Artificial Intelligence in Medicine (2004) **32**, 195–216

ELSEVIER

ARTIFICIAL INTELLIGENCE IN MEDICINE

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

Evolving rule-based systems in two medical domains using genetic programming

Athanasios Tsakonas[a], Georgios Dounias[a,*], Jan Jantzen[b], Hubertus Axer[c], Beth Bjerregaard[d], Diedrich Graf von Keyserlingk[e]

[a]*Department of Financial and Management Engineering, University of the Aegean, 31 Fostini St., 82100 Chios, Greece*
[b]*Technical University of Denmark, Oersted-DTU Automation, Dk-2800 Kongens Lyngby, Denmark*
[c]*Department of Neurology, Friedrich-Schiller-University Jena, Philosophenweg 3, D-07743 Jena, Germany*
[d]*Herlev University Hospital, DK-2730 Herlev, Denmark*
[e]*Department of Anatomy I, RWTH Aachen, Pauwelsstr. 30, D-52057 Aachen, Germany*

Received 15 May 2002; received in revised form 6 July 2003; accepted 27 February 2004

KEYWORDS
Hybrid intelligence;
Genetic programming;
Grammar driven GP;
Genetic-fuzzy systems;
Inductive machine learning;
Medical decision making;
Aphasia;
Pap-smear test

Summary

Objective: To demonstrate and compare the application of different genetic programming (GP) based intelligent methodologies for the construction of rule-based systems in two medical domains: the diagnosis of aphasia's subtypes and the classification of pap-smear examinations.
Material: Past data representing (a) successful diagnosis of aphasia's subtypes from collaborating medical experts through a free interview per patient, and (b) correctly classified smears (images of cells) by cyto-technologists, previously stained using the Papanicolaou method.
Methods: Initially a hybrid approach is proposed, which combines standard genetic programming and heuristic hierarchical crisp rule-base construction. Then, genetic programming for the production of crisp rule based systems is attempted. Finally, another hybrid intelligent model is composed by a grammar driven genetic programming system for the generation of fuzzy rule-based systems.
Results: Results denote the effectiveness of the proposed systems, while they are also compared for their efficiency, accuracy and comprehensibility, to those of an inductive machine learning approach as well as to those of a standard genetic programming symbolic expression approach.

* Corresponding author. Tel.: +30 2710 35454; fax: +30 2710 93464.
E-mail addresses: tsakonas@stt.aegean.gr (A. Tsakonas), g.dounias@aegean.gr (G. Dounias), jj@oersted.dtu.dk (J. Jantzen), hubertus.axer@med.uni-jena.de (H. Axer), bebj@herlevhosp.kbhamt.dk (B. Bjerregaard), dkeyserlingk@ukaachen.de (D.G. von Keyserlingk).

0933-3657/$ – see front matter
doi:10.1016/j.artmed.2004.02.007

Conclusion: The proposed GP-based intelligent methodologies are able to produce accurate and comprehensible results for medical experts performing competitive to other intelligent approaches. The aim of the authors was the production of accurate but also sensible decision rules that could potentially help medical doctors to extract conclusions, even at the expense of a higher classification score achievement.

1. Introduction

1.1. Background and literature review

The incorporation of computational intelligence in medical diagnosis is a continuously growing field with a large number of medical applications. Many of the medical diagnosis procedures can be assigned directly to intelligent data classification tasks. These classification procedures can be divided in two types, concerning the number of categories that each time are classified. The first classification type separates the data between only two classes (known as binary classification or two-class task), and the second type classifies the data between more than two classes (multi-class task). For example, there are methods for intelligent classification that handle efficiently the two-class task, such as the AdaBoost and the support vector machines. Any multi-class problem can be substituted by, more than one, two-class problems. Such an approach is to build independent classification rules for each of the classes and then run these competitive rules simultaneously [1]. However, it is not clear how the case of possible classification conflicts between some of the rules would be dealt with. Moreover, it is not clear, how the absence of a positive classification result, would be dealt with, among that kind of competitive crisp rule-bases. This problem is addressed in this paper where the construction of separating rules between two classes (or two groups of classes) is performed in a hierarchical way, creating a cooperative crisp rule-base, rather a competitive one, which always results in one class. The popularity that the rule-based solutions have gained nowadays might be explained by the natural decision method that humans often follow. The latter conclusion is demonstrated for example in the medical field, where physicians usually follow a strategy represented by a complex classification tree [2], proving that medical decision making often resembles to the approaches that this paper is concerned.

The other way of handling multi-class tasks, in order to build a rule-base, is by using directly a multi-class approach. Among the methodologies of this type, inductive decision trees and genetic programming (GP) have been used with success in the past, although the complexity of the classification task often is increased when more than two classes are separated. These multi-class methods can be further divided in two types. The first type, constructs crisp cooperative and hierarchical classification rule-trees, such as Quinlan's inductive decision trees [3]. Although fast and robust, this algorithm is however restricted in terms of each rule's (tree branch) premise set, where the expression evaluated is an inequality between an attribute (input variable) and a value (number). Apparently, a more generic methodology could involve, in the rules' premise sets, the incorporation of more complex comparisons, such as combinations of expressions including more than one attributes and values. The second type of multi-class methods, constructs fuzzy competitive or cooperative rule-bases using for example, heuristic [4] or genetic programming [5,6] techniques. The idea is addressed in [5,7] where the genetic programming approach is used to build a decision tree-like output.

Genetic programming, equipped with a proper function set, has been proved capable in finding optimal solutions in a reasonable time for a variety of classification tasks [7], thus it was naturally selected for the approaches employed in this work. Genetic programming is an extension to the inspiration of genetic algorithms (GA) [8], where the main problem of GA concerning the fixed problem definition, is avoided by using variable-length trees instead of fixed-sized individuals. Moreover, the GP theory enabled the use of functional tree-nodes that offered powerful intelligent tools like the symbolic regression problem solving [7]. An extension to the concept of standard GP is the strongly typed or type-constrained GP [9,10]. By using the latter approach, it is possible to construct two-valued logic (modus ponens) expressions, which preserve a satisfactory rate of success when used as classification rules in a number of application domains. In order to express these rules into a type-constrained GP, a grammar is usually adapted.

The first model described in this paper is a crisp rule-based discrimination system, which is successfully applied into two medical domains. Namely, we implemented a simple grammar that produces

''Boolean'' expressions for the separation between two classes. The grammar is used to restrict the structures of GP individuals. The separation functionality remains into the GP programs by assigning proper code segments to GP-program nodes. Complete classification for a medical case, comes upon heuristic combination of the extracted rules. The second model described in this paper attempts to incorporate fuzzy logic in rule-based medical decision making. The traditional (crisp) logic, although effective in a number of application domains, is often proved inadequate to handle classifying problems in a number of problems. Thus, in these areas, the fuzzy logic models are usually preferred. The simpler fuzzy rule-based (FRB) classifier can be considered the Mamdani model using the min–max criterion [11]. This model can be considered as a set of competitive rules. Each rule has an antecedent set, a fuzzy inference system and a consequent set. Antecedent sets are used to fuzzify the inputs using membership functions, namely to translate a number into a fuzzy linguistic term. The fuzzy inference system assigns the proper value to the rule called firing strength and the consequent set is used to characterize the output with a fuzzy linguistic term. When a fuzzy classifier is used, the consequent set assigns a specific class to the output. The rule with the maximum strength is supposed to fire, namely to give the decision output. While simple to implement, the fuzzy rule base has to be provided by a training procedure when domain knowledge does not exist. Various computational intelligence-related methods have been developed for this reason, including neural networks (NN) [12], genetic algorithms [13] and hybrid or, heuristic methods [4]. Specifically, genetic algorithms [8] have been used either for the determination of the rule bases or the membership functions or, both [14]. Consequently, genetic programming was used for the training of fuzzy rule-based systems [15].

In their work [16], Alba et al. describe the use of the GP as a search methodology in the cart-centering problem, in order to produce valid fuzzy rule-based systems, which can be directed then to a fuzzy controller. The ability however of the GP to maintain functional nodes inside the program structure, enables us to incorporate the functionality of a fuzzy rule-based system directly into the GP-tree architecture. Thus, a GP-program can behave like a fuzzy rule-based classifier. This advance is applied and tested in the first of the medical fields addressed here, the aphasia domain. We implemented the grammar in such a way that it generally satisfies the grammar used in [16]. The GP tree-nodes were given functionality, aiming both at simulating a Mamdani-FRB model using the min–max criterion, and at producing a useful output for the determination of the fitness value, during the training phase. Results from both crisp and fuzzy models demonstrate the capability of the GP-training approach to apply feature extraction and generate cooperative or competitive rule bases.

The paper is organized as follows: Section 1 is an introduction to our approach and contains details on the intelligent methodologies used of this work. Section 2 describes the aphasia medical domain and presents our discussion and the results of our implementations. Section 3 describes the pap-smear test, a discussion and the results obtained by our models. Finally, conclusions and suggestions for further research are presented in Section 4.

1.2. Genetic programming as a search methodology

Genetic and evolutionary algorithms are commonly used in various domains where a direct search method (e.g. back-propagation in neural networks) cannot be applied due to the nature of the problem [5,17,18]. In Fig. 1, is shown a simple program and its representation in genetic programming coded structures. In genetic programming, a population of random trees is initially generated, representing programs. Then, the genetic operations (crossover, mutation, etc.) on these trees are performed.

There are generally, four types of operators in genetic programming: crossover, mutation, reproduction and inversion [17,19]. Each candidate solution is evaluated based on this fitness measure. According to [7], when the classification of a new case is attempted, four results may derive, namely true positive (*tp*), false positive (*fp*), true negative (*tn*) and false negative (*fn*). From the above classification definitions, two well known measures showing the expression's efficiency can be constructed, sensitivity and specificity. Furthermore, we often refer to the term ''rule-strength'' in order to express the performance of a particular rule or diagnostic path or step, for a certain class to which

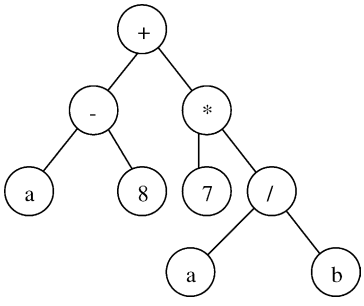


Figure 1 Tree representation of an example program: ($a - 8$) + 7(a/b).

this rule applies. In fact, rule-strength partially contributes (in a somewhat complicated manner) in the overall classification accuracy and when applied on test (i.e. new) data, it also represents the probability of correct classification of a test case according to this rule or path.

In addition to the above, Koza [7], based on Matthews results [20], suggests that the proper measure for such classification problems is the correlation:

$$C = \frac{(tp \times tn) - (fn \times fp)}{\sqrt{(tn + fn) \times (tn + fp) \times (tp + fn) \times (tp + fp)}} \quad (1)$$

Through the current experiments, the above correlation (1) was used, as the most complete measure for the fitness value. The correlation ranges in [−1,+1]. In order to have values in [0,1] the following conversion (2) was made [7]:

$$F = \frac{1}{2}(1 + C) \quad (2)$$

This fitness value ensures that subsequent generations will have better trees (programs). We selected to incorporate into fitness values one additional factor, the simplicity. This factor is proposed and explained in [1] and its value is defined in (3), as:

$$\text{simplicity} : := S = \frac{M - 0.5N - 0.5}{M - 1} \quad (3)$$

where M stands for the maximum size of trees allowed in the present application (in nodes), and N stands for the examined solution's size (in nodes). In [1], it is proposed a direct multiplication of this factor to fitness. However, in our experiments we observed that the spread of this factor from 0.5 to 1 although it helps producing small (simple) expressions, it restricts the search space. The modified simplicity introduced here, receives the following value according to (4):

$$\text{ModSimplicity} : := D = \frac{M - RN - (1 - R)}{M - 1} \quad (4)$$

where
$R ::= (100 - H)/100$ and $H : := 1.005(100 - 100/T)$.

