Global Journal of Computer Science and Technology

Diagnosing Parkinson by using Artificial Neural Networks and Support Vector Machines

DAVID GIL A, MAGNUS JOHNSON B
[a]Computing Technology and Data Processing, University of Alicante, Spain
[b]Lund University Cognitive Science, Sweden

***Abstract*- Parkinson's Disease (PD) is the second most common neurodegenerative a²iction only surpassed by Alzheimer's Disease (AD). Moreover, it is expected to increase in the next decade with accelerating treatment costs as a consequence. This situation leads us towards the need to develop a Decision Support System for PD. In this paper we propose methods based on ANNs and SVMs to aid the specialist in the diagnosis of PD. Data recorded during 195 examinations carried out on 31 patients was used to verify the capacity of the proposed system. The results show a high accuracy of around 90%.**

Key words: Parkinson diagnosis, Parkinson, mental disorder, expert systems in medicine, artificial intelligence in medicine, artificial neural networks, support vector machines.

I. INTRODUCTION

Parkinson's Disease (PD) is the second most common neurodegenerative affliction after Alzheimer's disease (AD). Studies from Olmsted County (Mayo Clinic) [Elbaz et al., 2002] have computed the lifetime risk of developing Parkinson's disease to 2 percent for men and 1.3 percent for women. The greater incidence in men is repeatedly con¯rmed.

PD is a progressive neurological disorder characterised by tremor, rigidity, and slowness of movements. It is associated with progressive neuronal loss in the substantia nigra and other brain structures. Non-motor features, such as dementia and dysautonomia, occur frequently, especially in advanced stages of the disease. Diagnosis depends on the presence of two or more cardinal motor features such as rest tremor, bradykinesia, or rigidity [Hughes et al., 1992]. Functional neuroimaging holds the promise of improved diagnosis and allows assessment in early disease [Piccini and Whone, 2004]. Two studies draw attention to the di±culties in the diagnosis of the disease in the early stages [Tolosa et al., 2006]. In a prospective clinicopathological study, Rajput [Rajput et al., 1991] showed that initial clinical diagnosis within 5 years from the disease onset was correct in 65% of the cases. After a mean duration of 12 years, the ¯nal diagnosis of PD by the clinician was confirmed with autopsy in 76% of cases. Similarly, among 800 patients in the Deprenyl and Tocopherol Antioxidative Therapy for PD study with mild early parkinsonism [Jankovic et al., 2000] judged to have PD, 89% were later reported to have an alternative diagnosis on the basis of multi-factorial, clinical diagnostic criteria.

Having so many factors to analyze to diagnose PD, specialist normally makes decisions by evaluating the current test results of their patients. Moreover, the previous decisions made on other patients with a similar condition are also done by them. These are complex procedures, especially when the number of factors that the specialist has to evaluate is high (high quantity and variety of these data). For these reasons, PD diagnosis involves experience and highly skilled specialists.

The use of classifier systems in medical diagnosis is increasing gradually. Recent advances in the field of artificial intelligence have led to the emergence of expert systems and Decision Support Systems (DSS) for medical applications.
Moreover, in the last few decades computational tools have been designed to improve the experiences and abilities of doctors and medical specialists in making decisions about their patients. Without doubt the evaluation of data taken from patients and decisions of experts are still the most important factors in diagnosis. However, expert systems and different Artificial Intelligence (AI) techniques for classification have the potential of being good supportive tools for the expert. Classification systems can help in increasing accuracy and reliability of diagnoses and minimizing possible errors, as well as making the diagnoses more time efficient [Akay, 2008].

Some of the related work about using AI techniques to aid in PD diagnosis and other types of mental disorder are [Cohen, 1994] [Cohen, 1998] [BjÄorne and Balkenius, 2005] [Berdia and Metz, 1998] [Ivanitsky and Naumov, 2008] [Loukas and Brown, 2004].

Motivated by the usefulness of such an expert system or DSS, the aim of this work is to propose a method to aid the specialist in the diagnosis of PD, thus increasing accuracy and reducing costs. The quantity and variety of the data recorded during examinations makes AI tools useful to improve the final diagnosis. AI tools are also useful in retrospective studies. Nowadays such historical studies are easier, better and more precise due to the increased use of automated tools that allow storage and retrieval of large

volumes of medical data. The proposal is to build a system using Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs). These two classifiers, which are widely used for pattern recognition, should provide a good generalization performance in the diagnosis task. The usage of such classifiers would reinforce and complement the diagnosis of the specialists and their methods in the diagnosis tasks.

