberg-Marquardt method [51], [52].

Aiming to classify defects in railroad switch machines, the authors in [2] proposed a type-1 and singleton fuzzy logic system trained by the conjugate gradient method (CG

T1-FLS), resulting in a higher performance and convergence speed in comparison with previous classification techniques based on Bayes, the artificial neural network and the type-1 and singleton fuzzy logic system trained by steepest descent method (SD T1-FLS). However, this work showed a higher dependence of user's parameters and higher complexity due to the calculation of full Hessian matrix. In order to overcome this problem, the type-1 and singleton fuzzy logic systems trained by scaled conjugate method (SCG T1-FLS and SCG T1-FLSMO) is proposed. Avoiding the computing of Hessian matrix, the SCG T1-FLS and SCG T1-FLSMO approximate the multiplication of the Hessian matrix by the directional vector. This fact can result in round-off problems and can limit the capability to reach a global minimum of a cost function.

In an attempt to improve the performance of the SCG T1-FLS and SCG T1-FLSMO, this thesis also propose the calculation of the $\mathbf{H}\mathbf{v}$ based on the differential operator $\mathcal{R}\{\cdot\}$ [53], in which \mathbf{H} and \mathbf{v} are the Hessian matrix and directional vector, respectively. It results in T1-FLS trained by SCG with the differential operator $\mathcal{R}\{\cdot\}$ [47], [48], [54]. The acronym adopted for these models are SCGR T1-FLS and SCGR T1-FLSMO. The main idea behind of the SCGR T1-FLS is to compute exactly $\mathbf{H}\mathbf{v}$, and, as a consequence, to yield a higher performance and convergence speed in comparison with the SCG T1-FLS. This increase of convergence speed is a remarkable results if a limited number of epochs is used. As a matter of fact, an effective training is a very difficult task to be accomplished and the situation may worsen when a specific application demands a periodic update of the parameters of the model (i.e. the classification technique needs to be retrained to cover new patterns). In this regard, the increase of the convergence speed during the training phase of the design of a classification technique without decreasing its accuracy is an interesting problem to be addressed as it was very-well addressed in chapter 1.

In this context, this chapter introduces the SCG T1-FLS and SCG T1-FLSMO, which is capable of reducing the dependence of user's parameters, increasing the classification performance and decreasing the computational complexity during the training phase through the approximation of $\mathbf{H}\mathbf{v}$. Moreover, it introduces the use of the differential operator $\mathcal{R}\{\cdot\}$ to compute the exact value of $\mathbf{H}\mathbf{v}$ and to come up with additional increase on the convergence speed, during the training phase and under a limited number of epochs.

This chapter is organized as follows: Section 3.1 explains the limitations of the main gradient training methods and formulate the problem; while Section 3.2 introduces the SCG training method and its principal concepts. Also it is presented the training algorithm used in T1-FLS and T1-FLSMO by SCG and SCGR methods, as well as, how to use the differential operator $\mathcal{R}\{\cdot\}$ to calculate $\mathbf{H}\mathbf{v}$ for T1-FLS and T1-FLSMO.

3.1 PROBLEM FORMULATION

Considering the FLS introduced in chapter 2 T1-FLSMO, together with singleton fuzzification, max-product composition, product implication¹ and height defuzzification, its output is described by equation (2.6). The best classification rate is achieved when the cost function given by equation (2.10) is minimized, which is directly related to the values of parameters $m_{F_k^l}$, $\sigma_{F_k^l}$ and θ_l^t , which are obtained at the end of training phase.

Different methods can be devised to update the parameters of a T1-FLSMO. In general, they are based on a local iterative process which uses the gradient of the cost function $\nabla \mathbf{J}(\mathbf{w}^{(q)})$ to search in its neighborhood a $\mathbf{w}^{(q+1)}$ which minimizes the cost function. The CG methods are also based on this iterative process but choose the search direction and step size more carefully by using second-order information [2] [55] and/or using the line-search techniques [56] [57]. The authors in [2] discussed a comparison between SD and CG methods applied to the training procedure of a type-1 and singleton FLS. As expected, the CG methods were faster than the SD method by achieving a higher accuracy. However, this work showed the higher existing dependence of user's parameters (such as the step-length $\alpha(q)$, and the parameters used to obtain the next search direction $\beta(q)$) and higher computation complexity, especially when the use of CG method requires the calculation of the Hessian matrix.

The issues aforementioned occur in training procedure of the methods. In addition, when employed a procedure which require less parameters defined by user, the time used to design the model consequently is reduced. This reduction, also entails that the specialist and not specialist users having the same chance to achieve the convergence of the method, independently from its knowledge. Moreover, the second-order training methods which computes the Hessian matrix has a large computational cost and memory usage do not be attractive for large-scale problems. In order to contour these issues, Section 3.2 introduces the SCG training method, and usage of the differential operator $\mathcal{R}\{\cdot\}$ to T1-FLSMO.

3.2 T1-FLSMO TRAINED BY SCALED CONJUGATE GRADIENT

The idea to be exploited is to introduce the SCG method for training T1-FLSMO. The SCG method was firstly proposed in [36] and further improved in [51] it consists in exploiting a conjugate direction does not need any line-search technique [45, 58]. As a consequence, the dependence of user's parameters are reduced. The main contribution of the SCG method is to estimate the term $\mathbf{s}^{(\gamma)} = \mathbf{H}(\mathbf{w}^{(q)}) \mathbf{v}^{(\gamma)}$ in the current epoch γ , where $\mathbf{v}^{(\gamma)}$ is a directional vector and $\mathbf{H}(\mathbf{w}^{(q)})$ is the Hessian matrix, which is obtained by the derivation of the gradient vector of the cost function $\nabla \mathbf{J}(\mathbf{w}^{(q)})$ in relation to the

¹ t-norm

fuzzy parameters $m_{F_k^l}(q)$, $\sigma_{F_k^l}(q)$, $\theta_l^t(q)$. The term $\mathbf{s}^{(q)}$ can be approximated as follows:

$$\mathbf{s}^{(\gamma)} \approx \mathbf{H}(\mathbf{w}^{(q)}) \mathbf{v}^{(\gamma)} \approx \frac{\nabla \mathbf{J}(\mathbf{w}^{(q)} + \varepsilon \mathbf{v}^{(\gamma)}) - \nabla \mathbf{J}(\mathbf{w}^{(q)})}{\varepsilon}, 0 < \varepsilon \ll 1. \quad (3.1)$$

Note that equation (3.1) avoids computing the Hessian matrix, thereby reducing the computational complexity and memory usage [51].

If the value of the vector $\mathbf{w}^{(q)}$ results in a negative definite Hessian matrix, then the use of a negative step-length $\alpha^{(q)}$ have to be avoided. Basically, a positive scale parameter $\lambda \in \mathbb{R}^+$ is added to the diagonal of $\mathbf{H}(\mathbf{w}^{(q)})$. If λ is sufficiently large, the Hessian matrix become positive definite and, as a consequence, it yields a positive $\alpha^{(q)}$. Taking a large value for λ implies a small step size in the direction of search vector $\mathbf{v}^{(\gamma)}$, that is, the first-order information will predominate over the second-order information. In a similar way, if the scale parameter λ has a small value, the second-order information will have a more prominent influence than the first-order one on the value of $\alpha^{(q)}$. To allow the adaptation of λ during the training procedure, SCG includes steps inherited from trust region techniques. Basically, it decreases λ in regions where the quadratic model is a good local approximation, and increases λ in regions where the quadratic approximation is poor. A detailed description of all necessary steps for implementing SCG can be found in [47, 51].

The algorithm used to implement the SCG method, which was proposed in [51] define $-\nabla \mathbf{J}(\mathbf{w}^{(q)})$ as the initial update direction when ($q = 1$), computing all the next directions conjugated from it. However, $-\nabla \mathbf{J}(\mathbf{w}^{(1)})$ can not point to the correct direction that minimizes the cost function. Thus, the algorithm presented in [51] assumes small values for $\alpha^{(\gamma)}$, resulting $\Delta^{(\gamma)} = 0$. This fact avoids the convergence of the parameters of the model. To overcome this problem, the SCG [51] method is modified as follows: when $\Delta^{(\gamma)} = 0$, it is assumed that $\mathbf{v}^{(\gamma)} = \mathbf{r}^{(\gamma)} = -\nabla \mathbf{J}(\mathbf{w}^{(q+1)})$ where q is a index representing the chosen initial direction given by q -th pattern. In other words, the actual direction is changed by the negative value of the gradient of the next pattern. Figure 3 shows the roadmap of algorithm used to implement the modified version of the SCG method, which is used for training T1-FLSMO. It is important to emphasize that the SCG method must be trained only with the batch mode [51].

```

Input: Weight vector  $\mathbf{w}^{(1)}$ ,
scalars  $0 < \epsilon \leq 10^{-4}$ ,  $0 < \lambda^{(1)} \leq 10^{-6}$  and
 $\bar{\lambda}^{(1)} = 0$ .
1 Output: Minimum weight vector ( $\mathbf{w}^{(\gamma+1)}$ ).
2  $\gamma = 1$  and success=true;
3  $q = 1$ ;
4 Set  $\mathbf{v}^{(\gamma)} = \mathbf{r}^{(\gamma)} = -\nabla \mathbf{J}(\mathbf{w}^{(q)})$ ;
5 if success = true then // calculate the
second order information
6    $\epsilon^{(\gamma)} = \epsilon / |\mathbf{v}^{(\gamma)}|$ ;
7    $\mathbf{s}^{(\gamma)} =$ 
    $(\nabla \mathbf{J}(\mathbf{w}^{(\gamma)} + \epsilon \mathbf{v}^{(\gamma)}) - \nabla \mathbf{J}(\mathbf{w}^{(\gamma)})) / \epsilon^{(\gamma)}$ ;
8    $\delta^{(\gamma)} = \mathbf{v}^{(\gamma)T} \mathbf{s}^{(\gamma)}$ ;
9 end
10 // scale the  $\delta^{(\gamma)}$ 
11  $\delta^{(\gamma)} = \delta^{(\gamma)} + (\lambda^{(\gamma)} - \bar{\lambda}^{(\gamma)}) |\mathbf{v}^{(\gamma)}|^2$ ;
12 if  $\delta^{(\gamma)} \leq 0$  then // make Hessian matrix
positive definite
13    $\bar{\lambda}^{(\gamma)} = 2 (\bar{\lambda}^{(\gamma)} - \delta^{(\gamma)} / |\mathbf{v}^{(\gamma)}|^2)$ ;
14    $\delta^{(\gamma)} = -\delta^{(\gamma)} + \lambda^{(\gamma)} |\mathbf{v}^{(\gamma)}|^2$ ;
15    $\lambda^{(\gamma)} = \bar{\lambda}^{(\gamma)}$ ;
16 end
17 // Calculate the step size
18  $v^{(\gamma)} = \mathbf{v}^{(\gamma)T} \mathbf{r}^{(\gamma)}$ ;
19  $\alpha^{(\gamma)} = v^{(\gamma)} / \delta^{(\gamma)}$ ;
20 // Calculate the comparison parameter;
21  $\Delta^{(\gamma)} =$ 
    $2\delta^{(\gamma)} [\mathbf{J}(\mathbf{w}^{(\gamma)}) - \mathbf{J}(\mathbf{w}^{(\gamma)} + \alpha^{(\gamma)} \mathbf{v}^{(\gamma)})] / v^{(\gamma)2}$ ;
22 if  $\Delta^{(\gamma)} > 0$  then // a successful reduction
in error can be made
23    $\mathbf{w}^{(\gamma+1)} = \mathbf{w}^{(\gamma)} + \alpha^{(\gamma)} \mathbf{v}^{(\gamma)}$ ;
24    $\mathbf{r}^{(\gamma+1)} = -\nabla \mathbf{J}(\mathbf{w}^{(\gamma+1)})$ ;
25    $\bar{\lambda}^{(\gamma)} = 0$ , success = true;
26   if  $\gamma \bmod N = 0$  then // restart the
algorithm
27      $\mathbf{v}^{(\gamma+1)} = \mathbf{r}^{(\gamma+1)}$ ;
28   else
29      $\beta^{(\gamma)} = (|\mathbf{r}^{(\gamma+1)}|^2 - \mathbf{r}^{(\gamma+1)T} \mathbf{r}^{(\gamma)}) / v^{(\gamma)}$ ;
30      $\mathbf{v}^{(\gamma+1)} = \mathbf{r}^{(\gamma+1)} + \beta^{(\gamma)} \mathbf{v}^{(\gamma)}$ ;
31   end
32   if  $\Delta \geq 0.75$  then // reduce the scale
parameter
33      $\lambda^{(\gamma)} = \frac{1}{4} \lambda^{(\gamma)}$ ;
34   end
35 else if  $\Delta^{(\gamma)} = 0$  then // change the initial
direction
36    $q = q + 1$  and go to line 2;
37 else
38    $\bar{\lambda}^{(\gamma)} = \lambda^{(\gamma)}$ ;
39   success = false;
40 end
41 if  $\Delta_k < 0.25$  then // increase the scale
parameter
42    $\lambda^{(\gamma)} = \lambda^{(\gamma)} + (\delta^{(\gamma)} (1 - \Delta^{(\gamma)}) / |v^{(\gamma)}|^2)$ ;
43 end
44 if  $r^{(\gamma)} \neq 0$  then
45    $\gamma = \gamma + 1$  and go to line 3;
46 end
47 terminate and return  $\mathbf{w}^{(\gamma+1)}$  as desired minimum
weight vector;

```

Figure 3: The algorithm of the proposed SCG method for training T1-FLS and T1-FLSMO.

