

# Nearest Neighbor Based Pap-Smear Cell Classification Using Tabu Search for Feature Selection

Yannis Marinakis<sup>1</sup> and George Dounias<sup>2</sup>

<sup>1</sup> Department of Production Engineering and Management  
Technical University of Crete, University Campus, 73100 Chania  
Phone: +30-28210-37282, E-mail: marinakis@ergasya.tuc.gr

<sup>2</sup> Department of Financial & Management Engineering  
University of the Aegean, 31 Fostini Str., 82100 Chios, Greece  
Phone: +30- 22710-35454, E-mail: g.dounias@aegean.gr

**ABSTRACT:** The problem of classification consists of using some known objects, usually described by a large vector of features, to induce a model that classifies others into known classes. Selecting the right set of features for classification is one of the most important problems in designing a good classifier. In this paper, a tabu search algorithm is proposed for the solution of the feature selection problem. Tabu Search is a well known and effective metaheuristic algorithm that was first introduced for the solution of combinatorial optimization problems. This algorithm is combined with a number of nearest neighbor based classifiers. The algorithm is tested in two sets of data for Pap-Smear Cell Classification. The first one consists of 917 images of Pap smear cells and the second set consists of 500 images, classified carefully by cyto-technicians and doctors. Each cell is described by 20 features, and the cells fall into 7 classes but a minimal requirement is to separate normal from abnormal cells, which is a 2 class problem.

**KEYWORDS:** Tabu Search, Pap-Smear Cell Classification, Nearest Neighbor Classifiers, Feature Subset Selection Problem

## INTRODUCTION

In recent years, there has been an increasing need for novel data-mining methodologies that can analyze and interpret large volumes of data. Classification is one of the most frequently encountered decision making tasks of human activity and many problems in business, medicine, science and industry can be treated as classification problems. Selecting the right set of features for classification is one of the most important problems in designing a good classifier. The basic feature selection problem is an optimization problem, with a performance measure for each subset of features to measure its ability to classify the samples. The problem is to search through the space of feature subsets to identify the optimal or near-optimal one with respect to the performance measure. When classifying objects with a subset of features, the main goal is to obtain an accurate subset of features that allows classification within an acceptable margin of error. Thus, the goal is to maximize the classification performance of a given procedure over all possible subsets. In the literature many successful feature selection algorithms have been proposed. These algorithms can be classified into two categories based on whether features are done independently of the learning algorithm used to construct the classifier. If feature selection depends on learning algorithm, the approach is referred to as a wrapper model. Otherwise, it is said to be a filter model. Filters, such as mutual information (MI), are based on the statistical tools. Wrappers assess subsets of features according to their usefulness to a given classifier. Unfortunately, finding the optimum feature subset has been proved to be NP-hard [11]. Many algorithms are, thus, proposed to find the suboptimum solutions in comparably smaller amount of time [8]. The Branch and Bound approaches (BB) [14], the Sequential Forward/Backward Search (SFS/SBE) [1],[3] and the Filters approaches [4] deterministically search for the suboptimum solutions. One of the most important of the filter approaches is the Kira and Rendell's Relief algorithm [10]. Stochastic algorithms, including Simulated Annealing (SA) [16], Scatter Search algorithms [12] and Genetic Algorithms (GA) [4] are of great interest recently because they often yield high accuracy and are much faster.

In this paper, the Tabu Search approach is proposed for the solution of the Feature Subset Selection Problem. In the classification phase of the proposed algorithm, a number of variants of the Nearest Neighbor classification method are used [5]. In order to assess the efficacy of the proposed methodology, the algorithm is used for the Pap-Smear Cell

Classification task. The term ‘‘Pap-Smear’’ refers to samples of human cells stained by the so-called Papanicolaou method. The Papanicolaou method is a medical procedure to detect pre-cancerous cells in the uterine cervix. The performance of the proposed algorithm is tested using two data sets of images of Pap-smear cells distributed unequally on 7 different classes. Each cell is described by 20 features extracted from pictures of single human cells. The rest of the paper is organized as follows: In the next section, a detailed analysis of the proposed classification algorithm is presented. In section 3, an analytical description of the Pap-smear classification task is given. In section 4, the application of the proposed algorithm to the Pap-smear classification problem is presented while in the last section conclusions and future research are given.

## THE PROPOSED CLASSIFICATION ALGORITHM

In this paper, an algorithm is proposed that uses in the classification phase, the nearest neighbor classifiers and for the feature selection problem, the tabu search approach. Initially, the performance measures used and the description of the cross validation procedure used for testing the accuracy of the classifier are described. Afterwards, a detailed analysis of the nearest neighbor classifiers and of the tabu search method is given.

## PERFORMANCE MEASURES AND CROSS VALIDATION

In order to estimate the solution of the classifiers, a number of performance measures are calculated. In a **2-class problem** if the actual class of a sample is the  $N$ , the estimated class is denoted by  $T_N$  (True Negative) if the sample is classified in its actual class and  $F_P$  (False Positive) if the model misclassifies the sample. On the other hand, the estimated class is denoted by  $T_P$  (True Positive) when the sample is classified correctly in class  $P$  and by  $F_N$  (False Negative) if the sample is misclassified in the class  $N$ . In Table I the definitions are presented.

Estimated Class	Actual Class	
	$N$	$P$
$\hat{N}$	$T_N$	$F_N$
$\hat{P}$	$F_P$	$T_P$

Table I: Definitions of the classified and the misclassified samples.

