In **2006 IEEE International Conference on Fuzzy Systems, World Congress on Computational Intelligence**, Vancouver, Canada, IEEE, June 2006, pp.231-236.

2006 IEEE International Conference on Fuzzy Systems
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Air quality assessment using Fuzzy Lattice Reasoning (FLR)

Ioannis N. Athanasiadis, *Associate Member, IEEE,* Vassilis G. Kaburlasos, *Member, IEEE*

***Abstract*—Accurate and on-line decision-making is required by decision support systems including those ones used for environmental information management. This paper focuses on air quality assessment and demonstrates the added value of applying data mining techniques in operational decision-making. More specifically, the application of Fuzzy Lattice Reasoning (FLR) classifier is investigated. An enhanced FLR learning algorithm is presented that employs a sigmoid valuation function for introducing tunable non-linearities. The FLR classifier is applied here beyond the unit-hypercube. The FLR with a sigmoid positive valuation function demonstrates an improved performance on a dataset from the region of Valencia, Spain regarding an environmental problem. Descriptive decision-making knowledge (i.e. rules) for classification is also induced.**

I. Introduction

AMBIENT AIR QUALITY assessment and management is characterized by both complexity and uncertainty mainly due to the difficulties of atmospheric chemistry and physics and the stochastic processes involved in air pollutant generation. These boundaries raise the major obstacles in building simple models capable for dependable prediction. In most cases, decision making relies on human expertise and experience, as analytical models are too complex and slow for operational decision support. Legislation in Europe and the US define environmental quality indicators, which could be communicated to the public on-time (or even in advance) for informing population about air quality, especially in urban areas.

As a consequence, simple, yet dependable prediction models are required for achieving both the requirements of accurate air quality assessment and capabilities for fast decision making (in contrast with the analytical complex models). These properties can be realized by learning from data, using knowledge discovery techniques [1], [2]. In this context, quantitative data-driven decision support models are challenged by difficulties in handling dynamic and uncertain real-world environments.

This paper tackles the problem of operational decision support in air quality assessment by utilizing a machine learning approach. Specifically, the Fuzzy Lattice Reasoning (FLR) classifier is presented here and used for prediction of ozone pollutant concentration levels, in a rural area in Valencia, Spain. A simplified version of the FLR classifier has been presented previously implemented as a neural network, namely σ-FLN [6], [7]. This work focuses on the 'learning algorithm' rather than on a specific implementation. An additional novelty here is the employment of a non-linear positive valuation function. Finally, a real world application is demonstrated regarding an environmental air quality assessment. The proposed methods produce better results comparatively.

II. Mathematical Background

This section summarizes briefly the required mathematics.

A *lattice* L is a partially ordered set (*poset*), so that any two of its elements $a,b\in$L have both a greatest lower bound (or *meet*) denoted by $a\wedge b:=inf\{a,b\}$ and a least upper bound (or *join*) denoted by $a\vee b:=sup\{a,b\}$. A lattice L is called *complete* when each of its subsets has both a least upper bound and a greatest lower bound in L. A non-void complete lattice has a least element and a greatest element denoted by O and I, respectively.

The Cartesian product L= $L_1\times\ldots\times L_N$ of N *constituent* lattices $L_1,\ldots,L_N$ is a lattice [1]. In a product lattice L= $L_1\times\ldots\times L_N$ inclusion can be defined as:
$(x_1,\ldots x_N) \le (y_1,\ldots,y_N)$ if and only if $x_1 \le y_1,\ldots, x_N \le y_N$.

The *meet* in a product lattice L= $L_1\times\ldots\times L_N$ is given by $(x_1,\ldots x_N)\wedge(y_1,\ldots,y_N)= (x_1\wedge y_1,\ldots,x_N\wedge y_N)$, whereas the *join* in L is given by
$(x_1,\ldots x_N)\vee(y_1,\ldots,y_N)= (x_1\vee y_1,\ldots,x_N\vee y_N)$ [1], [4].