Here, T stands for the size of the training set, M stands for the maximum size of trees as above, and N stands for the examined solution's size also as above. Consequently, the fitness value used, corresponds to the following formula (5):

$$F = \frac{1}{2}(D(1 + C)) \quad (5)$$

Our experimentation with the crisp rule-based models proved that this model is capable in restricting the solution size without, at the same time, restricting the search space.

As the algorithm allocates a large proportion of computer memory during the training phase, the current genetic methodology used, has adopted a steady-state genetic process [21,34], instead of the classic paradigm described by Koza that contains two populations. The tournament selection [22] was used, as this is the most widely used among the genetic software. We also used a kill tournament process that replaces the worst of two randomly selected individuals. Finally, we selected to apply crossover 70% of the time, mutation 20% of the time, and straight copy 10% of the time. The crossover used in our approach is a subtree-crossover.

1.3. Genetic programming for the generation of crisp rule-based systems

The approach that constructs crisp rule-based systems is based upon the theory that a rule would be more comprehensible for humans if it contained logical (boolean) expressions, rather than a mathematical formula that is extracted by standard GP routines. Thus, it could be easily interpretable and, possibly could be able to extract useful knowledge for the experts. Therefore, we selected the following operators as candidates to be part of an expression:

(a) *IfGT (arg1, arg2): if arg1 > arg2 returns true (1), else returns false (0)*
(b) *IfLT (arg1, arg2): if arg1 < arg2 returns true (1), else returns false (0)*
(c) *IfGTE (arg1, arg2): if arg1 ≥ arg2 returns true (1), else returns false (0)*
(d) *IfLTE (arg1, arg2): if arg1≤ arg2 returns true (1), else returns false (0)*
(e) *IfBT (arg1, arg2, arg3): returns true (1) if arg1 < arg2 and arg2 < arg3 (between)*
(f) *IfBTE (arg1, arg2, arg3): returns true (1) if arg1≤ arg2 and arg2≤ arg3 (between or equal)*
(g) *AND (arg1, arg2): returns true (1) if arg1 is true (1) and arg2 is true (1), else returns false (0)*
(h) *OR (arg1, arg2): returns true (1) if arg1 is true (1) or arg2 is true (1), else returns false (0)*
(i) *NOT (arg1): returns true (1) if arg1 is true (0), else returns false (0)*

Our genetic programming approach for producing crisp rule-based systems can be described within the following steps:

For each class-i, i = 1, 2, ..., k,

1. Discriminate class-i from the remaining ones i + 1, ..., i + k − 1.

2. Apply genetic programming for the discrimination of the two subsets formed through step 1.
3. Extract the discriminating rule for steps 1, 2 and check if all, k classes have been attached to a discriminating rule. If yes, then build a complete rule-base, else repeat process for the remaining classes to be discriminated.
4. If the overall classification performance of the rule-base is adequate then stop, else check other meaningful class combinations for obtaining discriminating rules, then apply genetic programming to these combinations and add the arising extra rules to the existing rule-base.

According to the nature of the data and the difficulty to separate between specific classes, modifications were made, when needed, to the above methodology. For example, the step 1, shown above, may describe the separation between groups of classes, instead of single ones. Although this procedure leads finally to a more complex rule network (see Appendix A), it enables the production of more simple rules that, in most cases encountered, were comprehensible by experts.

1.4. Genetic programming for the generation of fuzzy rule-based systems

In common implementations of GP, two types of nodes exist, which determine correspondingly two sets. Koza et al. [5,7] defines as functional nodes those, which take arguments and terminal nodes those, which do not take arguments. Although this simple characterization is sufficient to produce numerical expressions, there are cases where a stricter tree-hierarchy has to be defined. This hierarchy is used to describe the form of valid programs, thus reducing the search space of the GP [23]. Among various approaches to guide the tree architecture [24,25], the incorporation of a grammar [26] offers the advantage of producing trees containing no ''introns'' (i.e. segments of code inside a program producing no effect in the program's evaluation). Hence, such a grammar is usually adopted, which describes the programs' structure, in order to ensure the validity of new individuals. A common notation to express these grammars is the Backus Naur Form (BNF). The BNF grammar consists of terminal nodes and non-terminal nodes and is represented by the set $\{N, T, P, S\}$ where N is the set of non-terminals, T is the set of terminals, P is the set of production rules and S is a member of N corresponding to the starting symbol.

For example, consider a grammar expressing simple trees, which can produce the expression $F = (a - 8) + 7(a/b)$. It will be composed by the following sets:

$$N = \{EXPR, OP\}, \qquad T = \{-, *, /, a, b, 7, 8\},$$
$$S = <EXPR>$$

Then, P is expressed as shown in Table 1:

It should be noted that the use of the terms terminal and non-terminal in a BNF grammar, is not corresponding to what Koza defines as terminal and function. Rather, a function (a non-terminal node in terms of the GP tree architecture) is expressed as terminal in a BNF grammar as it is seen in Table 1. To avoid confusion, the use of the terms GP-function and GP-terminal (instead of the ambiguous terms function and terminal) has been proposed by [27] and is adopted throughout this paper. The construction of the production rules is the most critical point in the creation of a BNF grammar, since these rules express the permissible structures of an individual. In this paper, the BNF grammars are used for the representation of crisp and fuzzy rule-based classifiers into individuals. Additionally, the nodes of a program are active, by means of implementing a fuzzy inference rule-based system. Our intention was also to ensure that the resulting tree could be directly driven later in a fuzzy controller or fuzzy software (that it builds competitive fuzzy *if—then* classifying rules). The definition of the grammar is shown in Table 2.

In this example, the grammar describes a model of a fuzzy system with four inputs and one output. One of the inputs is considered to have different value range than the others. Thus, two groups of antecedent sets (groups A and B) are constructed, each of them covering the different value range of the according inputs, having also a different number of antecedent sets. Alternatively, we suggest that normalization could be applied to the input data during preprocessing. Based on the above grammar, a sample program is shown in Fig. 2. The contour section corresponds to the following rule:

If X1 is large and X2 is small then Y is class 1.

As stated previously, in order to implement a Mamdani fuzzy model with the min—max criterion, we select the minimum weight of antecedent sets

Table 1 BNF grammar of simple example program trees

Grammar used for a simple example tree	
<EXPR>	::= <EXPR> <OP> <EXPR> | <VAR> | <NUMBER>
<OP>	::= − | * | /
<VAR>	::= *a* | *b*
<NUMBER>	::= 7 | 8

Table 2 BNF grammar of program trees (trees are in a prefix notation), with words in bold denoting valid program nodes

Grammar used for the GP-tree	
<TREE>	::= <RL> | <RULE>
<RL>	::= **RL** <TREE> <TREE>
<RULE>	::= **RULE** <COND> <CLASS>
<COND>	::= <IF_A> | <IF_B> | <AND>
<IF_A>	::= **IF_A** <INP_A> <FS_A>
<IF_B>	::= **IF_B** <INP_B> <FS_B>
<AND>	::= **AND** <COND> <COND>
<CLASS>	::= **THEN** <OUT> <CLASS_VALUE>
<FS_A>	::= **S_A** | **M_A** | **L_A**
<FS_B>	::= **VS_B** | **S_B** | **M_B** | **L_B** | **VL_B**
<INP_A>	::= **X1**
<INP_B>	::= **X2** | **X3** | **X4**
<CLASS_VALUE>	::= **CLASS1** | **CLASS2** | **CLASS3** ...
<OUT>	::= **Y**

for each rule and the rule with the maximum weight is the one that fires.

The GP-functions used to describe the fuzzy methodology correspond to the words with bold in Table 2. In Table 3 we present their functionality.

It is worth to note that the fuzzification happens in *IF_A* and *IF_B* nodes. For example, if the implementation uses Gaussian membership, then for a given Gaussian range *a* (standard for each of the *IF_A* and *IF_B* nodes), a center *c* = *arg2* and a value *x* = *arg1*, the node output will be given by the following formula (6):

$$n = e^{-1/2((x-c)/a)^2} \tag{6}$$

The *THEN* node returns 1 if for the examining example the output (*arg1*) belongs to the class

Table 3 Functions adapted in GP-implementation for simulation of a Mamdani-classifier behavior

Function	Operation
RL (arg1, arg2)	If absolute (arg1) > absolute (arg2) then return arg1; else return arg2
RULE (arg1, arg2)	Return arg1*arg2
IF_A (arg1, arg2)	Fuzzify (arg1), based on the (arg2) value, return weight
IF_B (arg1, arg2)	Fuzzify (arg1), based on the (arg2) value, return weight
AND (arg1, arg2)	Return minimum (arg1, arg2)
THEN (arg1, arg2)	If arg1 = arg2 then return 1; else return −1
L_A, M_A, L_B, etc.	Return a constant value (e.g 0 for L_A, 10 for M_A, 500 for L_B etc.)
CLASS1, CLASS2, etc.	Return a constant value (e.g. 1 for CLASS1, 2 for CLASS2, etc.)
X1, X2, etc.	System inputs (assuming a numerical value)
Y	System output (assuming a numerical value)

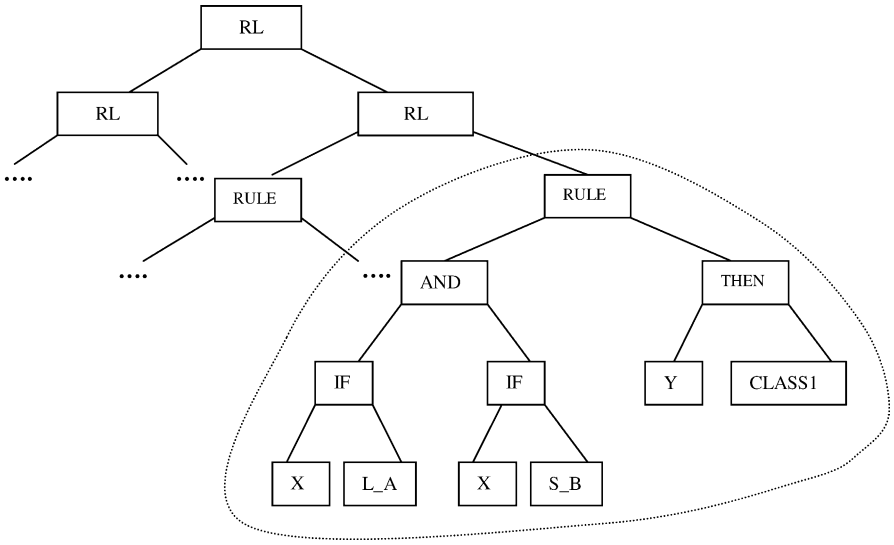


Figure 2 Genetic programming tree architecture of fuzzy rule-based classifying systems.

described by *arg2*, else it returns −1. The reason to use this methodology, together with the RL working, is to be able to know (when the tree evaluation is complete) whether the rule, which fired, was true or false. If the fired rule describes a false consequent set, the program value will be negative. While a program describes a full rule base, during the training phase will produce either positive or negative values based on the training set records. The fitness value is comprised here by the following Eq. (7):

$$F = \sum_{t=0}^{n-1} (1 : f_t > 0 | 0 : f_t \leq 0) \tag{7}$$

where, F is the program fitness, t is a record in the training set, n is the number of training records, and f_t is the program output for the record t. Incorporation of the simplicity factor was not encountered here, while large solutions, due to the grammar definition, can still be easily interpreted.