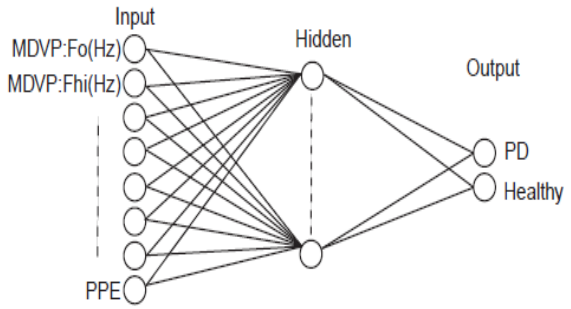

Fig.1. The architecture of the MLP network (input layer, hidden layer and output layer). The input layer represents the input data (the input data is described in section 4.1). The usage of a hidden layer enables the representation of data sets that are not linearly separable. The output layer represents the classification result. The weights and the threshold of the MLP are calculated during an adaptation process.

(ANNs) and Support Vector Machines (SVMs). These two classifiers, which are widely used for pattern recognition, should provide a good generalization performance in the diagnosis task. The usage of such classifiers would reinforce and complement the diagnosis of the specialists and their methods in the diagnosis tasks.

The remaining part of the paper is organized as follows: First, we explain the Parkinson data set used in the experimentation; second, we give a brief description of ANNs; Third, we describe the basic concepts of SVMs; Fourth, we describe our testing of the system and analyze the results; Finally, we draw
the relevant conclusions and suggest future lines of research.

II. Multilayer Perceptron

In this study we have used a Multi-Layer Perceptron (MLP) network with two layers. A two-layer MLP network is a fully connected feedforward neural network consisting of an input layer (which is not counted since its neurons are only for representation and thus do no processing), a hidden layer, and an output layer (healthy or ill) which represents the classification result (see figure 1) [Ripley, 1996] [Haykin, 1998][Bishop]. Each neuron (see figure 2) in the input and hidden layers is connected to all neurons in the next layer by weighted connections. These neurons (see figure 2) compute weighted sums of their inputs and adds a threshold. The resulting sums are used to calculate the activity of the neurons by applying a sigmoid activation function.

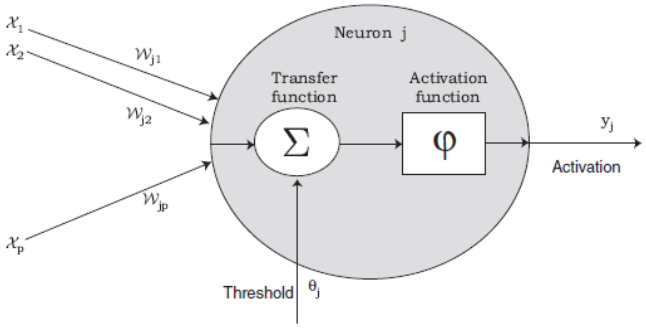

Fig. 2. A neuron in the hidden or output layer in the MLP. In the experimentation section the number hidden neurons in the MLP will be established.

This process is defined as follows:

$$\nu_j = \sum_{i=1}^{p} w_{ji} x_i + \theta_j \quad , \qquad y_j = f_j(\nu_j) \tag{1}$$

where V_j is the linear combination of inputs *x1; x2; :::; xp;* and the threshold θ_i, *wji* is the connection weight between the input *xi* and the neuron *j*, and *fj* is the activation function of the j_{th} neuron, and *yj* is the output. The sigmoid function is a common choice of the activation function. It is defined as:

$$f(t) = \frac{1}{1 + e^{-t}} \tag{2}$$

A single neuron in the MLP is able to linearly separate its input space into two subspaces by a hyper plane defined by the weights and the threshold. The weights define the direction of this hyper plane whereas the threshold term μj offsets it from origo.

The MLP network uses the backpropagation algorithm [Rumelhart et al., 1986], which is a gradient descent method, for the adaptation of the weights. This algorithm works as follows

The backpropagation MLP is a supervised ANN. This means the network is presented with input examples as well as the corresponding desired output. The backpropagation

algorithm adapts the weights and the thresholds of the neurons in a way that minimizes the error function E

$$E = \frac{1}{2}\sum_{p=1}^{n}(d_p - y_p)^2$$

where y_p is the actual output and d_p the desired output for input pattern p.

The minimization of E can be accomplished by gradient descent, i.e. the weights are adjusted to change the value of E in the direction of it's the negative gradient. The exact updating rules can be calculated by applying derivatives and the chain rule (for the weights between the input and the hidden layer).