The results of the models are analyzed based on the Root Mean Squared Error, the Error of Group 1, the Error of Group 2 and the Overall Error. The first measure is calculated from the formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^M (y_i - \hat{y}_i)^2}{M}} \quad (1)$$

where  $M$  is the number of samples in the data set, the  $\hat{y}_i$  is the classifier model output and the  $y_i$  is the true class of the test sample  $i$ . Denoting the number of the false negative samples by  $F_N$ , the number of the true negative samples by  $T_N$ , the number of false positive samples by  $F_P$  and the number of true positive samples by  $T_P$ , we define the following relative performance measures in percentage.

$$\text{The Error for Group 1:} \quad F_N \% = \frac{F_N}{T_P + F_N} 100 \quad (2)$$

$$\text{The Error for Group 2:} \quad F_P \% = \frac{F_P}{T_N + F_P} 100 \quad (3)$$

$$\text{The Overall Error:} \quad OE \% = \frac{F_N + F_P}{T_P + F_N + T_N + F_P} 100. \quad (4)$$

In a classification problem having more than two classes, the classification errors  $F_N\%$ ,  $F_P\%$  and  $OE\%$  using the definitions above do not make sense. The error of a  $C$  class problem can be described using a  $C \times C$  confusion matrix. The element  $x_{ij}$  in row  $i$  and column  $j$  describes the number of samples, of true class  $j$  classified as class  $i$ , i.e. all truly classified samples are placed in the diagonal and the remaining misclassified samples in the upper and lower triangular parts. The confusion matrix describes the number of errors, but an error rate is obtained by scaling each column to a

norm of 100%. Table II shows the confusion matrix of a 7-class problem where the classes 1,2, and 3 belong to class  $N$  and the classes 4, 5, 6, and 7 belong to class  $P$  and, thus, the classification errors  $F_N\%$ ,  $F_P\%$  and  $OE\%$  of the 2-class problem can easily be identified for the 7-class problem.

Estimated Class	Actual Class	
	1-3	4-7
1-3	$T_N$	$F_N$
4-7	$F_P$	$T_P$

Table II. Confusion matrix for a 7-class problem

**Cross validation** is an established technique for estimating the accuracy of a classifier. It is normally performed either using a number of random test/train partitions of the data or using  $s$ -fold cross validation. In this paper, in order to test the efficiency of the proposed method a  $s$ -fold validation procedure is utilized. Initially, the data set is divided into  $s$  disjoint groups containing approximately  $M/s$  samples each, where  $M$  is the number of the samples in the data set (for example if a 10-fold validation is used the data set is divided into 10 equal size disjoint groups). Next, each of these groups is systematically removed from the data set, a model is built from the remaining groups (the training set) and, then, the accuracy of the model is calculated using the hidden group (the test set). If the 10-fold validation is used, the procedure is repeated for ten times having different groups hidden, and, then, the average accuracies are measured.

## NEAREST NEIGHBOR CLASSIFIERS

Nearest Neighbor methods are among the most popular for classification [5]. They represent the earliest general methods proposed for this problem and were heavily investigated in the fields of statistics and pattern recognition. The nearest neighbor technique is a simple and appealing method to address classification problems. In this paper, a number of variations of the nearest neighbor rules are used. Initially, the classic **1 - Nearest Neighbor (1-nn)** method is used. The 1 - nn works as follows: In each iteration of the feature selection algorithm (section 2.3), a number of features are activated. For each sample of the test set its Euclidean Distance from each sample of the training set is calculated. The Euclidean Distance is calculated as follows:

$$D_{ij} = \sqrt{\sum_{l=1}^d |x_{il} - x_{jl}|^2} \quad (5)$$

where  $D_{ij}$  is the distance between the test sample  $x_{il}$  and the training sample  $x_{jl}$ , and  $l = 1, \dots, d$  is the number of activated features in each iteration. With this procedure the nearest sample from the training set is calculated. Thus, each test sample is classified in the same class that its nearest sample from the training set belongs to.

The previous approach may be extended to the **k-Nearest Neighbor (k-nn)** method, where we examine the  $k$ -nearest samples from the training set and, then, classify the test sample by using a voting scheme. The most common way is to choose the most representative class in the training set. Thus, the  $k$ -nn method makes a decision based on the majority class membership among the  $k$  nearest neighbors of an unknown sample. In other words every member among the  $k$  nearest has an equal percentage in the vote. However, it is natural to give more weight to those members that are closer to the test samples. This method is called **Weighted k Nearest Neighbor (wk-nn)**. In this method, the  $i$  neighbor receives weight

$$w_i = \frac{i}{\sum_{i=1}^k i} \quad (6)$$

Thus, the following hold:

$$w_k \geq w_{k-1} \geq \dots \geq w_1 > 0 \quad (7)$$

$$w_k + w_{k-1} + \dots + w_1 = 1. \quad (8)$$

## TABU SEARCH FOR THE FEATURE SUBSET SELECTION PROBLEM

The first metaheuristic used in this paper for the solution of the classification problem is called **Tabu-1nn**. The Tabu-1nn uses the Tabu Search method for solving the feature subset selection problem and the 1- Nearest Neighbor method as a classifier. **Tabu search (TS)** was introduced by Glover [6], [7] as a general iterative metaheuristic for solving combinatorial optimization problems. Computational experience has shown that TS is a well established approximation

technique, which can compete with almost all known techniques and which, by its flexibility, can beat many classic procedures. It is a form of local neighbor search. Each solution  $S$  has an associated set of neighbors  $N(S)$ . A solution  $S' \in N(S)$  can be reached from  $S$  by an operation called a *move*. TS moves from a solution to its best admissible neighbor, even if this causes the objective function to deteriorate. To avoid cycling, solutions that have been recently explored are declared *forbidden or tabu* for a number of iterations. The tabu status of a solution is overridden when certain criteria (*aspiration criteria*) are satisfied. Sometimes, *intensification* and *diversification* strategies are used to improve the search. In the first case, the search is accentuated in the promising regions of the feasible domain. In the second case, an attempt is made to consider solutions in a broad area of the search space.