Due to the considerably large parameter space, mutation plays an important role in our model, while in [16], only the crossover operator is necessary. Shrink mutation was proved valuable in reducing the code bloat caused by crossover operations, which often drove the solution in self-repeating rule bases. The initialization of the population was random with respect to the grammar rules. The applied restrictions were the following:

(a) During initialization, the root node must be a <TREE>[1] type node.
(b) Crossover is allowed only between equal type nodes.
(c) Node mutation is allowed only in terminal nodes and the new value must be of the same type.
(d) Shrink mutation is allowed only between <TREE> type nodes, or between <COND> type nodes.

As seen from the above, we applied random initialization based on the grammar production rules. Obviously, the optimal classification score is dependent on the proper selection of membership functions. To solve this problem, we may follow a variety of approaches. A histogram analysis of the values for each input may reveal areas where these values are concentrated. Alternatively, domain knowledge may help in the determination of membership functions. Another approach is to leave the genetic programming process to compute the membership functions. However, with the latter implementation, the search space for the GP increases dramatically, and we consider that it should be avoided if prior domain knowledge exists. An example of this approach could be a three-parameter function *IF_Gauss* (arg1, arg2, arg3) instead of the *IF_A* and *IF_B* functions, where *arg1* is the input value, *arg2* is the center of Gaussian and *arg3* is the width of the Gaussian. The arguments *arg2* and *arg3* may be derived by number nodes or as results of numerical operations implemented in sub-trees.

[1] See Table 2 for <TREE> and <COND> expressions.

1.5. Comparison with other intelligent approaches

A number of other techniques can be applied to either the aphasia or the pap-smear data, in order to acquire a model for efficient decision-making. However, due to the nature of the selected application domains (i.e. nominal values of the attributes used, objective nature of the data values), we selected to present comparative results from:

(a) Standard algorithms that perform top—down induction of decision trees using information entropy criteria.
(b) Common genetic programming for symbolic regression task.

The standard computational intelligence methodology used for producing automated domain-dependent expert knowledge by mining the medical data, belongs to the area of inductive machine learning [28,29]. Inductive learning tools and techniques have been widely applied during the last two decades in various domains [30] for their comprehensibility, as well as for their ability to generalize from processing with large databases and high complexity domains of application. Usually, but not exclusively, the inductive learning approaches construct decision trees [3], by applying an intelligent approach for reducing either the complexity of the search space, or the size of the tree produced, known as the "divide-and-conquer" approach. Regarding the handling of complexity by inductive decision trees, note for example that, according to combinatorial theory, there are more than 10^{13} ways to partition a set containing 20 items, while an inductive tree forms a competitive classifier for these items in only a few nodes and in a very short time, by observing a number of pre-classified examples for these items.

Quinlan's approach [3,28] is the most widely used in machine learning for its comprehensibility and simplicity in data processing. The present work applies the well-known Quinlan's approach called C4.5, [28], which generally works as follows:

Given (a) a set of observational statements (i.e. attribute value vectors) each of which is assigned to a certain class, and (b) a universe of classes, *find* a set of discriminative descriptions between classes.

The algorithm leads to the generation of a decision tree, in which leaves are class names and nodes represent attribute-based tests with branches for each possible outcome. Since all available cases belong to different classes, the algorithm attempts to split them into subsets, by the *divide and conquer* principle, see [28]. The quantitative criterion for splitting the set of the initial statements to subsets in order to form the tree, is based on information entropy measurements.

The symbolic regression task, performed by a common genetic programming procedure, produces mathematical expressions that can be used for discrimination between classes based on their value. For example, when classifying between two classes (or two groups of classes) zero or positive values may imply the first class and correspondingly, negative values may anticipate the second class.

2. The aphasia problem

2.1. Description of aphasia

Speech is the major instrument of communication in human beings. Loss or disturbance of speech is a severe handicap in daily living. Aphasia is a disturbance of comprehension and formulation of language. Because the human brain consists of vast neuronal networks located in many functional cerebral regions, the aphasic symptoms, which are produced by damage in these networks, can differ. Thus different major aphasia profiles can be distinguished:

Broca aphasia (also called motor or expressive aphasia): In Broca aphasia the disturbances of the expressive language functions are more prominent than disturbances of the receptive language functions. The patients speak non-fluently with labored, slow, and impaired articulation. One major symptom is agrammatism (or telegram style), which is a reduction of the sentences to only a few words. Nevertheless, the utterances of Broca aphasics make sense and comprehension of language may be affected less.

Wernicke's aphasia (also called sensory or receptive aphasia): The speech of Wernicke's aphasics is fluent and the articulation is good. In contrast the sentences do not have much sense because the patient produces both literal paraphasias (where sounds within the words are changed or left out) and verbal paraphasias (where wrong words are used). Some patients produce absolutely meaningless sentences (jargon) or words (neologisms). Comprehension and repetition is severely impaired.

Global aphasia (also called total aphasia): Global aphasia is a very severe language disturbance, where all language modalities are affected. Often no communication is possible at all.

Anomic aphasia: The spontaneous speech of anomic patients is fluent and grammatically correct, but these patients have difficulties in the retrieval of words. The word finding difficulties may generate pauses and circumlocutions. Comprehension and repetition are relatively normal.

Conduction aphasia: In conduction aphasia the repetition of spoken words is severely disturbed, whereas the comprehension is apparently good. Literal paraphasias are common (where sounds within the words are changed or left out).

2.2. Neurolinguistic test batteries

Clinical diagnosis of the type of aphasia is made from an expert through a free interview. This can lead to different evaluations and characterizations of the aphasia syndrome dependent on the individual exploration. To compare such evaluations inter-individually it is necessary to standardize such examinations. Standardized examinations are also useful for comparison of different test profiles of one patient to define, e.g. the benefit of a therapy. Major comprehensive language tests in English speaking countries are the Western aphasia battery (WAB) and the Boston diagnostic aphasia examination (BDAE). In German speaking countries the Aachen aphasia test (AAT) is the commonly used test battery. Because the AAT was used for evaluation of language function in this database this test will be described in detail.

2.3. The Aachen aphasia test

The first part is an evaluation of *spontaneous speech* [31]. Six sub-tests are used to characterize six different levels of spontaneous speech. Because the different aphasia types have different failures regarding these levels, spontaneous speech can be used for a fast diagnosis of the type of aphasia. There are six subtypes (P0—P5, range: 0—5 points). The "*token test*" (T0, range: 0—100) is a general test of comprehension of language. The patient has to choose the right token out of a set of tokens different in shape, color, or size. The "token test" has five subtests of increasing levels of difficulty (T1—T5, range: 0—10 points). The third test is a *test of repetition* (N0, range: 0—100). The patient has to repeat different sounds, words or sentences. It consists of five subtests (N1—N5, range: 0—30

points). The *test written language* (C0, range: 0—100) is an evaluation of reading and writing functions. It consists of three subtests (C1—C3, range: 0—30 points). The *confrontation naming test* (B0, range: 0—100) is an evaluation of the capability of the patient to describe things or situations or actions with the right words. It consists of four subtests (B1—B4, range: 0—30 points). The *comprehension test* (V0, range: 0—100) evaluates the possibility of the patient to understand words or sentences accurately. Its two subtests evaluate the processing of heard and read words or sentences (V1—V2, range: 0—60 points).

2.4. The aphasia problem methodology

The methodology suggested for the aphasia domain mainly applies genetic programming to the aphasia data used in [32]. Two models were constructed in order to classify aphasia cases. The first model is a crisp rule-based system featuring two separate submodels: the first one (denoted as CRBS-GP1, explained later) classifies between four aphasia subtypes and the second one (denoted as CRBS-GP2) tries to discriminate among the full data set. The first model uses a dataset similar to the one presented in the work of [33], in order to provide a comparative result between a neural network-based approach and the current one. This heuristic methodology is expanded to the second model (CRBS-GP2) among all aphasia subtypes. Then, a GP model for the production of fuzzy rule-based systems (denoted as FRBS-GP, also explained later) is implemented. The latter, is tested in the four-class problem, in order to obtain results comparable to previous works.

Two more standard methods are initially applied to the aphasia data. First, machine learning results, give a comparison measure. Then, a standard application of genetic programming symbolic regression process is implemented. These results are discussed in the last subsection of the aphasia section, in comparison to those acquired by the proposed approaches.

2.5. Results and discussion for the aphasia problem

2.5.1. Machine learning

Initially, several runs were performed with the use of C4.5 for various settings on a data set consisting of 262 cases, classified to all known kinds of aphasia mentioned in Section 2.1, above. A training set accuracy ranging from 65 to 89% was obtained with most misclassifications occurring in classes Broca, Wernicke and Residual (the algorithm was not able to generalize adequately over these classes in most experimentation settings). A 10-fold cross-validation was also attempted, showing a similar performance to the abovementioned, while the overall accuracy was ranging between 65 and 72%. Boosting, an algorithmic technique that normally generates better solutions with a less comprehensible outcome [34,35,36], did not really seem to improve the algorithm's performance. According to the medical experts some of the results were found too simple, maybe poor. It seems that more information is needed in order to become useful for medical doctors. The acquired rule set sometimes sounds correct to the experts but they would definitely expect more conditions to be examined simultaneously in most cases. When rules become more complex (i.e. 4—5 premise parts) on the other side, it is very difficult to give a definite opinion whether they seem correct or not.

One rule appears to the expert to be complete and good for Broca. This one is:

Rule 1 (*cover 39*): If $\{P3 > 2\}$ and $\{P5 \leq 2\}$ and $\{N0 \leq 76\}$ and $\{N3 > 12\}$ and $\{V0 > 27\}$ and $\{V1 > 38\} \rightarrow$ (then) class B (Broca) [0.902]

Then, another rule seems adequate for the diagnosis of Wernicke:

Rule 4 (*cover 29*): If $\{P1 > 3\}$ and $\{P2 \leq 4\}$ and $\{P5 > 2\}$ and $\{N5 \leq 22\}$ and $\{V0 \leq 62\}$ and $\{V1 > 38\} \rightarrow$ (then) class W (Wernicke) [0.935]

The experts have characterized three more produced rules as complete and correct. They all refer to Conduction aphasia, they cover 6, 8 and 14 cases respectively, and their probability of correct classification of a new case ranges between 81.3 and 90%. All rules referring to Global aphasia score badly. On the contrary, there are more rules referring to Broca that seem interesting, complete and correct to the expert. The aphasia subclass known as Residual aphasia seems to be related only to ''light symptoms''.

2.5.2. Standard genetic programming

As a second step, we applied a standard approach of genetic programming for forming decision trees of specific max length, by combining different operators/functions, in a way that the outcome is such, that represents accurately all the training data of aphasia used. In the standard GP approach, the data set was decided to consist of a selection of 146 cases, which described four (4) major types of aphasia. These types were Anomic aphasia (also denoted as class 1), Broca aphasia (class 2), Global aphasia (class 3), Wernicke aphasia (class 4).

We selected for the training set 74 cases and for the test set 72 cases (approximately 50% of the data was selected as test set, for a fair comparison). The

intention was to produce three rules in a form of a mathematical expression that, according to a heuristic classification scheme, would perform the following tasks:

(a) Rule 1 distinguishes class 1 from classes 2–4
(b) Rule 2 distinguishes class 2 from classes 3 and 4
(c) Rule 3 distinguishes class 3 from 4

By using this methodology, an expert could theoretically take a decision on a patient case, by applying the rules starting from Rule #1, until a positive result is found. If no positive result is found, the case is assumed to belong to class #4 (Wernike aphasia). This approach is expected to offer an advantage in the training process of such a classification system, especially for the extraction of Rules #2 and #3. Rules #2 and #3 will use a data set that will not contain cases from class 1 and cases from classes 1 and 2, respectively. The idea is that first the training process is accelerated with the use of a smaller training set and then, with this approach is very likely to discover a simpler rule.