III. SVM

In this section, the basic concepts of the SVM are described. More thorough descriptions can be found in [Burges, 1998] [Theodoridis and Koutroumbas, 2003] [Hsu et al., 2003]. A typical two class problem as the one shown in figure 3 is similar to the problem of diagnosing patients as either ill or healthy.

For a classification problem, it is necessary to first try to estimate a function

$$f : \Re^N \rightarrow \{\pm 1\}$$

using training data, which are l N-dimensional patterns x_i and class labels y_i, where $(x_1; y_1); :::; (x_l; y_l)$

$$\in \Re^N \times \{\pm 1\} \quad (4)$$

such that f will classify new samples $(x; y)$ correctly.

Given this classification problem the SVM classifier, as described by [Vapnik, 1995] [Guyon et al., 1992] [Cortes and Vapnik, 1995], satisfies the following conditions:

$$\begin{cases} \mathbf{w}^T\varphi(\mathbf{x}_i) + b \geq +1 & if \quad y_i = +1, \\ \mathbf{w}^T\varphi(\mathbf{x}_i) + b \leq -1 & if \quad y_i = -1, \end{cases} \quad (5)$$

which is equivalent to

$$y_i[\mathbf{w}^T\varphi(\mathbf{x}_i) + b] \geq 1, \quad i = 1, 2, ..., l. \quad (6)$$

Here training vectors x_i are mapped into a higher dimensional space by the function φ. The equations of (8) construct a hyper plane $\mathbf{w}^T\varphi(\mathbf{x}_i)+b = 0$ in this higher dimensional space that enables discrimination between the two classes (figure 3). Each of the two half-space defined by this hyper plane corresponds to one class, H_1 for y_i = +1 and H_2 for yi = -1. Therefore the SVM classifier corresponds to the decision functions:

$$y(\mathbf{x}) = sign[\mathbf{w}^T\varphi(\mathbf{x}_i) + b] \quad (7)$$

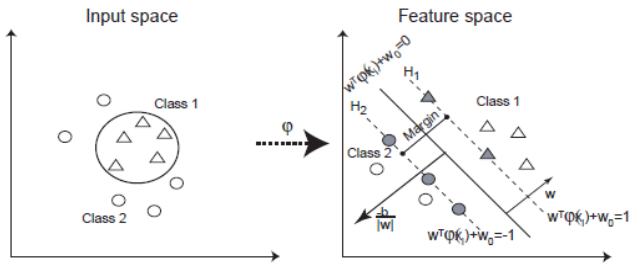

Fig.3.The mapping between input space and feature space in a two class problem with the SVM. Mapping the training data non-linearly into a higher dimensional feature space via function φ. $H1$ and $H2$ are parallel since they have the same normal w and perpendicular distance from the origin, $|\pm 1 - b|/\|\mathbf{w}\|$, and that no training points fall between them. The support vectors are the gray triangles and circles respectively located on H_1 and H_2. The distance from w to these support vectors is $1/||w||$ and the margin is simply $2=||w||$.

Thus the SVM finds a linear separating hyper plane with the maximal margin in this higher dimensional space. The margin of a linear classifier is the minimal distance of any training point to the hyper plane which is the distance between the dotted lines $H1$ and $H2$ and the solid line showed in figure 3. The points x which lie on the solid line satisfy $w^T \varphi(xi) + b = 0$, where w is normal to the hyper plane, $|b|/||w||$ is the perpendicular distance from the hyper plane to the origin, and $||w||$ is the Euclidean norm of w. $1/||w||$ is the shortest distance from the separating hyper plane to the closest positive (negative) example. Therefore, the margin of a separating hyper plane will be $1/||w||+1/||v||$. To calculate the optimal separating plane is equivalent to maximizing the separation margin or distance between the two dotted lines H_1 and H_2.

$H_1 : w^T \varphi(xi) + b = 1$ and $H_2 : wT \varphi(xi) + b = -1$ are parallel since they have the same normal w and perpendicular distance from the origin, $|1-b|/||w||$ for H_1 and $|-1-b|/||w||$ for H_2, and that no training points fall between them. Thus we expect the solution for a typical two dimensional problem to have the form shown in figure 3. Those training points which give equality in (9) are lying on one of the hyper planes H_1 and H_2 and are called support vectors. They are indicated in figure 3 by gray color.

For non-separable classes, the optimization process needs to be modified in an efficient and elegant manner. In mathematical terms, the maximal margin hyper plane for non-separable data is selected by minimizing the cost function:

$$J(\mathbf{w}|b, \xi) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \quad (8)$$

subject to the constraints:

$$y_i(\mathrm{w}^T\varphi(\mathrm{x}_i)+b)\geq 1-\xi_i,\ i=1,2,...,l. \quad (9)$$

Where,

$$\xi_i\geq 0,\quad i=1,2,...,l. \quad (10)$$

where the variables ξ_i are known as slack variables. Note that, the goal of the optimization task is to make the margin as large as possible and reduce the number of points with ξ > 0. The parameter *C* is a positive constant that controls the relative influence of the two competing terms.