In the following, an analytical description of the way the Tabu Search is implemented for the solution of the feature subset selection problem is given. The algorithm starts from an initial solution, i.e. an initial choice of the activated features. Feature selection vector is represented by a 0/1-bit string where 0 shows that the feature is not activated (not included in the solution) while 1 show that the feature is activated. Initially, only two features are activated and the RMSE of the solution is calculated with the 1-nn classifier, as described previously. Afterwards, a neighbor solution is generated. The neighbors are generated by randomly activated or deactivated a feature in the vector of features. Among the neighbors, the one with the best RMSE is selected and considered as a new current solution for the next iteration. Two restrictions in the implementation of the algorithm are used. The first one is that the feature vector is not allowed to have less than two features activated while the second one is the *Tabu Moves*. With the term Tabu Moves we mean that a tabu list is maintained to avoid returning to previously visited solutions. Thus, a feature that is added or deleted in this iteration from the solution, it is not allowed to return to the solution for a number of iteration equal with the size of the tabu list. The aspiration criterion that is used is a mechanism that overrides the tabu status of moves, meaning that if a move leads to a sufficient good solution even if it is tabu then the restriction of the tabu list is not activated and the move is allowed. We have, also, used an intensification and a diversification strategy. In the intensification strategy, the most promising regions of the search space are explored, i.e. the features with the less contribution in the quality of the solution are removed and the search is continued from the current solution. In the diversification strategy, the unexplored regions of the search space are explored in order to achieve a better solution. Two termination criteria are used. The first one is a prespecified maximum number of iterations and with the second termination criterion the algorithm stops when after some number of iterations there is not any improvement in the objective function value.

## PAP SMEAR CELL CLASSIFICATION

The metaheuristic algorithm presented in the previous sections is used for the classification of cells in cervical smears. Cervical smears are cytology specimens taken from the uterine cervix. A specimen is taken from the uterine cervix using a small brush, a cotton stick or wooden stick and transferred onto a thin, rectangular glass plate (slide). The specimen is stained using the Papanicolaou method which is a medical procedure to find pre-cancerous cells in the uterine cervix. This makes it possible to see characteristics of cells more clearly in a microscope. The purpose of the smear screening is to diagnose premalignant cell changes before they progress to cancer [2],[9],[13],[15]. The specimens most often contain cells from the *columnar epithelium* and the *squamous epithelium*. The columnar epithelium is located in the upper part of the cervix and the squamous epithelium in the lower part. Between these two is the *metaplastic epithelium*, also called the transformation zone or the **squamo-columnar junction**. It takes a skilled cytotechnician to differentiate between the different kinds of cells and it is a time consuming job as every glass slide can contain up to 300,000 cells. In the *squamous* epithelium there are 4 layers of cells: the basal, the parabasal, the intermediate, and finally the superficial layer. The *columnar* epithelium only contains a single layer of cells containing columnar cells and reserve cells. The *metaplastic* epithelium consists of reserve cells from the columnar epithelium. In *dysplastic* cells, the genetic information is somehow changed, and the cell will not divide as it should. This is a precancerous cell. Depending on which kind of cell that divides incorrectly, it is given diagnoses like *dysplasia* (i.e. disordered growth) and *carcinoma in situ*. The dysplastic cells are divided into *mild*, *moderate* and *severe* dysplastic.

## COMPUTATION RESULTS

### DATA AND PARAMETER DESCRIPTION

The classification task is performed using 2 databases, an old one with 500 cells and a new one with 917 cells, built by the Herlev University Hospital. The images were prepared and analyzed by the staff at the hospital using a commercial software package CHAMP (Dimac Imaging – [www.dimac-imaging.com](http://www.dimac-imaging.com)) for segmenting the images. The cells were

selected, not to collect a natural distribution, but to make a good collection of the important classes (7 classes). For both databases, 20 features are extracted (Table IV). The Pap-smear data fall into 7 classes (Table III), but a minimal requirement is to separate normal from abnormal, which is a 2-class problem.

<i>Old Data</i>	<i>New Data</i>
The old data contains 500 cells with the following distribution:	The new data contains 917 cells with the following distribution:
1. Normal - Columnar epithelial, 50 cells.	1. Normal - Super cial squamous epithelial, 74 cells.
2. Normal - Parabasal squamous epithelial, 50 cells.	2. Normal - Intermediate squamous epithelial, 70 cells
3. Normal - Intermediate squamous epithelial, 50 cells.	3. Normal - Columnar epithelial, 98 cells.
4. Normal - Super cial squamous epithelial, 50 cells.	4. Abnormal - Mild squamous non-keratinizing dysplasia, 182 cells.
5. Abnormal - Mild squamous non-keratinizing dysplasia, 100 cells.	5. Abnormal - Moderate squamous non-keratinizing dysplasia, 146 cells.
6. Abnormal - Moderate squamous non-keratinizing dysplasia, 100 cells.	6. Abnormal - Severe squamous non-keratinizing dysplasia, 197 cells.
7. Abnormal - Severe squamous non-keratinizing dysplasia, 100 cells.	7. Abnormal - Squamous cell carcinoma in situ intermediate, 150 cells.

Table III: Description of classes.

Feature number	Feature	Feature number	Feature
1	Nucleus area	11	Cytoplasm longest diameter
2	Cytoplasm area	12	Cytoplasm elongation
3	N/C ratio (Size of nucleus relative to cell size)	13	Cytoplasm roundness
4	Nucleus brightness	14	Nucleus perimeter
5	Cytoplasm brightness	15	Cytoplasm perimeter
6	Nucleus shortest diameter	16	Nucleus position
7	Nucleus longest diameter	17	Maxima in nucleus
8	Nucleus elongation	18	Minima in nucleus
9	Nucleus roundness	19	Maxima in cytoplasm
10	Cytoplasm shortest diameter	20	Minima in cytoplasm

Table IV: Features describing each cell.