A product lattice could combine diverse constituent lattices thus implying the potential to deal either separately and/or jointly, in any combination, with disparate types of data such as vectors of real numbers, propositions, (fuzzy) sets, events in a probability space, symbols, graphs, etc.

A *fuzzy lattice* is a pair ⟨L,μ⟩, where L is a lattice and (L×L,μ) is a fuzzy set with membership function μ: L×L→[0,1] such that $\mu(a,b) = 1$ if and only if $a\le b$ [6].

A *valuation function* v: L → R is defined on a lattice L as

Manuscript received February 14, 2006.

I. N. Athanasiadis is with the Dalle Molle Institute for Artificial Intelligence (USI/SUPSI), CH6928 Manno-Lugano, Switzerland (phone: +41 58 666 6660; fax: +41 58 666 6661; e-mail: ioannis@idsia.ch; homepage: www.athanasiadis.info).

V. G. Kaburlasos is with the Department of Industrial Informatics, Technological Educational Institution of Kavala, GR65404 Kavala, Greece (email: vgkabs@teikav.edu.gr).

This work was partially supported by the third European framework programme: Operational Programme in Education and Initial Vocational Training II, under project Archimedes I contract no. 04-3-001/1.

a real function that satisfies: $v(a)+v(b) = v(a\wedge b)+v(a\vee b)$, $a,b\in$L. A valuation function, is called *positive* if and only if $a<b \Rightarrow v(a)<v(b)$ [4]. A positive valuation function has been used in previous works for defining an *inclusion measure* (σ) in a complete lattice L.

In general, an *inclusion measure* on a lattice L is defined as a real function σ: L×L→[0,1], such that for each $a,b,x\in$L the following conditions are satisfied:

$\sigma(a,O) = 0, a\neq O$ (C0)

$\sigma(a,a) = 1, \forall a\in L$ (C1)

$a \leq b \Rightarrow \sigma(x,a) \leq \sigma(x,b)$ – *Consistency Property* (C2)

$a\wedge b < a \Rightarrow \sigma(a,b) < 1$. (C3)

Condition (C0) is required only for complete lattices. Given a lattice L and an inclusion measure σ: L×L→[0,1] it turns out that ⟨L,σ⟩ is a fuzzy lattice.

Another useful tool implied by a positive valuation in a general lattice L is a *metric distance* function d: L×L→R defined as $d(x,y)= v(x\vee y)-v(x\wedge y)$.

A positive valuation function v: L→R in a lattice L, with $v(O)=0$, is a sufficient condition for two inclusion measures:

$$k(a,b)= \frac{v(b)}{v(a\vee b)}, \text{ and} \quad \text{(IM1)}$$

$$s(a,b)= \frac{v(a\wedge b)}{v(a)}. \quad \text{(IM2)}$$

Ultimately, given a lattice L, for which a positive valuation function v: L→R can be defined, both ⟨L,k⟩ and ⟨L,s⟩ are fuzzy lattices.

The set of fuzzy lattices ⟨L,μ⟩ is called *framework of fuzzy lattices* [6]. The framework of fuzzy lattices has been used for decision-making in various applications [5]–[9]. This paper approaches the fuzzy lattice framework from a rule-based perspective presented below.

III. Fuzzy Lattice Reasoning (FLR) Classifier

Many data structures of practical interest are lattice ordered. The objective here is to present a classifier for inducing a rule-based inference engine from data.

A *fuzzy lattice rule* is a pair ⟨a,c⟩ where a is an element in a fuzzy lattice ⟨L,μ⟩ and c∈C is a categorical label. In this sense, a fuzzy lattice rule can be interpreted as the mapping a→c of a fuzzy lattice ⟨L,μ⟩ element a to a categorical label c.

