Prediction of parameters characterizing the state of a pollution removal biologic process

Stéphane Grieu, Adama Traoré, Monique Polit, Jesús Colprim

PREDICTION OF PARAMETERS CHARACTERIZING THE STATE OF A POLLUTION REMOVAL BIOLOGIC PROCESS

Stéphane Grieu*, Adama Traoré*, Monique Polit*,[1], Jesús Colprim**
*Laboratoire de Physique Appliquée et d'Automatique, Université de Perpignan
52 Avenue de Villeneuve, 66860 Perpignan, France
**Laboratori d'Enginyeria Quimica i Ambiental, Universitat de Girona
Campus de Montilivi, E-17071 Girona, Catalonia, Spain

Abstract
This work is devoted to the prediction, based on neural networks, of physicochemical parameters impossible to measure on-line. These parameters - the Chemical Oxygen Demand (*COD*) and the ammonia NH_4 - characterize the organic matter and nitrogen removal biological process carried out at the Saint Cyprien *WWTP* (France). Their knowledge make it possible to estimate the process quality and efficiency. First, the data are treated by *K-Means* clustering then by principal components analysis in order to optimize the Multi-Level Perceptron (*MLP*) learning phase. *K-Means* clustering makes it possible to highlight different operations within the Saint Cyprien treatment plant. The Principal Components Analysis (*PCA*) is used to eliminate redundancies and synthesizes the information expressed by a data set. With respect to the neural network used, these techniques facilitate the pollution removal process understanding and the identification of existing relations between the predictive variables and the variables to be predicted.

Keywords : Wastewater Treatment Plant (*WWTP*) ; Neuronal prédiction ; Multi-Level Perceptron (*MLP*) ; *K-Means* clustering ; Principal Components Analysis (*PCA*) ; Chemical Oxygen Demand (*COD*).

I. Introduction

Since the implementation of European Directive 91/271/CEE, nutrient removal of wastewater treatment plants (*WWTPs*) has been progressively adopted by existing and new facilities (Degrémont, 1989).
Nevertheless, when looking at nutrient removal, and specially nitrogen removal, on-line knowledge of influent quality could improve plant efficiency by implementing control schemes. It is possible to implement on-line ammonia or *COD* sensors, but all of these sensors are based on indirect measurements. The mechanism knowledge involved in wastewater treatment, related to carbon and nitrogen removal, has been widely studied since the presentation of *ASM1* in 1986 (Henze, *et al.*, 1986). From the actual process knowledge, it is possible to clearly identify variables affecting plant performances and the way the process is affected by variations in influent quality and quantity.

Neural networks, a statistical tool for data analysis (Hertz, *et al.*, 1991 ; Davalo and Naim, 1993), could be applied to establish a relation between variables describing a process state and different measured quantities. This relation depends, in a not always obvious way, on the predictive variables used. The principal characteristic of neural networks is their capacity to automatically establish relations between variables by means of a mechanism called training or learning (Moller, 1993 ; Charalambous, 1992). Neural networks are designed for a specific application and, after a training phase, able to generate a prediction, applying the relationship developed during the training period.

Artificial neural network models make it possible to develop non-linear empirical correlations. It is thus possible to connect a set of input variables, X_i $(1 \leq i \leq I)$, with a set of output variables, Y_k $(1 \leq k \leq K)$, assuming that we have N relevant experimental values for the couples $[X_i, Y_k]_n$ $(1 \leq n \leq N)$. As in all empirical models, the user must bear in mind that the regression models quality $Y = f(X)$, will depend on the relevance and the quality of the available experimental data used during the training phase. Moreover, getting a good prediction over some given points with a regression model, does not guarantee an important generalization capability or a good prediction for a new number of couples $[X_i, Y_k]_m$ $(N{+}1 \leq m \leq P)$. Keeping this in mind, a neural network must establish general mechanisms and be able to adapt them continuously to new and unknown situations by means of recalibration procedures. The neural networks true power is their ability to represent both linear and non-linear relationships and to learn these relationships directly from data. Traditional linear models are simply inadequate when they try to model data that contains non-linear characteristics.

This paper describes a procedure to estimate on-line influent and effluent *COD* and ammonia concentrations at the Saint Cyprien wastewater

[1] Corresponding author, Tel. 33 4 68 66 17 55, Fax 33 4 68 66 20 87

treatment plant (France). This procedure includes a Multi-Level Perceptron (*MLP*) neural network as estimation tool and data treatment by *K-Means* clustering and principal components analysis. Only easily obtained on-line data are considered in order to minimize the implementation effort in a real *WWTP*. Thus, assuming that in real *WWTPs*, influent flow, air flow to aeration basis and dissolved oxygen concentration in aerobic reactors are, most of the time, continuously measured, a global scheme of prediction using such on-line data will be presented to estimate influent and effluent concentrations.

II. Materials and methods

In the following paragraph, the materials and methods used to obtain, by application of the procedure, on-line values of *COD* and ammonia concentrations (influent and effluent) at the Saint Cyprien *WWTP* will be described.

1. *Saint Cyprien wastewater treatment plant*

The used experimental data were provided by the Saint Cyprien *WWTP* employees. Three towns are connected to this "activated sludge" plant able to treat the wastewater of 80000 inhabitants : Saint Cyprien (8653 inhabitants), Alenya (2339 inhabitants) and Latour Bas Elne (1711 inhabitants) (Eckenfelder and Musterman, 1995).

This *WWTP* is a weak charges biological plant (denitrification, biological dephosphatation with physicochemical complement). It has the particularity of being able to adapt operation to the seasonal population variations. The town of Saint Cyprien is a seaside resort and its population increases considerably during the summer season. Therefore the plant consists of parallel treatment modules characterizing its adaptability. At the end of the wastewater treatment sequence, a sludge composting unit allows the sludge re-use in agricultural applications.
The plant is divided into a water line and a sludge line. The water line consists of the primary treatment, the biological treatment tanks and the clarifiers. This line can be described in the following way (Degrémont, 1989) :

Water line

- **Primary treatment** : cleaning by tilted rake and grease and oil removal.
- **Biological treatment tanks :** 4 parallel tanks that can be used individually or jointly according to the period of the year (total volume of 15600 m^3).
- **Clarifier :** 2 clarifiers in parallel and individually usable (total volume of 3200 m^3).