It should be noted that from a medical point of view, it is worse to misclassify an abnormal cell as normal, than oppositely. Since we are looking for abnormal cells this is called a positive finding, while a normal cell is called a negative finding. An abnormal cell misclassified as normal is called a false negative finding, and it is important that a classifier minimizes the number of false negative. By analogy, a normal cell misclassified as abnormal is called false positive. In the proposed algorithms we tested different  $k$ 's. The first choice is the 2-fold validation where the set is divided in two equal parts. The other choices are the 3-fold (66% training set - 33% test set), 4-fold (75% - 25%), 5-fold (80% - 20%), 10-fold (90% - 10%) and the last is 20-fold (95% - 5%). The algorithm was implemented in Fortran 90 and was compiled using the Lahey f95 compiler on a Centrino Mobile Intel Pentium M 750 at 1.86 GHz, running Suse Linux 9.1. The maximum number of iterations for the tabu based metaheuristic is equal to 1000.

## 4.2 RESULTS OF 2-CLASS PROBLEM

Table V shows the errors (the Root Mean Square Error, the Error for group 1 and for group 2,  $F_N\%$  and  $F_P\%$ , and the Overall Error) for the proposed algorithms in the 2-class problem for both data sets. The best values of the errors for each data set in each Cross Validation are denoted with bold characters in the Table. We observe that the proposed algorithms give, for example, in the 2-fold Cross Validation, where the data set is divided in two equal parts and the half samples belongs to the training set while the other samples belongs to the test set, very good results as the RMSE is between 0,260023 - 0,284046 for the new data set and between 0,128387 - 0,154377 for the old data set. In this case, in the new data set although Tabu – 1nn has the best performance based on the RMSE, the  $F_P\%$  and the  $OE\%$ , the value of  $F_N\%$  is not as good as in the other methods, meaning is equal to 4,74119 in Tabu-1nn while for Tabu – w3nn is equal to 0,74184. As the value of  $s$  in the  $s$ -fold Cross Validation is increasing the results are improved taking into account the values of the four performance measures. For example, the best solution in the 10-fold Cross Validation has in the new data set the RMSE equal to 0,164553 and in the old data set has the RMSE equal to 0,02 while in the 20-fold Cross

Validation in the new data set the RMSE has been improved and is equal to 0,090369 and in the old data set in seven out of nine of the proposed methods all the performance measures are equal to 0. From this Table it can be observed that all methods give very good results and these results are almost identical for all methods. However, a higher performance is observed for the Tabu-1nn. It should, also, be noted that the value of the most significant performance measure, the  $F_N\%$ , is always smaller than 4,74119 and especially in 10-fold and 20-fold Cross Validation is smaller than 1 and for a number of methods is equal to 0.