3.2.1 The SCG method with the differential operator $\mathcal{R}\{\cdot\}$

It is well-known that equation (3.1) reduces the computational complexity due to the fact that it is not necessary to calculate the Hessian matrix $\mathbf{H}(\mathbf{w}^{(q)})$ [2]. Its principle is the same of finite differences and as a consequence, it can result in round off problems and, limit the capability for reaching the local minimum of the cost function. In order to avoid this problem, also is proposed to compute the exact value of $\mathbf{H}(\mathbf{w}^{(q)}) \mathbf{v}^{(\gamma)}$ for FLS by the differential operator $\mathcal{R}\{\cdot\}$ [53]. As a result, the T1-FLSMO trained by the SCG method with the differential operator $\mathcal{R}\{\cdot\}$ is called SGCR T1-FLSMO. The differential operator $\mathcal{R}\{\cdot\}$ uses the definition of the derivative to compute the exact value of $\mathbf{H}(\mathbf{w}^{(q)}) \mathbf{v}^{(\gamma)}$, which is given by

$$\mathbf{s}^{(\gamma)} = \mathbf{H}(\mathbf{w}^{(q)}) \mathbf{v}^{(\gamma)} = \lim_{r \rightarrow 0} \frac{\nabla \mathbf{J}(\mathbf{w}^{(q)} + r \mathbf{v}^{(\gamma)}) - \nabla \mathbf{J}(\mathbf{w}^{(q)})}{r}, \quad (3.2)$$

where $r \in \mathbb{R}$. From (3.2), we have

$$\mathbf{H}(\mathbf{w}^{(q)}) \mathbf{v}^{(\gamma)} = \mathcal{R}_{\mathbf{v}} \{ \nabla \mathbf{J}(\mathbf{w}^{(q)}) \} = \left. \frac{\partial}{\partial r} \mathbf{J}(\mathbf{w}^{(q)} + r \mathbf{v}^{(\gamma)}) \right|_{r=0}. \quad (3.3)$$

It is important to emphasize that the differential operator $\mathcal{R}\{\cdot\}$ has the same rules and properties of derivative. Through the use of the differential operator $\mathcal{R}\{\cdot\}$, the equations of $\mathbf{H}\mathbf{v}$ for T1-FLS and, T1-FLSMO are introduced.

Likewise mentioned at the end of chapter 2, when T1-FLSMO has a single output ($\Upsilon = 1$) it reduces to T1-FLS. Therefore, all statements from SCG algorithm introduced in this chapter for T1-FLSMO is also valid for T1-FLS. For sake of simplicity, the deductions to obtain $\mathbf{H}\mathbf{v}$ for T1-FLSMO and T1-FLS are presented in Appendix A and B, respectively.

4 NUMERICAL RESULTS

This chapter discusses performance analyses of the proposals described in chapters 2 and 3. To do so, data sets provided by UCI Machine Learning Repository [20] are used. The analyses are performed in terms of the following perspectives: a comparison between SCG and SCGR training methods and a comparison between the T1-FLSMO and T1-FLS to deal with MCP. In both comparisons, 5-fold cross-validation was performed [59]. In other words, each data set is randomly split into 5 subsets, then each subset is used as test sets for a classifier built on the remaining four subsets. For all FLS classifiers, the input variables were normalized to the $[-1, 1]$ interval and the batch mode was applied.

To carry out numerical analyses, all performance metrics are expanded to t-student distribution. The chosen performance metrics are: accuracy, mean squared error (MSE), Kohen's kappa coefficient [60] and F-score [61]. The Kohen's kappa and the F-score performance metrics are obtained by using the values of the confusion matrix of the classification (i.e. true positives, true negatives, false positives and false negatives). These metrics achieve a value equal to 1 if correct classification occurs and equal to 0 for wrong classification. For instance, in a binary data set with 1000 patterns, in which 950 is from class A and 50 is from class B, if the classifier output assign all the patterns to class A, the accuracy is 0.95; however, the F-Score and Kohen's kappa values are equal to 0, indicating the wrong classifications. It is assumed $\gamma = 200$ epochs for the training phase and there is no stop criteria during this phase. Moreover, thirty percent of the training data set was used to assemble the validation set, in which the parameters vector $\mathbf{w}^{(\gamma)}$ output of the classifier is the parameter vector which achieves the highest accuracy in validation check among all the all epochs.

The FLS classifiers are composed of two rules for each class, totalizing $M = 2Y$ number of rules for each classifier. The first rule for each class is heuristically created using the mean and the variance from all inputs of the respective class, which can be synthesized by

$$rule_1^t = \{m_{F_1^1}^t, \sigma_{F_1^1}^t, m_{F_2^1}^t, \sigma_{F_2^1}^t, \dots, m_{F_k^1}^t, \sigma_{F_k^1}^t\}, \quad (4.1)$$

in which $rule_1^t$ refers to the values of the membership functions related to the first rule from t -th class. The second rule is a modified version of the first rule. Basically, it has heuristically adopted that the mean value of the Gaussian membership functions are scaled by a value $U \in \mathbb{R}$. As a result,

$$rule_2^t = \{Um_{F_1^2}^t, \sigma_{F_1^2}^t, Um_{F_2^2}^t, \sigma_{F_2^2}^t, \dots, Um_{F_k^2}^t, \sigma_{F_k^2}^t\}, \quad (4.2)$$

represents the 2nd rule. In this work, $U = 0.2$ is heuristically adopted.

4.1 TRAINING METHOD COMPARISON

In order to investigate only the training methods, thereby it is selected only binary classification data sets, which are listed in table 1. For comparison purpose, SD T1-FLS [26] and CG T1-FLS [2] are compared.

The adopted step size for SD T1-FLS and CG T1-FLS is $\alpha = 0.01$. Note that, if a lower value of α is specified than the convergence rate is slow; however, a high value of α may result in a lack of convergence. In addition, the algorithm for CG T1-FLS follows the roadmap proposed by Fletcher and Reeves [62] to update the parameter $\beta(\gamma)$, which is used to determine the next direction. For both SCG T1-FLS and SCGR T1-FLS, the value of initial Lagrange coefficient is $\lambda_1 = 10^{-15}$ and $N = 10$. The infinitesimal increment is $\epsilon = 10^{-5}$ for SCG T1-FLS. The performance gains in term of accuracy and convergence speed is not relevant when other values are adopted, something that was heuristically observed carrying out several numerical simulations. The initial value for θ_l adopted are equal a ‘1’ to the rules regard to the class 1 and ‘-1’ to the rules from class 2.

Data Sets	Number of Samples	Input Features
1. Australian	106	7
2. Haberman	306	3
3. Heart	270	13
4. Ionosphere	351	34
5. Liver Disorders	345	6
6. Pima	768	8
7. Sonar	208	60

Table 1: Details of datasets to perform the training method analysis.

4.1.1 Convergence speed analysis

Figures 4, 5 and 6 show the performance in terms of convergence speed and MSE during the training phase, considering only relevant folds when a limited number of epochs applies. The k -th fold presented in the figures is the data set tested by one of the k subset, and trained by the remaining four. It can be noted that the proposals achieve higher convergence speed than SD T1-FLS and CG T1-FLS, owing the fact of SCG T1-FLS and SCGR T1-FLS use second order information and the trust-region technique during the training phase. In addition, the SCGR T1-FLS which uses the differential operator $\mathcal{R}\{\cdot\}$, obtained a slight improvement in terms of convergence speed in comparison to the SCG T1-FLS. It is worth noting that the SCG T1-FLS and the SCGR T1-FLS are consistent and offer similar convergence speed to other folds and data sets.

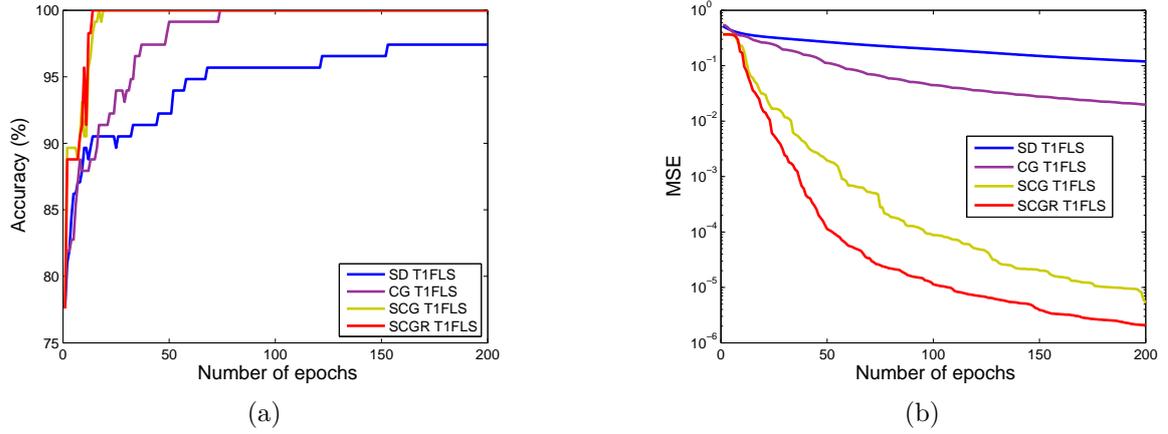


Figure 4: Sonar data set: performance of T1-FLS on 4th fold. (a) Accuracy. (b) MSE.

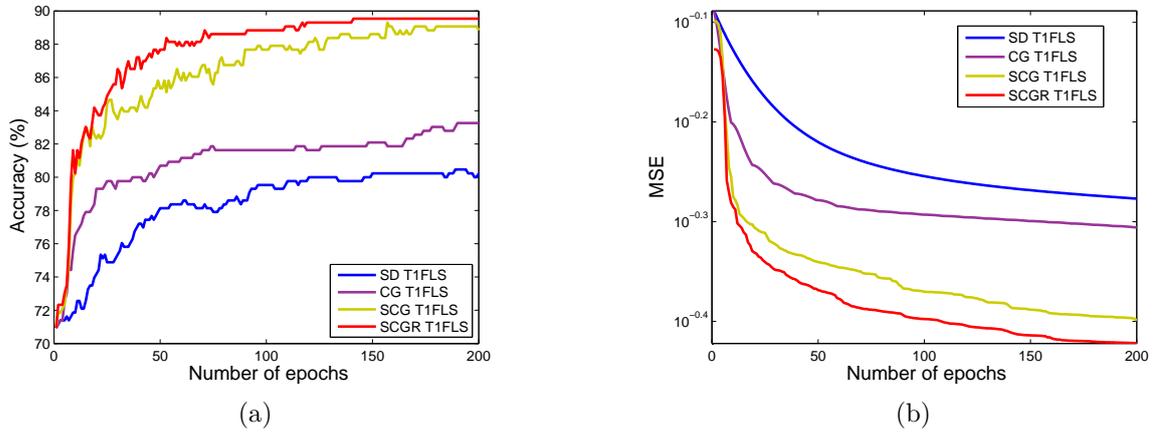


Figure 5: Pima data set: performance of T1-FLS on 1st fold. (a) Accuracy. (b) MSE.

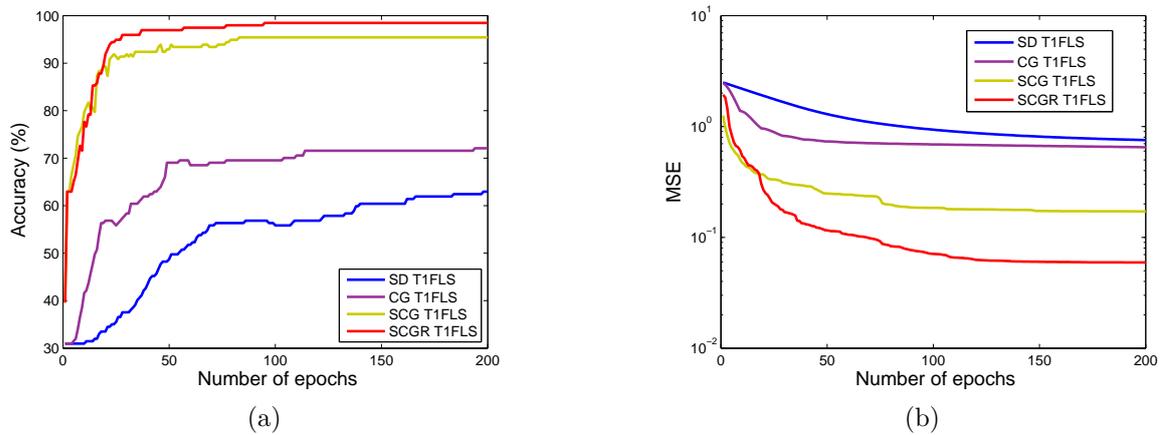


Figure 6: Ionosphere data set: performance of T1-FLS on 3rd fold. (a) Accuracy. (b) MSE.