Let a and b be two lattice L elements, and function k, as defined in (IM1) be a fuzzy membership function in L×L. The degree of truth of the fuzzy lattice rule a→c against an evidence (antecedent) b is:

$$\mu(b,a) = k(b,a) = \frac{v(a)}{v(b\vee a)}.$$

A *fuzzy lattice rule engine* $\mathcal{E}_{\langle L,\mu\rangle}$ can be thought of as a set of fuzzy lattice rules $\{a_i\rightarrow c_i: a_i\in\langle L,\mu\rangle\}$, that are commonly activated. *Reasoning* with a fuzzy lattice rule engine implies the calculation of the degree of truth for each one of engine's rules. For example consider the following engine that consists of three rules:

$\mathcal{E}_{\langle L,\sigma\rangle} = \{a_1\rightarrow c_1, a_2\rightarrow c_2, a_3\rightarrow c_3\}$,

where a_1, a_2, a_3, are elements of the fuzzy lattice ⟨L,σ⟩.

In response to an input element a_0, the engine will compute the following degrees of truth for each consequence: $c_1 = \sigma(a_0, a_1)$, $c_2 = \sigma(a_0, a_2)$, and $c_3 = \sigma(a_0, a_3)$. In this way, a fuzzy lattice reasoning engine can be used for generalization.

A. Fuzzy Lattice Rule induction (FLR classifier)

The task of inducing a fuzzy lattice rule engine is described as follows: Let M be a training set of partially ordered objects $\{u_1, u_2, \ldots, u_M\}\subseteq$U, each one of which is associated with a class label $c\in$C, where C=$\{c_1, c_2, \ldots, c_K\}$ is a set of K predefined labels (classes). The objective is to induce a set of fuzzy lattice rules that implement a function h:U→C, the latter associates an object $u\in$U to a classification label $c\in$C.

In general, the universe U of the training objects can include any type of complex data structures, as vectors of real numbers, graphs or sets. We are interested in applications where U is a lattice. Given a positive valuation function v: U→R, an inclusion measure σ: U×U→[0,1] can be defined in U, as shown above, for implying a fuzzy membership function μ: U×U→[0,1]. In this respect, ⟨U,μ⟩ is a fuzzy lattice. The classifier to be built is equivalent to a map h': ⟨U,μ⟩→C, which is a set of fuzzy lattice rules.

Each object u of the training set is an element of U and each training pair ⟨u,c⟩ can be expressed as a fuzzy lattice rule ⟨u,c⟩, where u is an element of the fuzzy lattice ⟨U,μ⟩ and c is the corresponding class. This means that the instances of a training set could be treated as fuzzy lattice rules. For example consider the simple case where the universe of the training instances is a closed interval of real numbers [O,I]. Then any training pair (x,c) where x∈[O,I] and c∈C can be expressed as a fuzzy lattice rule consisted from a trivial point lattice mapping it to class c, i.e. ⟨x,c⟩ ≡ ⟨[x,x],c⟩. Likewise for alternative universes of discourse.

A naive fuzzy lattice rule classifier that can be induced directly from a set of M training pairs $\{(u_1,c_1), \ldots, (u_M,c_M)\}$, $u_i\in$U, and $c_i\in$C, is the one that memorizes all training instances as fuzzy lattice rules. Given a positive valuation function v, each training element u_i is an element of the fuzzy lattice ⟨U,σ⟩, where σ is an inclusion measure defined by either (IM1) or (IM2). In this way, the most simple fuzzy lattice rule engine will consist at most of M (trivial) rules and will be:

$\mathcal{E} = \{u_1\rightarrow c_1, \ldots, u_i\rightarrow c_i, \ldots, u_M\rightarrow c_M\}$,

where u∈⟨U,σ⟩ and $c_i\in$C.

The training process for inducing a fuzzy lattice rule engine is based on joining 'similar' lattice rules pointing to the same class for formulating lattice rules of higher size,