	NEW DATA				OLD DATA			
	RMSE	$F_N\%$	$F_P\%$	$OE\%$	RMSE	$F_N\%$	$F_P\%$	$OE\%$
<i>2-fold Cross Validation (50% Training Set – 50% Test Set)</i>								
Tabu – 1nn	<b>0,260023</b>	<b>4,74119</b>	<b>12,39669</b>	<b>6,761186</b>	0,140705	1,333333	3	2
Tabu – 3nn	0,26204	2,37257	19,42149	6,870594	0,154377	2,666667	2	2,4
Tabu – 5nn	0,264153	2,075396	20,66116	6,979527	<b>0,128387</b>	<b>1,666667</b>	<b>2</b>	<b>1,8</b>
Tabu – 8nn	0,27024	0,889769	25,20661	7,3068	0,133956	1,333333	2,5	1,8
Tabu – 10nn	0,284046	0,889769	28,09917	8,069564	0,140705	1	3,5	2
Tabu – w3nn	0,278197	0,74184	27,27273	7,743005	0,154377	1,666667	3,5	2,4
Tabu – w5nn	0,266172	2,076712	21,07438	7,088697	0,140705	2	2	2
Tabu – w8nn	0,268251	0,740084	25,20661	7,197629	0,138438	2	2	2
Tabu – w10nn	0,266172	0,88933	24,38017	7,088697	0,138438	2,333333	1,5	2
<i>3-fold Cross Validation (66% Training Set – 33% Test Set)</i>								
Tabu – 1nn	<b>0,213638</b>	<b>2,222222</b>	<b>11,14712</b>	<b>4,579449</b>	0,132853	1,666667	1,997588	1,798812
Tabu – 3nn	0,240132	2,666667	14,44959	5,779135	0,138942	2,333333	1,500075	1,998413
Tabu – 5nn	0,228482	1,777778	14,86111	5,233758	0,113938	1	1,997588	1,398408
Tabu – 8nn	0,230258	0,888889	17,74691	5,342334	0,114771	0	4,975124	1,996008
Tabu – 10nn	0,229904	0,740741	18,15844	5,341976	0,145854	0,666667	4,492688	2,198014
Tabu – w3nn	0,237495	3,111111	12,79835	5,670917	0,132878	0,666667	3,505201	1,800014
Tabu – w5nn	0,230784	2,962963	11,98045	5,343048	0,139765	2	1,997588	1,998413
Tabu – w8nn	0,230784	2,666667	12,80864	5,343048	0,120027	1,333333	1,99005	1,598009
Tabu – w10nn	0,223177	1,925926	13,62654	5,015536	<b>0,089353</b>	<b>0,666667</b>	<b>1,99005</b>	<b>1,197605</b>
<i>4-fold Cross Validation (75% Training Set – 25% Test Set)</i>								
Tabu – 1nn	<b>0,208316</b>	<b>2,371267</b>	<b>9,911202</b>	<b>4,361116</b>	0,107967	0,666667	2	1,2
Tabu – 3nn	0,230544	2,369505	13,62705	5,342225	0,121066	1	2,5	1,6
Tabu – 5nn	0,230444	2,225099	14,05738	5,342225	<b>0,092713</b>	<b>1</b>	<b>1,5</b>	<b>1,2</b>
Tabu – 8nn	0,229618	1,038145	17,34973	5,3427	0,101975	0,333333	3	1,4
Tabu – 10nn	0,263987	1,333122	22,71858	6,978356	<b>0,092713</b>	<b>0,666667</b>	<b>2</b>	<b>1,2</b>
Tabu – w3nn	0,240204	2,074528	16,09973	5,778906	0,131443	0,666667	3,5	1,8
Tabu – w5nn	0,223262	2,520957	11,98087	5,015189	<b>0,092713</b>	<b>1,333333</b>	<b>1</b>	<b>1,2</b>
Tabu – w8nn	0,23288	2,373028	14,05055	5,451396	0,115074	1	2	1,4
Tabu – w10nn	0,230372	1,781312	15,28005	5,3427	<b>0,092713</b>	<b>1</b>	<b>1,5</b>	<b>1,2</b>
<i>5-fold Cross Validation (80% Training Set – 20% Test Set)</i>								
Tabu – 1nn	<b>0,203104</b>	<b>2,518518</b>	<b>8,690476</b>	<b>4,14469</b>	0,074641	0,666667	1,5	1
Tabu – 3nn	0,211221	1,333333	13,23129	4,470777	0,074641	1,333333	0,5	1
Tabu – 5nn	0,213751	1,62963	12,81463	4,579473	0,074641	1	1	1
Tabu – 8nn	0,226317	1,037037	16,55612	5,125327	0,094641	1,333333	1	1,2
Tabu – 10nn	0,221381	1,037037	15,71429	4,906747	0,094641	0,666667	2	1,2
Tabu – w3nn	0,228462	1,481481	15,7398	5,234616	0,108284	1,333333	2	1,6
Tabu – w5nn	0,216151	1,925926	12,38946	4,688762	<b>0,06</b>	<b>1</b>	<b>1</b>	<b>1</b>
Tabu – w8nn	0,213264	1,777778	12,43197	4,58066	0,074641	1	1	1
Tabu – w10nn	0,213758	1,037037	14,48129	4,580066	0,094641	1	1,5	1,2
<i>10-fold Cross Validation (90% Training Set – 10% Test Set)</i>								
Tabu – 1nn	<b>0,164553</b>	<b>1,18525</b>	<b>7,45</b>	<b>2,835643</b>	0,042426	0,666667	0,5	0,6
Tabu – 3nn	0,197351	0,443371	13,66667	3,927377	<b>0,02</b>	<b>0,333333</b>	<b>0,5</b>	<b>0,4</b>
Tabu – 5nn	0,197349	0,298507	14,08333	3,927377	0,028284	0,666667	0	0,4
Tabu – 8nn	0,223651	0,149254	18,61667	5,016722	0,028284	0	1	0,4
Tabu – 10nn	0,218049	0,149254	17,78333	4,798137	0,028284	0	1	0,4
Tabu – w3nn	0,225876	0,296313	18,61667	5,125418	0,028284	0	1	0,4
Tabu – w5nn	0,197683	0,59482	13,26667	3,927377	<b>0,02</b>	<b>0,666667</b>	<b>0</b>	<b>0,4</b>

Tabu – w8nn	0,208206	0,298507	15,73333	4,363354	0,028284	0,666667	0	0,4
Tabu – w10nn	0,208206	0,298507	15,73333	4,363354	0,034142	0,333333	1	0,6
<i>20-fold Cross Validation (95% Training Set – 5% Test Set)</i>								
Tabu – 1nn	<b>0,090369</b>	<b>0,748663</b>	<b>3,717949</b>	<b>1,52657</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
Tabu – 3nn	0,151264	0,151515	9,51923	2,615942	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
Tabu – 5nn	0,151818	0	9,51923	2,507246	0,01	0,333333	0	0,2
Tabu – 8nn	0,19207	0	14,45513	3,811594	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
Tabu – 10nn	0,18905	0	14,07051	3,705314	0,01	0	0,5	0,2
Tabu – w3nn	0,182943	0	13,17308	3,487923	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
Tabu – w5nn	0,156694	0,147059	10,32051	2,835749	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
Tabu – w8nn	0,156694	0	10,73718	2,835749	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
Tabu – w10nn	0,160979	0	10,73718	2,833333	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

Table V: Results of the algorithms in the 2-class problem

The selection of a set of appropriate input feature variables is an important issue in building a good classifier. The purpose of feature variable selection is to find the smallest set of features that can result in satisfactory predictive performance. Because of the curse of dimensionality, it is often necessary and beneficial to limit the number of input features in a classifier in order to have a good predictive and less computationally intensive model. In the Pap-smear cell classification problem analysed in this paper, there are  $2^{20}$  possible feature combinations. The objective of the computational experiments is to show the performance of the proposed algorithm in searching for a reduced set of features with low RMSE. In Table VI, the average number of features that each algorithm selects for both data bases are presented. It can be seen that in all cases the number of features selected are fewer than the total number of features used in order to describe a cell. For the new data set, the minimum average number of features used is equal to 6,5 observed in Tabu-1nn while the maximum average number of features used is equal to 12,33 observed in Tabu-w3nn and Tabu-w10nn. For the old data set, the minimum average number of features used is equal to 7,4 observed in Tabu-1nn while the maximum average number of features used is equal to 14,67 observed in Tabu-10nn.