The sludge line and the complementary treatments are the water line continuation. The sludge line deals with sludge dehydration and composting. The complementary treatments are the filtration of treated water and the ultraviolet disinfection. They can be described in the following way (Durand and Jacquinot, 1995) :

Sludge line and complementary treatments

- **Sludge dehydration** : 3 presses placed in a closed and ventilated room (increase by 16 % for dryness).
- **Sludge composting** : compost prepared to be used in agriculture starting from dehydrated sludge and wood plates mixture.
- **Treated water filtration** : 2 sand filters. Total capacity : 150 m^3/h.
- **UV disinfection** for bacterial decontamination with 2 possible files of respectively 850 m^3/h and 150 m^3/h throwing out water into the “canal d’Elne”.

2. *Neural network structure*

Linear approximation networks are too restrictive to treat such a process and nonlinear approximation networks offer much greater capacity.
Thus, the used neural network is a multi-level Perceptron. This type of neural network is known as a supervised network because it requires a desired output in order to learn.
It is based on the classical Perceptron network introduced by Rosenblatt (1957). It includes a number of active neurones, the basic building block, connected together to form a network. Its functionality resides in weights associated with links between neurones.
When several levels of neurones are connected one after another, the problem is moving into the soft models area. All levels cannot be linear. The number of parameters to identify increases, and the model becomes essentially non-linear.

The *MLP* and many others neural networks learn using an algorithm called “backpropagation”. With this algorithm, the input data are repeatedly presented to the neural network. With each presentation the neural network output is compared to the desired output (the target) and an error is computed. This error is then fed back (backpropagated) to the network and used to adjust the weights such that the error decreases with each iteration and the neural model gets closer and closer to producing the desired output.
The minimization algorithm uses an iterative process and various weight values are explored with the aim of minimizing the quadratic criterion on the sum of squared error made during the learning phase (Hornik, *et al.*, 1989).

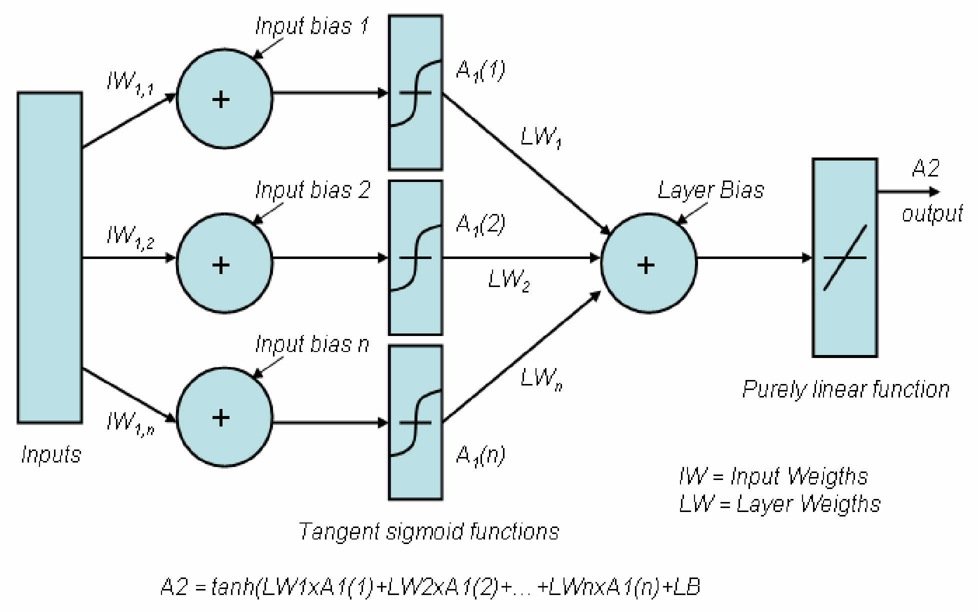


Figure 1. Multi-level Perceptron structure

It is very difficult to know which training algorithm will be the fastest for a given problem. It will depend on many factors, including the problem complexity, the number of data points in the training set, the number of weights in the network... Several training methods were used, but the Levenberg-Marquardt (*LM*) algorithm proved to be the fastest and the most robust. The error in the *LM* algorithm decreases much more rapidly with time than in the other algoritms used. It is particulary adapted for networks of moderate size and has memory reduction feature for use when the training set is large (Hagan and Menhaj, 1994).

The Learning rate parameter may also play an important role in the network convergence, depending on the application and the network architecture. This parameter is used to increase the chance of avoiding the training process from being trapped in a local minimum instead of the global minimum (Hamed, *et al.*, 2004). It is determined using an empirical approach.

The chosen *MLP* structure is depicted on Figure 1. In linear systems, there is no real benefit to cascading multiple layers : the equivalent weight matrix for the total system is simply the layers weight matrices product.

The situation is quite different if nonlinear hidden neurones are inserted between the input and the output layers. In this case, it seems natural to assume that the more layers are used, the greater power the networks possesses. However, it is not the case in practice. According to our tests, an excessive number of hidden layers often proves to be improductive. It causes slower convergence in the backpropagation learning because intermediate neurones not directly connected to output neurones have very small weight changes and learn very slowly. Two elements can be advanced :

- The error signals are numerically degraded when propagating across too many layers.
- Extra layers tend to create additional local minima.

Thus, it is essential to identify the proper number of layers. According to our tests, a *MLP* with only one hidden layer is adequate as universal approximator of any nonlinear functions and is used in the present study.

The overfitting problem that can occur during the learning phase has an extremely negative effect on the network generalization capacity, which becomes unable to predict good behaviours. One of this research objectives is to prevent overfitting (Schittenkopf, *et al.*, 1997). The prediction procedure thus utilizes a *PCA* and a *K-Means* clustering in order to optimize the network training.

The influence on the learning phase of the two following user-definable parameters was also studied :

- The number of iterations completed during the training phase.
- The number of neurones placed in the network hidden layer.

Although the relationship between the network performance and its hidden layer size is not well understood, a principle can be used as a guide : the principle of generalization versus convergence. Generalization means the network ability to produce good results with a data set that has not been seen during the learning phase. Convergence is the ability to learn the training data.

The number of iterations to effect during the training phase and the number of neurones to place in the network hidden layer are very important parameters. They increase the network calculation capacity during the learning process. However our network can become specialised on the learning data and only on those.
Thus, the objective is to use as many hidden neurones and iterations as needed for convergence without inhibiting the ability of the network to generalize. So the network will be able to focus on the important features in the data rather than fitting the noise, an inherent component of any environmental field data set (Schittenkopf, *et al.*, 1997 ; El-Din and Smith, 2002).