The classification was applied using the result of the mathematical expression. If the result was zero (0) or positive, then a *true* value was considered, while negative result denoted *false* outcome. In the training set we obtained an overall classification accuracy of 100%. In detail, the corresponding classification accuracy of each produced rule for the test set is shown in the Table 4.

The types of operators used in the above genetic process were addition, subtraction, multiplication and protected division (denoted also as pdiv). Protected division returns the value of one (1) when the denominator equals zero (0), in order to achieve closure in genetic programs. We also included the hyperbolic function (tanh), a rather popular transfer function in neural network approaches. The resulted formulas are shown in Table 5.

As it is observable, these results are not easily interpretable in terms of medical decision making in practice, still they appear to be more accurate among the other genetic programming approaches for aphasia in this paper. Moreover, they may potentially help the medical doctor in revealing the features that were promoted for the construction of these mathematical expressions. For example, Rule #3 makes use of only three features promoting only variables C3, P4 and P5 among the full feature set.

Table 4 Classification accuracy for the standard GP model on aphasia data

Rule #	Correct classification in the test set (%)
1	96.49
2	80
3	95.83
Overall (worst case)	90.77

2.5.3. CRBS-GP: Genetic programming for the production of crisp rule-based systems

2.5.3.1. First GP-model: CRBS-GP1. We performed nine (9) runs for each rule, using a population of 10,000 programs (i.e. trees). The algorithm execution was terminated when the fitness value was becoming larger than (0.999). Every 1/10 of a generation is recorded the population's best solution, and a check in the test set is performed. According to [7], throughout the execution of the algorithm, the expression that obtains good performance in the training and also in the test set is selected. Not always we obtain this expression at the end of the execution. In most cases, aphasia experiments have shown that the algorithmic execution ends with a rather complicated expression, which performs excellent in the training set, but has poor performance in the test set (these are typical cases of over-fitting). The following simple rules are proposed to distinguish among the four types of aphasia:

(a) Rule #1: *If N*5 (*repetition of sentences*) *is greater than or equal to* 23 *then there is Anomic aphasia*, which classifies correctly the 97.33% of training cases and the 95.15% of test cases representing Anomic aphasia.
(b) Rule #2a: *If P*5 (*syntactic structure*) *is greater than or equal to P*3 (*semantic structure*) *then*

Table 5 Standard GP symbolic regression formulas for the aphasia domain

Rule #	Mathematical expression
1	((pdiv ((tanh ((tanh (tanh (V0))) * ((tanh ((tanh ((pdiv (N1, P5)) − T5)) * N1)) * ((pdiv (N1, T2)) − T5)))), P5)) * (31 − N5))
2	(((pdiv ((pdiv ((−58 * P5), (pdiv (((−79 + (pdiv (9, V0))) + (tanh (tanh (C2)))), ((−76 + (pdiv (9, 30)))+ (tanh (tanh (C2)))))))), ((P1 * T4) − (T4 * P5))))+ (T4 + P5)) * (pdiv ((pdiv ((P1 * T4), V0)), C3)))
3	((tanh (C3 * P4)) − P5)

there is Broca aphasia, which classifies correctly the 84.13% of training cases representing Broca aphasia and the 86.2% of test cases.

(c) Rule #3: *If P5 (syntactic structure)is less than or equal to 1 then there is Global aphasia*, which classifies correctly the 100% of training cases and the 97.30% of test cases, representing Global aphasia.

However, it should be noted that more accurate (and complex) formulas have been obtained for Rule #2a (which distinguishes Broca from Global and Wernicke). The most accurate of these, classifies correctly the 100% of the abovementioned Broca training cases and the 91.38% of corresponding test cases. Its type is given below:

- Rule #2b: *(IfLTE (IfGT (IfBTE (IfLT T0 V0)(IfGTE (IfBT P5 B4 B3)(IfGTE B3 T2)) B3)(IfBTE P3 P5 P1))(IfBTE (IfGT T3 P3)(NOT (IfBT B3 V081))(NOT C3))*

When the above formula returns zero (0), presence of Broca aphasia is assumed. Another type with comparative performance, but less complex is the following:

- Rule #2c: *(OR (IfBTE T4 C3 T4) (OR (IfBTE P3 P5 P1) (OR (IfBTE 3 P5 P1) (IfLTE C3 P2))))*

Rule 2c classifies correctly 100% of the training cases and 86.20% of the test cases. When zero is returned, presence of Broca aphasia is assumed. As it may be observed, the above expressions can be further simplified while parts of these formulas produce no explicit result (usually referred as introns).

2.5.3.2. Second GP-model: CRBS-GP2. The second model intends to extend the previous model, aiming at obtaining a global classifier between all types of aphasia. The available data were separated again in two halves. The first half was used to train the system and the second half to test it. After each termination of the training procedure, the expression obtaining best results in both, the training and the test data, was selected. With this selection, the intention was to choose those expressions that could generalize adequately, thus avoiding solutions overfitted to the training set [7]. Table 6 shows the data used for each of the abovementioned tasks. Despite the fact that Transcortical aphasia cases were very limited, we attempted to introduce rules for this class, too. Finally in this model, Residual aphasia, although not consisting a certain subtype of aphasia, was also considered for classification and rule production, since it consists a different type of diagnosis within the entire aphasia database.

Table 6 Training set and test set records of each class for CRBS-GP2 on aphasia data

Class #	Training set records	Test set records
Anomic	12	12
Broca	21	21
Wernicke	24	23
Conduction	10	9
Residual	12	12
Transcortical	4	2
Global	17	16

After data separation into training and test groups, the rules presented in Table 7 were extracted. This selection was made in order to be able to apply a global classifier (by subsequent application of rules) as it is presented in Fig. 6 (see Appendix A). Both training and test data were randomly created in such a way that they would fairly contain an adequate number of cases of all classes.

After the training process, the rules achieving the best performance both in training and in test data (indicating generalizing abilities or, success in test data) are presented in Table 8. Sensitivity and specificity of value one (1) denote perfect performance, whereas value of zero (0) denotes poor performance (see Table 8). Simplicity values are normalized in [0,1], with value of zero (0) denoting a 150-node expression and value of one (1) denoting a single node expression.

Table 7 Rules and separation intention for CRBS-GP2 on aphasia data, with the classes shown for each rule denoting also the corresponding training and test set used for that rule

Rule #	Separation class	(Class) to have separation from
1	Global	All others
2	Anomic	Broca and Wernicke
3	Anomic	Transcortical
4	Anomic	Conduction
5	Anomic	Residual
6	Broca	Wernicke
7	Broca	Transcortical
8	Broca	Residual
9	Broca	Conduction
10	Wernicke	Transcortical
11	Transcortical	Conduction and Residual
12	Wernicke	Residual
13	Wernicke	Conduction
14	Conduction	Residual

Table 8 Aphasia results for CRBS-GP2, with fitness of 1 denoting a perfect fit on the selected set

Rule #	Training data				Test data				Simplicity
	Fitness	Sensitivity	Specificity	Missed cases	Fitness	Sensitivity	Specificity	Missed cases	
1	0.982175	1	0.988095	1	0.924842	0.875	0.974684	4	1
2	1	1	1	0	0.912479	0.9	0.955556	3	0.375839
3	1	1	1	0	0.787879	0.909091	0.666667	2	0.986577
4	1	1	1	0	1	1	1	0	0.95302
5	1	1	1	0	0.91986	0.846154	1	2	0.912752
6	1	1	1	0	1	1	1	0	0.939597
7	1	1	1	0	0.465497	0.95	0	3	0.959732
8	1	1	1	0	0.968807	1	0.923077	1	0.959732
9	0.963134	1	0.909091	1	0.920635	0.952381	0,888889	2	0.926175
10	1	1	1	0	0.845782	1	0.5	1	0.959732
11	1	1	1	0	0.898862	1	0.954545	1	0.986577
12	1	1	1	0	0.936594	0.956522	0.916667	2	0.926175
13	0.960985	0.96	1	1	0.838887	0.88	0.857143	4	0.852349
14	1	1	1	0	1	1	1	0	0.966443

Sensitivity and specificity of each rule, are important to be known, while in a subsequent application of these rules, the overall accuracy of the final diagnosis is derived by these values depending on the partial outcome of each rule (e.g. true/false). For example, if for the classification of a case, three (3) rules are to be applied subsequently, and their partial outcomes are ''no'', ''no'' and ''yes'' correspondingly, then the probability of correct classification for a new (test) case according to the above decision path, derives by multiplying the specificity values of the first two rules, times the sensitivity value of the third rule. These derived rules are presented in Table 9. The format of their presentation follows the S-expression type (Lisp-like or prefix notation), a common practice in presentations of GP-results.

These rules might be applied individually, depending on the diagnosis requirements. However, they are sufficient in order to provide a full-classification scheme. Therefore, the flowchart in Fig. 6 (see Appendix A) is proposed. The percentages in parentheses after each rule represent the correct classification in both training and test set.

The obtained results are considered adequately accurate and thus, they could be possibly useful, for the construction of an automated advisory tool performing aphasia classification of new cases.

Table 9 Extracted aphasia rules for the CRBS-GP2 model

Rule #	Formula
1	(P5)
2	(AND 6 (IfBTE (OR N5 T1) (IfLTE (NOT (AND C0 (OR (IfLT (NOT (IfGT (OR (OR (IfLTE (NOT (IfGT (IfGT (OR B4 C2) (IfGTE (IfBTE N3 N549) (IfGTE B1 B2))) (IfBT N5 N1 B3))) (NOT V2)) (IfBT B3 (IfGTE 111 N4) T1)) T1) (IfBTE P1 N1 N5))) (OR (NOT (IfBT (OR B4 C2) B1 N5)) (IfLT (NOT (IfBT T5 B1 N5)) (IfLTE (IfLTE (IfBT T5 (IfBT (NOT (OR C2 T1)) (IfGTE 113 N5) T1) N5) V2) N5)))) (OR (IfLT P5 P3) T1)))) (IfBTE (IfLTE T1 B3) B38)) (NOT (IfBT T5 B1 N5))))
3	(IfGT (T5 C3))
4	(IfGT (IfBT N5 N3 B1) (IfLTE C1 N4))
5	(OR (IfLT (IfLT (IfLT P3 T3) (IfLT B3 C3)) (IfLTE 93 B0)) (NOT T5))
6	(AND (IfLTE P5 P1) (IfGTE (NOT C3) (IfGT 3 P5)))
7	(IfGTE (IfGT N4 V2) (IfGT T0 C3))
8	(OR (IfLTE 26 B4) (IfLTE 25 N5))
9	(IfLT (IfLT (IfBT C2 N3 C1) (IfLT P5 P3)) (IfLT P0 P1))
10	(IfGT (IfGT N4 B2) (IfLTE N5 C1))
11	(IfLTE (T3 C3))
12	(IfBTE (IfLTE (IfLTE P1 P2) (IfGT T4 P2)) (IfGTE P4 T5) N0)
13	(AND (IfLT N4 C2) (IfBTE (OR T1 (IfGTE N3 B1)) (NOT (NOT (IfGT (OR -32 C3) (IfLTE N3 B4)))) (NOT (IfLT B3 N3))))
14	(IfGTE (NOT P4) (IfGT 75 N0))

Table 10 GP parameters for the aphasia problem using the FRBS-GP approach

Genetic programming parameters	
Available data records (concerning four major types of aphasia)	145
Data used for training	78 (54% of the available data, containing fairly distributed cases from all classes)
Data used for testing	67 (46% of the available data, containing fairly distributed cases from all classes)
Population	2000 individuals
GP implementation	Steady-state grammar-driven GP
Selection	Tournament with elitist strategy
Tournament size	6
Crossover rate	0.6
Overall mutation rate	0.4
Node mutation rate (proportional to overall mutation rate)	0.4
Shrink mutation rate (proportional to overall mutation rate)	0.6
Killing anti-tournament size	2
Maximum allowed formula size	200 tree-nodes per individual

Comprehensibility is also rated adequate for each independent rule, in most cases. Generalization ability is also rated high, as the final outcome is easily represented in a flow diagram and seems to be able to classify new aphasia cases by only using a rather limited number of attributes and conditions of relatively abstractive form.