	<i>2-fold</i>	<i>3-fold</i>	<i>4-fold</i>	<i>5-fold</i>	<i>10-fold</i>	<i>20-fold</i>
<b>NEW DATA</b>						
Tabu – 1nn	<b>6,5</b>	12	9,5	11,2	10,9	10,1
Tabu – 3nn	10,5	11	8,25	9	11,2	10,25
Tabu – 5nn	10	11	10,25	10	10,9	9,55
Tabu – 8nn	9	11,33	9,5	10,2	10	9,3
Tabu – 10nn	9	10,67	9,25	10,2	9,6	9,5
Tabu – w3nn	10,5	<b>12,33</b>	11	9,2	10,2	9,75
Tabu – w5nn	11,5	12	10	9,4	9,8	9,9
Tabu – w8nn	11	12	9,5	9	10,2	9,35
Tabu – w10nn	9	<b>12,33</b>	8,5	12,2	11,1	9,55
<b>OLD DATA</b>						
Tabu – 1nn	10,5	11	10,75	10	9,5	<b>7,4</b>
Tabu – 3nn	11	10,33	10,5	12,2	10,5	9,4
Tabu – 5nn	11	11	11,5	12,2	9,6	9,25
Tabu – 8nn	13,5	13	11,75	12,8	10,5	9
Tabu – 10nn	14	<b>14,67</b>	11,5	9,6	9,6	8,3
Tabu – w3nn	9,5	10,67	12,25	10,8	9,8	9,4
Tabu – w5nn	11	10,33	11,25	13,4	9,7	9,85
Tabu – w8nn	11	11,33	11	12	10,5	9,15
Tabu – w10nn	13	13,67	12	11,6	10,8	8,85

Table VI: Results of the algorithms (average number of features used) for the 2-class problem.

## RESULTS OF 7-CLASS PROBLEM

Table VII shows the errors (the Root Mean Square Error and the Overall Error) for the proposed algorithms in the 7-class problem for both data sets (as it has already been mentioned the  $F_N\%$  and the  $F_P\%$  do not make sense in the 7-class problem). The best values of the errors for each data set in each Cross Validation are denoted with bold characters in the Table. We observe that the proposed algorithms give, for example, in the 2-fold Cross Validation, where the data

set is divided in two equal parts and the half samples belongs to the training set while the other samples belongs to the test set, very good results as the RMSE is between 1,0416 – 1,116924 for the new data set and between 0,970155 – 1,297558 for the old data set. As the value of s in the s-fold Cross Validation is increasing the results are improved taking into account the values of the four performance measures. For example, the best solution in the 10-fold Cross Validation has in the new data set the RMSE equal to 0,905763 and in the old data set has the RMSE equal to 0,659465 while in the 20-fold Cross Validation in the new data set the RMSE has been improved and is equal to 0,776461 and in the old data set is equal to 0,487488. From this Table it can be observed that all methods give very good results and these results are almost identical for all methods. However, a higher performance is observed for the Tabu-1nn.

	NEW DATA		OLD DATA	
	RMSE	OE%	RMSE	OE%
<i>2-fold Cross Validation (50% Training Set – 50% Test Set)</i>				
Tabu – 1nn	1,097725	6,979527	<b>0,970155</b>	<b>2,4</b>
Tabu – 3nn	1,107526	7,743481	1,297558	4,8
Tabu – 5nn	1,075632	7,307037	1,049504	2,8
Tabu – 8nn	1,07402	7,088459	1,076184	3,2
Tabu – 10nn	1,064957	8,615416	1,032272	2,4
Tabu – w3nn	1,116924	8,615654	1,191824	3,2
Tabu – w5nn	1,093116	7,52514	1,067658	2,4
Tabu – w8nn	1,069031	8,180162	1,015378	2,6
Tabu – w10nn	<b>1,0416</b>	<b>6,761186</b>	1,027794	2,8
<i>3-fold Cross Validation (66% Training Set – 33% Test Set)</i>				
Tabu – 1nn	<b>0,998415</b>	<b>5,234115</b>	0,901895	2,397614
Tabu – 3nn	1,11935	7,851352	1,05101	3,399226
Tabu – 5nn	1,104866	6,978463	0,965862	2,796816
Tabu – 8nn	1,106326	6,871674	0,926921	2,199216
Tabu – 10nn	1,116397	7,087753	0,870997	2,199216
Tabu – w3nn	1,126345	7,634559	0,948671	2,799221
Tabu – w5nn	1,114004	6,869174	0,902844	2,397614
Tabu – w8nn	1,103898	6,434873	<b>0,851216</b>	<b>1,99721</b>
Tabu – w10nn	1,129295	7,633844	0,923595	2,198014
<i>4-fold Cross Validation (75% Training Set – 25% Test Set)</i>				
Tabu – 1nn	<b>0,962652</b>	<b>5,233529</b>	0,818468	1,4
Tabu – 3nn	1,070987	6,65132	0,922433	2,6
Tabu – 5nn	1,042917	6,870609	0,818359	1,6
Tabu – 8nn	1,022539	6,652269	0,802134	1,4
Tabu – 10nn	1,070661	7,087526	<b>0,794862</b>	<b>1,2</b>
Tabu – w3nn	1,043391	6,541675	0,98445	3,4
Tabu – w5nn	1,030383	6,215588	0,851283	1,8
Tabu – w8nn	1,034986	7,088475	0,844032	2
Tabu – w10nn	1,022443	6,215113	0,822172	1,6
<i>5-fold Cross Validation (80% Training Set – 20% Test Set)</i>				
Tabu – 1nn	0,977585	4,909717	0,828779	1,2
Tabu – 3nn	0,996275	5,779876	0,838493	1,4
Tabu – 5nn	1,00079	6,432644	0,743451	1,6
Tabu – 8nn	0,975546	5,887978	<b>0,727123</b>	<b>0,8</b>
Tabu – 10nn	<b>0,966245</b>	<b>5,669399</b>	0,731373	1,2
Tabu – w3nn	1,017272	5,88976	0,856501	2
Tabu – w5nn	1,000737	5,99608	0,814177	1,4
Tabu – w8nn	0,979248	6,105963	0,776908	1,6
Tabu – w10nn	0,980239	5,125921	0,741778	1
<i>10-fold Cross Validation (90% Training Set – 10% Test Set)</i>				
Tabu – 1nn	<b>0,905763</b>	<b>4,252269</b>	0,669753	0,6
Tabu – 3nn	0,929426	5,017917	0,760973	1,4
Tabu – 5nn	0,908669	4,363354	0,727018	1,4
Tabu – 8nn	0,916278	4,36216	0,722991	1,4
Tabu – 10nn	0,911791	4,359771	0,680138	0,8