3. *K-Means* clustering

K-Means clustering is also known as mobile centers clustering. The *K-Means* algorithm divides the observations constituting a data file in *K* independent clusters and specifies to which clusters each observation is assigned. Each data file observation is considered as an object localized in space. The algorithm implements a partition based on the fact that objects belonging to a cluster are as close as possible and as far as possible from objects pertaining to other clusters. The used distance parameter is the squared euclidean distance (Jain, *et al.*, 1999).

Each cluster is defined by its objects and its centroid or center. The cluster centroid is the point where the sum of all the distances between it and all the objects is minimal. The algorithm requires to know the number of clusters and breaks up into three principal stages :

- an initialisation stage to randomly (or by means of a preset method) determine the centers,
- a reallocation phase to assign each observation to its nearest center,
- a centring phase to replace the centers by the centers of gravity.

These stages are repeated up to convergence. The algorithm is iterative and minimizes the sum of the distances between each object and the cluster centroid (Jain and Dubes, 1988).
The procedure can be synthesized in the following way : an object O_i belongs to the class characterized by the nearest center of gravity. The *K-Means* clustering procedure can be summarized as depicted in Figure 2 :

1) Cut out the data in *K* subsets $P_1, ..., P_K$

2) Calculate the centers of gravity $C_1, ..., C_K$ for each subset $P_1, ..., P_K$

3) Update the subsets $P_1, ..., P_K$ such as any object O_i will be put in P_j if C_j is the center of gravity nearest to O_i

4) Return to stage (2) if one or more P_j subsets were modified during stage (3)

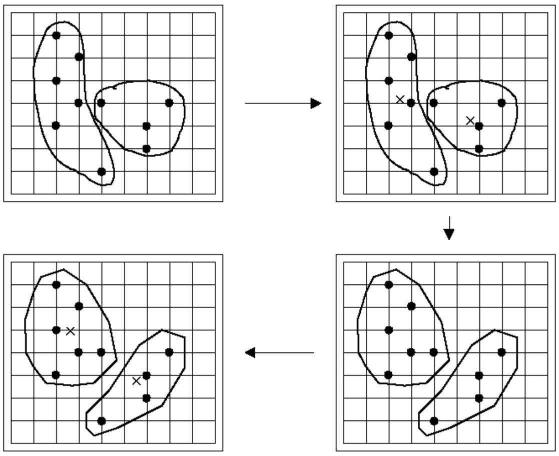


Figure 2. Example of *K-Means* clustering

4. *Principal Components Analysis*

The principal components analysis (*PCA*) is a factorial analysis method which enables to reduce a complex system of correlations to a smaller number of dimensions. During a phenomenon study, several variables can be correlated or be of weak interest. In this case, it is judicious to use multivariate statistical methods. The *PCA* main purpose is to reduce the number of variables in order to obtain new synthetic variables and to carry out geometrical representations. This reduction will only be possible if the initial variables are not independent and have nonnull correlation coefficients. They must be quantitative, heterogeneous and not globally independent (Morineau and Aluja-Banet, 1998 ; Cibois, 1983).

When p quantitative variables $X_1, ..., X_p$ are correlated, the expressed information is characterized by some redundancy. Principal components analysis has as objective to extract from a redundant list of p variables $X_1, ..., X_p$ a nonredundant list of K new variables or factors $f_1, ..., f_k$ $(k \leq p)$. This extraction is made in the simplest possible way, i.e. linearly : factors are built like linear combinations of variables and reciprocally. So, the *PCA* is a factorial method because the number of variables reduction is not effected by a simple selection, but constructing new synthetic variables obtained by combining initial variables. The *PCA* proceeds by reduction of the variables space dimension while eliminating correlations between initial variables.

In this linear context, the non-redundancy condition of the factors list $f_1, \ldots, f_k$ ($k \leq p$), is expressed by the non-correlation between factors. Each factor explains a part of the variability observed on the original variables. It makes it possible to summarize the information contained in the p initial variables thanks to a number of factors lower than p. This represents an appreciable information compression. Any correlation matrix $R\ (p \times p)$ can be analyzed and decomposed with principal components analysis. Of course, a matrix decomposition supposes that the correlations are suitable measurements of the variables connection : these connections have to be linear. In the opposite case, it is judicious to try to linearize these relations by means of adequate transformations before carrying out a *PCA*.

5. *Structure of the procedure*

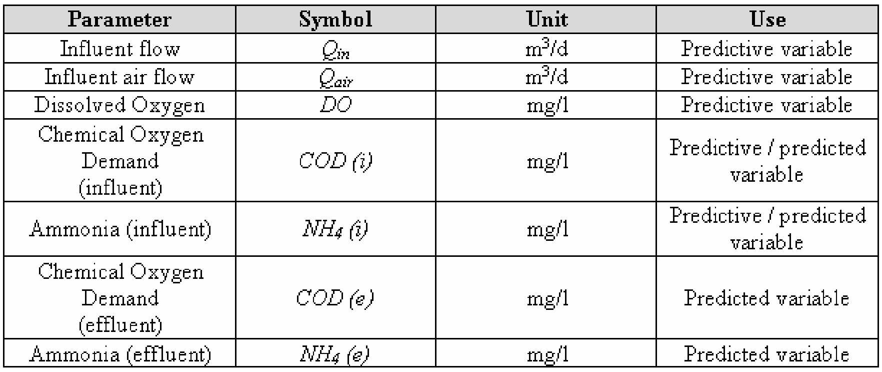
Data used to carry out this study describe the Saint Cyprien *WWTP* operation during the year 2002. It is important to note that this work depends on the form and quantity of measurements and analysis provided by the Saint Cyprien *WWTP*. Thus, the procedure is adapted, as well as possible, to available data, always keeping in mind the work initial objective, i.e. to estimate the state of a pollution removal process from on-line data. Table 1 presents the used parameters. Influent flow (Q_{in}), influent air flow (Q_{air}) and dissolved oxygen (*DO*) are easily on-line measured parameters at the Saint Cyprien wastewater plant but only available as daily average data. They were used, because of their predictive character, as neural network inputs.

The influent flow is the water quantity entering the plant. The influent air flow is the air quantity injected in the biological treatment tanks and necessary to an efficient pollution removal. The dissolved oxygen is an influential ecological parameter and can be considered as an indicator of pollution and biological activity. The parameters to be predicted (*COD* and NH_4), obtained from laboratory analysis carried out on samples (on average twice a week during the year 2002), are specific measurements. Daily average data are sufficient to obtain exploitable and significant predictions for parameters characterizing the various processes, because the studied systems have a very slow evolution rate. Evolution to a critical state being relatively slow, the use of daily profile measurements is not essential.