4.1.2 Classification rate analysis

Table 2 lists the classification rate for the proposed SCG T1-FLS and SCGR T1-FLS in comparison with the models in [2, 26]. Based on the mean and deviation values of the results, one can note that the best performance for all investigated data sets, during the training phase are achieved by the SCG T1-FLS and SCGR T1-FLS. However, in test phase all methods achieves similar performance, with exception in Ionosphere data set, in which the SCGs T1-FLS attain the best performance. The Cohen’s Kappa and F-Score metrics accentuate this observation. Another important result is that the SCG and SCGR methods achieve the lower values of MSE, especially in Australian, Heart and Ionosphere data sets.

Data set	Method	Training accuracy %	Test accuracy (%)	Training MSE	Training Cohen’s Kappa	Test Cohen’s Kappa	Training F-score
Australian	SD T1-FLS [26]	89.22 (1.20)	85.07 (5.20)	0.34 (0.04)	0.78 (0.03)	0.70 (0.06)	0.89 (0.01)
	CG T1-FLS [2]	89.64 (1.11)	83.04 (6.64)	0.34 (0.02)	0.79 (0.03)	0.65 (0.06)	0.89 (0.02)
	SCG T1-FLS	91.97 (1.24)	81.59 (10.17)	0.28 (0.03)	0.80 (0.03)	0.62 (0.07)	0.90 (0.03)
	SCGR T1-FLS	91.87 (1.71)	83.04 (6.32)	0.27 (0.03)	0.78 (0.03)	0.65 (0.06)	0.89 (0.02)
Haberman	SD T1-FLS [26]	77.08 (3.44)	74.49 (7.07)	0.66 (0.07)	0.28 (0.10)	0.21 (0.17)	0.63 (0.07)
	CG T1-FLS [2]	77.78 (4.35)	74.17 (6.14)	0.64 (0.08)	0.28 (0.10)	0.18 (0.17)	0.63 (0.06)
	SCG T1-FLS	82.11 (3.08)	73.84 (6.23)	0.55 (0.05)	0.23 (0.11)	0.26 (0.16)	0.59 (0.12)
	SCGR T1-FLS	80.94 (4.64)	75.16 (6.08)	0.56 (0.05)	0.21 (0.11)	0.24 (0.17)	0.58 (0.13)
Heart	SD T1-FLS [26]	88.34 (3.14)	81.11 (8.91)	0.35 (0.05)	0.59 (0.06)	0.62 (0.11)	0.80 (0.24)
	CG T1-FLS [2]	90.20 (2.00)	82.96 (7.03)	0.31 (0.06)	0.60 (0.06)	0.65 (0.10)	0.80 (0.25)
	SCG T1-FLS	97.35 (0.88)	80.00 (9.08)	0.11 (0.03)	0.70 (0.05)	0.59 (0.11)	0.85 (0.19)
	SCGR T1-FLS	97.48 (0.73)	80.74 (8.06)	0.10 (0.03)	0.71 (0.05)	0.61 (0.11)	0.85 (0.19)
Ionosphere	SD T1-FLS [26]	63.93 (3.92)	64.68 (2.32)	0.76 (0.02)	0.36 (0.06)	0.37 (0.10)	0.64 (0.04)
	CG T1-FLS [2]	72.67 (2.68)	68.08 (5.53)	0.64 (0.02)	0.28 (0.08)	0.21 (0.14)	0.61 (0.12)
	SCG T1-FLS	96.55 (4.62)	87.73 (7.90)	0.11 (0.13)	0.88 (0.03)	0.73 (0.08)	0.94 (0.05)
	SCGR T1-FLS	96.45 (7.21)	88.29 (8.73)	0.09 (0.15)	0.90 (0.03)	0.73 (0.08)	0.95 (0.09)
Liver Disorders	SD T1-FLS [26]	73.99 (2.81)	65.22 (7.80)	0.76 (0.05)	0.43 (0.07)	0.25 (0.12)	0.71 (0.03)
	CG T1-FLS [2]	75.34 (24.41)	67.54 (10.22)	0.67 (0.49)	0.53 (0.06)	0.32 (0.12)	0.77 (0.03)
	SCG T1-FLS	84.87 (2.14)	64.64 (5.53)	0.50 (0.06)	0.49 (0.06)	0.25 (0.12)	0.74 (0.06)
	SCGR T1-FLS	86.74 (2.86)	64.06 (8.74)	0.45 (0.06)	0.53 (0.06)	0.25 (0.12)	0.76 (0.05)
Pima	SD T1-FLS [26]	79.12 (2.04)	73.96 (4.83)	0.57 (0.05)	0.52 (0.04)	0.41 (0.08)	0.76 (0.02)
	CG T1-FLS [2]	80.88 (3.54)	73.31 (5.66)	0.53 (0.06)	0.53 (0.04)	0.39 (0.08)	0.77 (0.02)
	SCG T1-FLS	86.56 (2.37)	73.57 (3.99)	0.43 (0.05)	0.54 (0.04)	0.39 (0.08)	0.77 (0.05)
	SCGR T1-FLS	86.70 (3.46)	73.70 (4.81)	0.43 (0.06)	0.54 (0.04)	0.41 (0.08)	0.77 (0.04)
Sonar	SD T1-FLS [26]	96.74 (2.93)	72.08 (11.66)	0.14 (0.05)	0.85 (0.04)	0.42 (0.14)	0.93 (0.10)
	CG T1-FLS [2]	98.45 (3.46)	75.95 (9.83)	0.08 (0.12)	0.92 (0.03)	0.50 (0.13)	0.96 (0.08)
	SCG T1-FLS	99.66 (0.62)	76.96 (7.05)	0.01 (0.02)	0.93 (0.03)	0.53 (0.13)	0.96 (0.04)
	SCGR T1-FLS	98.79 (2.98)	74.56 (9.92)	0.04 (0.11)	0.88 (0.04)	0.48 (0.14)	0.94 (0.07)

Table 2: Performance in terms of the mean and deviation for each metric with 95% t-student confidence interval.

4.2 CLASSIFICATION MODEL ANALYSIS COMPARISON

The aim of this section is to investigate the performance of FLS classification models when MCP is considered. In this context, T1-FLS using the OvA decomposition strategy [26] and T1-FLSMO are compared. The selected data sets to carry out this analysis are listed in Table 3.

The adopted step size for SD T1-FLS and SD T1-FLSMO is $\alpha = 0.01$. Considering SCG T1-FLSMO and SCGR T1-FLSMO, the initial values of Lagrange coefficient is $\lambda_1 = 10^{-15}$ and $N = 10$. For SCG T1-FLSMO, the infinitesimal increment is $\epsilon = 10^{-5}$. A length- Υ vector codifies the output of the data sets, assigning 1 to the element with

the presence of the pattern and -1 in the element that does not have the presence of the pattern. The performance gains in term of accuracy and convergence speed is not relevant when other values are assumed. The initial parameters of θ_i^t are equal to '1' for rules of 't' class in the output 't' and '-1' to remaining rules. To perform the analysis of the computational complexity, a comparison is carried out by the quantities of the classifier update parameters for a hypothetical problem with 3, 5 and 10 classes, varying the number of rules between one and fifty (i.e. $M \in [1, 50]$).

Data set	Number of samples	Input features	Number of classes	Total number of rules	
				T1-FLS + OvA	T1-FLSMO
Contraceptive	1,473	9	3	12	6
Ecoli	336	8	8	32	16
Iris	150	4	3	12	6
Wine	178	13	3	12	6

Table 3: Details of data sets used to perform the classification proposal analysis.

4.2.1 Convergence speed analysis

Figures 7, 8, 9 and 10 show the convergence speed and the MSE in terms of the epochs number for T1-FLSMO. Note that only the most representative k -th fold is shown. Despite the data sets were well mapped by T1-FLSMO, it may be noted that higher convergence speeds are achieved by SCG methods than SD methods, owing to the use of second order information during the training phase. This result reinforces the conclusions obtained in subsection 4.1.2.

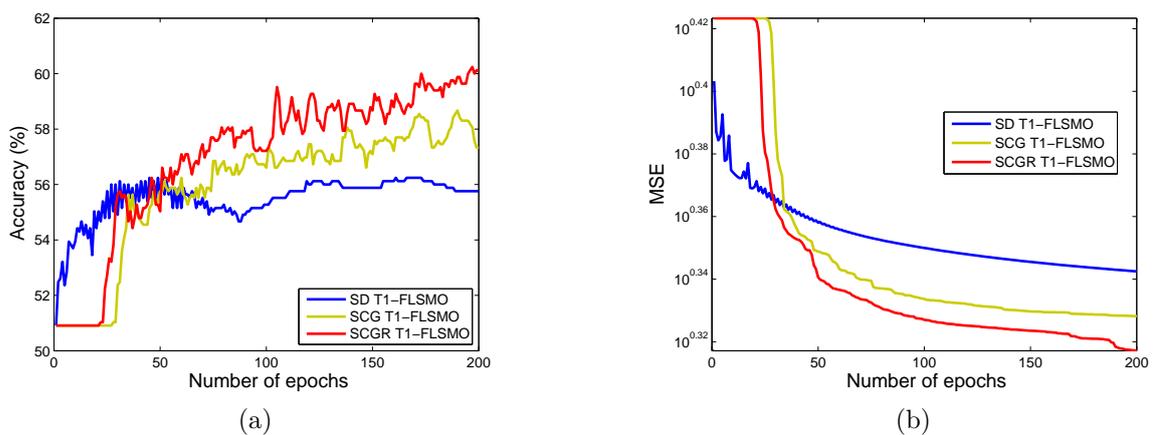


Figure 7: Contraceptive data set: performance of T1-FLSMO on 4th fold. (a) Accuracy. (b) MSE.

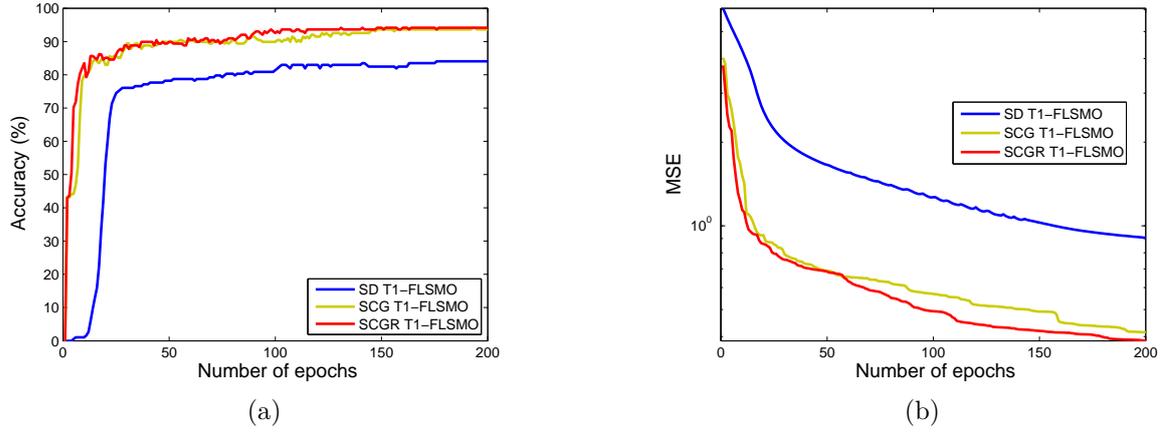


Figure 8: Ecoli data set: performance of T1-FLSMO on 3rd fold. (a) Accuracy. (b) MSE.

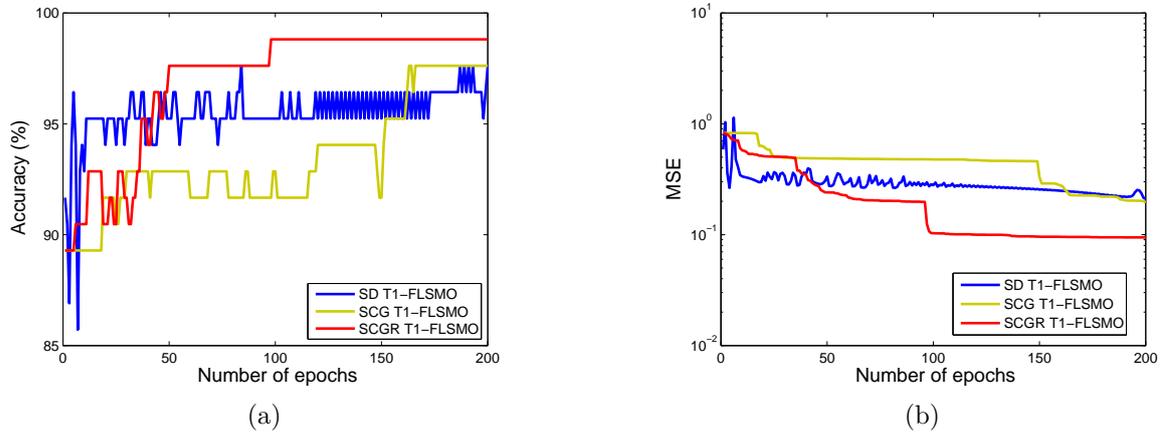


Figure 9: Iris data set: performance of T1-FLSMO on 4th fold. (a) Accuracy. (b) MSE.