When no linear separation of the training data is possible, SVM can work in combination with kernel techniques so that the hyper plane defining the SVM corresponds to a nonlinear decision boundary in the input space. If the data is mapped to some other (possibly infinite dimensional) Euclidean space using a mapping Φ(x), the training algorithm only depends on the data through dot products in such a Euclidean space, i.e. on functions of the form $\Phi(\mathrm{x})\cdot\Phi(\mathrm{x}')$.

If a kernel function *K* is defined as:

$$K(\mathrm{x},\mathrm{x}')=\Phi(\mathrm{x})\cdot\Phi(\mathrm{x}') \quad (11)$$

then, it is not necessary to know the Φ function during the training process. In the test phase, an SVM is used by computing dot products of a given test point x with *w*, or more specifically by computing the sign of:

$$f(\mathrm{x})=\sum_{i=1}^{s}y_i\Phi(s_i)\cdot\Phi(\mathrm{x})+b=\sum_{i=1}^{s}y_iK(s_i,\mathrm{x})+b \quad (12)$$

where *si* are support vectors.

Figure 3 shows the basic idea of the SVM in which the use of kernels in SVM enables the mapping of the data into some other dot product space (called feature space) F via a nonlinear transformation.

$$\Phi:\Re^N\rightarrow F \quad (13)$$

and perform the above linear algorithm in F. Note that, all the points belonging to a given class remain at a given side of the separating hyper plane and the data becomes linearly separable. In the input space, the hyper plane corresponds to a non-linear decision function whose form is determined by the kernel.

We now provide a description of the tools to construct nonlinear classifiers. We substitute $\Phi(\mathrm{x}_i)$ for each training example x_i, and perform the optimal hyper plane algorithm in *F*. Since we are using kernels, we will end up with a nonlinear decision function of the form.

$$y(\mathrm{x})=sign\left[\sum_{i=1}^{l}y_iK(\mathrm{x},\mathrm{x}')+b\right]. \quad (14)$$

There are many possibilities to define a function to be used as a Kernel. However, typical examples of kernels used in SVM, which have been successfully applied to a wide variety of applications, are linear, polynomial, radial basis functions and the hyperbolic tangent:

$$Linear\ Kernel: k(\mathrm{x},\mathrm{x}')=\mathrm{x}\cdot\mathrm{x}' \quad (15)$$

$$Polynomial\ Kernel: k(\mathrm{x},\mathrm{x}')=(\mathrm{x}\cdot\mathrm{x}'+c)^d \quad (16)$$

$$RBF\ Kernel: k(\mathrm{x},\mathrm{x}')=\exp\left(-\frac{\gamma(\mathrm{x}-\mathrm{x}')^2}{\sigma}\right) \quad (17)$$

$$Sigmoid\ Kernel: k(\mathrm{x},\mathrm{x}')=\tanh(\gamma(\mathrm{x}\cdot\mathrm{x}')+c) \quad (18)$$

Beside the possible kernels defined in this section there are others and much of the most current research is oriented to improve and to increase the efficiency of the SVM method. In subsections 4.3 and 4.4 the peculiarities of the kernels used in our experiments will be explained. These are the linear and the puk kernel.

IV. Experimentation

4.1 Parkinson data

The Parkinson database used in this study is taken from the University of California at Irvine (UCI) machine learning repository[Asuncion and New- man, 2007] [Little et al., 2007] [Little et al., 2008]. It was used for training and testing experiments. The reason to use these sets of data is that the data sets of this website have been donated from hospitals. These data have been studied by many professionals of artificial intelligence departments.The dataset is composed of a range of biomedical voice measurements from 31 people, 23 with PD. Each column in table 1 is a particular voice measure, and each row corresponds to one of 195 voice recordings of these individuals ("name" column). Table 1 shows the fields of this database and a brief description of each input variable.

Table 1

List of measurement methods applied to acoustic signals recorded from each subject.