COD and NH_4 are physicochemical parameters, very difficult to measure on-line, characterizing a biological process of organic mater and nitrogen removal. Their knowledge allows to estimate the process quality and efficiency. *COD* represents the oxygen quantity necessary for chemically eliminating the organic matter. NH_4 presence in water is a human or industrial contamination sign.

As previously mentioned, the Saint Cyprien *WWTP* is a weak charges biological station able to adapt its working to seasonal population fluctuations. Saint Cyprien is a tourist resort and its population rises sharply during the summer season. Consequently, the plant can function in a much lower mode than its maximum capacity (15 to 20 %) or, during certain periods, in a very close mode (90 to 95 %). Differences between day-time and night-time operations are also characteristic of the studied process (considerable reduction in the quantity of water to be treated during the night due to a reduced human activity). A classification by *K-Means* clustering allows us to highlight these significant differences in operation and to carry out fractional training which greatly facilitates the phenomena comprehension by the neural network.

The data treatment by *K-Means* clustering and principal components analysis is carried out to improve the necessary effectiveness of the neural network learning phase. This effectiveness is difficult to obtain because of the strong non-linearity characterizing the biological phenomena and the data limited quantity. Jointly to the network intrinsic parameters optimisation, it allows to minimize overfitting.



Parameter	Symbol	Unit	Use
Influent flow	Q_{in}	m^3/d	Predictive variable
Influent air flow	Q_{air}	m^3/d	Predictive variable
Dissolved Oxygen	*DO*	mg/l	Predictive variable
Chemical Oxygen Demand (influent)	*COD (i)*	mg/l	Predictive / predicted variable
Ammonia (influent)	NH_4 *(i)*	mg/l	Predictive / predicted variable
Chemical Oxygen Demand (effluent)	*COD (e)*	mg/l	Predicted variable
Ammonia (effluent)	NH_4 *(e)*	mg/l	Predicted variable

Table 1. Parmeters used for the prediction

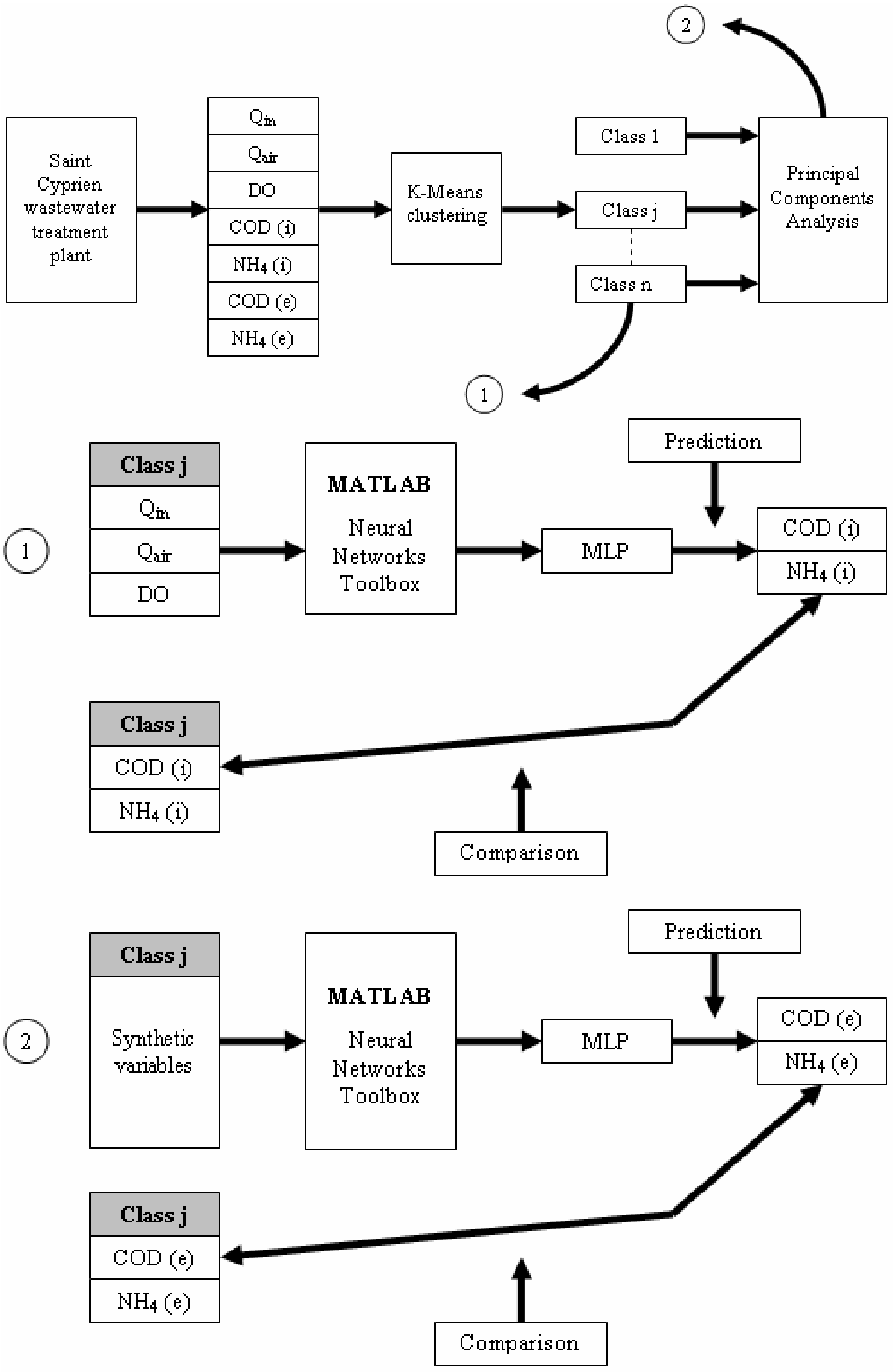
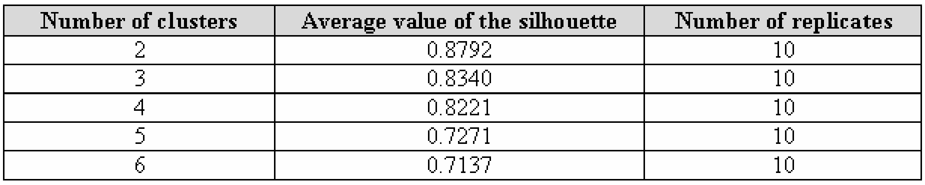


Figure 3. General scheme of the prediction various stages

As mentioned above, data treatment by *K-Means* clustering makes it possible to define various operation classes characterizing the pollution removal process. Particularly, these classes are defined by the seasonal variations in plant operation. According to clustering results, the network learning phase is conducted separately for each class and is so more precise and effective.