2.5.4. FRBS-GP: Genetic programming for the production of fuzzy rule-based systems

Since different physicians perform the aphasia's score ratings, fuzziness in these ratings is to be expected [37]. This diagnosis of aphasia's subtypes has gained recently the focus of various computational intelligence implementations [33,38]. Thus, our intention was to use this data in order to construct a Mamdani-fuzzy classifier. We decided to use Gaussian membership functions, due to resemblance to the normal distribution, which may portray the scoring's deviations from characteristic values. For inputs P0—P5 three (3) Gaussian membership functions were used: small, medium and large, having centers in 0, 2.5 and 5 respectively, and a width of 2.5. For the rest of the parameters, eleven (11) Gaussian membership functions were available by the grammar, each of them having width of 10 and center ranging from 0 to 100 (by a step of 10 values for incrementing the model's efficiency). By using two different sets of membership functions in the system design, it is not necessary to normalize the data. However, normalization should be applied for practical reasons related to the size of the genetic programming function set (e.g. if every parameter had a different value range). The system parameters are presented in Table 10. In order to compare the model's efficiency with previous works in this domain, we selected to classify four of the major aphasia's subtypes: Broca, Wernicke, Global and Anomic aphasia. The training set size and the test set size were formed in a way similar to these previous works.

Fig. 3 shows the best individual's fitness during training. The classification results are presented in Table 11. They are presented together with previous research using neural networks [33].

The solution obtained from the FRBS-GP approach was accomplished after 1786 generations. It consists of a set of eleven (11) fuzzy competitive rules in the if—then form, which classifies correctly 88.5% (69 out of 78 cases) of the training set. The *Rx* fuzzy set's name can be translated as *"is about X"*. For example, the *R40* fuzzy set has a meaning of *"is about 40"*, the *R0* fuzzy set has a meaning of *"is about 0"* etc. The above rules belong to a fuzzy rule base. For every new patient case examined, the first step corresponds to the fuzzification of the input data. Then each of the rules below is assigned a weight and the rule with the maximum weight

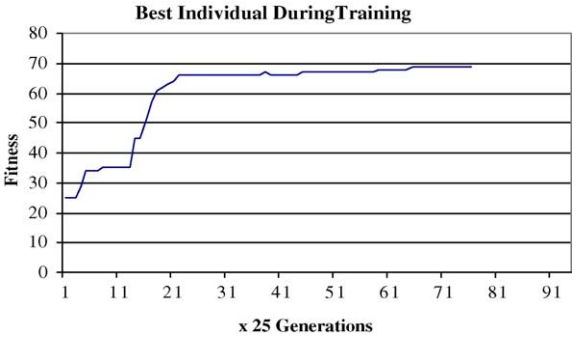


Figure 3 Fitness of the best individual during training for the FRBS-GP approach.

Table 11 Comparison between various NN and GP models in the aphasia domain

Methodology	Classification accuracy in test set (%)
NN (spontaneous speech model)	86.0
NN (comprehensive model)	92.4
Machine learning	68.5
Standard-GP	90.8
CRBS-GP1a	79.8
CRBS-GP1b	84.6
FRBS-GP	79.1

''fires'', offering the output of the system. Thus, these rules are part of a fuzzy mechanism and they should not be used without a fuzzy classifier. They can give though, a rough explanation of the system's decision methodology.

(a) If B2 is R40 and B1 is R40 and V0 is R60 and C3 is R0 then Broca aphasia
(b) If B4 is R0 and D B1 is R40 and C1 is R0 then Broca aphasia
(c) If B4 is R0 and C1 is R0 and C1 is R0 and B2 is R50 then Broca aphasia
(d) If B1 is R40 and C1 is R0 then Wernicke aphasia
(e) If B4 is R0 and B2 is R40 and N0 is R100 then Wernicke aphasia
(f) If V0 is R60 and V0 is R60 and B2 is R50 then Wernicke aphasia
(g) If N2 is R40 and B2 is R50 and V0 is R60 and B2 is R40 and P1 is medium and C1 is R0 and V0 is R40 and C1 is R0 then Wernicke aphasia
(h) If B2 is R40 and V0 is R40 then Wernicke aphasia
(i) If C0 is R0 and P1 is Medium and N2 is R40 then Global aphasia
(j) If N0 is R100 then Anomic aphasia
(k) If B2 is R40 and B1 is R40 and C1 is R0 then Anomic aphasia

The above fuzzy rule base classified correctly the 79.1% of the test data. As it can be seen, these rules can be further simplified, while identical antecedent sets exist in different rules which classify the same class.

2.6. Comparison between models for the aphasia problem

As it is shown in Table 10, the results are competitive to other GP-implementations. The accuracy of these rules in the training data approaches 100% and it ranges between 79.8 and 84.6% in test data. Since this paper's approach had 88.5% accuracy in the training data and 79.1% in the test data, it seems that, in this application domain, rather the fuzzy rules are those providing more robust generalization.

The GP results remain however less accurate than the best neural net model. Despite that fact, the suggested approach offers an additional advantage over those of NN's (where prior domain knowledge is generally needed) and it was used in that case for the selection of inputs. The Standard-GP model is referred to the system of Section 2.5.2. The model CRBS-GP1a, is composed by Rules #1, #2a, #3 and the CRBS-GP1b consists of the Rules #1, #2b, #3 (see Section 2.5.3.1 and Section 2.5.3.2, respectively).

3. The pap-smear test

3.1. Description of the pap-smear test problem

Using a small brush, a cotton stick or wooden stick, a specimen is taken from the uterine cervix and transferred onto a thin, rectangular glass plate (slide). The specimen (smear) is stained using the Papanicolaou method. This makes it possible to see characteristics of cells more clearly in a microscope. The purpose of the smear screening, is to diagnose pre-malignant cell changes before they progress to cancer. Smears contain mainly two types of cells: squamous epithelial cells and columnar epithelial cells (Fig. 4). The columnar epithelium is found in the upper part of cervix, and the squamous epithelium in the lower part (Fig. 5). The screening of smears is done by a cyto-technologist and/or cytopathologist. It is time consuming, as each slide may contain up to 300,000 cells.

The columnar epithelium consists of a single layer of cells, resting on the basal membrane.

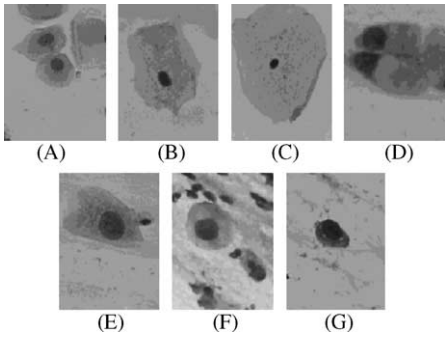


Figure 4 Some of the cells found in cervix: (A) parabasal denoted PARA, (B) intermediate denoted INTER, (C) superficial squamous epithelia denoted SUPER, (D) columnar epithelium denoted CYL, (E—F) mild, moderate and severe non-keratinizing dysplasia (source: [8]).

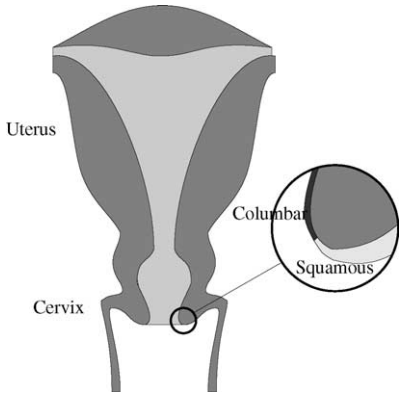


Figure 5 Schematic drawing of the uterus and the cervix. The drawing also shows the transformation zone where the exocervical squamous epithelium meets the endocervical columnar epithelium (source: [8]).

Underneath the columnar epithelium are the reserve cells, which can multiply to produce squamous metaplasia. The nucleus is located at the bottom of the cytoplasm. When viewed from the top, the area of the nucleus will seem large when compared to the area of its cytoplasm. Viewed from the side, the cytoplasm will seem larger (Fig. 4(d)). The area of the nucleus is $\sim$50 μm^2 and it is darker than the surrounding cytoplasm. The squamous epithelium is divided into four layers; the basal, parabasal, intermediate and superficial layer.

The cells of the basal layer lie on the basal membrane, and they produce the cells of the overlying layers. The most mature cells are found in the superficial layer. Cells of the basal and parabasal layers are round, with nuclei of $\sim$50 μm^2 and cytoplasm of 200–300 μm^2 (Fig. 4(a)). Cells of the intermediate and superficial layers have small nuclei of 20–35 μm^2 and large cytoplasm of 800–1600 μm^2 (Fig. 4(b) and (c)).

Dysplasic cells are cells that have undergone precancerous changes. They generally have larger and darker nuclei and have a tendency to cling together in large clusters. Squamous dysplasia is divided into three classes: mild, moderate, and severe (Fig. 4(e–g)). Mild dysplasic cells have enlarged and light nuclei. For moderate dysplasic cells, the nuclei are larger and darker. The nuclei may have begun to deteriorate, which is seen as a granulation of the nuclei. In the last stage of pre-cancerous changes, severe dysplasia, the nuclei are large, dark and often deformed. The cytoplasm of severe dysplasia is dark and small when compared to the nuclei. More details for the pap-smear problem, as well as other attempts to develop efficient intelligent approaches on the pap-smear problem can be found in [39–42].

3.2. The pap-smear problem methodology

The entire data set consists of 450 pap-smear cases belonging to seven (7) different classes. These classes are defined and symbolized as follows:

Class 1: columnar epithelium (CYL), *class* 2: parabasal (PARA), *class* 3: intermediate (INTER), *class* 4: superficial squamus epithelia (SUPER), *class* 5: mild non-keratinizing dysplasia (DYS), *class* 6: Moderate DYS, *class* 7: severe DYS.

For the machine learning approach we used the above division of the data in seven (7) classes, while for most of the genetic programming approaches we decided to unify classes 5–7 into one single class, as it seems that there is some confusion in discriminating among the three (3) dysplasia classes (the characterization as mild, moderate and severe, is related to the number per sample of the dysplasic cells found, and some times is confusing).

Specifically, according to the *PST-GP* model presented in Section 1.3, for the pap-smear database, k (i.e. number of classes) has the value of five (5). Thus four simple rules in total are needed initially for discriminating among the five classes. Then four (4) extra rules are discovered and added to the final rule-base, until an adequate classification performance is obtained for the pap-smear data. In the following sections, it is shown that the rules are characterised by a high degree of generalisation, they are compact, clear and effective for classifying new (test) cases. It is worth to note also that, as always in evolutionary programming approaches, several competitive configurations may exist, that could possibly classify the examined data set in a superior way, but the overall genetic-based approach is a time-consuming process, demanding high computing power and repeated experimentation. In that sense, the current results although powerful and meaningful should rather be considered indicative. As in the aphasia domain, the crisp rule-based system is compared with machine learning and standard GP results, which are primarily presented and analysed.

