



**HAL**  
open science

# Evolutionary feature selection applied to artificial neural networks for wood veneer classification

Marco Castellani, Hefin Rowlands

► **To cite this version:**

Marco Castellani, Hefin Rowlands. Evolutionary feature selection applied to artificial neural networks for wood veneer classification. *International Journal of Production Research*, 2008, 46 (11), pp.3085-3105. 10.1080/00207540601139955 . hal-00512967

**HAL Id: hal-00512967**

**<https://hal.science/hal-00512967>**

Submitted on 1 Sep 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**Evolutionary feature selection applied to artificial neural networks for wood veneer classification**

Journal:	<i>International Journal of Production Research</i>
Manuscript ID:	TPRS-2006-IJPR-0437.R1
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	27-Nov-2006
Complete List of Authors:	Castellani, Marco; Universidade Nova Lisboa, Centro de Inteligência Artificial (CENTRIA), Departamento Informática Rowlands, Hefin; University of Wales, Centre for Enterprise and Quality Management
Keywords:	EVOLUTIONARY ALGORITHMS, COMPUTER VISION, NEURAL NETWORK APPLICATIONS, PATTERN RECOGNITION, INSPECTION
Keywords (user):	feature selection



# Evolutionary feature selection applied to artificial neural networks for wood veneer classification

MARCO CASTELLANI\*<sup>†</sup> and HEFIN ROWLANDS<sup>‡</sup>

<sup>†</sup>Centro de Inteligência Artificial (CENTRIA), Departamento Informática,  
Universidade Nova Lisboa, 2829-516 Caparica, Portugal

<sup>‡</sup>Centre for Enterprise and Quality Management, University of Wales, Newport, Allt-yr-yn Campus, PO Box 180, NP20 5XR Newport, UK

## ABSTRACT

This paper presents the application of *FeaSANNT*, an evolutionary algorithm for optimisation of artificial neural networks, to the training of a multi-layer perceptron for identification of defects in wood veneer. Given a fixed artificial neural network structure, *FeaSANNT* concurrently evolves the input feature vector and the network weights. The novelty of the method lies in the implementation of the embedded approach in an evolutionary feature selection paradigm. Experimental tests show that the proposed algorithm produces high-performing solutions with robust learning results. Significant reduction of the set of veneer features is obtained. Experimental comparisons are made with a previous method based on statistical filtering of the input features and a standard genetic wrapper algorithm. In the first case, *FeaSANNT* greatly reduces the feature set with no degradation of the neural network accuracy. Moreover,

---

\* Corresponding author. Email: mcas@ftc.unl.pt

1  
2  
3  
4 *FeaSANNT* entails lower design costs since feature selection is fully automated. In the  
5  
6 second case, the proposed algorithm achieves superior results in terms of identification  
7  
8 accuracy and reduction of the feature set. *FeaSANNT* involves also lower computational  
9  
10 costs than the standard evolutionary wrapper approach and eases the algorithm design  
11  
12 effort. Limited overlapping is observed between the patterns of features selected by the  
13  
14 three algorithms. This result suggests that the full feature set contains mainly redundant  
15  
16 attributes.  
17  
18  
19  
20  
21  
22

23 **Keywords:** neural nets, evolutionary algorithms, feature selection, pattern recognition,  
24  
25 computer vision  
26  
27  
28  
29  
30  
31  
32  
33  
34

### 35 NOTATION

36	
37	
38	FeaSANNT feature selection and artificial neural network training
39	
40	ANN artificial neural network
41	
42	MLP multi-layer perceptron
43	
44	EA evolutionary algorithm
45	
46	BP backpropagation
47	
48	
49	ANNT artificial neural network training
50	
51	
52	basicBP backpropagation rule with momentum term - fixed duration
53	
54	valBP backpropagation rule with momentum term - variable duration
55	
56	
57	
58	
59	
60	

## 1 Introduction

Plywood is made of thin layers of wood, called veneers, joined together using an adhesive. Defects of the veneer are identified by human inspectors as the sheets are transported to assembly on a conveyor. The task is extremely stressful and demanding and mis-classifications are frequent. Two studies conducted on inspectors in wood mills reported identification accuracies ranging from a more optimistic 68% (Huber et al., 1985) estimate to a more conservative 55% (Polzleitner and Schwingshagl, 1992) measure.

An automatic visual inspection system (Pham and Alcock, 1999a) was developed for this application by the Intelligent Systems Lab of Cardiff School of Engineering at the University of Wales, UK and the Wood Research Institute of Kuopio, Finland. Fig. 1 outlines the system. Monochrome images of the veneer are pre-processed by algorithms that locate defect areas (Pham and Alcock, 1999b) where a set of numerical descriptors is extracted for further analysis. Seventeen statistical attributes of the local grey level distribution were initially deemed relevant for defect identification (Pham and Alcock, 1999c). Twelve defects of the veneer can be distinguished from clear wood giving 13 possible classes. For each data sample, a classifier takes the 17-dimensional vector of image features and decides to which of the thirteen classes the pattern belongs.

On the recognition of wood veneer defects, the best results were obtained using artificial neural network (ANN) (Pham and Liu, 1995) classifiers. In particular, Packianather (Packianather, 1997; Packianather and Drake, 2000) reported 85% identification rates using a Multi-Layer Perceptron (MLP) (Pham and Liu, 1995).

On the basis of statistical analysis of the patterns of defective veneer, Packianather (1997) suggested that reduction of the feature vector is beneficial. The study examined

1  
2  
3  
4 the intra-class and the inter-class variation amongst the attributes and lead to  
5  
6 improvement of the classifier compactness and accuracy upon rejection of 6 of the 17  
7  
8 data features.  
9

10  
11 The statistical method used by Packianather (1997) is an example of *filter* approach  
12 (Blum and Langley, 1997) to feature selection. **The filter approach selects variables**  
13 **by ranking them according to statistical, geometrical or information theoretical**  
14 **measures.** Despite being the least computationally intensive, such approach is not  
15  
16 guaranteed to produce optimal results since it does not take into account the interaction  
17  
18 between the input vector and the classifier (Yang and Honavar, 1998).  
19

20  
21 A popular alternative method for feature selection is the *wrapper* approach (Blum and  
22  
23 Langley, 1997), that is, to directly evaluate subsets of features by running the induction  
24  
25 algorithm and taking the classification accuracy as fitness measure. Compared to the  
26  
27 filter approach, the wrapper approach involves a more severe computational effort since  
28  
29 it entails extensive experimentation guided by hill climbing of the optimisation  
30  
31 landscape (Vafaie and DeJong, 1995; Blum and Langley, 1997). Due to the locality of  
32  
33 the search procedure, algorithms following this approach are likely to be deceived by  
34  
35 noise or get trapped into secondary peaks of performance. Global search techniques are  
36  
37 more suitable to search the noisy, multimodal and deceptive optimisation surface of the  
38  
39 feature space.  
40  
41  
42  
43  
44  
45  
46  
47  
48

49  
50 This study focuses on the application of evolutionary algorithms (EAs) (Fogel, 2000) to  
51  
52 the selection of the set of attributes for the ANN wood veneer defect identification  
53  
54 system.  
55

56  
57 A large literature supports the use of EAs for feature selection (e.g. Vafaie and DeJong,  
58  
59 1995; Yao, 1999; Smith and Bull, 2003; Zhang et al., 2005), the wrapper approach  
60

1  
2  
3  
4 being the common implementation. The main problem in the design of evolutionary  
5 feature selection algorithms for ANN classifiers is the time complexity of the evaluation  
6 procedures for the solutions. This slowness adds to the slowness of EAs, thus severely  
7 constraining the extent of the evolutionary search.  
8  
9

10  
11  
12  
13 The evolutionary *FeaSANNT* algorithm was recently developed for automatic feature  
14 selection and ANN training. Similarly to other algorithms in the literature, the proposed  
15 method relies on the global nature of the evolutionary search to escape sub-optimal  
16 peaks of performance. However, *FeaSANNT* is characterised by a distinctive approach  
17 based on the concurrent evolution of the feature vector and the ANN weights. This  
18 *embedded* approach (Blum and Langley, 1997) allows considerable savings of  
19 computation time and simplifies the algorithm design.  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29

30 This paper presents the application of the *FeaSANNT* algorithm to the wood veneer  
31 defect identification problem. Experimental comparisons are made with a standard  
32 evolutionary feature selection algorithm and the statistical approach employed by  
33 Packianather (1997).  
34  
35  
36  
37  
38  
39

40 Section 2 introduces the problem domain. Section 3 presents the proposed algorithm.  
41 Section 4 describes the experimental design. Section 5 presents the experimental results.  
42 Section 6 analyses the computational overheads of the proposed algorithm. Section 7  
43 discusses the results. Section 8 concludes the paper and proposes areas for further  
44 investigation.  
45  
46  
47  
48  
49  
50

## 51 52 53 **2 Problem Domain**

54  
55 **The goal of this study is to design an MLP classifier that correctly recognises**  
56 **instances of veneer defects. Defect areas can be identified and separated from clear**  
57 **wood following image segmentation.**  
58  
59  
60

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

The segmentation procedure was the work of Pham and Alcock (1996). Digitised images of the veneer sheets consist of 512x512 pixels. A feature extraction window of size 60 pixels in the horizontal direction and 85 pixels in the vertical direction (roughly corresponding to a 3cm square on the sheet) is centred around likely defective areas. The grey level values and their frequencies are recorded from the feature extraction window.