Tabu – w3nn	0,912102	5,019111	0,753263	0,6
Tabu – w5nn	0,933394	4,910416	0,672604	1
Tabu – w8nn	0,922104	4,363354	0,663337	1
Tabu – w10nn	0,910576	4,142379	<b>0,659465</b>	<b>1</b>
<i>20-fold Cross Validation (95% Training Set – 5% Test Set)</i>				
Tabu – 1nn	<b>0,776461</b>	<b>3,048309</b>	<b>0,487488</b>	<b>0</b>
Tabu – 3nn	0,855476	4,140097	0,658098	0,4
Tabu – 5nn	0,821645	3,485507	0,595209	1
Tabu – 8nn	0,821604	4,031401	0,625383	0,6
Tabu – 10nn	0,82703	3,705314	0,627029	0,4
Tabu – w3nn	0,88653	4,036232	0,644507	0,2
Tabu – w5nn	0,817966	2,949275	0,599635	0,2
Tabu – w8nn	0,832524	3,821256	0,575656	0,2
Tabu – w10nn	0,806831	3,594203	0,592631	0,4

Table VII: Results of the algorithms in the 7-class problem

In the 7-class problem, the same objective with the one of the 2-class problem concerning the number of features selected (i.e., to have a high performance of the proposed algorithm in searching for a reduced set of features with low RMSE) is tested. In Table VIII, the average number of features that each algorithm selects for both data bases are presented. It can be seen that in all cases the number of features selected are fewer than the total number of features used in order to describe a cell. For the new data set, the minimum average number of features used is equal to 7.67 observed in Tabu-w10nn while the maximum average number of features used is equal to 12,67 observed in Tabu-1nn. For the old data set, the minimum average number of features used is equal to 8.67 observed in Tabu-3nn while the maximum average number of features used is equal to 14 observed in Tabu-5nn and Tabu-w8nn.

	<i>2-fold</i>	<i>3-fold</i>	<i>4-fold</i>	<i>5-fold</i>	<i>10-fold</i>	<i>20-fold</i>
<b>NEW DATA</b>						
Tabu – 1nn	12,5	<b>12,67</b>	10,5	11,6	10,5	10,15
Tabu – 3nn	9,5	10	9,5	9,6	10,6	11,3
Tabu – 5nn	9	9,33	9,25	9,2	11,2	9,9
Tabu – 8nn	8,5	8	9,5	11,4	10,1	9,85
Tabu – 10nn	8,5	8,33	8,25	11,8	8,7	9,8
Tabu – w3nn	10,5	12	13	10	11,7	10,8
Tabu – w5nn	9	9,5	10,25	9	11,1	11,05
Tabu – w8nn	8,5	10	10	11,4	10,8	10,2
Tabu – w10nn	9	<b>7,67</b>	10,75	11,6	9,8	10,3
<b>OLD DATA</b>						
Tabu – 1nn	10	13,7	11,3	10,2	10	9,15
Tabu – 3nn	12	<b>8,67</b>	12,3	13,6	10,9	10,1
Tabu – 5nn	<b>14</b>	13	13,3	13,4	11,2	9,9
Tabu – 8nn	12,5	13	12,3	12	11,3	9,4
Tabu – 10nn	11	9,33	13,3	13,8	11	10,4
Tabu – w3nn	12,5	11,7	10,3	10	10,8	10,3
Tabu – w5nn	11,5	11,7	11,3	10	11,3	9,25
Tabu – w8nn	<b>14</b>	11	<b>14</b>	12,4	12	10
Tabu – w10nn	13	13,3	12,5	12,8	11	10,2

Table VIII: Results of the algorithms (average number of features used) for the 7-class problem.

In Table IX, the times that each feature was selected in the optimal solutions of all algorithms are presented. The five most important features that are used in the algorithms for the two Datasets and for both 2-class and 7-class problems are typed with bold letters. As it can be seen, the three most important features are the third feature (N/C ratio (Size of nucleus relative to cell size)) as it was selected totally 1138 times, i.e. in the 71,84% in all solutions, the fifth feature (Cytoplasm brightness) as it was selected totally 1090 times, i.e. in the 68,81% in all solutions and the fourth feature (Nucleus brightness) as it was selected totally 962 times, i.e. in the 60,73% in all solutions. The five less important features that are used in the algorithms for the two Datasets and for both 2-class and 7-class problems are typed with italic letters. As it can be seen, the three less important features are the ninth feature (Nucleus roundness) as it was selected totally only 503 times, i.e. in the 31,75% in all solutions, the eighth feature (Nucleus elongation) as it was

selected totally 576 times, i.e. in the 36,36% in all solutions and the fiftieth feature (Cytoplasm perimeter) as it was selected 671 times, i.e. in the 42,36% in all solutions.