Box 1.
The Induction of a Fuzzy Lattice Reasoning Classifier

Step-0: Let $\mathcal{E}_{\langle L,\sigma\rangle}=\{a_1\to c_1,\ldots, a_R\to c_R\}$ be a fuzzy lattice rule engine.
Note that $\mathcal{E}_{\langle L,\sigma\rangle}$ could initially be empty, i.e. R=0.
Step-1: Present the next training pair (u,c), in the form of a fuzzy lattice rule $u\to c$ to the initially "set" rules in $\mathcal{E}_{\langle L,\sigma\rangle}$.
Step-2: If no more rules in $\mathcal{E}$ are "set" then: Include input rule $u\to c$ in $\mathcal{E}$; R←R+1; Go to Step-1. Else, compute the fuzzy degree of inclusion $\sigma(u\le a_r)$, $l\in\{1,\ldots,R\}$ of antecedent u to the antecedents of all the "set" rules in $\mathcal{E}_{\langle L,\sigma\rangle}$.
Step-3: Competition among the "set" rules in $\mathcal{E}_{\langle L,\sigma\rangle}$. Winner is the rule $a_J\to c_J$, where J= $\arg\max_{r\in\{1,\ldots R\}} \sigma(u\le a_r)$.
Step-4: The *Assimilation Condition*: Test whether both $diag(u\vee a_J)$ is less than a maximum user-defined threshold size D_{crit} and c = c_J.
Step-5: If the *assimilation condition* is satisfied then replace the antecedent a_J of the winner rule $a_J\to c_J$ by the join-lattice $u\vee a_J$, i.e. with the rule $u\vee a_J\to c_J$. Else, "reset" the winner rule $a_J\to c_J$, and go to Step-2.

and potentially higher ability for generalization. A 'similarity' is computed in an inclusion measure sense. The training procedure for inducing a rule engine $\mathcal{E}_{\langle L,\sigma\rangle}$ is accomplished in a single iteration over all training instances as shown in Box 1.

Previous work has employed for tuning the dimensionless *vigilance parameter*:

$$\rho_{crit}=\frac{N}{N+D_{crit}} \Leftrightarrow D_{crit}=\frac{N(1-\rho_{crit})}{\rho_{crit}}$$

instead of D_{crit}. Note that ρ_{crit} varies in the interval [0.5,1] for any number of dimensions N as shown in [6]. In the experiments below ρ_{crit} has been employed.

The decision making process (testing phase) of an induced fuzzy rule engine $\mathcal{E}_{\langle L,\sigma\rangle}$ of size R, involves competition of its rules over an evidence (antecedent) $x\in U$, of unknown label. In an iterative process, the element x is presented to each rule of the engine: $a_r\to c_r$, and the inclusion measure $\sigma(x,a_r)$ is calculated. Finally, x is assigned to the category c_J, where J = $\arg\max_{r\in\{1,\ldots R\}} \sigma(x\le a_r)$.

In principal, the Fuzzy Lattice Reasoning (FLR) classifier can be applied on any universe of partially ordered objects, as that of $\Re^N$, graphs or sets. Note that a similar lattice algorithm, namely *Find-S algorithm*, has been presented in a machine learning context (Mitchell 1997) but without an employment of positive valuation functions.

B. FLR classifier with a sigmoid valuation function

The application of a FLR classifier depends on an appropriate valuation function in U, which is used to compute an inclusion measure such as those defined in (IM1), (IM2). This remark holds also for other decision making schemes built upon the framework of fuzzy lattices that map data to lattices. Typically, prior works [5]–[10] have focused in forming fuzzy lattices from numerical datasets by employing linear valuation functions. In cases that data reside in the N–dimensional unit hypercube $I^N=[0,1]\times[0,1]\times\ldots\times[0,1]$, the positive valuation function selected for each consistuent lattice is the simple function $v_i(x)=x$. In other cases, where data reside in $\Re^N$ the training dataset can be formulated as $T=[O_1,I_1]\times[O_2,I_2]\times\ldots\times[O_N,I_N]$, and the positive valuation function for each constituent lattice is $v_i(x)=(x-O)/(I-O)$, which linearly scales T to the N–dimensional unit hupercube I^N.