A set of seventeen features are extracted from each window for veneer defect identification. These features are listed in Table 1. Features 7, 14 and 15 are designed to detect dark defects. Features 8, 16 and 17 are designed to detect bright defects. Features 1 to 12, 14 and 16 are first-order features. First-order features are tonal attributes and they are calculated directly from the grey level histogram of the window. Features 13, 15 and 17 are second-order features. Second-order features are textural attributes and they are obtained from the image itself by thresholding and edge-detection. Further details on the feature extraction process can be found in (Packianather, 1997; Drake and Packianather, 1998)

Twelve defects of the veneer can be distinguished, namely bark, streaks, pin knots, coloured streaks, curly grain, discoloration, rotten knots, roughness, sound knots, splits, holes and worm holes. Data patterns must be classified into one of thirteen classes corresponding to the twelve possible defects and clear wood.

A set of 232 pre-classified numerical data representing vectors of statistical features and associated plywood defects is available. All the classes are represented by 20 examples, except for curly grain (16 examples), holes and worm holes (8 examples each).



### 3 *FeaSANNT*: Feature Selection and ANN Training Algorithm

The *FeaSANNT* (Feature Selection and ANN Training) algorithm uses the embedded approach for the selection of the data attributes for ANN systems. The choice of such approach, still a largely unexplored area in the evolutionary feature selection for ANNs, is motivated by the possibility of obtaining high performing solutions at reduced computational costs.

#### 3.1 General Overview

*FeaSANNT* architecture is sketched in fig. 2. The algorithm comprises a feature selection module and an ANN training module that act concurrently on the same pool of individuals.

Both the modules are expected to benefit from their co-occurrence. On the one hand, manipulation of the input vector modifies the error surface thus helping the ANN training module to escape local peaks or flat areas of fitness. On the other hand, similar input vectors require similar weight settings, thus allowing modification of the feature vector without major disruption of the ANN behaviour.

The genotype of each individual is composed of two chromosomes (Fogel, 2000), namely a binary string representing the data attributes and a real-valued string representing the MLP weights.

Each generation the fitness of the population is assessed, then a cycle of the feature selection module and a cycle of the ANN training module are executed. Fitness ranking (Fogel, 2000) is used to select the pool of reproducing individuals. A new population is produced through genetic recombination, mutation and BP training of the individuals. New solutions replace old ones via generational replacement (Fogel, 2000). The

1  
2  
3  
4 procedure is repeated until a pre-defined number of iterations has elapsed and the fittest  
5  
6 solution of the last generation is picked.  
7

8  
9 Genetic crossover is operated only in the feature selection module. The choice against  
10  
11 recombination of the MLP weight encodings is motivated by the lack of knowledge  
12  
13 about which functional units to swap. On the contrary, the distributed nature of the  
14  
15 knowledge base in connectionist systems supports the argument against point-to-point  
16  
17 exchanges of genetic material amongst solutions. Against the appropriateness of the  
18  
19 crossover operator weighs also the competing convention problem (Thierens et al.,  
20  
21 1993), namely the many-to-one mapping from the representation of the solutions (the  
22  
23 genotype) to the actual ANN (the phenotype). This problem leads to disruption of the  
24  
25 genetic search.  
26  
27  
28

29  
30 The backpropagation (BP) (Rumelhart and McClelland, 1986) rule is included into the  
31  
32 ANN training module to support the MLP training procedure. This operator acts on the  
33  
34 weights of the decoded individuals so as to reduce classification error. The changes are  
35  
36 stored into the genotype. The algorithm therefore uses “Lamarckism” (Aboitiz, 1992),  
37  
38 that is, the permanent storing in the genotype of behaviours learned by the phenotype.  
39  
40  
41

### 42 43 **3.2 Input Feature Selection Module** 44

45  
46 This module selects from an initial broad set of input features that subset which  
47  
48 maximises the ANN performance.  
49

50  
51 Candidate solutions are encoded in the chromosome representing the set of pattern  
52  
53 attributes. This chromosome corresponds to a binary mask of length equal to the  
54  
55 number of the full feature set and defines which attributes are fed to the ANN. The  
56  
57 feature selection module manipulates the input mask via the two customary genetic  
58  
59  
60

1  
2  
3  
4 operators of bit-flip mutation (Fogel, 2000) and two-point crossover (Fogel, 2000). The  
5  
6 feature selection algorithm is akin to a standard genetic algorithm (Fogel, 2000).  
7  
8  
9

### 10 **3.3 Neural Network Connection Weight Optimisation Module**

11  
12 This module evolves the ANN weight encodings to minimise classification error. Each  
13  
14 solution is characterised by the real-valued chromosome that encodes the setting of the  
15  
16 connection weights. Evolution is achieved via two operators, namely mutation and the  
17  
18 BP algorithm.  
19  
20  
21

22 Genetic mutations slightly modify the weights of the nodes of a solution. For each  
23  
24 weight, the perturbation is randomly sampled with uniform probability from an interval  
25  
26 of pre-defined width.  
27  
28

29 The BP rule is introduced as a deterministic mutation operator to speed up the learning  
30  
31 process. If selected, an individual undergoes one cycle of BP learning over the whole  
32  
33 training set. Because BP learning is computationally expensive, the operator is used  
34  
35 with a moderate rate of occurrence.  
36  
37

38 Weights corresponding to switched off features are still processed by the ANN training  
39  
40 module. However, the only alteration on such weights comes from the mutation  
41  
42 operator, which is a zero-mean random perturbation. The BP operator has no effect  
43  
44 since there is no signal (i.e. zero signal) passing through the connection. The genetic  
45  
46 drift of these genes is therefore expected to be extremely modest.  
47  
48  
49

50 The weight optimisation module can be run independently from the feature selection  
51  
52 procedure and used as an ANN training algorithm on its own. It is henceforth referred as  
53  
54 the *ANNT* (ANN Training) algorithm.  
55  
56  
57  
58  
59  
60

### 3.4 Fitness Evaluation Procedure

The fitness of the candidate solutions is evaluated on their classification accuracy on the training set of examples. To encourage the creation of compact solutions, whenever the fitness score of two individuals is equal, preference is given to the solution using the smallest feature set.

There are cases where the difference in accuracy between some of the solutions is negligible in comparison to the spread of the population. In such cases, it is more efficient to consider those solutions as equally performing and give preference to the ones using the smallest feature sets.

The proposed algorithm considers the accuracy of two individuals to be equal when the difference is less than one standard deviation of the average population accuracy. That is, the population is divided into a number of bins of width equal to

$$width = \max \left[ std\_dva \cdot \left( 1 - \frac{gen}{duration} \right), \frac{best - worst}{popsize} \right] \quad (1)$$

where  $width$  is the width of the bin,  $std\_dva$  is the standard deviation of the population fitness,  $gen$  is the current evolutionary cycle,  $duration$  is the length of evolutionary procedure,  $best$  and  $worst$  are the classification accuracies of respectively the best and the worst individual and  $popsize$  is the population size.

The first bin is centred around the best performing solution while the centres of the remaining bins are calculated according to the following formula:

$$centre_i = best - i \cdot width \quad (2)$$

where  $centre_i$  is the centre of the  $i^{\text{th}}$  bin and  $i$  is an integer number ( $i=1, \dots, n$ ) that is progressively increased until all the population is grouped.

The proposed procedure aims at cutting the noise that affects the evaluation of the candidate solutions. As the algorithm proceeds, the width of the bins is shrunk to shift the emphasis on finer differences of accuracy. For each fitness evaluation test, equation (1) limits the number of bins to a value that is no greater than the population size.

Solutions are awarded the following pair of measures as fitness score:

$$fitness_j = \{n - i, size_j\} \quad (3)$$

where  $fitness_j$  is the fitness score of the  $j^{\text{th}}$  member of the population,  $i$  is the bin where the  $j^{\text{th}}$  solution lies,  $n$  is the total number of bins and  $size_j$  expresses the dimensionality of the MLP input layer (i.e. the number of selected features).