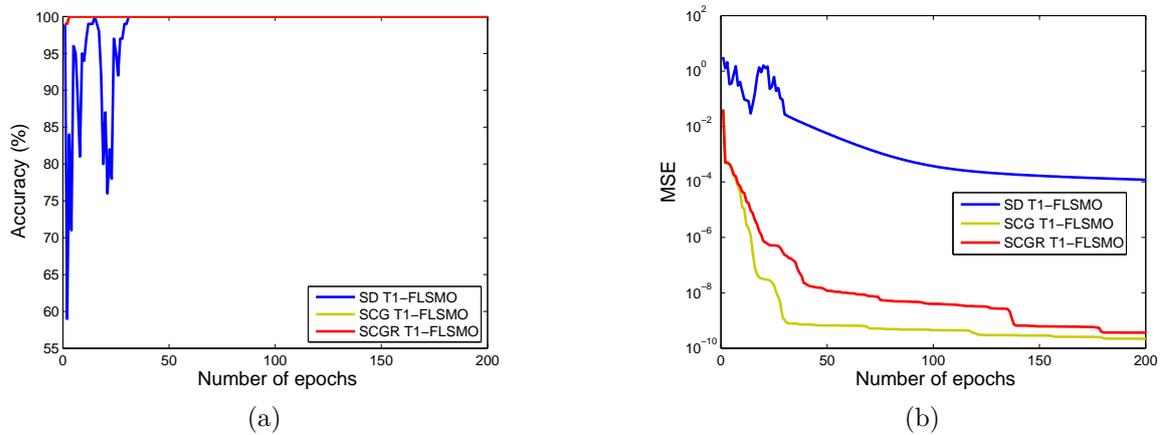


Figure 10: Wine data set: performance of T1-FLSMO on 5th fold. (a) Accuracy. (b) MSE.

4.2.2 Classification rate analysis

In table 4 are listed the classification rates and performance metrics of T1-FLSMO and T1-FLS [26] based on the OvA decomposition strategy. The training accuracy shows that T1-FLSMO offers lower performance in Contraceptive and Ecoli data sets. On the remaining data sets, its performance was similar to T1-FLS with the OvA decomposition strategy. Focusing on the test accuracy and the Test Cohen’s Kappa metric, similar performance are noted. Concerning in MSE values, it is concluded the T1-FLS with OvA decomposition strategy achieves the lowest values. This fact can be explained by the T1-FLS with OvA decomposition strategy has a specialist classifier for each class, while the T1-FLSMO uses the same classifier for all classes.

Data set	Method	Training accuracy %	Test accuracy (%)	Training MSE	Training Cohen’s Kappa	Test Cohen’s Kappa	Training F-score
Iris	SD T1-FLS [26]	96.73 (1.51)	95.00 (5.70)	2.96 (0.00)	0.95 (0.03)	0.92 (0.05)	0.97 (0.01)
	SD T1-FLSMO	97.14 (1.41)	93.33 (6.25)	0.23 (0.11)	0.93 (0.03)	0.90 (0.06)	0.93 (0.06)
	SCG T1-FLS	100.00 (0.00)	97.50 (2.21)	0.03 (0.04)	1.00 (0.00)	0.96 (0.04)	1.00 (0.00)
	SCG T1-FLSMO	98.21 (2.04)	94.17 (7.54)	0.13 (0.14)	0.94 (0.03)	0.91 (0.05)	0.96 (0.04)
	SCGR T1-FLS	100.00 (0.00)	97.50 (2.21)	0.01 (0.01)	1.00 (0.00)	0.96 (0.04)	1.00 (0.00)
	SCGR T1-FLSMO	98.21 (0.91)	96.67 (3.61)	0.11 (0.05)	0.94 (0.03)	0.95 (0.04)	0.96 (0.04)
Contraceptive	SD T1-FLS [26]	49.43 (0.57)	48.33 (2.09)	3.00 (0.00)	0.19 (0.03)	0.17 (0.05)	0.37 (0.00)
	SD T1-FLSMO	56.61 (1.43)	54.92 (2.30)	2.21 (0.02)	0.32 (0.03)	0.29 (0.05)	0.52 (0.03)
	SCG T1-FLS	63.61 (1.13)	55.67 (1.19)	1.95 (0.06)	0.43 (0.03)	0.30 (0.05)	0.62 (0.01)
	SCG T1-FLSMO	59.35 (2.37)	55.46 (2.51)	2.08 (0.09)	0.35 (0.03)	0.30 (0.05)	0.56 (0.02)
	SCGR T1-FLS	64.02 (1.65)	55.06 (1.15)	1.93 (0.06)	0.44 (0.03)	0.30 (0.05)	0.62 (0.02)
	SCGR T1-FLSMO	60.17 (1.46)	55.67 (1.95)	2.06 (0.06)	0.35 (0.03)	0.31 (0.05)	0.56 (0.02)
Wine	SD T1-FLS [26]	100.00 (0.00)	96.62 (1.73)	0.05 (0.02)	1.00 (0.00)	0.95 (0.05)	1.00 (0.00)
	SD T1-FLSMO	99.60 (0.73)	93.24 (4.33)	0.03 (0.06)	0.99 (0.01)	0.90 (0.06)	0.93 (0.04)
	SCG T1-FLS	100.00 (0.00)	97.17 (2.68)	0.00 (0.00)	1.00 (0.00)	0.96 (0.04)	1.00 (0.00)
	SCG T1-FLSMO	99.39 (1.20)	94.37 (2.72)	0.05 (0.09)	0.97 (0.02)	0.91 (0.06)	0.98 (0.03)
	SCGR T1-FLS	100.00 (0.00)	97.17 (2.68)	0.00 (0.00)	1.00 (0.00)	0.96 (0.04)	1.00 (0.00)
	SCGR T1-FLSMO	99.60 (0.73)	95.49 (3.34)	0.03 (0.06)	0.98 (0.01)	0.93 (0.05)	0.99 (0.01)
Ecoli	SD T1-FLS [26]	88.56 (2.91)	83.25 (5.54)	1.02 (0.24)	0.84 (0.03)	0.77 (0.06)	0.72 (0.07)
	SD T1-FLSMO	82.58 (6.55)	79.91 (6.26)	1.04 (0.30)	0.74 (0.04)	0.71 (0.07)	0.38 (0.10)
	SCG T1-FLS	94.15 (1.00)	82.51 (8.35)	0.45 (0.06)	0.92 (0.02)	0.76 (0.06)	0.79 (0.11)
	SCG T1-FLSMO	91.76 (2.41)	84.00 (11.38)	0.53 (0.12)	0.86 (0.03)	0.78 (0.06)	0.60 (0.19)
	SCGR T1-FLS	94.68 (1.15)	83.62 (7.47)	0.41 (0.06)	0.93 (0.02)	0.77 (0.06)	0.83 (0.16)
	SCGR T1-FLSMO	92.55 (1.73)	83.63 (8.13)	0.50 (0.11)	0.86 (0.03)	0.77 (0.06)	0.62 (0.19)

Table 4: Performance in terms of the mean and deviation for each metric with 95% t-student confidence interval.

4.2.3 Computational complexity analysis

Figure 11 shows a comparison of the number fuzzy update parameters for T1-FLS with OvA and T1-FLSMO, in terms of the number of rules between one and fifty in a data set with ten input features ($P = 10$). From figure, it can be stated that T1-FLSMO reduces computational complexity significantly when the number of rules increases and especially when the number of classes grows. Additionally, from Table 3 and 4 it is noted that T1-FLSMO has similar performance to T1-FLS, but it is obtained by using a half of number of rules demanded by the T1-FLS.

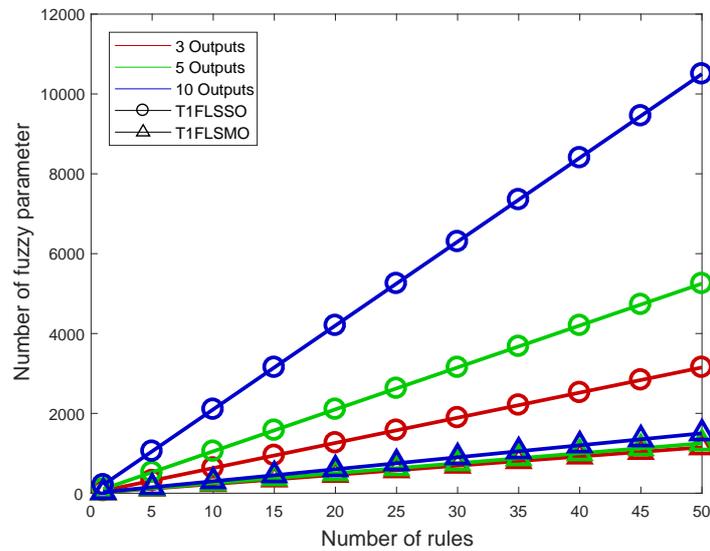


Figure 11: Comparison of the quantity of update parameters for T1-FLS with OvA and T1-FLSMO for different quantities of outputs for a data set with 10 input features.

5 CONCLUSION

This thesis has investigated FLS to deal with MCP by introducing two contributions. First is a new fuzzy model which deals with MCP without the use of binary decomposition strategy (T1-FLSMO) and second, the use of the SCG and SCGR methods for training T1-FLSMO and T1-FLS. Aiming to validate the proposals, well-known classification data sets provided by UCI Machine Learning Repository were adopted.

The chapter 2 has addressed in to reduce the classifier complexity when the FLS deals with multiclass classification problems. The FLS applied in MCP requires a binary decomposition strategy due to the fact that FLS has only a single output. The use of binary decomposition strategy rises the number of the classifier for dealing with MCP, mainly when the number of the class growth. In order to avoid the use this strategy, the T1-FLS is extended to have a multiple outputs (T1-FLSMO). The computational simulations and numerical results, showed that T1-FLSMO reduces significantly the number of total fuzzy rules used in MCP when compared with T1-FLS, consequently, reducing the computational complexity. The performance of the T1-FLS with OvA strategy is better than T1-FLSMO in almost all data sets on training phase in terms of accuracy and MSE values. However, the performance of both methods resulted in similar testing performance which justifies the usage of T1-FLSMO in MCP. Also, T1-FLSMO demands a half of the number rules used by T1-FLS with the same performance for the chosen data sets. Additionally, SCG T1-FLSMO and SCGR T1-FLSMO resulted in the fast convergence rate and the best performance in comparison with SD-T1FLSMO.

Chapter 3 has introduced the SCG training method for T1-FLSMO and its particular case T1-FLS. The SCG method avoids the use of Hessian matrix approximating the $\mathbf{H}\mathbf{v}$ using the calculus of two gradients per iteration. This method reduces the dependence of user's parameters to guarantee the convergence during training phase. In attempt to improve the performance of SCG, also, was proposed the SCGR method, which calculates the exact value of $\mathbf{H}\mathbf{v}$ through the differential operator $\mathcal{R}\{\cdot\}$. The computational of the exact $\mathbf{H}\mathbf{v}$ in SCG is important to avoid round-off problem, and to increase the capability of the method to reach a global minimum of the cost function. The numerical results showed that SCG T1-FLS and SCGR T1-FLS resulted in fast convergence rates than SD T1-FLS and the CG T1-FLS. Additionally, these proposed models achieve higher classification rate in training and test phase, and fewer values of MSE if a limited number of epochs applies. Highlighting the highest value of Cohen's Kappa and F-Score metrics achieved by SCGR in all datasets, the proposed models turn out to be a very attractive option due to the higher performance in T1-FLSMO and T1-FLS (i.e. achieving higher classification rate in few epochs) than those obtained with SD T1-FLS and CG-T1FLS.

5.1 FUTURE WORKS

Future efforts can be addressed in order to:

- extend the concept of T1-FLSMO to the interval and singleton type-2 fuzzy logic system [63].
- apply the SCG method for the interval and singleton type-2 fuzzy logic system reducing the dependence of user's parameters and to achieve a fast convergence speed avoiding to compute the full hessian matrix.
- apply the differential operator $\mathcal{R}\{\cdot\}$ in the interval and singleton type-2 fuzzy logic system, allowing to use the SCGR method and any second-order algorithm which makes use of the $\mathbf{H}\mathbf{v}$.