Field name	Description
name	ASCII subject name and recording number
MDVP:Fo(Hz)	Average vocal fundamental frequency
MDVP:Fhi(Hz)	Maximum vocal fundamental frequency
MDVP:Flo(Hz)	Minimum vocal fundamental frequency
MDVP:Jitter(%)	Five measures of variation in fundamental frequency
MDVP:Jitter(Abs)	
MDVP:RAP	
MDVP:PPQ	
Jitter:DDP	
MDVP:Shimmer	Six measures of variation in amplitude
MDVP:Shimmer(dB)	
Shimmer:APQ3	
Shimmer:APQ5	
MDVP:APQ	
Shimmer:DDA	
NHR	Two measures of ratio of noise to tonal components in the voice
HNR	
RPDE	Two nonlinear dynamical complexity measures
D2	
DFA	Signal fractal scaling exponent
spread1	Three nonlinear measures of fundamental frequency variation
spread2	
PPE	
Status	Output - Health of the subject (1) - Parkinson's, (0) - healthy

In this study we have used the Weka program package. The Weka program package is a JAVA software package from the University of Waikato, New Zealand [Witten and Frank, 2005] issued under the GNU General Public License. This

software has been used and referenced in many works and projects [Massarelli et al., 2009] [Huang et al., 2009] [Ahmed et al., 2008] [Tari et al., 2008].

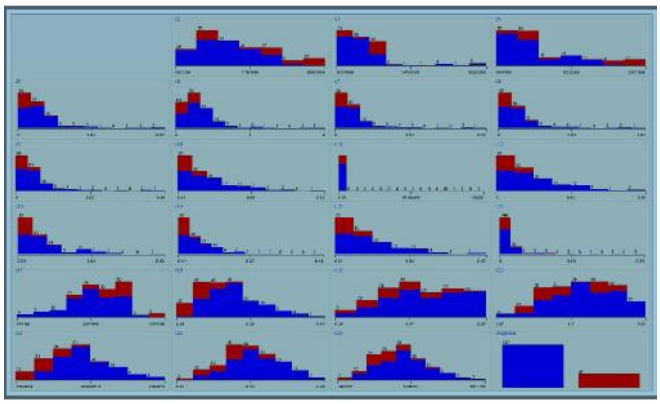

Fig.4. Relations input data and output (diagnosis). Every histogram in the above figure show the relation between each one of the input and output (the diagnosis) data. The goal is to provide a fast visual perception to appreciate the degree of influence between a specific input data, which has been measured by the specialists, and the final diagnosis.

The first step of the experimentation is to load the data. Figure 4 shows the relation between every input field and the diagnosis. That relation reflects just how small or big the influence between a specific input data and the final diagnosis is. The method to evaluate our system is to obtain some measures as classification accuracy, sensitivity, specificity, positive predictive value, negative predictive value and a confusion matrix. A confusion matrix [Kohavi and Provost, 1998] contains information about actual and predicted classifications done by a classification system.

4.2 MLP

In this section we test the Parkinson database by using MLP. For the construction of the architecture of the MLP we proceed as follows:

a) Layer 1 corresponds directly to the input vector, that is, all the parameters fields of the patient's record.

b) Layer 2 (the hidden layer). The number of hidden neurons for this layer is the most elaborated question in the network's architecture. This number represents a trade of between performance and the risk of over fitting. In fact, the number of neurons in a hidden layer will significantly influence the ability of the network to generalize from the training data to the unknown examples [Pal and of Nottingham , GB]. By doing some experiments we discovered that:

- With a low number of neurons for this layer training and test sets performed badly.
- With a high number of neurons the training set performed well. However there is a high risk of over fitting.
- The optimal solution for this layer was found to be 13 neurons.

Therefore, the best solution for this hidden layer has been found with 13 neurons.

c) Layer 3 (the output layer) (ill and healthy patients). Table 2, 3 and 4 show the confusion matrix for a two class classifier. Classification accuracy, sensitivity, specificity, positive predictive value and negative predictive value can be defined by using the elements of the confusion matrix.

Table 2

Definition of the confusion matrix with the value for every measure for the MLP classifier:

Actual	Predicted	
	Positive	Negative
Positive	True positive (TP)=138	False negative (FN)=9
Negative	False positive (FP)=6	True negative (TN)=42