Too many variables used as neural network inputs can cause overfitting. A principal components analysis was thus carried out after the clustering in order to synthesize the information without important losses. This *PCA* was interesting to reduce the number of network inputs, used for the prediction of *COD* (effluent) and *NH_4* (effluent) without information loss. For the prediction of *COD* (influent) and *NH_4* (influent), the number of input variables being less important, the *PCA* impact on the obtained results quality will be less sensitive.



Number of clusters	Average value of the silhouette	Number of replicates
2	0.8792	10
3	0.8340	10
4	0.8221	10
5	0.7271	10
6	0.7137	10

Table 2. Optimal number of clusters

Figure 3 represents the whole used procedure. It consists of four successive stages and can be synthesized in the following way :

- **Stage 1** : data treatment by *K-Means* clustering to highlight significant differences in operation.
- **Stage 2** : data treatment by *PCA* to synthetize the information.
- **Stage 3** : prediction by *MLP* neural network of *COD* (influent) and NH_4 (influent), using as inputs Q_{in}, Q_{air} and *DO*.
- **Stage 4** : prediction by *MLP* neural network of *COD* (effluent) and NH_4 (effluent), using as inputs Q_{in}, Q_{air}, *DO* and stage 3 results.

III. Results

This paragraph presents the results obtained with the proposed procedure including *K-Means* clustering, principal components analysis and neural network prediction.

1. K-Means clustering results

Two parameters have to be taken into account to carry out this classification : the desired number of clusters and the initial position of the various centers (or centroids). The optimal number of clusters is not easy to fix a priori, so the best possible choice was determined empirically.

Like the majority of numerical minimization techniques, the solution provided by the *K-Means* clustering algorithm depends on the centroids initial position. This position can greatly influence the obtained results quality.

It is thus possible for the algorithm to reach a local minimum and for any redistribution of points to a new cluster to cause an increase in the total sum of distances between the points and the centroid. In order to minimize this problem, it is advisable to carry out several replicates based on random initial centroids positions. The best solution is retained.

The *K-Means* clustering result is presented using a silhouette representation. The optimal number of clusters is empirically determined by evaluating the clustering quality on the average value of the silhouette. The higher it is the more separate and compact are the clusters. Ten replicates were carried out for a *K* fixed number of clusters ranging between 2 and 6. The average values obtained are presented in the table 2.

Figure 4 shows two compact and perfectly separate clusters. The majority of individuals presents very high silhouette values (ranging between 0.8 and 1). This confirms the perfect clusters separation determined by *K-Means* clustering. The average value is equal to 0.8792, which is very high, and confirms the visual impression given by the representation.

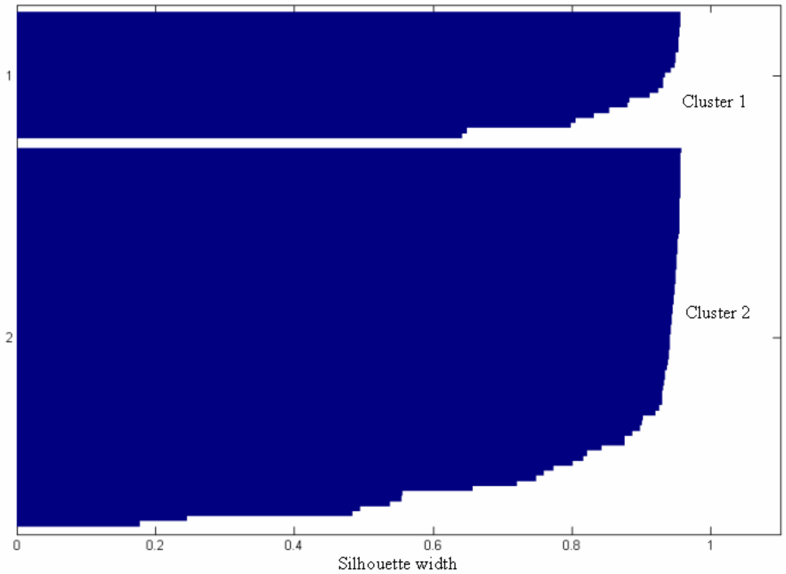


Figure 4. *K-Means* clustering results for $K = 2$

The classification obtained by *K-Means* clustering highlights the existence of two subsets in the data file obtained from the Saint Cyprien *WWTP* and characteristic of its operation during 2002. The two obtained classes clearly distinguish the plant working during the summer period characterized by an increase of the water quantity to be treated and the pollution to be removed (class 1) from the working during the rest of the year (class 2).

Class 1 is made up of 25 days : the totality of summer days for which the station functions almost at its maximum capacity (4 tanks) and a few days more marked by an increase in the influent flow due to strong rains. Class 2 is composed of 75 days, distributed on the remainder of the year, for which the plant functions clearly below its maximum capacity (2 tanks). The *K-Means* clustering technique allowed to insulate the two principal operation types in the Saint Cyprien plant, and to confirm the very clear separation existing in the data file characterizing these operations. These results give the possibility of carrying out a specific neuronal training operation for each class in order to facilitate the network comprehension of the mechanisms expressed by the data, and to consider its capability to react and face up to various behaviors.

2. *Principal components analysis results*

As part of the prediction procedure, principal components analysis is used in order to reduce the number of network inputs. The set of variables is used for its predictive character and the *PCA* enables to reduce its dimension in order to support the network learning process without notable information loss. The technique was applied to each of the two data classes, used to predict *COD* (effluent) and NH_4 (effluent). The data file is composed of the following variables : Q_{in}, Q_{air}, *DO*, *COD* (influent) and NH_4 (influent). These quantitative variables have a high degree of correlation and thus a redundant character. The *PCA* allows us to extract from this set of correlated variables a nonredundant list of new (synthetic) variables. These synthetic variables are linear combinations of the initial quantitative variables. They make it possible to minimize the risks of overfitting and the consequences of a too low number of examples characterizing each class defined by *K-Means* clustering.