REFERENCES

- [1] Z. Bingul and O. Karahan, "A fuzzy logic controller tuned with pso for 2 dof robot trajectory control," *Expert Systems with Applications*, vol. 38, no. 1, pp. 1017 – 1031, 2011.
- [2] E. P. de Aguiar, F. M. A. Nogueira, R. P. Amaral, D. F. Fabri, S. C. Rossignoli, J. G. Ferreira, M. M. B. R. Vellasco, R. Tanscheit, P. C. S. Vellasco, and M. V. Ribeiro, "Eann 2014: A fuzzy logic system trained by conjugate gradient methods for fault classification in a switch machine," *Neural Computing and Applications*, vol. 27, no. 5, pp. 1175–1189, Jul. 2016.
- [3] H. Izakian and A. Abraham, "Fuzzy c-means and fuzzy swarm for fuzzy clustering problem," *Expert Systems with Applications*, vol. 38, no. 3, pp. 1835 – 1838, 2011.
- [4] A. Azadeh, M. Saberi, and O. Seraj, "An integrated fuzzy regression algorithm for energy consumption estimation with non-stationary data: A case study of iran," *in proc 7th International Conference on Sustainable Energy Technologies*, vol. 35, no. 6, pp. 2351 – 2366, 2010.
- [5] X. Kong, C. Lin, Y. Jiang, W. Yan, and X. Chu, "Efficient dynamic task scheduling in virtualized data centers with fuzzy prediction," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1068 – 1077, 2011.
- [6] "Solving multi-class problems with linguistic fuzzy rule based classification systems based on pairwise learning and preference relations," *Fuzzy Sets and Systems*, vol. 161, no. 23, pp. 3064 – 3080, 2010.
- [7] S. Abe, "Fuzzy support vector machines for multilabel classification," *Pattern Recognition*, vol. 48, no. 6, pp. 2110 – 2117, 2015.
- [8] J. Hang, J. Zhang, and M. Cheng, "Application of multi-class fuzzy support vector machine classifier for fault diagnosis of wind turbine," *Fuzzy Sets and Systems*, vol. 297, no. Supplement C, pp. 128 – 140, 2016.
- [9] J.-S. R. Jang and C.-T. Sun, *Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice-Hall, Inc., 1997.
- [10] D. Petković, Žarko Čojbašić, V. Nikolić, S. Shamshirband, M. L. M. Kiah, N. B. Anuar, and A. W. A. Wahab, "Adaptive neuro-fuzzy maximal power extraction of wind turbine with continuously variable transmission," *Energy*, vol. 64, pp. 868 – 874, 2014.
- [11] M. Sarma and K. K. Sarma, "Dialect identification from assamese speech using prosodic features and a neuro fuzzy classifier," *in proc. 2016 3rd International Conference on Signal Processing and Integrated Networks*, Feb. 2016, pp. 127–132.
- [12] S. Ghosh, S. Biswas, and D. S. and, "A novel neuro-fuzzy classification technique for data mining," *Egyptian Informatics Journal*, vol. 15, no. 3, pp. 129 – 147, 2014.
- [13] Y. H. Lin and M. S. Tsai, "Non-intrusive load monitoring by novel neuro-fuzzy classification considering uncertainties," *IEEE Transactions on Smart Grid*, vol. 5, no. 5, pp. 2376–2384, Sep. 2014.

- [14] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, “Ten years of genetic fuzzy systems: current framework and new trends,” *Fuzzy Sets and Systems*, vol. 141, no. 1, pp. 5–31, 2004.
- [15] S. Elhag, A. Fernández, A. Bawakid, S. Alshomrani, and F. Herrera, “On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on intrusion detection systems,” *Expert Systems with Applications*, vol. 42, no. 1, pp. 193 – 202, 2015.
- [16] T. Nguyen, A. Khosravi, D. Creighton, and S. Nahavandi, “Classification of healthcare data using genetic fuzzy logic system and wavelets,” *Expert Systems with Applications*, vol. 42, no. 4, pp. 2184 – 2197, 2015.
- [17] A. Jozi, T. Pinto, I. Praça, F. Silva, B. Teixeira, and Z. Vale, “Energy consumption forecasting using genetic fuzzy rule-based systems based on mogul learning methodology,” in *Proc. IEEE Manchester PowerTech*, Jun. 2017, pp. 1–5.
- [18] I. Rodrêguez-Fdez, M. Mucientes, and A. Bugarên, “A genetic fuzzy system for large-scale regression,” in *in Proc. IEEE International Conference on Fuzzy Systems*, Jul. 2016, pp. 1421–1428.
- [19] J. M. Mendel, “Uncertain rule-based fuzzy logic system: introduction and new directions,” 2001.
- [20] M. Lichman, “UCI machine learning repository,” 2013, <http://archive.ics.uci.edu/ml>.
- [21] P. Levinger, D. T. H. Lai, R. Begg, K. Webster, J. Feller, and W. Gilleard, “The application of multiclass svm to the detection of knee pathologies using kinetic data: a preliminary study,” in *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, Dec. 2007, pp. 589–594.
- [22] R. V. Sharan and T. J. Moir, “Comparison of multiclass svm classification techniques in an audio surveillance application under mismatched conditions,” in *in proc 19th International Conference on Digital Signal Processing*, Aug. 2014, pp. 83–88.
- [23] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang, “Large-scale image classification: fast feature extraction and svm training,” in *in Proc. IEEE Conference on Computer Vision and Pattern Recognition* , 2011, pp. 1689–1696.
- [24] B. Zhao, F. Li, and E. P. Xing, “Large-scale category structure aware image categorization,” in *Advances in Neural Information Processing Systems 24*, 2011, pp. 1251–1259.
- [25] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, “Good practice in large-scale learning for image classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 507–520, Mar. 2014.
- [26] E. P. de Aguiar, F. M. A. Nogueira, R. P. Amaral, D. F. Fabri, S. C. Rossignoli, J. G. Ferreira, M. M. B. R. Vellasco, R. Tanscheit, P. C. S. Vellasco, M. V. Ribeiro, and P. Vellasco, “Classification of events in switch machines using bayes, fuzzy logic system and neural network,” in *Proc. International Conference Engineering Applications of Neural Networks*, Sep. 2014, pp. 81–91.

- [27] E. P. de Aguiar, F. M. de A. Nogueira, M. M. B. R. Vellasco, and M. V. Ribeiro, “Set-membership type-1 fuzzy logic system applied to fault classification in a switch machine,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2703–2712, Oct. 2017.
- [28] S. Knerr, L. Personnaz, and G. Dreyfus, “Single-layer learning revisited: a stepwise procedure for building and training a neural network,” *Neurocomputing: algorithms, architectures and applications*, vol. 68, no. 41-50, p. 71, 1990.
- [29] R. Anand, K. Mehrotra, C. K. Mohan, and S. Ranka, “Efficient classification for multiclass problems using modular neural networks,” *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 117–124, 1995.
- [30] E. L. Allwein, R. E. Schapire, and Y. Singer, “Reducing multiclass to binary: A unifying approach for margin classifiers,” *Journal of Machine Learning Research*, vol. 1, pp. 113–141, Dec. 2000.
- [31] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
- [32] M. Elkano, M. Galar, J. A. Sanz, A. Fernández, E. Barrenechea, F. Herrera, and H. Bustince, “Enhancing multiclass classification in farc-hd fuzzy classifier: On the synergy between n -dimensional overlap functions and decomposition strategies,” *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 5, pp. 1562–1580, Oct. 2015.
- [33] A. Fernández, M. Calderón, E. Barrenechea, H. Bustince, and F. Herrera, “Solving multi-class problems with linguistic fuzzy rule based classification systems based on pairwise learning and preference relations,” *Fuzzy Sets and Systems*, vol. 161, no. 23, pp. 3064 – 3080, 2010.
- [34] S. Haykin, *3rd Ed. Adaptive Filter Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.
- [35] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
- [36] M. Moller, “A scaled conjugate gradient algorithm for fast supervised learning,” *DAIMI Report Series*, vol. 19, no. 339, 1990.
- [37] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1996.
- [38] R. Battiti, “First- and second-order methods for learning: Between steepest descent and newton’s method,” *Neural Computation*, vol. 4, no. 2, pp. pp. 141–166, 1992.
- [39] D. G. Luenberger and Y. Ye, *Linear and nonlinear programming*. Springer, 2015, vol. 228.
- [40] A. J. Shepherd, *Second-order methods for neural networks: Fast and reliable training methods for multi-layer perceptrons*. Springer Science & Business Media, 2012.
- [41] M. Bazaraa, H. Sherali, and C. Shetty, *Nonlinear Programming: Theory and Algorithms*. Wiley, 2006.
- [42] H. A. Moghaddam and M. Matinfar, “Fast adaptive lda using quasi-newton algorithm,” *Pattern recognition letters*, vol. 28, no. 5, pp. 613–621, 2007.

- [43] L. P. Ricotti, S. Ragazzini, and G. Martinelli, “Learning of word stress in a sub-optimal second order back-propagation neural network,” in *IEEE International Conference on Neural Networks*, vol. 1, Jul. 1988, pp. 355–361.
- [44] Y. Le Cun, I. Kanter, and S. A. Solla, “Eigenvalues of covariance matrices: Application to neural-network learning,” *Physical Review Letters*, vol. 66, no. 18, p. pp. 2396, 1991.
- [45] Y.-H. Dai, “Conjugate gradient methods with armijo-type line searches,” *Acta Mathematicae Applicatae Sinica (English Series)*, vol. 18, no. 1, pp. pp. 123–130, 2002.
- [46] Y.-H. Dai and C.-X. Kou, “A nonlinear conjugate gradient algorithm with an optimal property and an improved wolfe line search,” *SIAM Journal on Optimization*, vol. 23, no. 1, pp. pp. 296–320, 2013.
- [47] E. P. dos Santos and F. J. Von Zuben, “Efficient second-order learning algorithms for discrete-time recurrent neural networks,” in *Recurrent neural networks: design and applications*. CRC Press, 1999.
- [48] M. V. Ribeiro, C. A. Duque, and J. M. T. Romano, “An interconnected type-1 fuzzy algorithm for impulsive noise cancellation in multicarrier-based power line communication systems,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 7, pp. pp. 1364–1376, Jul. 2006.
- [49] T. Steihaug, “The conjugate gradient method and trust regions in large scale optimization,” *SIAM Journal on Numerical Analysis*, vol. 20, no. 3, pp. 626–637, 1983.
- [50] M. A. Wolfe, *Numerical methods for unconstrained optimization: An introduction*. Van Nostrand Reinhold, 1978.
- [51] M. F. Moller, “A scaled conjugate gradient algorithm for fast supervised learning,” *Neural networks*, vol. 6, no. 4, pp. 525–533, 1993.
- [52] B. Cetişli and A. Barkana, “Speeding up the scaled conjugate gradient algorithm and its application in neuro-fuzzy classifier training,” *Soft Computing*, vol. 14, no. 4, p. pp. 365, 2009.
- [53] B. A. Pearlmutter, “Fast exact multiplication by the hessian,” *Neural computation*, vol. 6, no. 1, pp. 147–160, 1994.
- [54] P. Campolucci, M. Simonetti, A. Uncini, and F. Piazza, “New second-order algorithms for recurrent neural networks based on conjugate gradient,” in *IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence*, vol. 1, May. 1998, pp. 384–389.
- [55] W. W. Hager and H. Zhang, “A survey of nonlinear conjugate gradient methods,” *Pacific journal of Optimization*, vol. 2, no. 1, pp. 35–58, 2006.
- [56] Y.-H. Dai and C.-X. Kou, “A nonlinear conjugate gradient algorithm with an optimal property and an improved wolfe line search,” *SIAM Journal on Optimization*, vol. 23, no. 1, pp. pp. 296–320, 2013.
- [57] Y.-H. Dai, “Conjugate gradient methods with armijo-type line searches,” *Acta Mathematicae Applicatae Sinica*, vol. 18, no. 1, pp. pp. 123–130, 2002.

- [58] Y.-Y. Chen and S.-Q. Du, “Nonlinear conjugate gradient methods with wolfe type line search,” in *Abstract and Applied Analysis*. Hindawi Publishing Corporation, 2013.
- [59] P. Refaeilzadeh, L. Tang, and H. Liu, *Cross-Validation*. Springer US, 2009.
- [60] S. Stehman, “Estimating the kappa coefficient and its variance under stratified random sampling,” *Photogrammetric Engineering and Remote Sensing*, vol. 62, no. 4, pp. pp. 401–407, 1996.
- [61] M. Sokolova, N. Japkowicz, and S. Szpakowicz, *Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation*, A. Sattar and B.-h. Kang, Eds. Springer Berlin Heidelberg, 2006.
- [62] R. Fletcher and C. M. Reeves, “Function minimization by conjugate gradients,” *The computer journal*, vol. 7, no. 2, pp. pp. 149–154, 1964.
- [63] E. P. Aguiar, R. P. F. Amaral, M. M. B. R. Vellasco, and M. V. Ribeiro, “Computing derivatives in interval type-2 fuzzy logic systems trained by steepest descent method for fault classification in a switch machine,” in *IEEE International Conference on Fuzzy Systems*, Naples/Italy, Jul. 2017.