The first fitness measure is proportionally related to the classification accuracy. That is, the best performing solution (grouped into the first bin) has an accuracy score equal to  $n-1$ . All the solutions within half bin width from the accuracy of the best individual obtain the same score. The solutions grouped into the second bin obtain an accuracy score equal to  $n-2$ , and so forth until the last bin where solutions achieve a score equal to  $0$ . Solutions having the same accuracy score (i.e. belonging to the same bin) are ranked according to the number of selected features by the fitness ranking procedure.

## 4 Experimental Design

The proposed algorithm is applied to the wood veneer classification problem. For comparison, six tests are performed on the same problem using combinations of different feature selection and ANN training algorithms. In the first three tests, the MLP classifier is trained using the full feature set and employing respectively two versions of the BP rule and *ANNT*. In the following two tests, the BP rule and *ANNT* are used to train the classifier using the reduced feature set suggested by Packianather (1997). In the last test, a standard genetic wrapper algorithm is employed. In all the cases, the learning algorithms act on the same MLP architecture that is employed by *FeaSANNT*.

### 4.1 Full Feature Set

The first set of experiments is performed using the full feature set. The results of the learning trials are used as a baseline for comparison with the feature selection procedures.

In the first experiment, the MLP is trained using the standard BP rule with a momentum term. The learning procedure is run for a fixed number of iterations on the training set of examples. This algorithm is called henceforth *basicBP*.

The second experiment replicates the above procedure using the *ANNT* algorithm. The algorithm is run for a fixed number of iterations and the fittest individual of the last generation is chosen as the solution. This test evaluates the efficacy of *ANNT*, which is the MLP training module of *FeaSANNT*.

The third experiment uses again the BP rule with a momentum term but changes the stopping criterion for the training of the classifier. In this case, the training set is randomly divided into an MLP training subset and a validation subset. The classifier is trained using the MLP training subset and the learning accuracy is monitored on the

1  
2  
3  
4 validation subset. As the classification accuracy on the validation subset stops  
5  
6 improving, the learning procedure is terminated. This algorithm is called henceforth  
7  
8 *valBP*. This test evaluates the efficacy of the MLP training procedure of the genetic  
9  
10 wrapper algorithm.  
11  
12

#### 13 14 15 **4.2 Feature Selection through Filter Approach**

16  
17 The second set of experiments is performed using the reduced set of attributes that is  
18  
19 selected by Packianather (1997) using statistical analysis of the data patterns. Also in  
20  
21 this case, two tests are performed, the first training the ANN with the *basicBP* rule and  
22  
23 the second using the *ANNT* algorithm.  
24  
25  
26

#### 27 28 **4.3 Feature Selection through Evolutionary Wrapper Approach**

29  
30 The last control experiment is performed using a standard evolutionary wrapper  
31  
32 approach.  
33

34  
35 The algorithm uses the feature selection module of *FeaSANNT* to select the data  
36  
37 attributes, and the BP rule with a momentum term to train the MLP classifier. The  
38  
39 whole system is equivalent to a standard genetic wrapper algorithm.  
40

41  
42 The algorithm keeps the overall structure of *FeaSANNT*. Since the evolutionary  
43  
44 procedure concerns only the input vector, individuals are characterised by the sole  
45  
46 binary string defining the ANN input mask. This genotype is manipulated by the  
47  
48 mutation and crossover operators of the feature selection module. Individuals are  
49  
50 selected for reproduction according to the fitness ranking procedure and the offspring  
51  
52 population replaces the parent population via generational replacement.  
53  
54

55  
56 The fitness of the individuals is evaluated on the learning results of the *valBP* algorithm.  
57  
58 That is, for each solution an MLP classifier is randomly initialised and trained using  
59  
60

1  
2  
3  
4 those attributes of the training patterns that correspond to enabled locations of the  
5 genotype. The wrapper algorithm employs the same procedure used by *FeaSANNT* (see  
6  
7 Section 3.4) to assign the fitness measure of an individual.  
8  
9

10  
11 Each generation of the evolutionary algorithm, the training set of examples is randomly  
12 divided into an MLP training subset and a validation subset. All the MLPs are trained  
13 on the former and their learning accuracy is monitored on the latter. Since the MLP  
14 training subset and the fitness validation subset are re-initialised at every evolutionary  
15 cycle, the evolutionary procedure uses the whole training set of examples.  
16  
17

18 The genetic wrapper procedure is run for a pre-defined number of iterations. At the last  
19 generation, the population is evaluated and the fittest individual is picked as the final  
20 solution.  
21  
22

## 23 **5 Experimental Settings and Results**

24 Input data are normalised according to the Mean-Variance procedure. For each learning  
25 trial, the database is randomly partitioned into a training set including 80% of the  
26 examples (**185 data patterns**) and a test set including the remaining 20% (**47 data**  
27 **patterns**). The classifiers are trained on the former and the final learning result is  
28 evaluated on the latter. To reduce the danger of overfitting, the order of presentation of  
29 the training samples is randomly reshuffled for every learning cycle of the algorithm  
30 under evaluation.  
31  
32

33 Due to the unbalanced distribution of the wood veneer defect examples and the small  
34 size of some categories, the distribution of the training data set is balanced by  
35 duplicating randomly picked members of the smaller classes. **The balanced training**  
36 **set contains 208 data patterns, corresponding to 16 examples per class.**  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60



1  
2  
3  
4 The best performing ANN structure is experimentally found to have one hidden layer of  
5 30 units, a configuration very similar to the one suggested by Packianather et al. (2000).  
6  
7

8  
9 The learning parameters characterising the various algorithms are manually optimised  
10 according to experimental trial and error. Once the parameters are fixed, each algorithm  
11 is run and the classification accuracy of the final solution is estimated on the test set of  
12 examples. This procedure is repeated 20 times with different random initialisations. The  
13 learning accuracy of the final solutions is estimated as the average test accuracy of the  
14 20 independent learning trials.  
15  
16  
17  
18  
19  
20  
21  
22

23 Table 2 summarises the parameter settings for the different algorithms.  
24  
25

### 26 **5.1 ANN Training Algorithms Settings**

27

28  
29 The learning parameters of the *basicBP* and *ANNT* algorithms are experimentally set to  
30 maximise the learning results.  
31  
32

33  
34 The *valBP* algorithm is run until the classification accuracy stops improving. Since  
35 oscillations of the BP training process and noisy evaluations may lead to premature  
36 termination of the algorithm, the evolution of the classification accuracy is monitored  
37 using an average measure of the past readings. That is, every ten BP cycles the accuracy  
38 of the classifier is tested on the validation subset and the result is compared with the  
39 average of the last 20 readings. If the current accuracy measure is better than the  
40 average of the last 20 readings, the present MLP configuration is stored and the  
41 algorithm is continued. If no improvement is recorded, the algorithm is terminated and  
42 the best configuration of the previous evaluation is taken as the final solution. The  
43 monitoring process covers therefore a span of 200 learning cycles sampled every ten  
44 cycles. Experimental evidence shows that shorter monitoring periods are detrimental for  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

1  
2  
3  
4 the performance of the algorithm, while longer cycles don't bring appreciable  
5  
6 improvements of accuracy.  
7  
8  
9

## 10 **5.2 Feature Selection Algorithms Settings**

11  
12 Experimental tests show that the performance of the *FeaSANNT* algorithm is robust to  
13  
14 reasonable variations of the search parameters.  
15

16  
17 Due to the lengthy training of the MLP classifier, the genetic wrapper procedure is  
18  
19 limited to a smaller population than *FeaSANNT* and a reduced number of learning  
20  
21 cycles. The choice for the population size and the evolution length is the result of a  
22  
23 trade off between exploration of the search space and algorithm running time. To  
24  
25 sustain the exploration of the feature space, the genetic wrapper algorithm uses a higher  
26  
27 input vector mutation rate than *FeaSANNT*. Finally, the size of the MLP training and  
28  
29 fitness validation subsets used by the *valBP* algorithm are set to 80% and 20%,  
30  
31 respectively, of the training set of examples.  
32  
33  
34  
35  
36

## 37 **5.3 Control Algorithms – Learning Results**

38  
39 The results of the seven learning tests (*FeaSANNT* plus the six control tests) are  
40  
41 reported in Table 3. For each experiment, the table reports the mean and the standard  
42  
43 deviation of the MLP accuracy, the number of selected features and the duration of the  
44  
45 learning procedure. The number of learning cycles for the *valBP* algorithm is estimated  
46  
47 from the average duration of the 20 learning trials. Accuracy results report the  
48  
49 percentage of correctly classified examples of the test set.  
50  
51  
52

53  
54 The three tests conducted using the full feature set show a good performance of the  
55  
56 *basicBP* and the *ANNT* algorithms. Both the procedures give accurate and consistent  
57  
58  
59  
60