3.3. Results and discussion for the pap-smear test problem

3.3.1. Machine learning

We conducted several experiments with the use of C4.5 for various settings (data set: 450 cases), which gave accuracy on the training data, ranging from 94.6 to 99.8%. The classifier was obtained rapidly and its performance could be characterized very high. However, the accuracy on the test data (10-fold cross-validation experiments) showed a considerable lower performance, with the accuracy

ranging this time, between 66.8 and 70%. Misclassification appeared to be somewhat uniform among all classes (perhaps due to an increased confusion among classes 5—7, as shall be stated later). Boosting did not really affect the algorithm's performance (accuracy in test set raised up to 73.0%, but with considerable loss of comprehensibility of the model). The most important rules proved to be the following (achieving a 95.2—97.9% probability of correct classification of a test case, by the application of the specific rule applied):

Rule 5 (*cover 40*): If {K/C $>$ 0.04364768} and {Kerne_Ycol $\leq$ 0.4} and {Cyto_Ycol $\leq$ 0.66} and {CytoMax $>$ 49} $\rightarrow$ (then) class 2 (PARA) [0.952]

Rule 8 (*cover 45*): If {K/C $\leq$ 0.04364768} and {KerneLong $>$ 8.23} and {CytoLong $>$ 52.39} and {KernePeri $>$ 27.56} $\rightarrow$ (then) class 3 (INTER) [0.979]

The meaning of the symbolic names are as follows: K/C is the ratio between nucleus area and cytoplasm area, Kerne_Ycol is the nucleus intensity, Cyto_Ycol is the cytoplasm intensity, CytoMax is the number of maxima in the cytoplasm, KerneLong is the nucleus longest diameter, CytoLong is the cytoplasm longest diameter, and KernePeri is the nucleus perimeter. The measurement unit is microns or 10^{-6} m (see more details for the definitions of the decision variables in the Appendix A, Fig. 7).

3.3.2. Standard genetic programming

As a second step, we applied standard genetic programming for the production of discriminating mathematical expressions. We selected to use for the training set 90% of the data. The rest 10% composed the test set. All seven (7) classes were used to discriminate the pap-smear data, according to the standard GP-approach. The intention was to produce six (6) rules, in a form of a mathematical expression that would perform the following heuristic classification scheme:

(a) Rule 1 distinguishes class 1 from classes 2—7
(b) Rule 2 distinguishes class 2 from classes 3—7
(c) Rule 3 distinguishes class 3 from classes 4—7
(d) Rule 4 distinguishes class 4 from classes 5—7
(e) Rule 5 distinguishes class 5 from classes 6 and 7
(f) Rule 6 distinguishes class 6 from class 7

Similarly to the aphasia domain, the classification was applied using the result of the mathematical expression. Thus, when the result was zero (0) or positive, then we considered a true value, and when then result was negative we suggested a false outcome.

Table 12 Pap-smear rule-strength for the standard GP procedure

Rule #	Correct classification in the test set (%)
1	95.91
2	97.72
3	94.87
4	100
5	72.41
6	68.42

For all the produced rules in the training set, the rule-strength ranged from 94.9 to 100% except for Rules #5 and #6, which means that the overall accuracy on the training data approaches exceeded 90%. The above results lead to the conclusion that additional information might needed than the information contained in our entire aphasia database, in order to discriminate between classes #5—#7. For example, the number of dysplasic cells contained into a certain area might be useful information. This is the reason why in the next genetic programming approach (Section 3.3.3) described below, we selected to unify classes #5—#7 into a single class (that of *dysplasia)*. In detail, the corresponding classification accuracy for the test data is shown in Table 12.

It is seen from the table above, that the classification accuracy drops dramatically when trying to discriminate between dysplasia classes. When we treat these three dysplasia classes as one single class, the overall classification accuracy for the set of the four remaining rules reaches 88.91%. As in the aphasia problem, the types of operators used in the genetic process were addition, subtraction, multiplication, protected division and the hyperbolic function. The resulted formulas are presented in prefix format (Lisp-like) below, where ADD denotes addition, SUB denotes subtraction, MUL denotes multiplication, DIV denotes division and TANH denotes the hyperbolic tangent (for the meaning of decision variables refer to Appendix A, Fig. 7).

- *Rule 1*: (MUL (SUB (TANH (SUB Ky CM)) (MUL (ADD CR (ADD (SUB (TANH (SUB (SUB (MUL (SUB CS KS) KP) (ADD KL (ADD (MUL (TANH (MUL KS CY)) KC) KS))) Cy)) KC) (DIV KS 79)))) Kp)) (SUB (MUL (DIV CR 73) (SUB KP (MUL Km CS))) (SUB CY KC)))
- *Rule 2*: (SUB (SUB (TANH (ADD (ADD (TANH CY) (SUB (DIV KM KY) KA)) (ADD (TANH (DIV Kx (MUL (ADD Cm KA) (TANH Km)))) (TANH (DIV (ADD CP 73) (TANH (TANH (DIV CY (DIV 108 KY)))))))) CY) (ADD CY (TANH (TANH (DIV (ADD Cm Km) (DIV 75 KY))))))

- *Rule 3*: (MUL (TANH (DIV (MUL (SUB KA CL) (SUB (TANH (ADD CE (MUL (ADD (SUB KP 43) (MUL CL (TANH (ADD (SUB KP 43) (MUL KR KC))))) KC))) KR)) (MUL (ADD (TANH (MUL CL KC)) CE) CR))) (TANH (ADD (ADD (SUB KP 43) (MUL CL KC)) (MUL 124 KC))))
- *Rule 4*: (DIV KA CS)
- *Rule 5*: (ADD (SUB (DIV (DIV (MUL KP KL) (ADD (SUB Kx Ky) (DIV 59 KS))) CM) (TANH (SUB (MUL CE (SUB (MUL KP CS) CM)) Cy))) (DIV (MUL KP KS) (ADD (SUB CA 44) (SUB (DIV (ADD CA (DIV (MUL KP KS) (SUB CS KS))) (SUB (MUL CS KS) Ky)) Cy))))
- *Rule 6*: (SUB (DIV (SUB (TANH (SUB (SUB KM (SUB (DIV (DIV Cy KC) Ky) (TANH (ADD (DIV Ky (ADD PI CM)) Km)))) Km)) (SUB (DIV Ky (ADD PI CM)) KE)) (SUB (SUB Ky (DIV (MUL PI Cy) (SUB (SUB 84 KM) Km))) Km)) (TANH (SUB Km (SUB (SUB 79 KM) KM))))

Generally, the process did not produce interpretable results for medical doctors although these results obtained a satisfactory level of classification accuracy compared to other genetic programming approaches for the pap-smear data set. An exception should be considered Rule #4, which appears a surprisingly high probability of correct classification in new data (100%), while it's meaning is simple and comprehensible by experts (KA/CS represents the nucleus area divided by the cytoplasm shortest diameter). In fact this rule was not unknown to medical staff, while it was already used to characterize class 4 cells (superficial) between other types of cells and thus, the standard GP-procedure just revealed this criterion, in other words it seems to have been able to generalize adequately over the pap-smear data.

3.3.3. PST-GP: Genetic programming for the production of a crisp rule-based system

The next step was to determine crisp decision rules. In this approach, the available data were separated in two halves. The first half was used for the training phase of the system and the second half for the test phase. After each termination of the training procedure, we selected the expression that managed to obtain good results also in the test set. This selection corresponds to our intention to select those expressions that could generalize, thus avoiding solutions over-fitted to the training sets. In total, for the training and the test phase, 50 cases were used as input to represent each of the classes 1—4 and other 250 cases for the representation of dysplasic cases (belonging to classes 5—7, all assumed and handled as one class in this experimentation). This was a result of the fact that most attempts to create generalizing rules (i.e. to have good performance in the test set) between classes 5—7, failed in the experiments performed. The training procedure followed in these cases, always resulted in over-fitting rules. Nevertheless, as noted in the previous subsection, the distinction between these classes usually depends on the number of the dysplasic cells found in the smear. This number (data not available to us) helps medical doctors to distinct a dysplasic case as mild, moderate, or severe dysplasia.

After the balancing of the available data into training and test groups, seven (7) rules were decided to be extracted in order to separate cells belonging to different classes, according to the following heuristic strategy (the classes shown for each rule denote also the corresponding training and test set for that rule):

- *Rule 1*: Separation of classes 1, 2 (CYL and PARA) from classes 3, 4 (INTER and SUPER)
- *Rule 2*: Separation of class 3 (INTER) from class 4 (SUPER)
- *Rule 3*: Separation of class 3 (INTER) from classes 5—7 (DYSPLASIC)
- *Rule 4*: Separation of class 1 (CYL) from class 2 (PARA)
- *Rule 5*: Separation of class 4 (SUPER) from classes 5—7 (DYSPLASIC)
- *Rule 6*: Separation of class 1 (CYL) from classes 5—7 (DYSPLASIC)
- *Rule 7*: Separation of class 2 (PARA) from classes 5— 7 (DYSPLASIC)

The rules that achieved better performance, after the training process, both in the training and test set, (proof of generalization) are presented in Table 13. As in the aphasia domain, in this problem, fitness of one (1) denotes a perfect fit on the selected set. Also, sensitivity and specificity of value one (1) denote perfect performance whereas value of zero (0) denotes poor performance. Simplicity values are normalized in [0,1] with a value of zero (0) denoting a 50-node expression and a value of one (1) denoting a single node expression.

Finally, these seven (7) rules acquired from the PST-GP approach, are presented below, where, Ky is the nucleus intensity, Cm is the cytoplasm maxima, KP is the nucleus position in cytoplasm, KM denotes nucleus maxima, CS is the cytoplasm's shortest diameter, CR is the cytoplasm's roundness, KC denotes the nucleus—cytoplasm ratio of areas, CY is the cytoplasm's intensity, CL is the cytoplasm's longest diameter, KA is the nucleus area, and KR represents the nucleus roundness):

Table 13 Pap-smear results for each of the rules of PST-GP

Rule #	Training data				Test data				Simplicity
	Fitness	Sensitivity	Specificity	Missed cases	Fitness	Sensitivity	Specificity	Missed cases	
1	1	1	1	0	0.98969	1	0.979592	1	0.986577
2	1	1	1	0	0.959933	0.923077	1	2	0.986577
3	1	1	1	0	1	1	1	0	0.932886
4	1	1	1	0	0.959933	0.923077	1	2	0.946309
5	1	1	1	0	1	1	1	0	0.986577
6	0.876202	0.833333	0.942308	10	0.891525	0.71875	0.988889	10	0.986577
7	1	1	1	0	0.952278	0.884615	0.991667	4	0.939597

- *Rule 1*: Ky $<$ Cm
- *Rule 2*: KP $>$ 28
- *Rule 3*: KM $<$ CS
- *Rule 4*: (CR $\leq$ KC) and ((Kp/CR) $\geq$ CY)
- *Rule 5*: CL $>$ KA
- *Rule 6*: KA $>$ 112
- *Rule 7*: KA $<$ (KM/KY)

In order to provide a complete classification scheme, the flowchart in Fig. 7 (see Appendix A) is constructed. The percentages in parenthesis accompanying each rule represent the correct classification that this rule obtains, according to the available data. The following eight (8) diagnostic steps correspond to the aforementioned rule diagram (Fig. 7). The sequence of execution of these steps is very important, as only if a higher-order step (i.e. rule) is not true, should the evaluation proceed to the next step.