Classif:
accuracy(%)

$$= \frac{TP+TN}{TP+FP+FN+TN} \text{x} 100 = 92.31\% \qquad (19)$$

Sensitivity(%)

$$= \frac{TP}{TP+FN} \text{x} 100 = 93.88\% \qquad (20)$$

Specificity(%)

$$= \frac{TN}{FP+TN} \text{x} 100 = 87.50\% \qquad (21)$$

Positive predictive value(%)

$$= \frac{TP}{TP+FP} \text{x} 100 = 95.83\% \qquad (22)$$

Negative predictive value(%)

$$= \frac{TN}{FN+TN} \text{x} 100 = 82.35\% \qquad (23)$$

4.3 SVM with linear kernel

In the next two sections we do some experimentation with the Parkinson database by using the SVM with different kernels in order to test the accuracy. The SVM produces better results than the MLP tested in the previous section. In particular we use a new algorithm for training the SVM: Sequential Minimal Optimization (SMO) which is a faster

training method for SVMs. SVMs have empirically been shown to have good generalization performance on a wide variety of problems. However, the use of SVMs is still limited to a small group of researchers. One possible reason is that training algorithms for SVMs are slow, especially for large problems. Another explanation is that SVM training algorithms are complex, subtle, and difficult for an average engineer to implement. Training a SVM requires the solution of a very large Quadratic Programming (QP) optimization problem. SMO breaks this large QP problem into a series of smallest possible QP problems [Platt, 1998] [Platt, 1999] [Keerthi et al., 2001]. This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default.

Table 3
Definition of the confusion matrix with the values for every measure of the SVM classifier and linear kernel.

Actual	Predicted	
	Positive	Negative
Positive	True positive (TP)=146	False negative (FN)=1
Negative	False positive (FP)=15	True negative (TN)=33

Classif: accuracy(%)

$$= \frac{TP+TN}{TP+FP+FN+TN} \text{x} 100 = 91.79\% \quad (24)$$

Sensitivity(%)

$$= \frac{TP}{TP+FN} \text{x} 100 = 99.32\% \quad (25)$$

Specificity(%)

$$= \frac{TN}{FP+TN} \text{x} 100 = 68.75\% \quad (26)$$

Positive predictive value(%)

$$= \frac{TP}{TP+FP} \text{x} 100 = 90.68\% \quad (27)$$

Negative predictive value(%)

$$= \frac{TN}{FN+TN} \text{x} 100 = 97.06\% \quad (28)$$

4.4 SVM with puk kernel

In this section we test the parkinson database with the SVM method by using an universal Pearson VII function based kernel (puk kernel) [ÄUstÄun et al., 2006]. This new method improves the accuracy of our system.

The applicability, suitability, performance and robustness of this alternative kernel in comparison to the commonly applied kernels is investigated by applying this to simulated as well as real-world data sets. From the outcome of these examinations, it was concluded that the PUK kernel is robust and has an equal or even stronger mapping power as compared to the standard kernel functions leading to an equal or better generalization performance of SVMs. In general, PUK can be used as a universal kernel that is able to serve as a generic alternative to the common linear, polynomial and RBF kernel functions [ÄUstÄun et al., 2007].

Table 4
Definition of the confusion matrix with the value for every measure of the SVM classifier and puk kernel.

Actual	Predicted	
	Positive	Negative
Positive	True positive (TP)=139	False negative (FN)=8
Negative	False positive (FP)=5	True negative (TN)=43

Classif: accuracy(%)

$$= \frac{TP+TN}{TP+FP+FN+TN} \text{x} 100 = 93.33\% \quad (29)$$

Sensitivity(%)

$$= \frac{TP}{TP+FN} \text{x} 100 = 94.56\% \quad (30)$$

Specificity(%)

$$= \frac{TN}{FP+TN} \text{x} 100 = 89.58\% \quad (31)$$

Positive predictive value(%)

$$= \frac{TP}{TP+FP} \text{x} 100 = 96.53\% \quad (32)$$

Negative predictive value(%)

$$= \frac{TN}{FN+TN} \text{x} 100 = 84.31\% \quad (33)$$

One of the reasons why such a high degree of accuracy is obtained, is due to the data cleaning procedure; the data preprocessing. The data preprocessing of the databases collected directly from a hospital or Health centers, is necessary in order to homogenize the data before applying them to artificial intelligence methods. Moreover, it increases the accuracy of the classification methods.

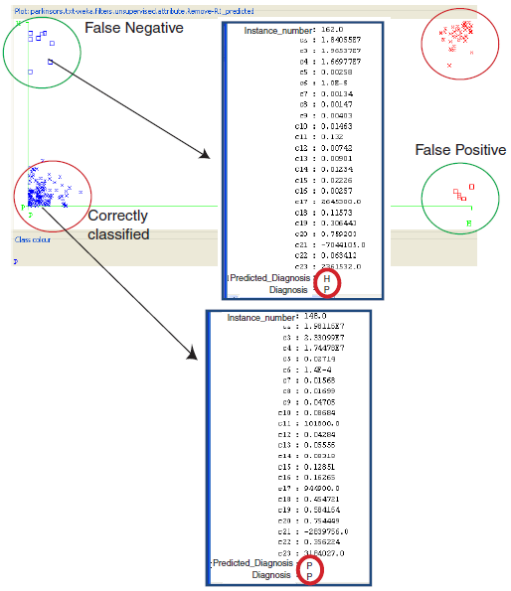

Fig. 5. The classifier errors. The two errors showed in the figure are the two instances of the Parkinson class. One of them is correctly classified (predicted Diagnosis and the real diagnosis coincide). However, the other one does not (predicted Diagnosis indicates H - Healthy whereas the real diagnosis is P - Parkinson). This is a false negative and the goal is to learn of these errors, try to find out what fields are implicating in that and to make the system robust.