In this paper, both linear and non-linear positive valuation functions are considered for applying the FLR classifier on a numerical dataset. The sigmoid function is an example of non-linear increasing function with range [0,1] that could be used as a positive valuation function for mapping an interval of real numbers to a fuzzy lattice. Specifically, in the case of data residing in I, a positive valuation function is defined by the sigmoid function

$$v_\lambda(x)=\frac{1}{1+e^{-\lambda(x-0.5)}}. \qquad \text{(V1)}$$

In the generic case of data residing of the interval [X_{min}, X_{max}], a positive valuation function is defined by the sigmoid function

$$v_\varsigma(x)=\frac{1}{1+e^{-\lambda(x-x_{med})}}, \qquad \text{(V2)}$$

where $x_{med}=\frac{X_{max}+X_{min}}{2}$, and $\lambda=\frac{\varsigma}{X_{max}-X_{min}}$, $\varsigma>0$. The single parameter λ, or the normalized equivalent ς can be used for tuning the slope of $v_\lambda(x)$ and $v_\varsigma(x)$. Figure 1 plots function $v_\varsigma(x)$ for various values of the normalized slope ς, in contrast with the linear valuation function $v(x)=\frac{x-X_{min}}{X_{max}-X_{min}}$, where $x\in[X_{min}, X_{max}]$.

The capacity of non-linear positive valuation functions to improve performance has been demonstrated lately in classification and regression applications [11]–[13]. In the following section we demonstrate experimental results of applying the FLR classifier on air quality data.

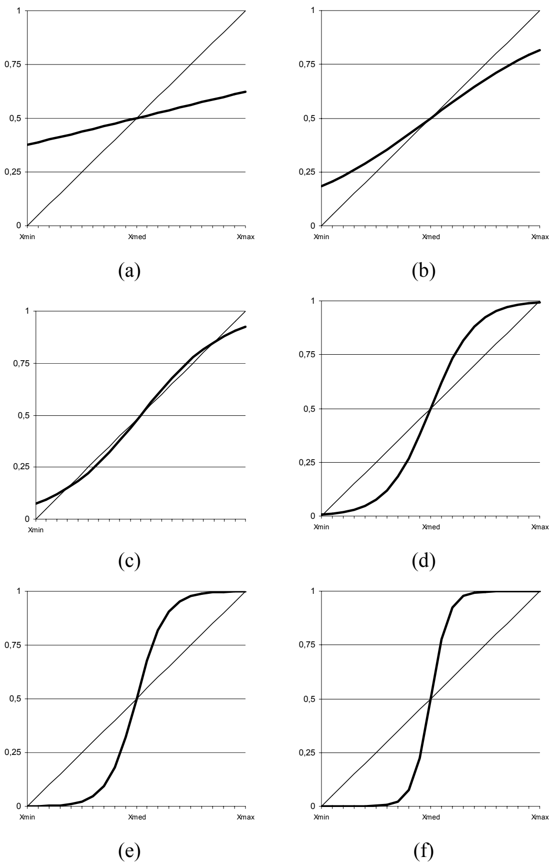

Fig. 1. Sigmoid positive valuation functions $v_\varsigma(x)$ illustrated in the interval $[X_{min}, X_{max}]$ in contrast with the linear positive valuation function $v_L(x)$ for various values of the normalized parameter ς. (a) ς=1, (b) ς=3, (c) ς=5, (d) ς=10, (e) ς=15, and (f) ς=25.

IV. Application On Air Quality Data

A. The problem and data preprocessing

The FLR classifier was applied on an air quality data set recorded in the region of Valencia, Spain. Eight variables, including both meteorological attributes and air-pollutant concentrations, have been sampled on a quarter-hourly basis during the year 2001. The target variable is the level of ozone concentration, a critical photochemical pollutant in urban areas, which is commonly used as an indicator of the overall ambient air quality. Ozone is a secondary pollutant formed as a result of catalytic reactions between pollutants emitted from industrial sources and automobiles. In the presence of sunlight (ultra-violet radiation) and, under suitable meteorological conditions, the precursors react photochemically to "produce" ozone. Due to the chemical reaction dynamics, the models analytical models for describing ozone formation in ambient air are very complex. To tackle this problem we employed a machine learning approach for estimating ozone concentration levels by classification in the available dataset.