1. If *KerneY* $<$ *CytoMin* and *KernePeri* $>$ 28 and *KerneMax* $<$ *CytoShort* then cell class is *INTER* (class 3), else:
2. If *KerneY* $<$ *CytoMin* and *KernePeri* $>$ 28 and *KerneMax* $\geq$ *CytoShort* then cell class is *DYS* (classes 5–7), else:
3. If *KerneY* $<$ *CytoMin* and *KernePeri* $\leq$ 28 and *CytoLong* $>$ *Kerne_A* then cell class is *SUPER* (class 4), else:
4. If *KerneY* $<$ *CytoMin* and *KernePeri* $\leq$ 28 and *CytoLong* $\leq$ *Kerne_A* then cell class is *DYS* (classes 5–7), else:
5. If *KerneY* $\geq$ *CytoMin* and [(*CytoRund* $\leq$ *KC*) and (*KernePos*/*CytoRund* $\geq$ *Cyto_Ycol*)] and Kerne_A $<$ (KerneMax/KerneYcol) then *PARA* (class 2)
6. If *KerneY* $\geq$ *CytoMin* and [(*CytoRund* $\leq$ KC) and (*KernePos*/*CytoRund* $\geq$ *Cyto_Ycol*)] and *Kerne_A* $\geq$ (*KerneMax*/*KerneYcol*) then *DYS* (classes 5–7)
7. If *KerneY* $\geq$ *CytoMin* and [not (*CytoRund* $\leq$ *KC*) and (*KernePos*/*CytoRund* $\geq$ *Cyto_Ycol*)] and *Kerne_A* $>$ 112 then *CYL* (class 1)
8. If *KerneY* $\geq$ *CytoMin* and [not (*CytoRund* $\leq$ *KC*) and (*KernePos*/*CytoRund* $\geq$ *Cyto_Ycol*)] and *Kerne_A* $\leq$ 112 then *DYS* (classes 5–7)

The rule-strength of each diagnostic step of the above PST-GP approach is 90.42% for steps 1 and 2, raises up to 97.96% for steps 3 and 4, reaches an 88.46% for step 5 and an almost prefect classification rate of 99.17% for step 6, drops down to 66.35% for step 7 and finally reaches 91.28% for the last diagnostic step. By rule-strength of a diagnostic step, we mean the correct classification of test data according to the rules of this step. For example, the first step separates intermediate cells from the rest of the cell types, and it classifies correctly as intermediate cells the 90.42% of the cases tested. Then the second step applies only if the first one fails to work as classifier, and the case is then classified as a class 5–7 (dysplasia), or it has to be tested by steps no. 3–8.

The acquired results are rated adequate in terms of comprehensibility, usability, accuracy, and ability to generalize. However, constraints such as *KerneY* $<$ *CytoMin* and *KerneMax* $\geq$ *CytoShort* make no physical sense, since they concern incompatible measurement units, e.g. it makes no sense to compare the nucleus intensity with the number of minima found in the cytoplasm (first inequality). Yet, the complete rule-based scheme used for classification, was not easily interpretable and it does not correspond to known diagnostic models used by pap-smear test experts.

4. Conclusions and further research

This paper presented GP-based hybrid intelligent methodologies for the construction of rule-based medical decision systems. Two medical domains were considered. Both domains, the aphasia and the pap-smear test databases, have not yet

obtained standard benchmarking classification rates. Furthermore, most records either in the aphasia database or in the pap-smear data set contains a number of quite subjective observations and estimations made by experts (i.e. medical doctors), a fact that renders both application domains difficult to standardise and model accurately. This is a main reason why we selected to test various intelligent methodologies in order to be able to draw conclusions and comparisons on both, the effectiveness of the proposed hybrid intelligent models and the appropriateness of the application domains. Results have shown that, generally, machine learning performs poor in both medical domains. Existing results in literature, using neural network models, offer the highest accuracy, however they require

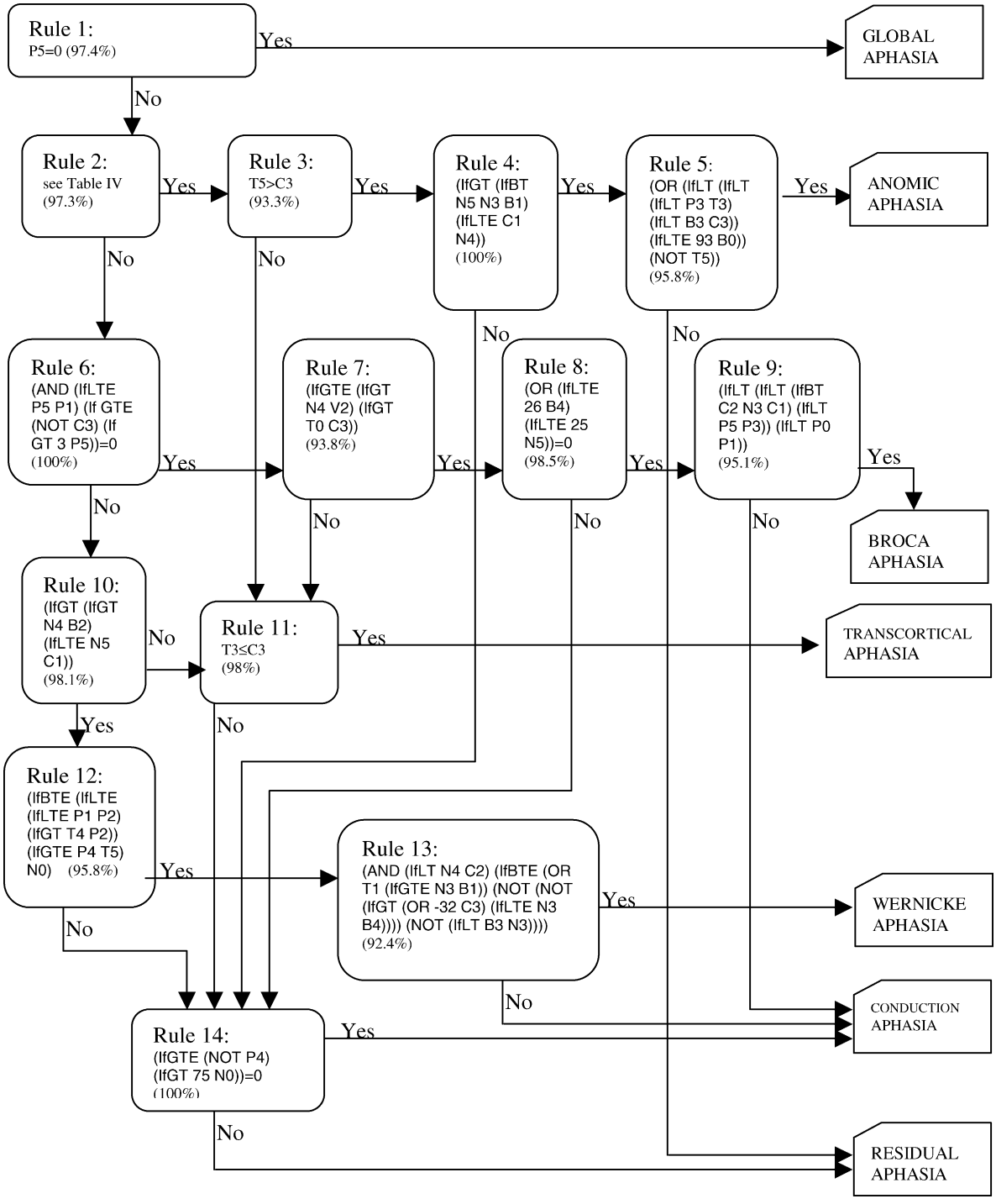
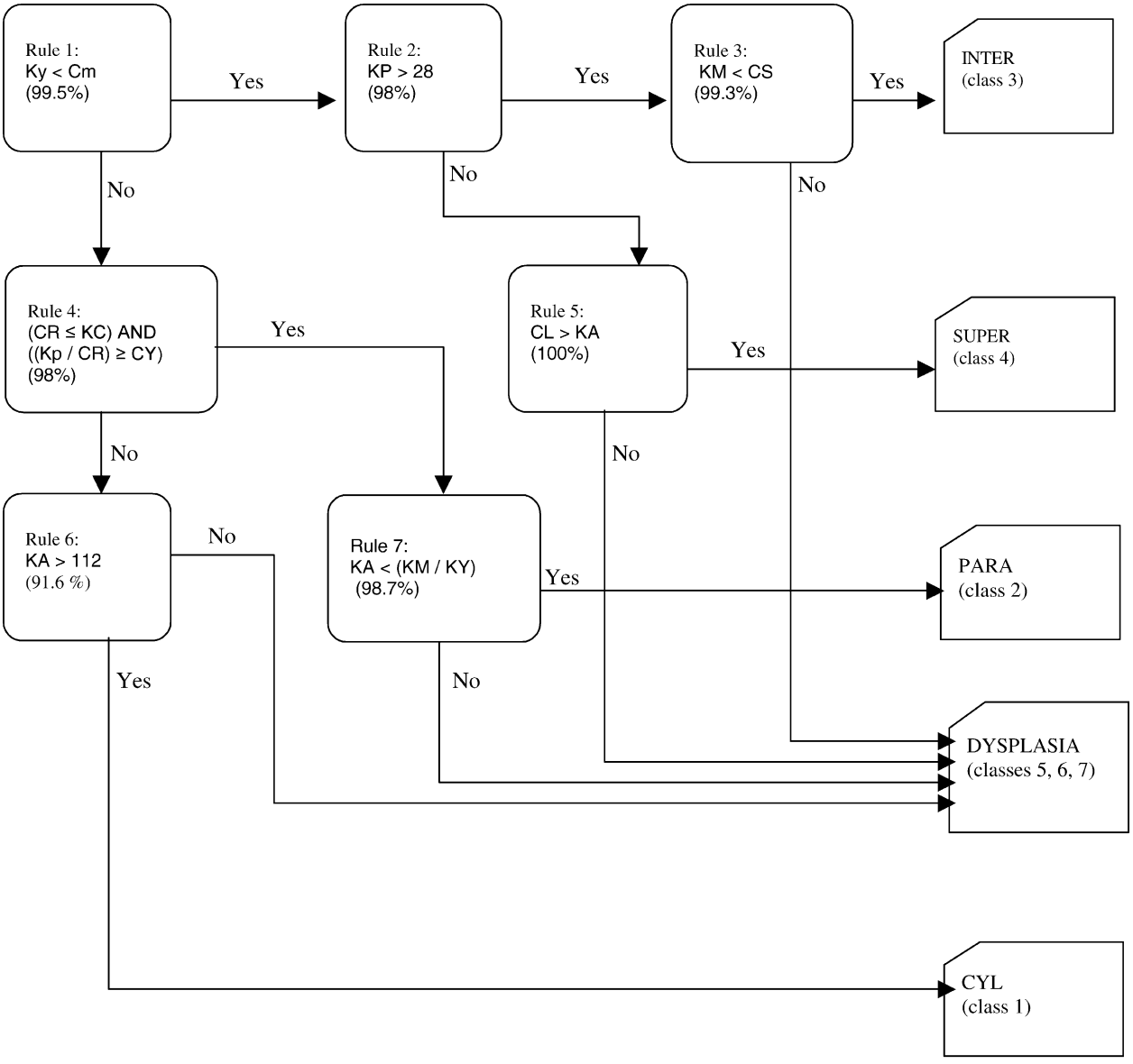


Figure 6 Rule flow-chart for classification of different types of aphasia corresponding to CRBS-GP2.

feature selection and configuration. Our standard—GP results, with a GP procedure for symbolic regression, offer the best accuracy among GP-solutions, though they lack interpretability. The accuracy of classification on test data of the acquired GP crisp and fuzzy rule bases is lower than the accuracy achieved by neural networks and higher than the accuracy obtained by machine learning. These GP



Legend:

Kerne_A nucleus area	Cyto_A cytoplasm area	K/C nucleus area / cytoplasm area	Kerne_Ycol nucleus intensity	Cyto_Ycol cytoplasm intensity	KerneShort nucleus shortest diameter	KerneLong nucleus longest diameter	KerneElong nucleus elongation	KerneRund nucleus roundness	CytoShort cytoplasm shortest diameter	CytoLong cytoplasm longest diameter	CytoElong cytoplasm elongation
KA	**CA**	**KC**	**KY**	**CY**	**KS**	**KL**	**KE**	**KR**	**CS**	**CL**	**CE**
CytoRund cytoplasm roundness	KernePer Nucleus perimeter	CytoPeri cytoplasm perimeter	KerneX nucleus x coordinate	KerneY nucleus y coordinate	CytoX cytoplasm x coordinate	CytoY cytoplasm y coordinate	KernePos nucleus position in cytoplasm	KerneMax nucleus maxima	KerneMin nucleus minima	CytoMax cytoplasm maxima	CytoMin cytoplasm minima
CR	**KP**	**CP**	**Kx**	**Ky**	**Cx**	**Cy**	**Kp**	**KM**	**Km**	**CM**	**Cm**

Figure 7 Rule flowchart for classification between different types of cervical cells corresponding to PST-GP.

crisp and fuzzy rule bases however, retain higher comprehensibility, as compared with the competitive methodologies, when criticized by medical experts.