1  
2  
3  
4 learning results, with no appreciable differences between the basic gradient descent  
5 method and the more sophisticated EA.  
6  
7

8  
9 The performance of the *valBP* algorithm is unsatisfactory in terms of learning accuracy  
10 and robustness. On average, the duration of the *valBP* procedure is much shorter than  
11 the manually optimised duration of the *basicBP* procedure. Premature termination of the  
12 learning process is therefore a likely reason for the poor performance of the *valBP* rule.  
13  
14

15  
16 It is interesting to notice that the stopping criterion used by Packianather (1997) for his  
17 implementation of the BP algorithm is similar to the stopping criterion of the *valBP*  
18 algorithm. Comparison of the learning results shows that the accuracy of the two  
19 algorithms is equivalent.  
20  
21

22  
23 The learning trials carried out using the set of attributes suggested by Packianather  
24 (1997) confirm that there are redundant features that can be removed without affecting  
25 the ANN performance. The reduction of the feature set causes only a minor decrease of  
26 the speed of convergence and robustness of the MLP training procedures.  
27  
28

29  
30 The last control experiment regards the genetic wrapper procedure. Compared to the  
31 results obtained by the *valBP* algorithm using the full feature set, the wrapper algorithm  
32 noticeably improves the accuracy and the consistency of the solutions. This observation  
33 confirms the capability of EAs to cope with noisy evaluations of the population fitness.  
34  
35 However, compared to the results of the learning trials performed using the full feature  
36 set and the feature subset suggested by Packianather (1997), the genetic wrapper  
37 procedure is inferior in terms of robustness and learning accuracy.  
38  
39

40  
41 The genetic wrapper algorithm selects on average 11.35 attributes, a figure close to the  
42 configuration suggested by Packianather (1997). Fig. 3 shows a sample evolution curve  
43 for the population average of the size of the feature set. The measurements are averaged  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

1  
2  
3  
4 over ten sample learning trials. The plot shows a trend towards more compact input  
5  
6 vector configurations, even though oscillations can be noticed in the learning curve in  
7  
8 proximity of the end of the evolution period. This behaviour suggests that the duration  
9  
10 of the algorithm is probably too short to allow convergence to the minimal solution.  
11  
12

#### 13 14 15 **5.4 *FeaSANNT* – Learning Results**

16  
17 The proposed algorithm obtains an average **92.98%** classification accuracy on the  
18  
19 test set of examples (**98.36%** on the training set). This figure is comparable within  
20  
21 one standard deviation to the accuracy results achieved by all the other algorithms  
22  
23 but the *valBP* algorithm, which clearly underperforms. In order to assess the  
24  
25 significance of the accuracy differences measured between *FeaSANNT* and the  
26  
27 other algorithms, a set of ANOVA tests are used. Tables 4a-e report the F-test  
28  
29 statistics and the critical value for a 5% alpha level of significance of five ANOVA  
30  
31 test performed taking the learning results of *FeaSANNT* and the learning results of  
32  
33 respectively the *basicBP*, *ANNT*, *Filter+basicBP* and *Filter+ANNT* algorithms. The  
34  
35 *valBP* algorithm is excluded from the comparison since it is the only one that  
36  
37 clearly underperforms. The tests reported in tables 4a-e reveal that there are  
38  
39 statistically significant differences between the accuracy results obtained by  
40  
41 *FeaSANNT* and the genetic wrapper algorithm. In this case, it can be concluded  
42  
43 that the proposed algorithm is superior to the genetic wrapper algorithm in terms  
44  
45 of learning accuracy. In all the other cases, no statistically significant differences  
46  
47 are revealed and the algorithms can be considered equally performing in terms of  
48  
49 accuracy results.  
50  
51

52  
53  
54  
55  
56  
57 Tables 5a-b present two typical confusion matrices of the classification results  
58  
59 obtained by the MLP at the end of the *FeaSANNT* learning procedure. Table 5a  
60

1  
2  
3  
4 reports the classification results on the training set of examples, while Table 5b  
5 reports the classification results on the test set of examples. Both the matrices  
6 report classification results averaged over 20 learning trials. For the three classes  
7 curly grain, holes and worm holes, the number of training set samples includes the  
8 patterns duplicated following the data balancing procedure. The confusion  
9 matrices highlight the good generalisation capabilities of the classifier. The most  
10 frequent the classification mistakes on the test set concern the three under-  
11 represented classes (curly grain, holes and worm holes). There seems to be no  
12 strong indication of systematic classification errors. In the training set, the most  
13 common classification error is curly grain being identified as sound knots. In the  
14 test set, the two most common classification errors are bark being identified as  
15 rotten knots and roughness as streaks. Interestingly, analysis of the inter-class  
16 variation performed by Packianather (1997) identifies bark, rotten knots, streaks,  
17 and curly grain as the classes that are the hardest to separate from the others.

18  
19 The standard deviation of the classification accuracy of the solutions produced by  
20 *FeaSANNT* is comparable to the distribution of the solutions produced by the genetic  
21 wrapper algorithm. This figure is slightly larger than the spread of the accuracy of the  
22 solutions produced by the *basicBP* and the *ANNT* algorithms on the fixed sets of  
23 attributes (full feature set and filter-reduced set). This result is probably due to the larger  
24 solution space (features and weights) that *FeaSANNT* and the genetic wrapper algorithm  
25 search.

26  
27 In terms of reduction of the input features, *FeaSANNT* generates solutions using on  
28 average less than 8 attributes, thus outperforming the other feature selection methods.  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

1  
2  
3  
4 For each of the three feature selection algorithms, Table 6 breaks down the frequency of  
5 selection for the 17 attributes. The first row gives for each feature on/off values  
6 according to the results of the feature selection procedure used by Packianather (1997).  
7  
8  
9  
10  
11  
12 The other two rows report frequency values calculated out of the 20 learning trials.

13  
14 **Comparison of the feature selection results shows that no definite conclusions can**  
15 **be taken on the usefulness of many attributes. Some features are frequently**  
16 **selected by all the algorithms, namely features number 7, 9, 11 13 and 17, while**  
17 **other attributes such as number 1, 2 are frequently not selected. However, there**  
18 **many cases where the three algorithms disagree on the selection result. This**  
19 **behavior suggests the presence of redundant and overlapping attributes, which can**  
20 **be used interchangeably without altering the performance of the classifier. As a**  
21 **consequence, several combinations of different attributes generate equally**  
22 **descriptive feature sets. Since the three algorithms use different feature selection**  
23 **criteria, different choices of equally performing feature sets are possible. The**  
24 **hypothesis that the full feature set is composed of several redundant attributes is**  
25 **also supported by the fact that, despite considerably reducing the set of input**  
26 **features, none of the feature selection algorithms improves the accuracies obtained**  
27 **by training the neural network using the full attribute set. This result indicates the**  
28 **presence of unnecessary and overlapping attributes, rather than conflicting or**  
29 **harmful ones.**  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49

50  
51 Fig. 4 shows the evolution of the average classification accuracy of *FeaSANNT*  
52 population and of its fittest individual on the training set of examples. The evolution of  
53 the classification accuracy of the fittest individual on the test set is also reported. Fig. 5  
54 reports the evolution of the size of the feature vector of the fittest individual. In both  
55  
56  
57  
58  
59  
60

1  
2  
3  
4 cases, for a better understanding of the algorithm behaviour the learning process is  
5 monitored over 10000 generations. The plots refer to average values over ten  
6 independent learning trials that were run during the algorithm optimisation phase. The  
7 curves show the standard EA learning pattern marked by a brisk initial improvement of  
8 the population fitness followed by slow convergence to the optimum values.  
9  
10  
11  
12  
13  
14  
15  
16

## 17 **6 Computational Issues**

18  
19 Compared to the customary wrapper approach to evolutionary feature selection, the  
20 proposed embedded approach minimises the computational effort for training the ANN  
21 solutions. As a result, larger computational resources are dedicated to a more thorough  
22 exploration of the feature space. To prove this point, the computational complexity of  
23 *FeaSANNT* is estimated in comparison to the complexity of the standard genetic  
24 wrapper algorithm used in the tests of the previous section.  
25  
26  
27  
28  
29  
30  
31  
32  
33

34 The structure of the two algorithms is similar except for the training and the evaluation  
35 of the MLPs. The genetic wrapper algorithm fully trains an MLP classifier for each  
36 candidate feature vector configuration, while *FeaSANNT* concurrently evolves the input  
37 vector and the ANN weights. Both algorithms use the BP rule.  
38  
39  
40  
41  
42

43 The most expensive operation in the BP training of an MLP is given by the forward  
44 processing of the training information and the backward propagation of the output  
45 errors. For each training example, the cost of this operation can be approximated as two  
46 passes (forward and backward) of the input signal through the ANN. For each BP cycle,  
47 the cost for passing the whole training set of examples will be henceforth used as the  
48 unit of computational cost. Table 7 summarises the estimated costs for the two  
49 algorithms expressed in ANN passes of the training set.  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