According to the obtained results (Tables 3 & 4), the number of preserved synthetic variables can vary but must always be representative of the information expressed by the set of initial variables. For class 1 the first three principal components which express 88.7 % of the system total variance are preserved. It is important to emphasize that more than 60 % of this total variance is expressed by the 1st principal component alone. The two principal components not selected to form part of the new synthetic data file express 8.32 % and 2.98 % of the original variance respectively. Their contribution is thus very low. The three new definite synthetic variables allow us to reduce considerably the data file dimension used for its predictive character, while preserving the information large majority. Figure 5 presents the results obtained by an histogram showing the total variance expressed by the principal axes (class 1).

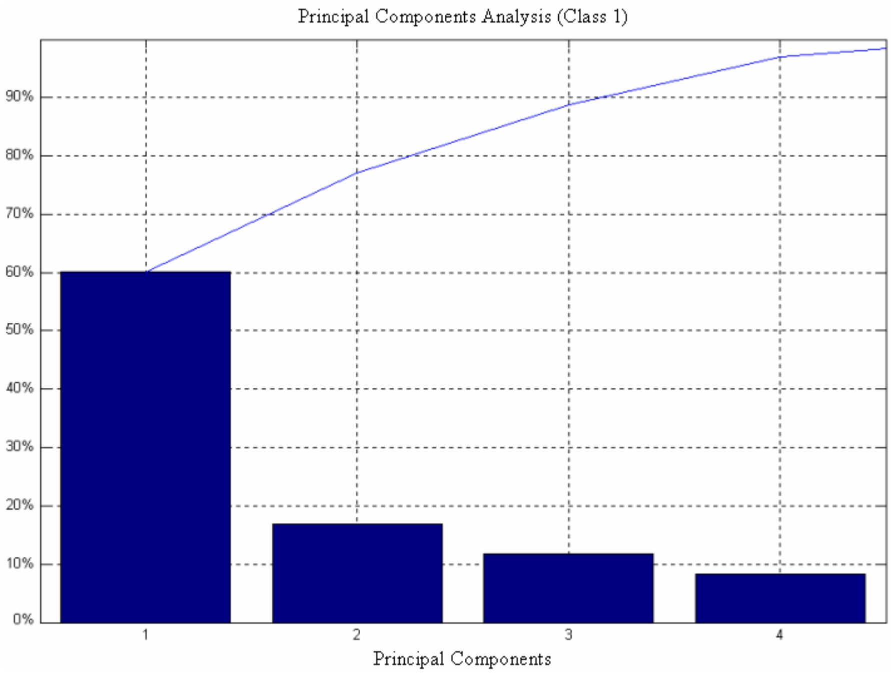
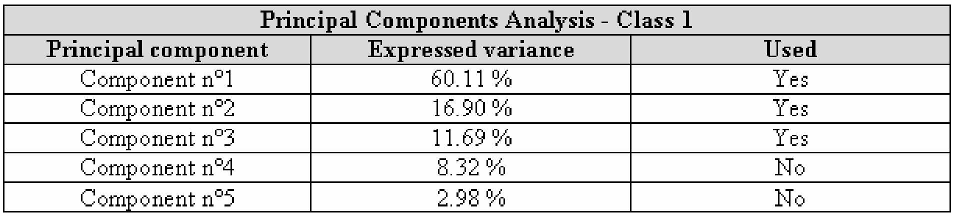
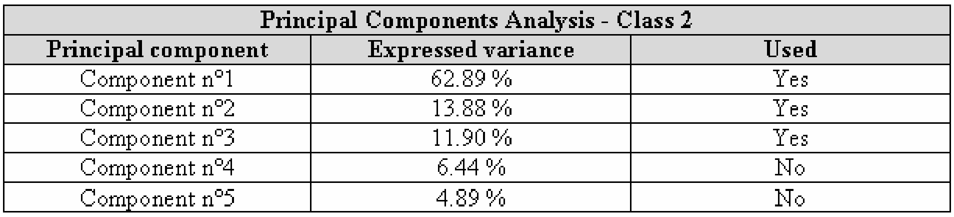


Figure 5. Histogram showing the variance expressed by the principal components (class 1)



Principal Components Analysis - Class 1		
Principal component	**Expressed variance**	**Used**
Component n°1	60.11 %	Yes
Component n°2	16.90 %	Yes
Component n°3	11.69 %	Yes
Component n°4	8.32 %	No
Component n°5	2.98 %	No

Table 3. Percentage of variance expressed by the principal components (class 1)



Principal Components Analysis - Class 2		
Principal component	**Expressed variance**	**Used**
Component n°1	62.89 %	Yes
Component n°2	13.88 %	Yes
Component n°3	11.90 %	Yes
Component n°4	6.44 %	No
Component n°5	4.89 %	No

Table 4. Percentage of variance expressed by the principal components (class 2)

Table 4 presents the variance expressed by each of the 5 principal components using class 2 data. We also preserve for class 2 the first three principal components which express 88.6 % of the system total variance (Figure 6). The first principal component expresses almost 63 % of the total variance, whereas not selected components express only 6.44 % and 4.89 % of this variance. As for class 1, the three synthetic variables obtained by principal components analysis make it possible to significantly reduce the data file dimension.

The results provided by the principal components analysis are satisfactory for the 2 classes defined by *K-Means* clustering, and make it possible to reduce significantly the data file dimension used as neural network input for the prediction of *COD* (effluent) and NH_4 (effluent) : i.e. from 5 variables to 3 synthetic variables.

The prediction tests carried out showed that a too large number of variables (observations) used as network inputs, combined with a too restricted number of examples (individuals) has a negative impact on the learning process and favours overfitting. The strongly nonlinear nature of the biological process to model amplifies the problem. The reduction of the data file dimension used for the learning phase could also be obtained by eliminating several variables but this would be to the detriment of the information they express. The principal components analysis allows to carry out this reduction while preserving almost all the information used for its predictive character (Philippeau, 1986).