In total there are available 35,040 data vectors, out of which 565 records have the ozone label missing, and thus where excluded in the analysis below. Values were missing in other attributes, and in total there are 6,020 records (that is around 17% of the total) with at least one missing value. In the following we used both the original dataset with missing values and a preprocessed one that included all records with no missing values. The attributes of the dataset are summarized in Table I.

B. Experimental results

For estimating the ozone concentration level we employed three classifiers in our experiments:

a. The C4.5 classifier
b. The FLR classifier, with a linear positive valuation function
c. The FLR classifier, with a sigmoid positive valuation function

Two series of experiments have been carried out: first, using the dataset without missing values and, second, the original set including the ones with missing values. In both aforementioned series of experiments the data collected from January 1, 2001 until mid June have been used for training, whereas the remaining data until year end have been used for testing. The corresponding numbers of data vectors available in classes 'low' and 'med', respectively, are shown in Table II.

First, the C4.5 classifier has been employed on a standard software platform (WEKA platform [15]), for generating decision trees, in which the internal nodes specify inequalities for the values of environmental attributes, moreover the tree leaves specify an output class. Initially, the C4.5 classifier has been applied on the data without missing values, without pruning, resulting in a decision tree with

1393 leaves (rules). The corresponding classification accuracy on the training set reached 94.8%, whereas on the testing set it was only 64.85% (Table III). Similar results have been obtained for the dataset with no missing values. Obviously, C4.5 overfits the training data, therefore two pruning methods have been employed: (1) Confidence Factor Pruning (CFP), and (2) Reduced Error Pruning (REP).

TABLE I
DATASET ATTRIBUTES

	Data Attribute Name	Symbol	Data Type	Units
1	Sulfur dioxide	SO_2	real number	μg/m^3
2	Nitrogen oxide	NO	real number	μg/m^3
3	Nitrogen dioxide	NO_2	real number	μg/m^3
4	Nitrogen oxides	NO_x	real number	μg/m^3
5	Wind velocity	VEL	real number	m/s
6	Temperature	TEM	real number	^{o}C
7	Relative humidity	HR	real number	%
8	Ozone Concentration Level	O_3	class label 'low' (0–60 μg/m^3) 'medium' (60–100 μg/m^3)	

TABLE II
DATASET STATISTICS

	Records in class low	Records in class medium
DATASET WITHOUT MISSING VALUES		
Training	6,865	4,761
Testing	12,256	5,138
DATASET WITH MISSING VALUES		
Training	9,472	6,074
Testing	13,483	5,446

Results are shown in Tables III and VI for selected pruning parameter values. The highest accuracy achieved on the testing split was 73.74% and 77.56% respectively for each dataset.

The FLR classifier has been implemented on the same software platform (WEKA) using both linear and sigmoid valuation functions. Initially, the FLR Classifier has been employed using a linear valuation function. In this case, the valuation function used was $v_i(x) = (x–O)/(I–O)$, where [O, I] are the minimum and maximum values of the training data in each dimension. Results are presented in Tables IV and VII for selected values of the vigilance parameter ρ. The FLR Classifier achieved a classification accuracy of 83.23% with only three rules for the dataset without missing values and 84.60% with 19 rules for the dataset with missing values. Note that the FLR classifier outperforms C4.5.

Then, experiments have been conducted for the FLR Classifier using the sigmoid function of Equation (V2) on both datasets. In this case the FLR Classifier has been tuned using two parameters: The vigilance parameter ρ and the slope parameter of the sigmoid valuation function ς. Results obtained by FLR with sigmoid valuation function are presented in Tables V and VIII. For the dataset without missing values the FLR with sigmoid positive valuation function achieved a classification accuracy of 85.22% with three rules. Note that using the sigmoid positive valuation function the best performance has improved by 2% without increasing the number of induced rules. For the dataset with missing values, the best accuracy improved by nearly 1%, again without increasing the number of rules, as shown in Tables VII and VIII.

V. DISCUSSION

This paper introduced the Fuzzy Lattice Reasoning (FLR) classifier and demonstrated its usage for assessing ambient air quality. Results obtained with FLR Classifier have compared favorably with the results obtained by C4.5 decision trees. The FLR classifier with linear positive valuation function, compared to C4.5, improved classification accuracy by 9.5% for the dataset without missing values and by 7% for the dataset with missing values. Furthermore, the employment of a sigmoid positive valuation function by the FLR classifier achieved further improvement without increasing the complexity (number of induced rules) of the model.