Specifically, the aphasia diagnosis problem consisted the first medical application domain to test, through the aphasia database maintained in Aachen Medical School, Department of Anatomy (Germany). Initially, two crisp models were constructed and used for the diagnosis of aphasia. Our methodology consisted of a genetic programming core and a supporting heuristic rule-based classification system. The first model was implemented only for the four (4) major types of aphasia. This approach enabled the authors to draw conclusions on the model's effectiveness as compared to previous intelligent techniques found in literature [37]. The results were comprehensible for medical experts and they could be characterized as almost equivalent to previous approaches [37]. They have also enabled the experts to draw conclusions or, to reconfirm known medical results in some cases. The other proposed crisp classification model was applied to the full range of aphasia's subtypes, in order to be able to operate as an assisting (black-box architecture oriented) decision tool. These latter results were not much comprehensible, however a relatively high accuracy was still obtained in the test data set.

As a next step, a genetic programming model for the construction of fuzzy rule-based systems was considered. Results were comparable to those of crisp rule-based systems, however the difference in classification accuracy between the training and the test data was significantly less ($\sim$10% for the FRBS-GP model, versus $\sim$20% for the CRBS-GP1 model). This outcome may denote the fuzziness involved in the data, offering more generalizing capabilities on the FRBS-GP model.

Our work on the pap-smear diagnosis problem proposed and tested a similar intelligent methodology for the construction of crisp rule-based medical decision systems. We initially combined the genetic programming search with a heuristic scheme for classification, in order to obtain a rule-based decision output. This domain was considered particularly suitable for genetic programming approaches, due to the completeness of the smear database and the numerical (and quite accurate) nature of the data. Results were rather comprehensible and could prove further useful for the construction of computer-based systems for medical assistance in pap-smear diagnosis. Comparison with other competitive approaches does not exist in literature for the same sample of data (available at: http://fuzzy.iau.dtu.dk/smear/). Our prime intention has always been to produce comprehensible and sensible rules that potentially help medical doctors to extract conclusions, often at the expense of a higher classification score achievement. Thus, we primarily promoted small-sized solutions throughout the whole training phase, sometimes receiving solutions even with lower classification accuracy, in order to finally get a reasonable rule-based scheme. In most cases this aim was achieved, producing rules with very small size. For example, 5/7 rules obtained from the application of the PST-GP approach on the pap-smear problem are very simple. Our future intention is to provide solutions using alternative methodologies or different model configurations in order to obtain transparent results. On the other hand, the validation of a methodology intended for medical assistance, makes real sense when tested in real world conditions, next to a medical expert, applied on newly acquired data corresponding to patient records. Finally, two other ways of using further the proposed hybrid scheme for the pap-smear problem could be the discovery of new diagnostic medical knowledge, as well as the construction of training computer programs for related novice medical staff.

Appendix A

Please see Figs. 6 and 7.

References

[1] Bojarczuk CC, Lopes HS, Freitas AA. Genetic programming for knowledge discovery in chest-pain diagnosis. IEEE Eng Med Biol 2000;38—44.
[2] Elstein AS, Shulman LS, Sprafta SA. Medical problem solving: an analysis of clinical reasoning. Cambridge: Harvard University; 1978.
[3] Quinlan JR. Induction of decision trees. Machine Learning 1986;1:81—106.
[4] Nauck D, Kruse R. Designing neuro-fuzzy systems through backpropagations. In: Pedrycz W, editor. Fuzzy modelling—paradigms and practice. Boston: Kluwer Academic Publishers; 1996. p. 203—31.
[5] Koza JR. Genetic programming—on the programming of computers by means of natural selection. Cambridge, MA, USA: The MIT Press; 1992.
[6] Wong ML. A flexible knowledge discovery system using genetic programming and logic grammars. Decision Support Syst 2001;31(4):405—28.
[7] Koza JR, Bennett III FH, Andre D, Keane MA. Genetic programming III: Darwinian invention and problem solving. San Fransisco: Morgan Kaufmann Publishers; 1999.
[8] Goldberg DE. Genetic algorithms in search optimization and machine learning. Reading, MA: Addison-Wesley; 1989.
[9] Montana DJ. Strongly typed genetic programming. Evolutionary Comput 1995;2(3):199—230.

[10] Gruau F. On using syntactic constraints with genetic programming. Cambridge, MA: MIT Press; 1996.
[11] Mamdani EH, Assilian S. An experiment in linguistic synthesis with a fuzzy logic controller. Int J Man-Machine Stud 1975;7(1):1—13.
[12] Jang J-.R. Neuro-fuzzy modeling for nonlinear dynamic system identification. In: Ruspini EH, Bonissone PP, Pedrycz W, editors. Handbook of fuzzy computation. Dirak House, Temple Back, Bristol, UK: Institute of Physics Publishing; 1998.
[13] Herrera F, Verdegay KL, editors. Genetic algorithms and soft computing. Heidelberg: Physica-Verlag; 1996.
[14] Li Y, Ng KC. Uniform approach to model-based fuzzy control system design and structural optimization. In: Herrera F, Verdegay KL, editors. Genetic algorithms and soft computing. Heidelberg: Physica-Verlag; 1996.
[15] Alba E, Cotta C, Troya JM. Type-constrained genetic programming for rule-base definition in fuzzy logic controllers. In: Koza JR, Goldberg DE, Fogel DB, Riolo RL, editors. Proceedings of the 1st Annual Conference on Genetic Programming. Cambridge, MA: The MIT Press; 1996. p. 255—60.
[16] Alba E, Cotta C, Troya JM. Evolutionary design of fuzzy logic controllers using strongly-typed GP. In: Proceedings of the IEEE International Symposium on Intelligent Control; 1996. p. 127—32.
[17] Koza JR. Genetic programming II—automatic discovery of reusable programs. Cambridge, MA: MIT Press; 1994.
[18] Langdon WB. Data structures and genetic programming. In: Angeline PJ, Kinnear Jr KE, editors. Advances in genetic programming, vol. 2. Cambridge, MA: MIT Press; 1996. p. 395—414.
[19] Angeline PJ, Kinnear Jr KE. Advances in genetic programming. Cambridge, MA: MIT Press; 1996.
[20] Matthews BW. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochem Biophys Acta 1975;405:442—551.
[21] Rogers A, Prügel-Bennett A. Modeling the dynamics of steady-state genetic algorithms. In: Banzhaf W, Reeves C, editors. Foundations of genetic algorithms. San Francisco: Morgan Kaufmann; 1999. p. 57—68.
[22] Blickle T, Theile L. A mathematical analysis of tournament selection. In: Eshelman LJ, editor. Proceedings of the 6th International Conference on Genetic Algorithms. Hillsdale, NJ: Lawrence Erlbaum Associates; 1995. p. 9—16.
[23] Janikow CZ. A methodology for processing problem constraints in genetic programming. Comput Math Appl 1996;32(8):97—113.
[24] Horner H. A C++ class library for GP, Technical Report published by the Vienna University of Economics Genetic Programming Kernel; 1996. Visit also: http://www.cs.bham.ac.uk/simwbl/biblio/gp-html/HelmutHorner.html (acc. 04/04/04).
[25] Paterson N, Livesey M. Evolving caching algorithms in C by GP. In: Koza JR, Deb K, Dorigo M, Fogel DB, Garzon M, Iba H, Riolo RL, editors. Proceedings of the 2nd Annual Conference on Genetic Programming. San Francisco, CA: Morgan Kaufmann; 1997. p. 262—7.
[26] Ryan C, Collins JJ, O'Neil M. Grammatical evolution: evolving programs for an arbitrary language. In: Banzhaf W, Poli R, Schoenauer M, Fogarty TC, editors. Genetic programming, lecture notes in computer science. Berlin: Springer-Verlag; 1998.
[27] Whigham P. Search bias, language bias and genetic programming. In: Koza JR, Goldberg DE, Fogel DB, Riolo RL, editors. Proceedings of the First Annual Conference on Genetic Programming. Stanford, CA: MIT Press; 1996 . p. 230—7.
[28] Quinlan JR. C4.5: Programs for machine learning. San Mateo, CA: Morgan Kaufmann; 1993.
[29] Mitchell TM. Machine learning. New York: McGraw-Hill; 1997
[30] Langley P, Simon H. Applications of machine learning and rule induction. Commun ACM 1995;38(11):55—64.
[31] Huber W, Poeck K, Weniger D. The Aachen aphasia test. In: Rose FC, editor. Advances in neurology, vol. 42: progress in aphasiology. New York: Raven; 1984.
[32] Axer H, Jantzen J, Berks G, Südfeld D, von Keyserlingk DG. The aphasia database on the web: description of a model for problems of classification in medicine. In: Proceedings of the European Symposium on Intelligent Techniques, ESIT 2000. Aachen: Verlag-Mainz; 2000. p. 104—10.
[33] Axer H, Jantzen J, von Keyserlingk DG. An aphasia database on the internet: a model for computer assisted analysis in aphasiology. Brain Language 2000;75:390—8.
[34] Quinlan JR. Boosting, bagging, and C4.5. In: Proceedings of the 13th National Conference on Artificial Intelligence. Portland Oregon: AAAI Press; 1996. pp. 725—730.
[35] Schapire R. The strategy of weak learnability. Machine Learning 1990;5(2):197—227.
[36] Freund Y, Schapire R. Experiments with a new boosting algorithm. In: Saitta L, editor. Proceedings of the 13th International Conference on Machine Learning. Morgan Kaufmann; 1996. p. 148—56.
[37] Jantzen J, Axer H, von Keyserlingk DG. Diagnosis of aphasia using neural and fuzzy techniques. In: Zimmermann H-J, Tselentis G, van Someren M, Dounias G, editors. Advances in computational intelligence and learning. Massachussets: Kluwer Academic Publishers; 2002. p. 461—74.
[38] Tsakonas A, Dounias G, Axer H, von Keyserlingk DG. Hybrid CI and Adaptive Schemes for handling the problem of aphasia diagnosis. In: Proceedings of the CD-rom EUNITE-01, European Symposium on Intelligent Technologies, Hybrid Systems and their Implementation on Smart Adaptive Systems. Verlag-Mainz; 2001.
[39] Meisels A, Morin C. Cytopathology of the uterus. 2nd ed. Chicago: American Society for Clinical Pathology Press—"ASCP" Press; 1997.
[40] Koss L. The application of PAPNET to diagnostic cytology. In: Lisboa P, Ifeachor E, Szczepaniak P, editors. Artificial neural networks in biomedicine. London: Springer; 2000. p. 51—68.
[41] Tsakonas A, Dounias G, Jantzen J, Bjerregaard B. A hybrid CI approach combining genetic programming and heuristic classification for pap-smear diagnosis. In: Proceedings of the CD-rom EUNITE-01, European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems. Verlag-Mainz; 2001.
[42] Byriel J. Neuro-fuzzy classification of cells in cervical smears. MSc Thesis. Department of Automation, Technical University of Denmark, Kongens, Lyngby, Denmark, 1999.