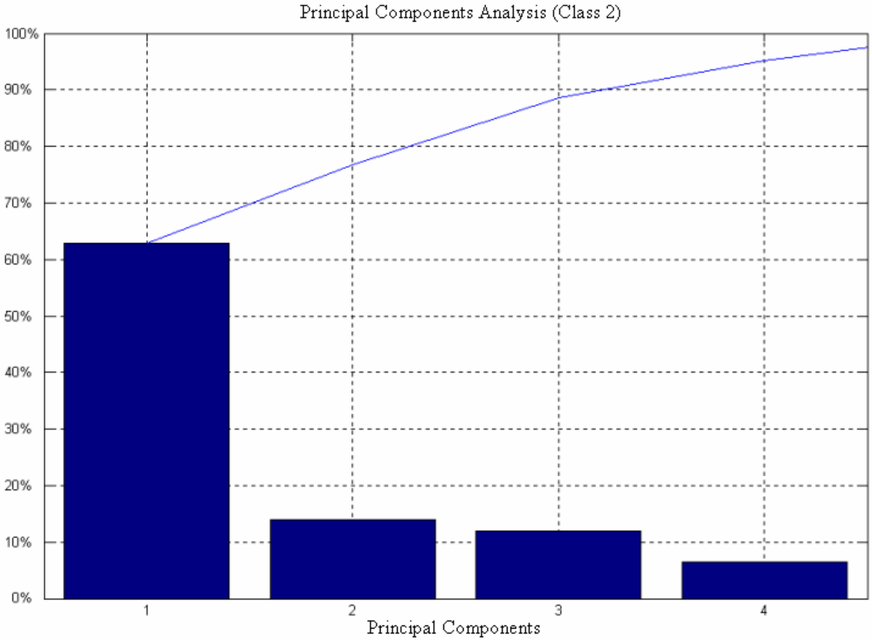


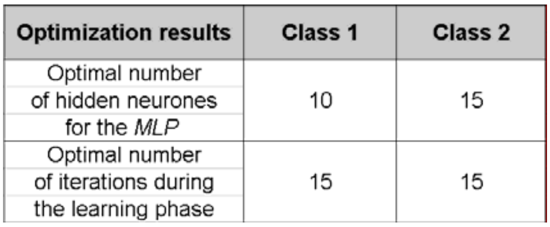
Figure 6. Histogram showing the variance expressed by the principal components (class 2)

3. *Influential parameters optimization*

It is important to optimize the number of iterations used during the learning phase and the number of neurones placed in the network hidden layer. These two parameters have an important influence on the learning process efficiency. This optimization completes the data treatment by *PCA* and *K-Means* clustering used to minimize the overfitting phenomenon. This optimization was carried out empirically, varying each parameter successively and independently for the two classes. The procedure was repeated several times, using different validation data subsets (every time that the procedure is repeated, the data file is randomly divided into two subsets), in order to eliminate a local minima possibility. This kind of problems is frequently encountered using back propagation algorithms. These influential parameters were considered as optimized when the average absolute error was the weakest (Grieu, *et al.*, 2003).

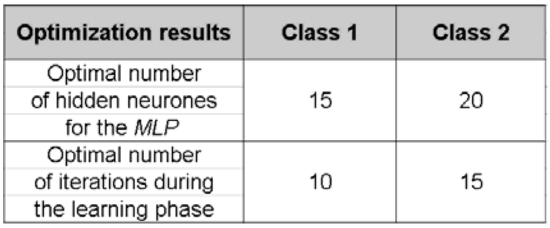
For both classes, the results show, independently of the used validation data subset, that if the number of hidden neurones is too weak, the network has not enough parameters and cannot exploit efficiently the information. On the other hand, if this number is too important, the number of model parameters increases and it becomes possible, during the optimization phase, to model relations which are only the result of statistical fluctuations in the training data file and not fundamental relations of dependence between variables. The results also show that a too high number of iterations leads to overfitting.

Tables 5 and 6 present the optimal number of hidden neurones and the optimal number of iterations for both classes. Figures 7 to 10 show these two parameters influence on the average absolute error during learning and validation phases for class 2 (only graphics for class 2 are presented but similar ones were obtained for class 1). They present the results obtained using a specific validation data subset but it is important to note that for all the tested distributions of available data between learning and validation subsets, optimization results were similar and very closed.



Optimization results	Class 1	Class 2
Optimal number of hidden neurones for the *MLP*	10	15
Optimal number of iterations during the learning phase	15	15

Table 5. Optimization results for *COD* and *NH_4* (influent) prediction



Optimization results	Class 1	Class 2
Optimal number of hidden neurones for the *MLP*	15	20
Optimal number of iterations during the learning phase	10	15

Table 6. Optimization results for *COD* and *NH_4* (effluent) prediction

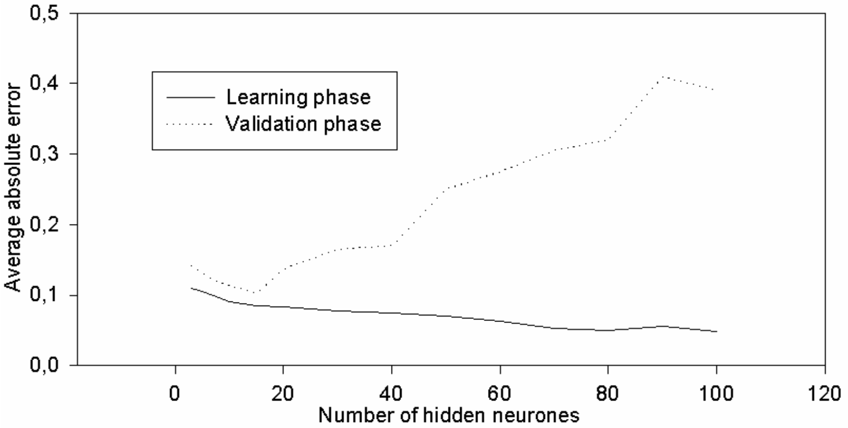


Figure 7. Number of hidden neurones influence on the *COD* (influent) and *NH_4* (influent) prediction - Class 2

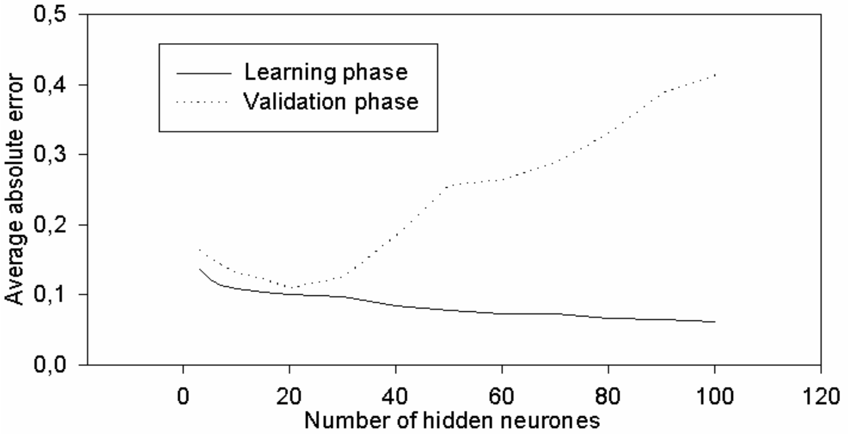


Figure 8. Number of hidden neurones influence on the *COD* (effluent) and NH_4 (effluent) prediction - Class 2