Appendix A – Deduction of the $\mathbf{H}(\mathbf{w}^{(q)}) \mathbf{v}^{(q)}$ from $\mathcal{R}\{\cdot\}$ operator for T1-FLSMO

Considering the algorithm presented in Figure 3, we know that $\mathbf{v}^{(q)} = \nabla \mathbf{J}(\mathbf{w}^{(q)})$. To compute the $\mathbf{H}(\mathbf{w}^{(q)}) \mathbf{v}^{(q)}$ with the differential operator $\mathcal{R}\{\cdot\}$, we can apply the $\mathcal{R}\{\cdot\}$ operator separately in each gradient of the cost function, as follow:

$$\mathcal{R}_{\mathbf{v}} \left\{ \nabla \mathbf{J}(\mathbf{w}^{(q)}) \right\} = \left[\mathcal{R}_{\mathbf{v}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} \mathcal{R}_{\mathbf{v}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} \mathcal{R}_{\mathbf{v}} \left\{ \nabla_{\theta_i^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\} \right], \quad (\text{A.1})$$

where

$$\nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) = \left[\frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial m_{F_1^1}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial m_{F_p^1}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial m_{F_1^M}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial m_{F_p^M}(q)} \right], \quad (\text{A.2})$$

$$\nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) = \left[\frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \sigma_{F_1^1}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \sigma_{F_p^1}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \sigma_{F_1^M}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \sigma_{F_p^M}(q)} \right] \quad (\text{A.3})$$

and

$$\nabla_{\theta_i^t} \mathbf{J}(\mathbf{w}^{(q)}) = \left[\frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \theta_1^1(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \theta_M^1(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \theta_1^t(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \theta_M^t(q)} \right]. \quad (\text{A.4})$$

For reasons of simplicity the following notations have been adopted:

$$e^t(q) = f_{s_{mo}}^t(\mathbf{x}^{(q)}) - y^t(q) \quad (\text{A.5})$$

and

$$e_{\theta_i^t}^t(q) = \theta_i^t(q) - f_{s_{mo}}^t(\mathbf{x}^{(q)}). \quad (\text{A.6})$$

• Deduction of the $\mathcal{R}_{\mathbf{v}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$

$\mathcal{R}_{\mathbf{v}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\begin{aligned} \mathcal{R}_{\mathbf{v}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} + \dots \\ &\dots \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} + \mathcal{R}_{\theta_i^t} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}, \end{aligned} \quad (\text{A.7})$$

The terms of $\mathcal{R}_{\mathbf{v}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ are given by

Deduction of $\mathcal{R}_{\mathbf{v}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$:

$\mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \mathcal{R}_{m_{F_i^j}} \left\{ \sum_{t=1}^{\Upsilon} \left([f_{s_{mo}}^t(\mathbf{x}^{(q)}) - y^t(q)] \cdots \right. \right. \\ &\quad \left. \left. \cdots [\theta_l^t(q) - f_{s_{mo}}^t(\mathbf{x}^{(q)})] a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \right\}. \end{aligned} \quad (\text{A.8})$$

Analyzing for $j \neq l$, we obtain

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \sum_{t=1}^{\Upsilon} \left(\sum_{j=1}^M \left(\sum_{i=1}^p (a_{F_i^j}(q) v_{m_i^j}) \cdots \right. \right. \\ &\quad \left. \left. \cdots (e_{\theta_j}^t(q) e_{\theta_l}^t(q) - e_{\theta_j}^t(q) e^t(q) - e^t(q) e_{\theta_l}^t(q)) \phi_l(\mathbf{x}^{(q)}) \right) \right). \end{aligned} \quad (\text{A.9})$$

if $j = l$ and $i \neq k$, then (A.9) is replaced by

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \sum_{t=1}^{\Upsilon} \left(\sum_{i=1}^p (a_{F_i^l}(q) v_{m_i^l}) \cdots \right. \\ &\quad \left. \cdots (e^t(q) e_{\theta_l}^t(q) - 2e^t(q) e_{\theta_l}^t(q) \phi_l(\mathbf{x}^{(q)}) + (e_{\theta_l}^t(q))^2 \phi_l(\mathbf{x}^{(q)})) \right). \end{aligned} \quad (\text{A.10})$$

and if $i = k$, then (A.10) is replaced by

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \sum_{t=1}^{\Upsilon} \left((a_{F_k^l}(q) v_{m_k^l}) \cdots \right. \\ &\quad \left. \cdots (e^t(q) e_{\theta_l}^t(q) - 2e^t(q) e_{\theta_l}^t(q) \phi_l(\mathbf{x}^{(q)}) + (e_{\theta_l}^t(q))^2 \phi_l(\mathbf{x}^{(q)})) \cdots \right. \\ &\quad \left. \cdots - \sum_{t=1}^{\Upsilon} (e^t(q) e_{\theta_l}^t(q)) \phi_l(\mathbf{x}^{(q)}) \frac{v_{m_k^l}}{\sigma_{F_k^l}^2} \right). \end{aligned} \quad (\text{A.11})$$

$\mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\begin{aligned} \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \mathcal{R}_{\sigma_{F_i^j}} \left\{ \sum_{t=1}^{\Upsilon} \left([f_{s_{mo}}^t(\mathbf{x}^{(q)}) - y^t(q)] \cdots \right. \right. \\ &\quad \left. \left. \cdots [\theta_l^t(q) - f_{s_{mo}}^t(\mathbf{x}^{(q)})] a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \right\}. \end{aligned} \quad (\text{A.12})$$

Analyzing for $j \neq l$, we obtain:

$$\begin{aligned} \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \sum_{t=1}^{\Upsilon} \left(\sum_{j=1}^M \left(\sum_{i=1}^p (b_{F_i^j}(q) v_{\sigma_i^j}) \cdots \right. \right. \\ &\quad \left. \left. \cdots (e_{\theta_j}^t(q) e_{\theta_l}^t(q) - e_{\theta_j}^t(q) e^t(q) - e^t(q) e_{\theta_l}^t(q)) \phi_j(\mathbf{x}^{(q)}) \right) \right). \end{aligned} \quad (\text{A.13})$$

if $j = l$ and $i \neq k$, then (A.13) is replaced by

$$\begin{aligned} \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \sum_{t=1}^{\Upsilon} \left(\sum_{i=1}^p (b_{F_i^l}(q) v_{\sigma_i^l}) \cdots \right. \\ &\quad \left. \cdots (e^t(q) e_{\theta_l}^t(q) - 2e^t(q) e_{\theta_l}^t(q) \phi_l(\mathbf{x}^{(q)}) + (e_{\theta_l}^t(q))^2 \phi_l(\mathbf{x}^{(q)})) \right) \end{aligned} \quad (\text{A.14})$$

and if $i = k$, then (A.14) is replaced by

$$\begin{aligned} \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \sum_{t=1}^{\Upsilon} \left((b_{F_k^l}(q) v_{\sigma_k^l}) \cdots \right. \\ &\quad \left. \cdots (e^t(q) e_{\theta_l}^t(q) - 2e^t(q) e_{\theta_l}^t(q) \phi_l(\mathbf{x}^{(q)}) + (e_{\theta_l}^t(q))^2 \phi_j(\mathbf{x}^{(q)})) \cdots \right. \\ &\quad \left. \cdots - 2 \sum_{t=1}^{\Upsilon} (e^t(q) e_{\theta_l}^t(q)) v_{\sigma_k^l} \phi_l(\mathbf{x}^{(q)}) \frac{(x_k^{(q)} - m_{F_k^l}(q))}{\sigma_{F_k^l}^3(q)}. \right. \end{aligned} \quad (\text{A.15})$$

$\mathcal{R}_{\theta_j^t} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\begin{aligned} \mathcal{R}_{\theta_j^t} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \mathcal{R}_{\theta_j^t} \left\{ \sum_{t=1}^{\Upsilon} \left([f_{smo}^t(\mathbf{x}^{(q)}) - y^t(q)] \cdots \right. \right. \\ &\quad \left. \left. \cdots [\theta_l^t(q) - f_{smo}^t(\mathbf{x}^{(q)})] a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \right\}. \end{aligned} \quad (\text{A.16})$$

Analyzing for $j \neq l$, we obtain

$$\begin{aligned} \mathcal{R}_{\theta_j^t} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \cdots \\ &\quad \cdots \sum_{t=1}^{\Upsilon} \left(\sum_{j=1}^M (\phi_j(\mathbf{x}^{(q)}) v_{\theta_j^t}) (e_{\theta_l}^t(q) - e^t(q)) \right) \end{aligned} \quad (\text{A.17})$$

if $j = l$ and $t = \Upsilon$, then A.17 is replaced by

$$\begin{aligned} \mathcal{R}_{\theta_j^t} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \cdots \\ &\quad \cdots v_{\theta_l^\Upsilon} (e_{\theta_l}^\Upsilon(q) \phi_l(\mathbf{x}^{(q)}) + (1 - \phi_l(\mathbf{x}^{(q)})) e^\Upsilon(q)) \end{aligned} \quad (\text{A.18})$$

•Deduction of $\mathcal{R}_v \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$

$$\begin{aligned} \mathcal{R}_v \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} + \dots \\ &\dots \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} + \mathcal{R}_{\theta_l^t} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}, \end{aligned} \quad (\text{A.19})$$

$\mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \mathcal{R}_{m_{F_i^j}} \left\{ \sum_{t=1}^{\Upsilon} \left([f_{s_{m_o}}^t(\mathbf{x}^{(q)}) - y^t(q)] \dots \right. \right. \\ &\dots \left. \left. [\theta_l^t(q) - f_{s_{m_o}}^t(\mathbf{x}^{(q)})] b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \right\}. \end{aligned} \quad (\text{A.20})$$

Analyzing for $j \neq l$, we obtain

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \sum_{t=1}^{\Upsilon} \left(\sum_{j=1}^M \left(\sum_{i=1}^p (a_{F_i^j}(q) v_{m_i^j}) \dots \right. \right. \\ &\dots \left. \left. (e_{\theta_j}^t(q) e_{\theta_l}^t(q) - e_{\theta_j}^t(q) e^t(q) - e^t(q) e_{\theta_l}^t(q)) \phi_l(\mathbf{x}^{(q)}) \right) \right) \end{aligned} \quad (\text{A.21})$$

if $j = l$ and $i \neq k$, then (A.21) is replaced by

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \sum_{t=1}^{\Upsilon} \left(\sum_{i=1}^p (a_{F_i^l}(q) v_{m_i^l}) \dots \right. \\ &\dots \left. \left(e^t(q) e_{\theta_l}^t(q) - 2e^t(q) e_{\theta_l}^t(q) \phi_l(\mathbf{x}^{(q)}) + (e_{\theta_l}^t(q))^2 \phi_l(\mathbf{x}^{(q)}) \right) \right). \end{aligned} \quad (\text{A.22})$$

and if $i = k$, then A.22 is replaced by

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) v_{m_k^l} \sum_{t=1}^{\Upsilon} \left(a_{F_k^l}(q) \dots \right. \\ &\dots \left. \left(e^t(q) e_{\theta_l}^t(q) - 2e^t(q) e_{\theta_l}^t(q) \phi_l(\mathbf{x}^{(q)}) + (e_{\theta_l}^t(q))^2 \phi_l(\mathbf{x}^{(q)}) \right) \right) \dots \end{aligned} \quad (\text{A.23})$$

$$\dots - 2 \sum_{t=1}^{\Upsilon} \left(e^t(q) e_{\theta_l}^t(q) \right) \phi_l(\mathbf{x}^{(q)}) \frac{(x_k^{(q)} - m_{F_k^l}(q))}{\sigma_{F_k^l}^3(q)}$$

$\mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\begin{aligned} \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \mathcal{R}_{\sigma_{F_i^j}} \left\{ \sum_{t=1}^{\Upsilon} \left([f_{s_{mo}}^t(\mathbf{x}^{(q)}) - y^t(q)] \cdots \right. \right. \\ &\quad \left. \left. \cdots [\theta_l^t(q) - f_{s_{mo}}^t(\mathbf{x}^{(q)})] b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \right\}. \end{aligned} \quad (\text{A.24})$$

Analyzing for $j \neq l$, we obtain

$$\begin{aligned} \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \sum_{t=1}^{\Upsilon} \left(\sum_{j=1}^M \left(\sum_{i=1}^p (b_{F_i^j}(q) v_{\sigma_i^j}) \cdots \right. \right. \\ &\quad \left. \left. \cdots (e_{\theta_j}^t(q) e_{\theta_l}^t(q) - e_{\theta_j}^t(q) e^t(q) - e^t(q) e_{\theta_l}^t(q)) \phi_j(\mathbf{x}^{(q)}) \right) \right) \end{aligned} \quad (\text{A.25})$$

if $j = l$ and $i \neq k$, then (A.25) is replaced by

$$\begin{aligned} \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \sum_{t=1}^{\Upsilon} \left(\sum_{i=1}^p (b_{F_i^l}(q) v_{\sigma_i^l}) \cdots \right. \\ &\quad \left. \cdots (e^t(q) e_{\theta_l}^t(q) - 2e^t(q) e_{\theta_l}^t(q) \phi_l(\mathbf{x}^{(q)}) + (e_{\theta_l}^t(q))^2 \phi_l(\mathbf{x}^{(q)})) \right) \end{aligned} \quad (\text{A.26})$$

and if $i = k$, then (A.26) is replaced by

$$\begin{aligned} \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) v_{\sigma_k^l} \sum_{t=1}^{\Upsilon} (b_{F_k^l}(q) \cdots \\ &\quad \cdots (e^t(q) e_{\theta_l}^t(q) - 2e^t(q) e_{\theta_l}^t(q) \phi_l(\mathbf{x}^{(q)}) + (e_{\theta_l}^t(q))^2 \phi_j(\mathbf{x}^{(q)}))) \cdots \\ &\quad \cdots - 3 \sum_{t=1}^{\Upsilon} (e^t(q) e_{\theta_l}^t(q)) \phi_l(\mathbf{x}^{(q)}) \frac{(x_k^{(q)} - m_{F_k^l}(q))^2}{\sigma_{F_k^l}^4(q)}. \end{aligned} \quad (\text{A.27})$$