1  
2  
3  
4 The *valBP* algorithm used by the genetic wrapper procedure trains an MLP until the  
5 classification accuracy stops improving. For each individual, the cost of this operation  
6 can be estimated by the sum of the costs for training the ANN and monitoring the  
7 evolution of its accuracy. These costs can be roughly deduced from the average duration  
8 of the *valBP* algorithm on the full data set (see Table 3). On average, the final MLP  
9 configuration is achieved after 783 learning cycles. The actual algorithm lasts 10 more  
10 cycles, that is, until the first time the classification accuracy is seen decreasing. Since  
11 the ANN is trained on a subset of examples containing 80% of the training data, the cost  
12 of fully training the MLP classifier amounts to  $(783+10) \times 0.8$  double (forward and  
13 backward) passes of the training set. In addition, there is the cost of monitoring every 10  
14 BP cycles the classification accuracy, that corresponds to  $78+1$  evaluations of the MLP  
15 performance. Each evaluation amounts to one forward pass of the validation subset of  
16 examples, which contains 20% of the training data. The total cost of monitoring the  
17 evolution of the classification accuracy amounts therefore to  $(78+1) \times 0.2$  passes of the  
18 training set. By multiplying this figure by the population size and by the number of  
19 evolutionary cycles, it is possible to calculate the total cost of the genetic wrapper  
20 procedure in terms ANN passes of the training set.

21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44 In the case of *FeaSANNT*, each solution has a 60% probability of undergoing one BP  
45 learning cycle. Each solution is evaluated once on the full set of training examples. On  
46 average, this amounts to 0.6 double (forward and backward) passes and one single pass  
47 of the full set of training examples per solution. Multiplied by the population size and  
48 by the average number of evolutionary cycles, this figure expresses the estimated  
49 computational cost of running *FeaSANNT* for the wood veneer defect identification  
50 problem.  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60



1  
2  
3  
4 The above calculations don't take into account lesser computational costs that affect  
5 negligibly the algorithm complexity, such as the cost for calculating the ANN weight  
6 updates at each BP cycle (wrapper and *FeaSANNT*) and the costs for the random genetic  
7 weight mutations (*FeaSANNT*).  
8  
9

10  
11  
12  
13 Table 7 shows that, given the current settings, *FeaSANNT* allows savings of roughly one  
14 third of the estimated computation cycles needed by the genetic wrapper algorithm.  
15  
16  
17

## 18 19 20 **7 Discussion**

21  
22 Experimental evidence shows that *FeaSANNT* is capable of robust learning results,  
23 characterised by high-performing solutions and a considerable reduction of the number  
24 of input features. **Robustness is defined as a solution that is not sensitive to large**  
25 **variations or changes in input parameters sometimes referred to as noise**  
26 **conditions. In the experiments conducted in this work, robustness is built into the**  
27 **FeaSANNT algorithm by virtue of the data patterns used for training the**  
28 **algorithm. As discussed in Section 2, virtually all classes are represented by 20**  
29 **examples with 185 data patterns used for learning as discussed in section 5. The**  
30 **accuracy of the solution when tested, indirectly represents the robustness of the**  
31 **algorithm as it is measuring the ability of the algorithm to classify "noisy" data**  
32 **input using the 47 data patterns.** On average, the evolved MLP classifiers recognise  
33 wood veneer defects with over 90% accuracy using less than half the full feature set.  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49

50 Compared to the currently used statistical feature selection method employed by  
51 Packianather (1997), *FeaSANNT* greatly reduces the feature set with no degradation of  
52 the ANN accuracy.  
53  
54  
55  
56

57 Compared to a standard genetic wrapper algorithm, *FeaSANNT* obtains superior results  
58 in terms of learning accuracy and reduction of input features. Two reasons explain the  
59  
60

1  
2  
3  
4 underperformance of the genetic wrapper algorithm, that is, inadequate exploration of  
5  
6 the feature space and unsatisfactory training of the MLP classifiers.  
7  
8

9 The poor exploration of the search space is related to the necessity of fully training an  
10  
11 MLP classifier for every evaluation of an input vector configuration. Given the  
12  
13 slowness of the BP rule, to maintain acceptable running times the genetic search must  
14  
15 be restricted to a smaller population size and a shorter evolution span. Nevertheless, the  
16  
17 genetic wrapper procedure implies larger computational overheads than the proposed  
18  
19 embedded algorithm. This problem is intrinsically related to the choice of the wrapper  
20  
21 approach. Faster ANN training methods can speed up the evaluation of the population  
22  
23 fitness. However, the same methods can likewise speed up the ANN training procedure  
24  
25 in an evolutionary embedded framework. Given equal genetic operators and ANN  
26  
27 training algorithms, an embedded feature selection approach such as *FeaSANNT* allows  
28  
29 considerable savings of computational resources. These resources can be allocated to a  
30  
31 more thorough exploration of the feature space or allow faster algorithm running times.  
32  
33  
34  
35  
36

37 The unsatisfactory performance of the ANN training procedure of the genetic wrapper  
38  
39 algorithm is due to the difficulty of setting a stopping criterion for the induction  
40  
41 algorithm. The results of the learning trials carried out using the full feature set and the  
42  
43 filter-reduced feature set confirm that different input vector configurations determine  
44  
45 different MLP learning speeds. Consequently, setting a fixed number of ANN training  
46  
47 cycles is a sub-optimal solution. At the same time, due to oscillations in the learning  
48  
49 process and noise in the monitoring of the learning curve, an adaptive stopping criterion  
50  
51 such as the one used for the *valBP* algorithm is difficult to optimise. Larger fitness  
52  
53 validation subsets would improve the evaluation of the solutions. Unfortunately, given  
54  
55  
56  
57  
58  
59  
60

1  
2  
3  
4 the limited database of wood veneer defect patterns, a large validation subset can't be  
5  
6 afforded without affecting the quality of the training process.  
7

8  
9 The optimisation of the ANN learning algorithm is a crucial issue in the design of  
10  
11 evolutionary wrapper algorithms. This problem doesn't arise if an embedded approach  
12  
13 is chosen, since the number of ANN training cycles coincides with the convergence of  
14  
15 the evolutionary procedure.  
16  
17

18  
19 Finally, none of the feature selection methods improves the MLP learning results  
20  
21 obtained using the full set of attributes. Indeed, analysis of the feature selection results  
22  
23 suggests the presence of superfluous attributes, rather than attributes that negatively  
24  
25 affect the separability of the classes. In such case, given that the *basicBP* algorithm or  
26  
27 *ANNT* are adequate to learn the task from the full attribute set, the removal of any  
28  
29 number of unnecessary features has no effect on the accuracy of the learning process.  
30  
31 However, by removing several superfluous features *FeaSANNT* allows to cut feature  
32  
33 extraction costs and image pre-processing times.  
34  
35  
36  
37

## 38 **8 Conclusions and Further Work**

39  
40  
41 The evolutionary *FeaSANNT* algorithm is applied to the selection of the feature set for  
42  
43 the MLP wood veneer defect classifier. The proposed algorithm concurrently performs  
44  
45 the two tasks of feature selection and ANN training. The novelty of the method lies in  
46  
47 the implementation of the embedded approach in an evolutionary feature selection  
48  
49 paradigm.  
50  
51

52  
53 Experimental evidence shows that *FeaSANNT* creates high-performing MLP solutions  
54  
55 with robust learning results **and good generalisation capabilities**. On average, the  
56  
57 proposed algorithm generates solutions using less than half the full feature set, thus  
58  
59  
60

1  
2  
3  
4 outperforming previous results based on statistical filtering of the data attributes.

5  
6  
7 *FeaSANNT* requires also lower design costs since feature selection is fully automated.

8  
9 Experimental tests show the superiority of *FeaSANNT* over a standard evolutionary  
10 wrapper feature selection algorithm. Analysis of the computational overheads prove  
11 that, when the same genetic operators and ANN training algorithms are used,  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
FeaSANNT is more economical than a genetic wrapper approach. Consequently, larger  
computational resources are allocated to the exploration of the feature space to obtain  
better learning results. The adoption of the embedded approach simplifies also the  
design of the evolutionary procedure, since it removes the need of setting the stopping  
criterion for the ANN learning algorithm.

Analysis of the patterns of selected features suggests the presence of overlapping or  
redundant attributes, rather than conflicting or harmful ones. This hypothesis is  
corroborated by the lack of observable improvements of the ANN classifiers accuracy  
following the application of the feature selection routines.