Features	<i>2-class problem</i>				<i>7-class problem</i>			
	NEW DATA		OLD DATA		NEW DATA		OLD DATA	
	Times Selected	Average (%)	Times Selected	Average (%)	Times Selected	Average (%)	Times Selected	Average (%)
1	<b>245</b>	<b>61,87</b>	<b>248</b>	<b>62,63</b>	229	57,83	193	48,74
2	189	47,73	212	53,54	197	49,75	212	53,54
3	<b>282</b>	<b>71,21</b>	<b>241</b>	<b>60,86</b>	<b>346</b>	<b>87,37</b>	<b>269</b>	<b>67,93</b>
4	<b>248</b>	<b>62,63</b>	211	53,28	<b>238</b>	<b>60,10</b>	<b>265</b>	<b>66,92</b>
5	<b>242</b>	<b>61,11</b>	<b>287</b>	<b>72,47</b>	<b>242</b>	<b>61,11</b>	<b>319</b>	<b>80,56</b>
6	203	51,26	226	57,07	182	45,96	210	53,03
7	<b>260</b>	<b>65,66</b>	206	52,02	<b>230</b>	<b>58,08</b>	224	56,57
8	123	31,06	168	42,42	135	34,09	150	37,88
9	78	19,70	165	41,67	120	30,30	140	35,35
10	183	46,21	150	37,88	210	53,03	212	53,54
11	150	37,88	201	50,76	187	47,22	224	56,57
12	197	49,75	188	47,47	153	38,64	217	54,80
13	177	44,70	160	40,4	<b>238</b>	<b>60,10</b>	<b>256</b>	<b>64,65</b>
14	228	57,58	106	26,77	218	55,05	170	42,93
15	175	44,20	158	39,9	187	47,22	151	38,13
16	158	39,90	<b>271</b>	<b>68,43</b>	152	38,38	<b>326</b>	<b>82,32</b>
17	219	55,30	168	42,42	188	47,47	166	41,92
18	214	54,04	214	54,04	223	56,31	202	51,01
19	192	48,48	170	42,93	191	48,23	187	47,22
20	202	51,01	<b>227</b>	<b>57,32</b>	219	55,30	191	48,23

Table IX: Results of the algorithms (times each feature was selected).

## CONCLUSIONS

In this paper, a classification algorithm is proposed for solving the Pap-smear cell classification problem. Different classifiers are used for the classification problem, based on the Nearest Neighbor classification rule (the 1-Nearest Neighbor, the k- Nearest Neighbor and the wk-Nearest Neighbor) and the Tabu Search approach is used for the Feature Selection Problem. The performance of the proposed algorithm is tested using two data sets of Pap-smear cells. The obtained results indicate the high performance of the proposed algorithm in searching for a reduced set of features (in almost all cases less than 50% of all features are used) with high accuracy and in achieving excellent classification of Pap-smear cells both in 2 classes and in 7 classes. Future research is intended to be focused in using different than the Nearest Neighbor classifiers and different algorithms for the Feature Selection Problem.

## REFERENCES

- [1] Aha D.W. and Bankert R.L., 1996, "A Comparative evaluation of sequential feature selection algorithms". In Artificial Intelligence and Statistics, Fisher D. and Lenx J.-H. (Eds.), Springer-Verlag, New York.
- [2] Byriel, J., 1999, "Neuro-fuzzy classification of cells in cervical smears". Master's Thesis, Technical University of Denmark: Oersted-DTU, Automation.
- [3] Cantu-Paz E., 2004, "Feature subset selection, class separability, and genetic algorithms". Genetic and Evolutionary Computation Conference, pp. 959-970.
- [4] Cantu-Paz E., Newsam S. and Kamath C., 2004, "Feature selection in scientific application". Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 788-793.
- [5] Duda, R.O. and Hart P. E., 1973, "Pattern Classification and Scene Analysis" John Wiley and Sons, New York.
- [6] Glover, F., 1989, "Tabu Search I", ORSA Journal on Computing, 1 (3), pp. 190-206.
- [7] Glover, F., 1990, "Tabu Search II", ORSA Journal on Computing, 2 (1), pp. 4-32.

- [8] Jain A. and Zongker D., 1997, "Feature selection: Evaluation, application, and small sample performance". IEEE Transactions on Pattern Analysis and Machine Intelligence, 19, pp. 153-158.
- [9] Jantzen, J., Norup, J., Dounias, G. and Bjerregaard B. (2006) "Pap-smear benchmark data for pattern classification". (submitted).
- [10] Kira K. and Rendell L., 1992, "A practical approach to feature selection". Proceedings of the Ninth International Conference on Machine Learning, Aberdeen, Scotland, pp. 249-256.
- [11] Kohavi R. and John G., 1997, "Wrappers for feature subset selection". Artificial Intelligence, 97, pp. 273-324.
- [12] Lopez F.G., Torres M.G., Batista B.M., Perez J.A.M. and Moreno-Vega. J.M., 2006, "Solving feature subset selection problem by a parallel scatter search". European Journal of Operational Research, 169, pp. 477-489.
- [13] Martin, E., 2003, "Pap-smear classification", Master's Thesis, Technical University of Denmark: Oersted-DTU, Automation,.
- [14] Narendra P.M. and Fukunaga K., 1977, "A branch and bound algorithm for feature subset selection". IEEE Transactions on Computers, 26(9), pp. 917-922.
- [15] Norup, J., 2005, "Classification of pap-smear data by transductive neuro-fuzzy methods", Master's Thesis, Technical University of Denmark: Oersted-DTU, Automation,.
- [16] Siedlecki W. and Sklansky J., 1988, "On automatic feature selection". International Journal of Pattern Recognition and Artificial Intelligence, 2(2), pp. 197-220.