$\mathcal{R}_{\theta_j^t} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\begin{aligned} \mathcal{R}_{\theta_j^t} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \mathcal{R}_{\theta_j^t} \left\{ \sum_{t=1}^{\Upsilon} \left([f_{s_{mo}}^t(\mathbf{x}^{(q)}) - y^t(q)] \cdots \right. \right. \\ &\quad \left. \left. \cdots [\theta_l^t(q) - f_{s_{mo}}^t(\mathbf{x}^{(q)})] b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \right\}. \end{aligned} \quad (\text{A.28})$$

Analyzing for $j \neq l$, we obtain

$$\begin{aligned} \mathcal{R}_{\theta_j^t} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \cdots \\ &\cdots \sum_{t=1}^{\Upsilon} \left(\sum_{j=1}^M (\phi_j(\mathbf{x}^{(q)}) v_{\theta_j^t}) (e_{\theta_l}^t(q) - e^t(q)) \right) \end{aligned} \quad (\text{A.29})$$

and if $j = l$ and $t = \Upsilon$, then (A.29) is replaced by

$$\begin{aligned} \mathcal{R}_{\theta_j^t} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) v_{\theta_l^\Upsilon} \cdots \\ &\cdots (e_{\theta_l}^\Upsilon(q) \phi_l(\mathbf{x}^{(q)}) + (1 - \phi_l(\mathbf{x}^{(q)})) e^\Upsilon(q)). \end{aligned} \quad (\text{A.30})$$

•Deduction of $\mathcal{R}_v \left\{ \nabla_{\theta_l^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$

$$\begin{aligned} \mathcal{R}_v \left\{ \nabla_{\theta_l^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\theta_l^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\} \cdots \\ &\cdots + \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\theta_l^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\} + \mathcal{R}_{\theta_j^t} \left\{ \nabla_{\theta_l^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\}. \end{aligned} \quad (\text{A.31})$$

$\mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\theta_l^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\theta_l^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\} = \mathcal{R}_{m_{F_i^j}} \left\{ [f_{s_{mo}}^t(\mathbf{x}^{(q)}) - y^t(q)] \phi_l(\mathbf{x}^{(q)}) \right\}. \quad (\text{A.32})$$

Analyzing for $j \neq l$, we obtain

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\theta_l^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \phi_l(\mathbf{x}^{(q)}) \sum_{j=1}^M \left(\sum_{i=1}^p (a_{F_i^j}(q) v_{m_i^j}) \cdots \right. \\ &\left. \cdots (e_{\theta_j}^t(q) - e^t(q)) \phi_j(\mathbf{x}^{(q)}) \right) \end{aligned} \quad (\text{A.33})$$

if $j = l$, then (A.33) is replaced by

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\theta_l^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \phi_l(\mathbf{x}^{(q)}) \sum_{i=1}^p (a_{F_i^l}(q) v_{m_i^l}) \cdots \\ &\cdots (e_{\theta_l}^t(q) \phi_l(\mathbf{x}^{(q)}) + (1 - \phi_l(\mathbf{x}^{(q)})) e^t(q)). \end{aligned} \quad (\text{A.34})$$

$\mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\theta_l^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\theta_l^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\} = \mathcal{R}_{\sigma_{F_i^j}} \left\{ [f_{s_{mo}}^t(\mathbf{x}^{(q)}) - y^t(q)] \phi_l(\mathbf{x}^{(q)}) \right\}. \quad (\text{A.35})$$

Analyzing for $j \neq l$, we obtain

$$\begin{aligned} \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\theta_l^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \phi_l(\mathbf{x}^{(q)}) \sum_{j=1}^M \left(\sum_{i=1}^p (b_{F_i^j}(q) v_{\sigma_i^j}) \cdots \right. \\ &\quad \left. \cdots (e_{\theta_j^t}^t(q) - e^t(q)) \phi_j(\mathbf{x}^{(q)}) \right) \end{aligned} \quad (\text{A.36})$$

if $j = l$, then (A.36) is replaced by

$$\begin{aligned} \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\theta_l^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \phi_l(\mathbf{x}^{(q)}) \sum_{i=1}^p \left((b_{F_i^l}(q) v_{\sigma_i^l}) \cdots \right. \\ &\quad \left. \cdots (e_{\theta_l^t}^t(q) \phi_l(\mathbf{x}^{(q)}) + (1 - \phi_l(\mathbf{x}^{(q)})) e^t(q)) \right). \end{aligned} \quad (\text{A.37})$$

$\mathcal{R}_{\theta_j^t} \left\{ \nabla_{\theta_l^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\mathcal{R}_{\theta_j^t} \left\{ \nabla_{\theta_l^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\} = \mathcal{R}_{\theta_j^t} \left\{ [f_{smo}^t(\mathbf{x}^{(q)}) - y^t(q)] \phi_l(\mathbf{x}^{(q)}) \right\}, \quad (\text{A.38})$$

$$\mathcal{R}_{\theta_j^t} \left\{ \nabla_{\theta_l^t} \mathbf{J}(\mathbf{w}^{(q)}) \right\} = \phi_l(\mathbf{x}^{(q)}) \sum_{j=1}^M (\phi_j(q) v_{\theta_j^t}). \quad (\text{A.39})$$

Appendix B – Deduction of the $\mathbf{H}(\mathbf{w}^{(q)}) \mathbf{v}^{(q)}$ from $\mathcal{R}\{\cdot\}$ operator for T1-FLS

Considering the algorithm presented in Figure 3, we know that $\mathbf{v}^{(q)} = \nabla \mathbf{J}(\mathbf{w}^{(q)})$. To compute the $\mathbf{H}(\mathbf{w}^{(q)}) \mathbf{v}^{(q)}$ with the differential operator $\mathcal{R}\{\cdot\}$, we can apply the $\mathcal{R}\{\cdot\}$ operator separately in each gradient of the cost function, as follow:

$$\begin{aligned} \mathcal{R}_{\mathbf{v}} \left\{ \nabla \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \left[\mathcal{R}_{\mathbf{v}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} \dots \right. \\ &\quad \left. \dots \mathcal{R}_{\mathbf{v}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} \mathcal{R}_{\mathbf{v}} \left\{ \nabla_{\theta_l} \mathbf{J}(\mathbf{w}^{(q)}) \right\} \right], \end{aligned} \quad (\text{B.1})$$

where

$$\begin{aligned} \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) &= \left[\frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial m_{F_1^1}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial m_{F_p^1}(q)}, \dots \right. \\ &\quad \left. \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial m_{F_1^M}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial m_{F_p^M}(q)} \right], \end{aligned} \quad (\text{B.2})$$

$$\begin{aligned} \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) &= \left[\frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \sigma_{F_1^1}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \sigma_{F_p^1}(q)}, \dots \right. \\ &\quad \left. \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \sigma_{F_1^M}(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \sigma_{F_p^M}(q)} \right] \end{aligned} \quad (\text{B.3})$$

and

$$\nabla_{\theta_l} \mathbf{J}(\mathbf{w}^{(q)}) = \left[\frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \theta_1(q)}, \dots, \frac{\partial \mathbf{J}(\mathbf{w}^{(q)})}{\partial \theta_M(q)} \right]. \quad (\text{B.4})$$

For reasons of simplicity the following notations have been adopted:

$$e(q) = f_s(\mathbf{x}^{(q)}) - y^{(q)}, \quad (\text{B.5})$$

$$e_{\theta_l}(q) = \theta_l(q) - f_s(\mathbf{x}^{(q)}), \quad (\text{B.6})$$

$$a_{F_k^l}(q) = \frac{x_k^{(q)} - m_{F_k^l}(q)}{\sigma_{F_k^l}^2(q)} \quad (\text{B.7})$$

and

$$b_{F_k^l}(q) = \frac{(x_k^{(q)} - m_{F_k^l}(q))^2}{\sigma_{F_k^l}^3(q)}. \quad (\text{B.8})$$

• **Deduction of the $\mathcal{R}_v \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$**

$\mathcal{R}_v \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\begin{aligned} \mathcal{R}_v \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} + \dots \\ &\dots \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} + \mathcal{R}_{\theta_j} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}, \end{aligned} \quad (\text{B.9})$$

The terms of $\mathcal{R}_v \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ are given by

Deduction of $\mathcal{R}_v \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$:

$\mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \mathcal{R}_{m_{F_i^j}} \left\{ [f_s(\mathbf{x}^{(q)}) - y^{(q)}] \dots \right. \\ &\dots \left. [\theta_l(q) - f_s(\mathbf{x}^{(q)})] \phi_l(\mathbf{x}^{(q)}) a_{F_k^l}(q) \right\}. \end{aligned} \quad (\text{B.10})$$

Analyzing for $j \neq l$, we obtain:

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \sum_{j=1}^M \left(\sum_{i=1}^p (a_{F_i^j}(q) v_{m_i^j}) \dots \right. \\ &\dots \left. (e_{\theta_j}(q) e_{\theta_l}(q) - e_{\theta_j}(q) e(q) - e(q) e_{\theta_l}(q)) \phi_j(\mathbf{x}^{(q)}) \right), \end{aligned} \quad (\text{B.11})$$

if $j = l$ and $i \neq k$, then (B.11) is replaced by

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \left(\sum_{i=1}^p (a_{F_i^l}(q) v_{m_i^l}) \dots \right. \\ &\dots \left. (e(q) e_{\theta_l}(q) - 2e(q) e_{\theta_l}(q) \phi_l(\mathbf{x}^{(q)}) + (e_{\theta_l}(q))^2 \phi_l(\mathbf{x}^{(q)})) \right) \end{aligned} \quad (\text{B.12})$$

and if $i = k$, then (B.12) is exchanged by

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= (a_{F_k^l}(q))^2 \phi_l(\mathbf{x}^{(q)}) \dots \\ &\dots v_{m_k^l} (e(q) e_{\theta_l}(q) - 2e(q) e_{\theta_l}(q) \phi_l(\mathbf{x}^{(q)}) + (e_{\theta_l}(q))^2 \phi_l(\mathbf{x}^{(q)})) \dots \\ &\dots - e(q) e_{\theta_l}(q) \phi_l(\mathbf{x}^{(q)}) \frac{v_{m_k^l}}{\sigma_{F_k^l}^2}. \end{aligned} \quad (\text{B.13})$$

$$\begin{aligned}
\mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} & \text{ is obtained as follows:} \\
\mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} & = \mathcal{R}_{\sigma_{F_i^j}} \left\{ [f_s(\mathbf{x}^{(q)}) - y^{(q)}] \dots \right. \\
& \left. \dots [\theta_l(q) - f_s(\mathbf{x}^{(q)})] \phi_l(\mathbf{x}^{(q)}) a_{F_k^l}(q) \right\}.
\end{aligned} \tag{B.14}$$

Analyzing for $j \neq l$, we obtain

$$\begin{aligned}
\mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} & = a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \sum_{j=1}^M \left(\sum_{i=1}^p (b_{F_i^j}(q) v_{\sigma_i^j}) \dots \right. \\
& \left. \dots (e_{\theta_j}(q) e_{\theta_l}(q) - e_{\theta_j}(q) e(q) - e(q) e_{\theta_l}(q)) \phi_j(\mathbf{x}^{(q)}) \right),
\end{aligned} \tag{B.15}$$

if $j = l$ and $i \neq k$, then (B.15) is replaced by

$$\begin{aligned}
\mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} & = a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \left(\sum_{i=1}^p (b_{F_i^l}(q) v_{\sigma_i^l}) \dots \right. \\
& \left. \dots (e(q) e_{\theta_l}(q) - 2e(q) e_{\theta_l}(q) \phi_l(\mathbf{x}^{(q)}) + (e_{\theta_l}(q))^2 \phi_l(\mathbf{x}^{(q)})) \right)
\end{aligned} \tag{B.16}$$

and if $i = k$, then (B.16) is replaced by:

$$\begin{aligned}
\mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} & = a_{F_k^l}(q) b_{F_k^l}(q) v_{\sigma_k^l} \phi_l(\mathbf{x}^{(q)}) \dots \\
& \dots (e(q) e_{\theta_l}(q) - 2e(q) e_{\theta_l}(q) \phi_l(\mathbf{x}^{(q)}) + (e_{\theta_l}(q))^2 \phi_l(\mathbf{x}^{(q)})) \dots \\
& \dots - 2v_{\sigma_k^l} \phi_l(\mathbf{x}^{(q)}) e(q) e_{\theta_l}(q) \frac{(x_k^{(q)} - m_{F_k^l}(q))}{\sigma_{F_k^l}^3(q)}.
\end{aligned} \tag{B.17}$$

$\mathcal{R}_{\theta_j} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\begin{aligned}
\mathcal{R}_{\theta_j} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} & = \mathcal{R}_{\theta_j} \left\{ [f_s(\mathbf{x}^{(q)}) - y^{(q)}] \dots \right. \\
& \left. \dots [\theta_l(q) - f_s(\mathbf{x}^{(q)})] \phi_l(\mathbf{x}^{(q)}) a_{F_k^l}(q) \right\}.
\end{aligned} \tag{B.18}$$