**Further work should address the investigation of other filter feature selection  
criteria such as mutual information or classifier-independent feature selection, and  
other non-evolutionary wrapper algorithms such as feedforward and backward  
wrapper methods (Blum and Langley, 1997).** By including the optimisation of the  
ANN structure in the evolutionary process, the solutions are expected to take further  
advantage from the reduction of the input vector. More compact solutions are likely to  
be obtained by tailoring the ANN structure to the evolved feature set. Further work in  
this direction is in progress and preliminary results are encouraging.

## References

Aboitiz, F., Mechanisms of adaptive evolution - Darwinism and Lamarckism restated, *Medical Hypotheses*, 1992, **38**(3), 194-202.

Blum, A. and Langley, P., Selection of Relevant Features and Examples in Machine Learning, *Artificial Intelligence*, 1997, **97**, 245-271.

Drake P.R., Packianather M.S., A decision tree of neural networks for classifying images of wood veneer, *International Journal Advanced Manufacturing Technology* 1998; **14**(4), 280–285.

Fogel, D.B., *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, 2000, (IEEE Press, New York)..

Huber, H.A., McMillin, C.W. and McKinney, J.P., Lumber Defect Detection Abilities of Furniture Rough Mill Employees, *Forest Products Journal*, 1985, **35**(11/12), 79-82.

Packianather, M.S., Design and Optimisation of Neural Network Classifiers for Automatic Visual Inspection of Wood Veneer, Ph.D. Thesis, Univ. of Wales, Cardiff, UK, 1997.

Packianather, M.S. and Drake, P.R., Neural Networks for Classifying Images of Wood Veneer. part 2, *International Journal Advanced Manufacturing Technology*, 2000, **16**, 424-433

Packianather, M.S., Drake, P.R. and Rowlands, H., Optimising the Parameters of Multilayered Feedforward Neural Networks through Taguchi Design of Experiments, *Quality and Reliability Engineering International*, 2000, **16**, 461-473

Pham, D. T. and Alcock, R. J., Automatic Detection of Defects on Birch Wood Boards, *Proceedings IMechE*, part E, 1996, **210**, 45-52.

1  
2  
3  
4 Pham, D.T. and Alcock, R.J., Recent Developments in Automated Visual Inspection of  
5 Wood Boards, *Advances in Manufacturing: Decision, Control and Information*  
6 *Technology*, Tzafestas ed., 80-88, 1999a, (Springler-Verlag, London).  
7  
8

9  
10  
11 Pham, D.T. and Alcock, R.J., Plywood Image Segmentation Using Hardware-Based  
12 Image Processing Functions, *Proceedings IMechE*, part B, 1999b, **213**, 431-434.  
13

14  
15  
16 Pham, D.T. and Alcock, R.J., Automated Visual Inspection of Wood Boards: Selection  
17 of Features for Defect Classification by a Neural Network. *Proceedings IMechE*, part E,  
18 1999c, **213**, 231-245.  
19  
20

21  
22  
23 Pham, D.T. and Liu, X., *Neural Networks for Identification, Prediction and Control*,  
24 1995, (Springler-Verlag, London).  
25  
26

27  
28 Polzleitner, W. and Schwingshagl, G., Real-Time Surface Grading of Profiled Wooden  
29 Boards, *Industrial Metrology*, 1992, **2**, 283-298.  
30  
31

32  
33 Rumelhart, D.E. and McClelland, J.L., *Parallel distributed processing: exploration in*  
34 *the micro-structure of cognition*, **1-2**, 1986, (Cambridge, MIT Press).  
35  
36

37  
38 Smith, M.G. and Bull, L., Feature Construction and Selection Using Genetic  
39 Programming and a Genetic Algorithm, *Proc. 6th European Conf. Genetic*  
40 *Programming*, Colchester, England, UK, 2003, 229-237.  
41  
42

43  
44 Thierens, D., Suykens, J., Vanderwalle, J. and De Moor, B., Genetic Weight  
45 Optimisation of a Feedforward Neural Network Controller, *Artificial Neural Networks*  
46 *and Genetic Algorithms*, Albrecht, R.F., Reeves, C.R., and Steele, N.C. eds., 1993,  
47 (Springler-Verlag, Wien), 658-663.  
48  
49

50  
51  
52 Vafaie, H. and DeJong, H., Genetic Algorithms as a Tool for Restructuring Feature  
53 Space Representations, *Proc. Int. Conf. on Tools with AI*, Herndon, VA, 1995, 8-11.  
54  
55  
56  
57  
58  
59  
60

1  
2  
3  
4 Yang, J. and Honavar, V., Feature Subset Selection Using a Genetic Algorithm, *Feature*  
5  
6  
7 *Extraction, Construction, and Subset Selection: A Data Mining Perspective*, Motoda H.  
8  
9 and Liu, H. eds., 1998, (Kluwer, New York), 1-19.

10  
11 Yao, X., Evolving Artificial Neural Networks. *Proc. IEEE*, 1999, **87**(9), 1423-1447.

12  
13 Zhang, P., Verma, B. and Kumar, K., Neural vs. Statistical Classifier in Conjunction  
14  
15 with Genetic Algorithm Based Feature Selection, *Pattern Recognition Letters*, 2005, **26**,  
16  
17 909-919.  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27

## 28 **Word count**

29  
30 **abstract 208 words**

31  
32 **article 6928 words (including references)**  
33  
34  
35  
36  
37

## 38 **LIST OF TABLES**

39  
40 **Table 1.** Statistical features.

41 **Table 2.** Parameter Setting of Multi-Layer Perceptron and Learning Algorithms.

42 **Table 3.** Experimental Results.

43 **Table 4.** ANOVA F-Test results.

44 **Table 5.** Confusion Matrices.

45 **Table 6.** Feature Selection Results

46 **Table 7.** Computational Complexity  
47  
48  
49  
50  
51

## 52 **LIST OF FIGURES**

53  
54 **Fig. 1.** The AVI System.

55 **Fig. 2.** FeaSANNT Architecture.

56 **Fig. 3.** Genetic Wrapper Algorithm – Input Vector Evolution Curve.

57 **Fig. 4.** FeaSANNT Evolution Curves – Classification Accuracy.

58 **Fig. 5.** FeaSANNT – Input Vector Evolution Curve.  
59  
60

Features	
1	Mean of grey levels ( $\mu$ )
2	Median of grey levels
3	Mode of grey levels
4	Standard deviation of the grey levels ( $\sigma$ )
5	Skewness of grey levels
6	Kurtosis of grey levels
7	Number of pixels with a grey level of less than or equal to 80
8	Number of pixels with a grey level of greater than or equal to 220
9	Grey level (p) for which there are 20 pixels below <sup>(a)</sup>
10	Grey level (s) for which there are 20 pixels above <sup>(b)</sup>
11	Histogram tail length on the dark side (q-p) <sup>(c)</sup>
12	Histogram tail length on the bright side (s-r) <sup>(d)</sup>
13	Number of edge pixels after thresholding a segmented window at mean value $\mu$ <sup>(e)</sup>
14	Number of pixels after thresholding at $\mu-2\sigma$
15	Number of edge pixels after thresholding at $\mu-2\sigma$ <sup>(e)</sup>
16	Number of pixels after thresholding at $\mu+2\sigma$
17	Number of edge pixels after thresholding at $\mu+2\sigma$ <sup>(e)</sup>
<p>(a) The grey level p is used as the lowest grey level to accommodate for potential noise pixels.            (b) The grey level s is used as the highest grey level to accommodate for potential noise pixels.            (c) q is the grey level below which there are 2000 pixels.            (d) r is the grey level above which there are 2000 pixels.            (e) the number of edge pixel is calculated after the application of a 3x3 Laplacian convolution mask to the thresholded sub-image.</p>	

**Table 1.** Statistical features.



Multi-Layer Perceptron	Settings				
Input nodes	17				
Output nodes	13				
Hidden layers	1				
Hidden nodes	30				
Act. function hidden nodes	Hyper-tangent				
Act. function output nodes	Sigmoidal				
Learning Algorithms	<i>basicBP</i>	<i>valBP</i>	<i>ANNT</i>	<i>wrapper</i>	<i>FeaSANNT</i>
Learning coefficient	0.1	0.1	n.a.	n.a.	n.a.
Momentum term	0.01	0.01	n.a.	n.a.	n.a.
Initial range MLP weights	[-0.05, 0.05]	[-0.05, 0.05]	[-0.05, 0.05]	n.a.	[-0.05, 0.05]
Trials	20	20	20	20	20
Generations	*	**	*	75	*
Population size	n.a.	n.a.	100	30	100
Feature mask crossover rate	n.a.	n.a.	n.a.	1.0	1.0
MLP weight mutation rate	n.a.	n.a.	0.25	n.a.	0.25
MLP weight mutation width	n.a.	n.a.	0.2	n.a.	0.2
BP mutation rate	n.a.	n.a.	0.6	n.a.	0.6
Feature mask mutation rate	n.a.	n.a.	n.a.	0.2	0.05
Masked features at start	n.a.	n.a.	n.a.	10% total	10% total
* experimentally optimised					
** on-line optimised					
n.a. not applicable					