By using these three types of tools, not only will it be possible to make comparisons between them to see which present the higher precision but also to complement them. In particular, MLP has a high value of "Positive predictive value" equal to 95.83%, SVM with linear kernel has the highest value of "Sensitivity" equal to 99.32% and "Negative predictive value" equal to 97.06%. Finally, SVM with kernel puk presents the highest values of "Classification of accuracy" equal to 93.33% and "Positive predictive value" equal to 96.53%.

V. Conclusions and Future Work

In this paper we have evaluated the performance of a classifier constructed by means of ANN and SVM.

The results presented by these three methods (MLP and SVM with the two kernel types) have both a high precision level of the confusion matrix regarding the different measurement parameters (accuracy, sensitivity, specificity, positive predictive value and negative predictive value).

The accuracy of the ANN and SVM were very good. They showed a high degree of certainty, above 90%. Furthermore, some of the parameters reach very high accuracy such as "Sensitivity" and "Negative predictive value" with 99.32% and 97.06% respectively.

Consequently, we propose a hybrid system combining ANN and SVM classifiers (SVM is tested with two different kernels). The goal is not only to establish a comparison between all of them but also to benefit from the highest accuracies of each classifier. The diagnosis must be reinforced and complemented in order to provide a better generalization in the same way that two or more specialists (or a specialist group) co-operate with each their methods, in order to obtain a final common diagnosis. As illustrated in Figure 5, our system allows finding out which instances are correctly or incorrectly classified. A future line of the system is an exhaustive study of all the fields, thus allowing us to determine why the errors occurred, and learning how to avoid this from happening in the future.

We have found that the outliers and the imbalanced data directly affected the classification performance and effectiveness of the classifiers. There are 147 registers with PD and 48 healthy ones. The accuracy of the classifiers will be improved by eliminating a number of outliers from both the minority and majority classes, and increasing the size of the minority class to the same size of the majority class.

Once the AI methods have been separately and/or individually tested, the next step will be to use a clustering method also called a metalearning. Metalearning algorithms take classifiers and turn them into more powerful learners with a higher generalization degree. They carry out the classifications either by averaging probability estimation or by voting and they always take advantage of every particular method.

VI. Acknowledgment

We want to express our acknowledgements to Max Little of the University of Oxford, who has created the database, in collaboration with the National Centre for Voice and Speech, Denver, Colorado, who recorded the speech signals. The original study published the feature extraction methods for general voice disorders.

References

B.A. Ahmed, M.E. Matheny, P.L. Rice, J.R. Clarke, and O.I. Ogunyemi. A comparison of methods for assessing penetrating trauma on retrospective multi-center data. *Journal of Biomedical Informatics*, 2008.

M.F. Akay. Support vector machines combined with feature selection for breast cancer diagnosis. *Expert Systems With Applications*, 2008.

A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. URL http://www.ics.uci.edu/$\sim$mlearn/{MLR}epository.html.