Analyzing for $j \neq l$, we obtain

$$\begin{aligned}
\mathcal{R}_{\theta_j} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} & = a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \dots \\
& \dots \left(\sum_{j=1}^M (\phi_j(\mathbf{x}^{(q)}) v_{\theta_j}) (e_{\theta_l}(q) - e(q)) \right),
\end{aligned} \tag{B.19}$$

if $j = l$ and $i \neq k$, then (B.19) is replaced by:

$$\begin{aligned} \mathcal{R}_{\theta_j} \left\{ \nabla_{m_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) v_{\theta_l} \cdots \\ &\cdots \left(e_{\theta_l}(q) \phi_l(\mathbf{x}^{(q)}) + (1 - \phi_l(\mathbf{x}^{(q)})) e(q) \right). \end{aligned} \quad (\text{B.20})$$

•Deduction of $\mathcal{R}_v \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$

$$\begin{aligned} \mathcal{R}_v \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} \cdots \\ &\cdots + \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} + \mathcal{R}_{\theta_j} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}. \end{aligned} \quad (\text{B.21})$$

$\mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \mathcal{R}_{m_{F_i^j}} \left\{ [f_s(\mathbf{x}^{(q)}) - y^{(q)}] \cdots \right. \\ &\left. \cdots [\theta_l(q) - f_s(\mathbf{x}^{(q)})] \phi_l(\mathbf{x}^{(q)}) b_{F_k^l}(q) \right\}. \end{aligned} \quad (\text{B.22})$$

Analyzing for $j \neq l$, we obtain

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \sum_{j=1}^M \left(\sum_{i=1}^p (a_{F_i^j}(q) v_{m_i^j}) \cdots \right. \\ &\left. \cdots (e_{\theta_j}(q) e_{\theta_l}(q) - e_{\theta_j}(q) e(q) - e(q) e_{\theta_l}(q)) \phi_j(\mathbf{x}^{(q)}) \right), \end{aligned} \quad (\text{B.23})$$

if $j = l$ and $i \neq k$, then (B.23) is replaced by

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \left(\sum_{i=1}^p (a_{F_i^l}(q) v_{m_i^l}) \cdots \right. \\ &\left. \cdots (e(q) e_{\theta_l}(q) - 2e(q) e_{\theta_l}(q) \phi_l(\mathbf{x}^{(q)}) + (e_{\theta_l}(q))^2 \phi_l(\mathbf{x}^{(q)})) \right) \end{aligned} \quad (\text{B.24})$$

and if $i = k$, then (B.24) is replaced by

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= b_{F_k^l}(q) a_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) v_{m_k^l} \cdots \\ &\cdots (e(q) e_{\theta_l}(q) - 2e(q) e_{\theta_l}(q) \phi_l(\mathbf{x}^{(q)}) + (e_{\theta_l}(q))^2 \phi_l(\mathbf{x}^{(q)})) \cdots \end{aligned} \quad (\text{B.25})$$

$$\cdots - 2e(q) e_{\theta_l}(q) \phi_l(\mathbf{x}^{(q)}) v_{m_k^l} \frac{(x_k^{(q)} - m_{F_k^l}(q))}{\sigma_{F_k^l}^3(q)}.$$

$\mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\begin{aligned} \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \mathcal{R}_{\sigma_{F_i^j}} \left\{ [f_s(\mathbf{x}^{(q)}) - y^{(q)}] \cdots \right. \\ &\quad \left. \cdots [\theta_l(q) - f_s(\mathbf{x}^{(q)})] \phi_l(\mathbf{x}^{(q)}) b_{F_k^l}(q) \right\}. \end{aligned} \quad (\text{B.26})$$

Analyzing for $j \neq l$, we obtain

$$\begin{aligned} \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \sum_{j=1}^M \left(\sum_{i=1}^p (b_{F_i^j}(q) v_{\sigma_i^j}) \cdots \right. \\ &\quad \left. \cdots (e_{\theta_j}(q) e_{\theta_l}(q) - e_{\theta_j}(q) e(q) - e(q) e_{\theta_l}(q)) \phi_j(\mathbf{x}^{(q)}) \right), \end{aligned} \quad (\text{B.27})$$

if $j = l$ and $i \neq k$, then (B.27) is replaced by

$$\begin{aligned} \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \left(\sum_{i=1}^p (b_{F_i^l}(q) v_{\sigma_i^l}) \cdots \right. \\ &\quad \left. \cdots (e(q) e_{\theta_l}(q) - 2e(q) e_{\theta_l}(q) \phi_l(\mathbf{x}^{(q)}) + (e_{\theta_l}(q))^2 \phi_l(\mathbf{x}^{(q)})) \right) \end{aligned} \quad (\text{B.28})$$

and if $i = k$, then (B.28) is replaced by

$$\begin{aligned} \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= (b_{F_k^l}(q))^2 \phi_l(\mathbf{x}^{(q)}) v_{\sigma_k^l} \cdots \\ &\quad \cdots (e(q) e_{\theta_l}(q) - 2e(q) e_{\theta_l}(q) \phi_l(\mathbf{x}^{(q)}) + (e_{\theta_l}(q))^2 \phi_l(\mathbf{x}^{(q)})) \cdots \\ &\quad \cdots - 3e(q) e_{\theta_l}(q) \phi_l(\mathbf{x}^{(q)}) v_{\sigma_k^l} \frac{(x_k^{(q)} - m_{F_k^l}(q))^2}{\sigma_{F_k^l}^4(q)}. \end{aligned} \quad (\text{B.29})$$

$\mathcal{R}_{\theta_j} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\begin{aligned} \mathcal{R}_{\theta_j} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \mathcal{R}_{\theta_j} \left\{ [f_s(\mathbf{x}^{(q)}) - y^{(q)}] \cdots \right. \\ &\quad \left. \cdots [\theta_l(q) - f_s(\mathbf{x}^{(q)})] \phi_l(\mathbf{x}^{(q)}) b_{F_k^l}(q) \right\}. \end{aligned} \quad (\text{B.30})$$

Analyzing for $j \neq l$, we obtain

$$\begin{aligned} \mathcal{R}_{\theta_j} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) \cdots \\ &\quad \cdots \left(\sum_{j=1}^M (\phi_j(\mathbf{x}^{(q)}) v_{\theta_j}) (e_{\theta_l}(q) - e(q)) \right) \end{aligned} \quad (\text{B.31})$$

if $j = l$, then (B.31) is replaced by

$$\begin{aligned} \mathcal{R}_{\theta_j} \left\{ \nabla_{\sigma_{F_k^l}} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= b_{F_k^l}(q) \phi_l(\mathbf{x}^{(q)}) v_{\theta_l} \cdots \\ &\cdots \left(e_{\theta_l}(q) \phi_l(\mathbf{x}^{(q)}) + (1 - \phi_l(\mathbf{x}^{(q)})) e(q) \right). \end{aligned} \quad (\text{B.32})$$

•Deduction of $\mathcal{R}_v \left\{ \nabla_{\theta_l} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$

$$\begin{aligned} \mathcal{R}_v \left\{ \nabla_{\theta_l} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\theta_l} \mathbf{J}(\mathbf{w}^{(q)}) \right\} + \cdots \\ &\cdots \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\theta_l} \mathbf{J}(\mathbf{w}^{(q)}) \right\} + \mathcal{R}_{\theta_j} \left\{ \nabla_{\theta_l} \mathbf{J}(\mathbf{w}^{(q)}) \right\}. \end{aligned} \quad (\text{B.33})$$

$\mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\theta_l} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ is obtained as follows:

$$\mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\theta_l} \mathbf{J}(\mathbf{w}^{(q)}) \right\} = \mathcal{R}_{m_{F_i^j}} \left\{ [f_s(\mathbf{x}^{(q)}) - y^{(q)}] \phi_l(\mathbf{x}^{(q)}) \right\}. \quad (\text{B.34})$$

Analyzing for $j \neq l$, we obtain

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\theta_l} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \phi_l(\mathbf{x}^{(q)}) \sum_{j=1}^M \left(\sum_{i=1}^p (a_{F_i^j}(q) v_{m_i^j}) \cdots \right. \\ &\left. \cdots (e_{\theta_j}(q) - e(q)) \phi_j(\mathbf{x}^{(q)}) \right), \end{aligned} \quad (\text{B.35})$$

if $j = l$, then (B.35) is replaced by

$$\begin{aligned} \mathcal{R}_{m_{F_i^j}} \left\{ \nabla_{\theta_l} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \phi_l(\mathbf{x}^{(q)}) \sum_{i=1}^p \left((a_{F_i^l}(q) v_{m_i^l}) \cdots \right. \\ &\left. \cdots (e_{\theta_l}(q) \phi_l(\mathbf{x}^{(q)}) + (1 - \phi_l(\mathbf{x}^{(q)})) e(q)) \right). \end{aligned} \quad (\text{B.36})$$

The equations for $\mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\theta_l} \mathbf{J}(\mathbf{w}^{(q)}) \right\}$ are obtained as follows:

$$\mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\theta_l} \mathbf{J}(\mathbf{w}^{(q)}) \right\} = \mathcal{R}_{\sigma_{F_i^j}} \left\{ [f_s(\mathbf{x}^{(q)}) - y^{(q)}] \phi_l(\mathbf{x}^{(q)}) \right\}. \quad (\text{B.37})$$

Analyzing for $j \neq l$, we obtain

$$\begin{aligned} \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\theta_l} \mathbf{J}(\mathbf{w}^{(q)}) \right\} &= \phi_l(\mathbf{x}^{(q)}) \sum_{j=1}^M \left(\sum_{i=1}^p (b_{F_i^j}(q) v_{\sigma_i^j}) \cdots \right. \\ &\left. \cdots (e_{\theta_j}(q) - e(q)) \phi_j(\mathbf{x}^{(q)}) \right), \end{aligned} \quad (\text{B.38})$$

if $j = l$ and $i \neq k$, then (B.38) is replaced by

$$\begin{aligned} \mathcal{R}_{\sigma_{F_i^j}} \left\{ \nabla_{\theta_l} \mathbf{J} \left(\mathbf{w}^{(q)} \right) \right\} &= \phi_l \left(\mathbf{x}^{(q)} \right) \sum_{i=1}^P \left(\left(b_{F_i^l} (q) v_{\sigma_i^l} \right) \cdots \right. \\ &\quad \left. \cdots \left(e_{\theta_l} (q) \phi_l \left(\mathbf{x}^{(q)} \right) + \left(1 - \phi_l \left(\mathbf{x}^{(q)} \right) \right) e(q) \right) \right). \end{aligned} \quad (\text{B.39})$$

$\mathcal{R}_{\theta_j} \left\{ \nabla_{\theta_l} \mathbf{J} \left(\mathbf{w}^{(q)} \right) \right\}$ is obtained as follows:

$$\mathcal{R}_{\theta_j} \left\{ \nabla_{\theta_l} \mathbf{J} \left(\mathbf{w}^{(q)} \right) \right\} = \mathcal{R}_{\theta_j} \left\{ \left[f_s \left(\mathbf{x}^{(q)} \right) - y^{(q)} \right] \phi_l \left(\mathbf{x}^{(q)} \right) \right\}. \quad (\text{B.40})$$

$$\mathcal{R}_{\theta_j} \left\{ \nabla_{\theta_l} \mathbf{J} \left(\mathbf{w}^{(q)} \right) \right\} = \phi_l \left(\mathbf{x}^{(q)} \right) \sum_{j=1}^M \left(\phi_j \left(\mathbf{x}^{(q)} \right) v_{\theta_j} \right). \quad (\text{B.41})$$

Appendix C – Publications

This Appendix presents the list of publications. Items marked with * refer to contributions related to the thesis.

The list of papers published in journal during the master period are as follow:

- * E. P. de Aguiar, F. M. A. Nogueira, R. P. F. Amaral, D. F. Fabri, S. C. Rossignoli, J. G. Ferreira, M. M. B. R. Vellasco, R. Transcheit, P. C. S. Vellasco, and M. V. Ribeiro, "Eann 2014: A fuzzy logic system trained by conjugate gradient methods for fault classification in a switch machine", *Neural Computing and Applications*, vol. 27, no.5 pp. 1175-1189, Jul. 2016.
- * E. P. de Aguiar, R. P. F. Amaral, M. M. B. R. Vellasco, M. V. Ribeiro, "An Enhanced Singleton Type-2 Fuzzy Logic System For Fault Classification in a Railroad Switch Machine", *Electric Power Systems Research*, 2017

The list of conference papers published during the master period are as follows:

- E. P. de Aguiar, R. P. F. Amaral, M. M. B. R. Vellasco, M. V. Ribeiro, "Computing derivatives in interval type-2 fuzzy logic systems trained by steepest descent method for fault classification in a switch machine", In: *IEEE International Conference on Fuzzy Systems*, 2017, Naples/Italy, IEEE International conference of fuzzy systems, Jul. 2017.
- R. A. Campos, R. P. F. Amaral, L. G. da Fonseca, M. L. L. Júnior, E. P. de Aguiar, "Classification of Short Circuit GMA Welding using Type-1 and Singleton Fuzzy Logic System", In: *CILAMCE Ibero-Latin American Congress on Computational Methods in Engineering*, Florianópolis, 2017, accepted for publication.
- A. Q. Z. Bouhid, R. P. F. Amaral, L. G. da Fonseca, E. P. de Aguiar, "Classification of Faults in Switch Machine using Type-1 and Non-singleton Fuzzy Logic System Trained by Hestenes and Stiefel's Conjugate Gradient Method", In: *CILAMCE Ibero-Latin American Congress on Computational Methods in Engineering*, Florianópolis, 2017, accepted for publication.