**Table 2.** Parameter Setting of Multi-Layer Perceptron and Learning Algorithms.

Algorithm	Accuracy	Std. deviation	Features	Learning cycles
<i>basicBP</i>	94.89	2.33	17.00	4000
<i>valBP</i>	82.34	7.72	17.00	783
<i>ANNT</i>	93.30	2.70	17.00	4000
<i>Filter+basicBP</i>	93.51	3.42	11.00	6000
<i>Filter+ANNT</i>	94.79	3.42	11.00	5000
<i>wrapper</i>	89.68	3.98	11.35	75
<i>FeaSANNT</i>	92.98	3.91	7.95	9000

**Table 3.** Experimental Results.

Source of variation	df	SS	MS	F	Critical
Among	1.00	36.67	36.67	3.54	4.41
Within	38.00	393.84	10.36		
Total	39.00	430.51	47.03		

a) *FeaSANNT vs basicBP*

Source of variation	df	SS	MS	F	Critical
Among	1.00	1.02	1.02	0.09	4.41
Within	38.00	428.93	11.29		
Total	39.00	429.94	12.31		

b) *FeaSANNT vs ANNT*

Source of variation	df	SS	MS	F	Critical
Among	1.00	2.83	2.83	0.21	4.41
Within	38.00	512.22	13.48		
Total	39.00	515.05	16.31		

c) *FeaSANNT vs Filter+basicBP*

Source of variation	df	SS	MS	F	Critical
Among	1.00	32.71	32.71	2.43	4.41
Within	38.00	512.22	13.48		
Total	39.00	544.93	46.19		

d) *FeaSANNT vs Filter+ANNT*

Source of variation	df	SS	MS	F	Critical
Among	1.00	108.76	108.76	6.98	4.41
Within	38.00	591.90	15.58		
Total	39.00	700.66	124.33		

e) *FeaSANNT vs wrapper*

**Table 4. ANOVA F-Test results.**

class	1	2	3	4	5	6	7	8	9	10	11	12	13	Samples
1	15.9	0	0	0	0	0.05	0	0	0	0	0	0.05	0	16
2	0	16	0	0	0	0	0	0	0	0	0	0	0	16
3	0	0	16	0	0	0	0	0	0	0	0	0	0	16
4	0	0	0	15.95	0	0	0	0.05	0	0	0	0	0	16
5	0	0	0	0	15.9	0	0.05	0	0	0.05	0	0	0	16
6	0	0	0	0	0	15.95	0	0	0	0	0	0	0.05	16
7	0	0	0	0	0	0	15.4	0.05	0	0.55	0	0	0	16
8	0	0	0	0	0	0	0	15.95	0	0	0.05	0	0	16
9	0	0	0	0	0	0	0	0	16	0	0	0	0	16
10	0	0	0	0	0	0	0.2	0	0	15.8	0	0	0	16
11	0	0	0	0	0	0	0	1.05	0.05	0	14.9	0	0	16
12	0	0	0	0	0	0	0	0	0	0	0	16	0	16
13	0	0	0	0	0	0	0	0	0	0	0	0	16	16

a) Training Set

class	1	2	3	4	5	6	7	8	9	10	11	12	13	Samples
1	3.4	0	0	0	0.1	0.4	0	0	0	0	0	0.1	0	4
2	0	4	0	0	0	0	0	0	0	0	0	0	0	4
3	0	0	3.9	0	0	0	0.1	0	0	0	0	0	0	4
4	0	0	0.05	3.7	0	0.05	0	0.15	0	0	0.05	0	0	4
5	0	0	0	0	4	0	0	0	0	0	0	0	0	4
6	0.15	0	0	0	0	3.75	0	0	0	0	0.05	0	0.05	4
7	0	0	0	0	0	0	3.5	0.15	0.05	0.3	0	0	0	4
8	0	0	0	0	0	0	0	3.8	0	0	0.2	0	0	4
9	0.05	0	0	0	0	0	0	0	3.9	0	0	0.05	0	4
10	0	0	0	0	0.2	0	0	0.05	0	3.6	0.1	0	0.05	4
11	0	0	0	0	0	0	0.05	0.05	0.15	0	2.75	0	0	3
12	0.1	0	0	0	0	0.1	0	0	0.05	0	0	1.75	0	2
13	0.1	0	0	0	0	0.05	0	0	0	0	0	0.05	1.8	2

b) Test Set

Legend: 1 bark, 2 clear wood, 3 coloured streaks, 4 discoloration, 5 pin knots, 6 rotten knots, 7 roughness, 8 sound knots, 9 splits, 10 streaks, 11 curly grain, 12 holes, 13 worm holes

**Table 5.** *FeaSANNT* Learning Results: Confusion Matrices.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Wrapper	0.35	0.5	0.6	0.9	0.8	0.8	0.65	0.25	0.7	0.6	0.95	0.75	0.9	0.65	0.7	0.55	0.7
<i>FeaSANNT</i>	0.45	0.25	0.1	0.45	0.15	0.3	0.95	0.75	0.8	0.4	0.95	0.1	0.6	0	0.95	0.15	0.6
Filter	0	0	0	1	1	1	1	0	1	1	1	0	1	1	0	1	1

**Table 6.** Feature Selection Results

For Peer Review Only

	Population size	Passes (BP)	Passes (Fitness)	Passes per step	Evolution steps	Total passes
wrapper	30	$(783+10) \times 2 \times 0.8$	$(78+1) \times 0.2$	38 538	75	2 890 350
<i>FeaSANNT</i>	100	0.6x2	1	220	9000	1 980 000

**Table 7.** Computational Complexity

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

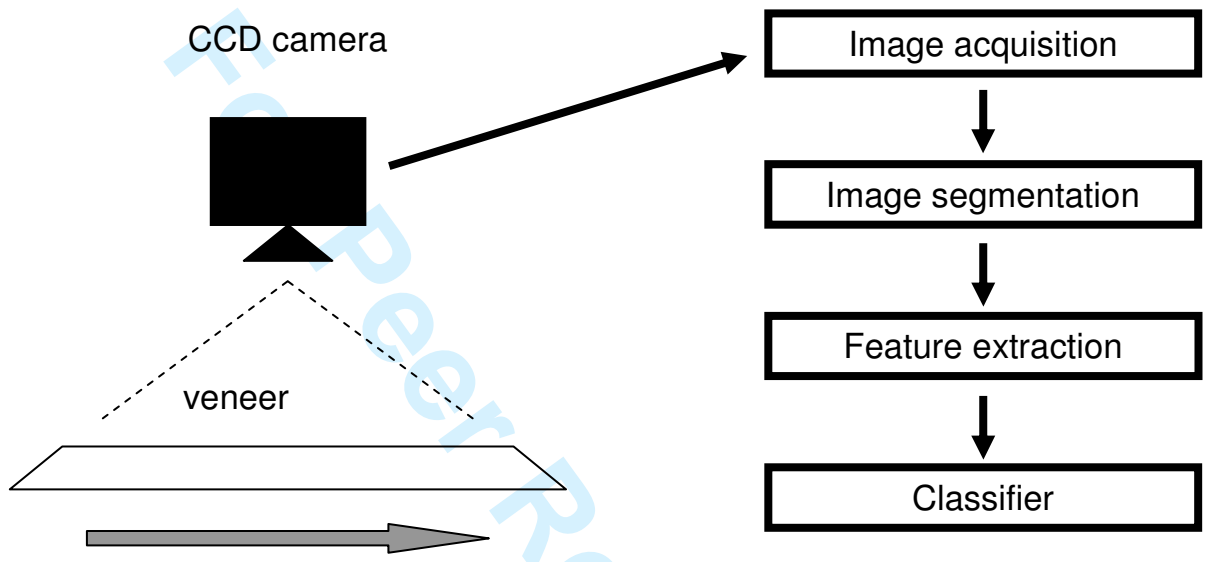


Fig. 1. The AVI System.