TABLE III
C4.5 RESULTS FOR THE DATASET WITHOUT MISSING VALUES

Pruning Method	Parameter value	Classification Accuracy (%) Training	Classification Accuracy (%) Testing	Number of Rules (Tree leaves)
Unpruned	–	94.80	64.85	1393
Confidence Factor Parameter: CF	0.1	91.33	67.31	575
	0.2	92.87	66.71	823
	0.3	93.92	67.40	1055
	0.4	94.10	67.39	1101
	0.5	94.31	67.19	1169
Reduced Error Pruning Parameter: no. of Folds	2	89.31	63.71	507
	10	89.01	71.85	465
	50	85.05	60.62	251
	100	**83.33**	**73.74**	**131**
	300	81.55	69.98	75
	500	77.73	72.48	31

TABLE IV
FLR WITH LINEAR VALUATION FUNCTION RESULTS FOR THE DATASET WITHOUT MISSING VALUES

Parameter value (ρ)	Classification Accuracy (%) Training Set	Classification Accuracy (%) Testing Set	Number of Rules
0.5	59.16	70.46	2
0.6	**64.73**	**83.23**	**3**
0.7	73.68	74.85	20
0.8	67.43	72.59	139

TABLE V
FLR WITH SIGMOID VALUATION FUNCTION RESULTS FOR THE DATASET WITHOUT MISSING VALUES

Parameter value (ς)	Parameter value (ρ)	Classification Accuracy (%) Training Set	Classification Accuracy (%) Testing Set	Number of Rules
1	0.5	59.16	70.46	2
	0.6	59.16	70.46	2
	0.7	59.16	70.46	2
	0.8	**62.73**	**85.22**	**3**
5	0.5	59.16	70.46	2
	0.6	65.40	82.70	3
	0.7	70.48	79.64	19
	0.8	67.53	78.72	40
10	0.5	59.16	70.46	2
	0.6	64.27	83.43	3
	0.7	65.77	74.89	34
	0.8	69.56	82.87	115
15	0.5	59.16	70.46	2
	0.6	64.73	83.24	3
	0.7	68.85	78.88	23
	0.8	70.39	81.54	112

TABLE VI
C4.5 RESULTS FOR THE DATASET WITH MISSING VALUES

Pruning Method	Parameter value	Classification Accuracy (%) Training	Classification Accuracy (%) Testing	Number of Rules (Tree leaves)
Unpruned	-	92.18	58.67	798
Confidence Factor Parameter: CF	0.1	89.14	60.26	279
	0.2	89.98	59.19	368
	0.3	90.81	59.44	463
	0.4	91.37	59.30	542
	0.5	91.59	59.32	598
Reduced Error Pruning Parameter: no. of Folds	2	88.14	64.91	318
	10	88.28	59.19	288
	50	85.44	60.17	144
	100	84.01	61.36	84
	300	**82.48**	**77.56**	**44**
	500	81.33	70.19	32

TABLE VII
FLR WITH LINEAR VALUATION FUNCTION RESULTS
FOR THE DATASET WITH MISSING VALUES

Parameter value (ρ)	Classification Accuracy (%) Training Set	Classification Accuracy (%) Testing Set	Number of Rules
0.5	60.99	71.22	5
0.6	60.99	71.22	8
0.7	**63.48**	**84.60**	**19**
0.8	69.00	66.54	43