S. Berdia and JT Metz. An arti¯cial neural network stimulating performance of normal subjects and schizophrenics on the Wisconsin Card Sorting Test. *Artificial Intelligence in Medicine*, 13(1-2):123{138, 1998.
CM Bishop. 1995, Neural Networks for Pattern, Recognition, Oxford: Oxford
University Press.
P. BjÄorne and C. Balkenius. A model of attentional impairments in autism: first steps toward a computational theory. *Cognitive Systems Research*, 6: 193{204, 2005.
C.J.C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121{167, 1998.
I. Cohen. Neural network analysis of learning in autism. *Neural Networks and Psychopathology: Connectionist Models in Practice and Research*, page 274, 1998.
IL Cohen. An arti¯cial neural network analogue of learning in autism. *Biol Psychiatry*, 36(1):5{20, 1994.
C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273{297, 1995.
A. Elbaz, J.H. Bower, D.M. Maraganore, S.K. McDonnell, B.J. Peterson, J.E.
Ahlskog, D.J. Schaid, and W.A. Rocca. Risk tables for parkinsonism and Parkinson's disease. *Journal of clinical epidemiology*, 55(1):25{31, 2002.
I. Guyon, B. Boser, and VN Vapnik. A training algorithm for optimal margin classifiers. In *Proc. of the 5th annual workshop of computational learning theory, ACM*, pages 144{152, 1992.
S. Haykin. Neural Networks: A Comprehensive Foundation, Englewoods Cli®s, 1998.
C.W. Hsu, C.C. Chang, C.J. Lin, et al. A practical guide to support vector classification, 2003.
S.H. Huang, L.R. Wulsin, H. Li, and J. Guo. Dimensionality reduction for knowledge discovery in medical claims database: Application to antidepressant medication utilization study. *Computer Methods and Programs in Biomedicine*, 93(2):115{123, 2009.
AJ Hughes, SE Daniel, L. Kilford, and AJ Lees. Accuracy of clinical diagnosis of idiopathic Parkinson's disease: a clinico-pathological study of 100 cases. *British Medical Journal*, 55(3):181{184, 1992.
GA Ivanitsky and RA Naumov. Recognition of ongoing mental activity with artificial neural network. *International Journal of Psychophysiology*, 69(3): 180{180, 2008.
J. Jankovic, A.H. Rajput, M.P. McDermott, and D.P. Perl. The evolution of diagnosis in early Parkinson disease, 2000.
SS Keerthi, SK Shevade, C. Bhattacharyya, and KRK Murthy. Improvements to Platt's SMO algorithm for SVM classi¯er design. *Neural Computation*, 13(3):637{649, 2001.
R. Kohavi and F. Provost. Glossary of terms. *Machine Learning*, 30(2/3): 271{274, 1998.
M.A. Little, P.E. McSharry, S.J. Roberts, D.A.E. Costello, and I.M. Moroz. Exploiting Nonlinear recurrence and Fractal scaling properties for voice disorder detection. *BioMedical Engineering OnLine*, 6(1):23, 2007.
M.A. Little, P.E. McSharry, E.J. Hunter, J. Spielman, and L.O. Ramig. Suit- ability of dysphonia measurements for telemonitoring of Parkinson? s disease. *IEEE transactions on bio-medical engineering*, 2008.
C. Loukas and P. Brown. Online prediction of self-paced hand-movements from subthalamic activity using neural networks in Parkinsons disease. *Journal of neuroscience methods*, 137(2):193{205, 2004.
I. Massarelli, M. Imbriani, A. Coi, M. Saraceno, N. Carli, and A.M. Bianucci. Development of QSAR models for predicting hepatocarcinogenic toxicity of chemicals. *European Journal of Medicinal Chemistry*, 2009.
M. Pal and University of Nottingham (GB). *Factors Influencing the Accuracy of Remote Sensing Classi¯cation: A Comparative Study*. University of Nottingham, 2002.
P. Piccini and A. Whone. Functional brain imaging in the differential diagnosis of Parkinson's disease. *Lancet Neurology*, 3(5):284{290, 2004.
J. Platt. Machines using sequential minimal optimization. *Advances in Kernel Methods-Support Vector Learning*, 1998.
J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. *Advances in Kernel Methods-Support Vector Learning*, 208, 1999.
AH Rajput, B. Rozdilsky, and A. Rajput. Accuracy of clinical diagnosis in parkinsonism{a prospective study. *The Canadian journal of neurological sciences. Le journal canadien des sciences neurologiques*, 18(3):275, 1991.
B.D. Ripley. *Pattern recognition and neural networks*. Cambridge university press, 1996.
D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533{536, 1986.
L. Tari, C. Baral, and S. Kim. Fuzzy c-means clustering with prior biological knowledge. *Journal of Biomedical Informatics*, 2008.
S. Theodoridis and K. Koutroumbas. Pattern Recognition (2nd), 2003.
E. Tolosa, G. Wenning, and W. Poewe. The diagnosis of Parkinson's disease. *Lancet Neurology*, 5(1):75{86, 2006.
B. ÄUstÄun, WJ Melssen, and LMC Buydens. Facilitating the application of Support Vector Regression by using a universal Pearson VII function based kernel. *Chemometrics and Intelligent Laboratory Systems*, 81(1):29{40, 2006.
B. ÄUstÄun, WJ Melssen, and LMC Buydens. Visualisation and interpretation of support vector regression models. *Analytica Chimica Acta*, 595(1-2):299{
309, 2007.
V.N. Vapnik. The Nature of Statistical Learning Theory [M], 1995.
I.H. Witten and E. Frank. Data Mining: Practical machine learning tools and techniques. *Morgan Kaufmann, San Francisco*, 2005.