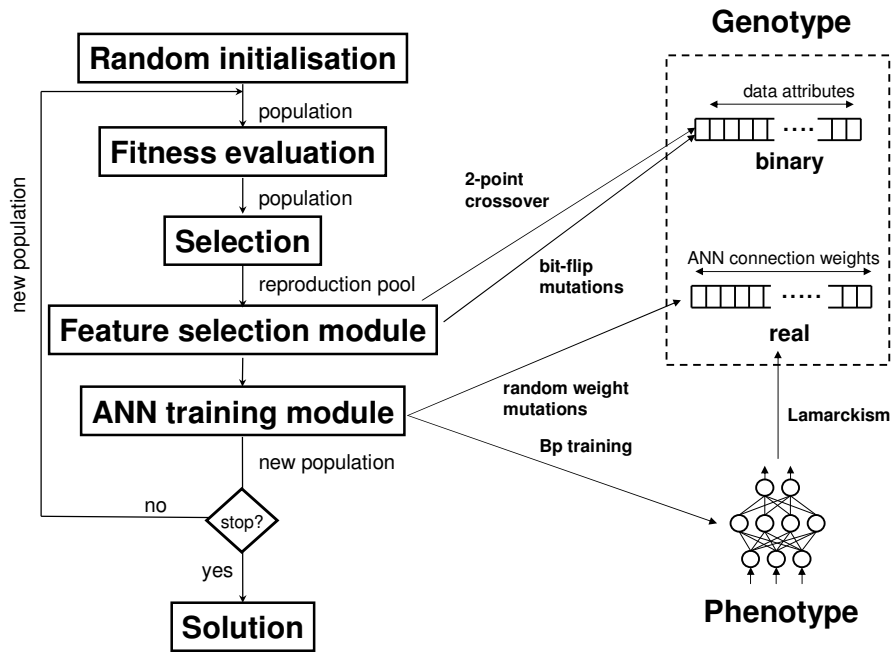


Fig. 2. FeaSANNT Architecture.



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

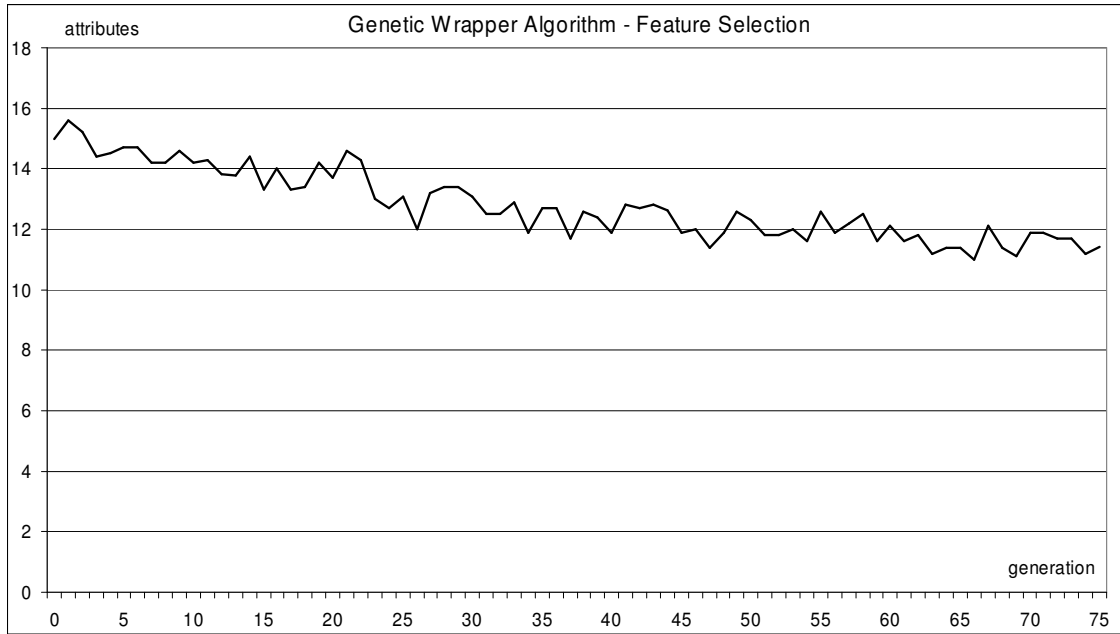
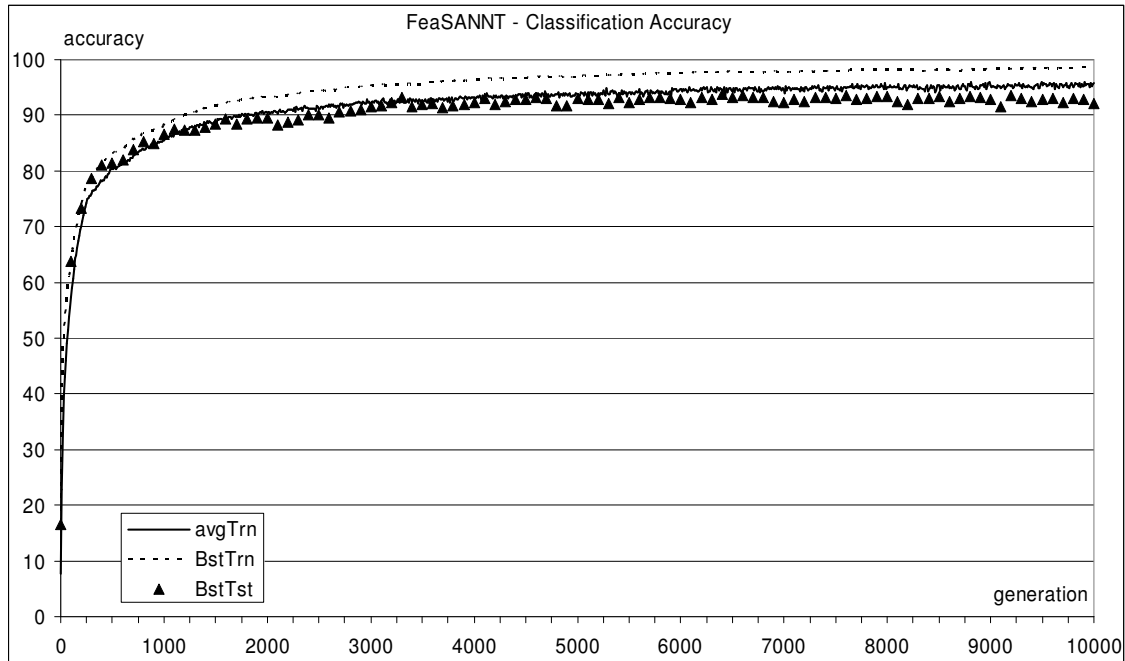


Fig. 3. Genetic Wrapper Algorithm – Input Vector Evolution Curve.



**Fig. 4.** FeaSANNT Evolution Curves – Classification Accuracy.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

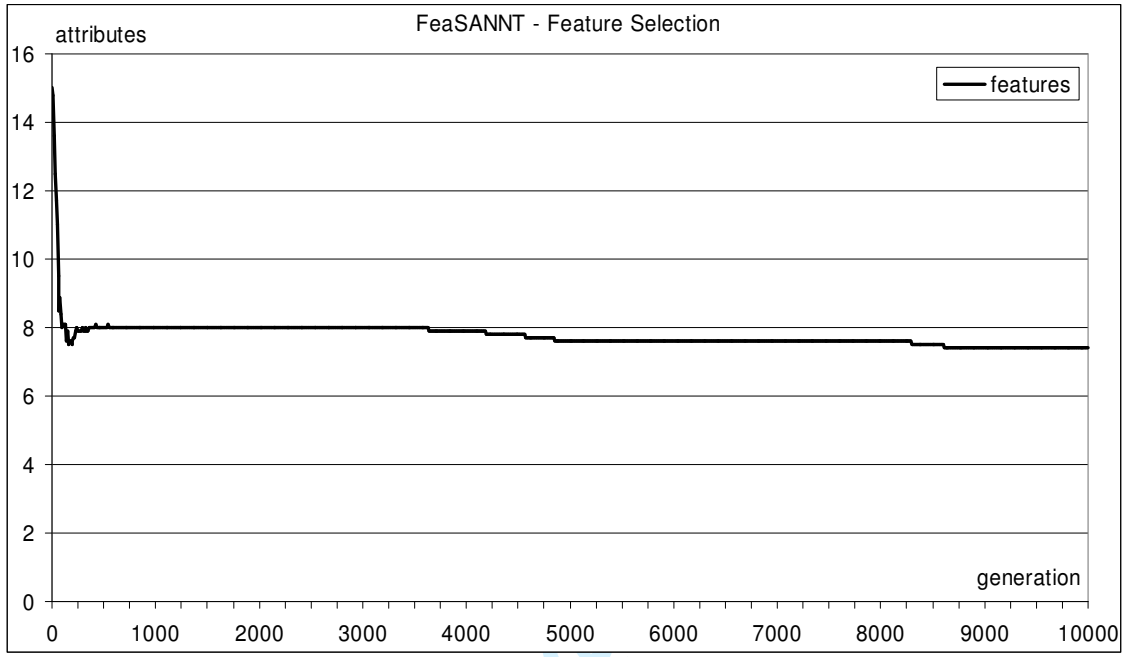


Fig. 5. FeaSANNT – Input Vector Evolution Curve.