TABLE VIII
FLR WITH SIGMOID VALUATION FUNCTION RESULTS
FOR THE DATASET WITH MISSING VALUES

Parameter value (ς)	Parameter value (ρ)	Classification Accuracy (%) Training Set	Classification Accuracy (%) Testing Set	Number of Rules
1	0.5	60.99	73.37	2
	0.6	60.99	73.37	2
	0.7	60.99	73.37	3
	0.8	60.99	73.37	4
5	0.5	60.99	71.22	4
	0.6	60.99	71.22	6
	0.7	60.99	71.23	9
	0.8	**65.34**	**85.53**	**19**
10	0.5	60.99	71.22	6
	0.6	60.99	71.23	9
	0.7	60.99	71.22	14
	0.8	63.55	82.55	26
15	0.5	60.99	71.22	6
	0.6	60.99	71.23	10
	0.7	60.99	71.23	17
	0.8	64.00	82.59	31

ACKNOWLEDGEMENT

The authors gratefully acknowledge Fundación Centro de Estudios Ambietales del Mediterráneo (CEAM) and IDI-Eikon, Valencia, Spain for providing with the air quality dataset.

REFERENCES

[1] I. N. Athanasiadis, P. A. Mitkas, "Supporting the Decision-Making Process in Environmental Monitoring Systems with Knowledge Discovery Techniques", *Knowledge-based Services for the Public Sector Symposium: Workshop III: Knowledge Discovery for Environmental Management*, H. Voss, M. Wachowicz, S. Dzeroski, and A. Lanza (eds), Bonn, Germany, KDnet, June 2004, pp.1-12

[2] I. N. Athanasiadis, K. Karatzas and P. Mitkas, "Contemporary air quality forecasting methods: A comparative analysis between statistical methods and classification algorithms", *5th International Conference on Urban Air Quality*, Valencia, March 2005.

[3] B.A. Davey, H.A. Priestley, *Introduction to Lattices and Order*, Cambridge University Press, Cambridge, UK, 1990.

[4] G. Birkhoff, Lattice Theory, Providence, RI: Colloquium Publications (American Mathematical Society), 1967.

[5] V.G. Kaburlasos, "Adaptive Resonance Theory with Supervised Learning and Large Database Applications", Ph.D. Dissertation, Dept. Elect. Eng., Univ. of Nevada, Reno, NV, 1992.

[6] V. G. Kaburlasos and V. Petridis. Fuzzy lattice neurocomputing (FLN) models. *Neural Networks*, 13(10):1145-1170, 2000.

[7] V. G. Kaburlasos and V. Petridis. Learning and decision-making in the framework of fuzzy lattices. In *New Learning Paradigms in Soft Computing*, L.C. Jain and J. Kacprzyk, Eds, vol. 84, pp. 55-96, 2002, Heidelberg, Germany: Physica-Verlag.

[8] V. Petridis, V.G. Kaburlasos, Fuzzy Lattice Neural Network (FLNN): A hybrid model for learning, *IEEE Transactions on Neural Networks*. 9 (5) (1998) 877-890.

[9] I.N. Athanasiadis, V.G. Kaburlasos, P.A. Mitkas, V. Petridis, Applying machine learning techniques on air quality data for real-time decision support, Proceedings of the 1st ICSC-NAISO Symposium on Information Technologies in Environmental Engineering (ITEE'2003), Gdansk, Poland, ICSC-NAISO Academic Press, June 2003, p.51.

[10] V. Petridis, V. G. Kaburlasos, "Learning in the Framework of Fuzzy Lattices", IEEE Transactions on Fuzzy Systems, vol. 7, no. 4, pp. 422-440, 1999.

[11] V.G. Kaburlasos, A. Kehagias, Novel fuzzy inference system (FIS) analysis and design based on lattice theory, part I: Working principles. *International Journal of General Systems*, 2006 (in press).

[12] V.G. Kaburlasos, A. Kehagias, Novel fuzzy inference system (FIS) analysis and design based on lattice theory, part II: Industrial system modeling. *IEEE Transactions on Fuzzy Systems*, 2006 (accepted).

[13] V.G. Kaburlasos, S.E. Papadakis, "Granular Self-Organizing Map (grSOM) for Structure Identification", *Neural Networks*, (in press; available online: Sept. 2005).

[14] T.M. Mitchell, Machine Learning, McGraw-Hill, New York, NY, 1997.

[15] I.H. Witten, E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufman Publishers, San Mateo, CA, 1999.