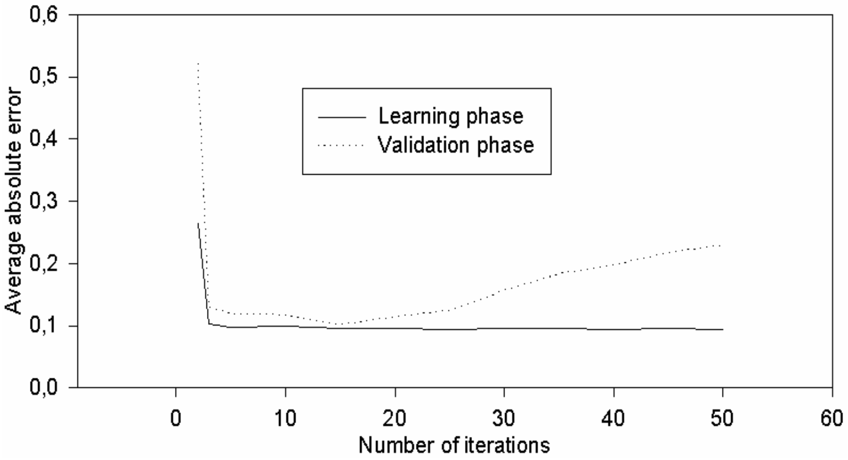


Figure 9. Number of iterations influence on the *COD* (influent) and NH_4 (influent) prediction - Class 2

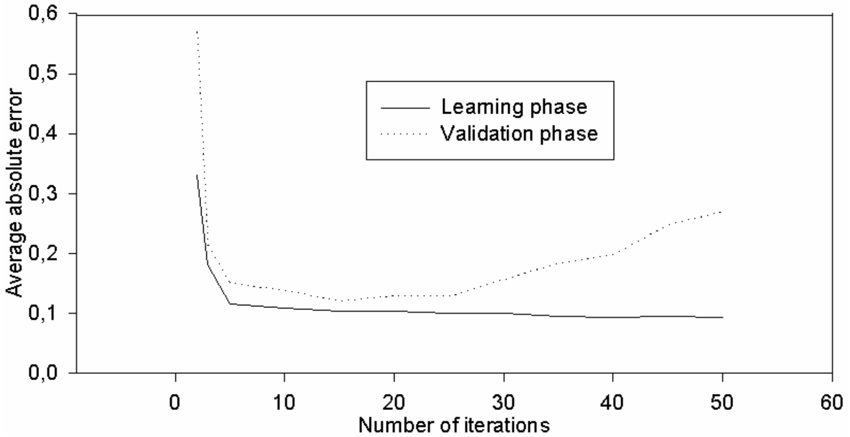


Figure 10. Number of iterations influence on the *COD* (effluent) and NH_4 (effluent) prediction - Class 2

4. Neuronal prediction results

This paragraph presents the results obtained for the *COD* and NH_4 prediction before the pollution removal treatment (influent) and after the pollution removal treatment (effluent), using class 1 and 2 data defined by *K-Means* clustering (only graphs for class 2 are presented , Figures 11 to 18).

The data distribution between learning and validation phases is significant. If the majority of available data is used during the learning phase, the quantity of data usable to test its effectiveness will be tiny and consequently non-representative of the whole distribution. Then, the performances will not characterize the learning smoothness. On the other hand, if the majority of data is used during the validation phase, the learning will be certainly very rough and the network generalization capacity limited. According to various tests, a suitable distribution was determined : 80 % of the data are reserved for the learning and 20 % for the validation. This distribution proved to be a good choice and is the one most frequently used.

Let’s note that data standardization was also carried out. Raw data obtained from the Saint Cyprien activated sludge *WWTP* consists of many process variables. Each of these variables is measured in different units having different magnitudes.
If raw data are fed into the multi-level perceptron, then variables having a larger magnitude are given unequal importance due to the nature of the weight update procedure. Standardizing the dynamic range of each variable in the input vector using data transformation techniques is required and ensures that any movements in a given direction in feature space are co-measurable, and this will improve the model performance.
As previously mentioned, the most effective backpropagation algorithm is, according to our tests, the "Levenberg-Marquardt" one. It was used for all the predictions. A suitable learning rate was determined to be 0.1 and was also used for all the neuronal predictions.

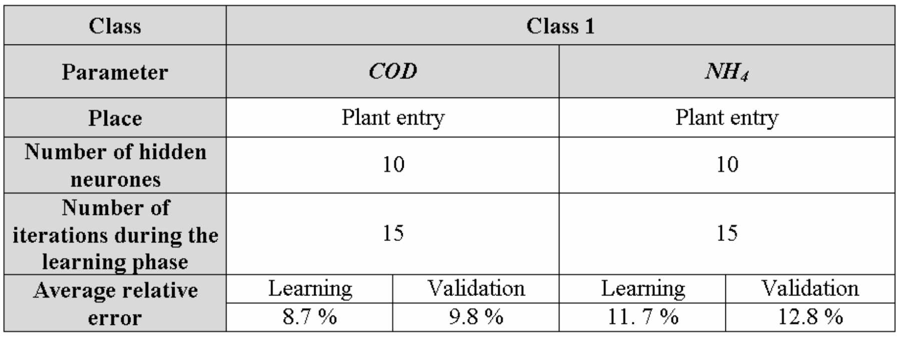
In agreement with the results obtained during the preliminary study, 10 neurones are placed in the network hidden layer and 15 iterations are carried out during the learning phase for the *COD* (influent) and NH_4 (influent) prediction (class 1). For the prediction based on class 2 data, the number of hidden neurones is changed to 15, while the number of iterations remains unchanged. For the *COD* (effluent) and NH_4 (effluent) prediction (class 1), 15 neurones are placed in the network hidden layer and 15 iterations are carried out during the learning phase. For class 2, the optimal number of hidden neurones is set to 20 and the optimal number of iterations to 15.

For each *WWTP* operation highlighted by *K-Means* clustering (class 1 and 2), prediction results can be considered as satisfactory, according to the work initial objective. Tables 7 to 10 synthesize the obtained results and present the average relative errors calculated for the *COD* (influent / effluent) and NH_4 (influent / effluent) prediction. The results are considered as satisfactory for the following reasons :

- The studied process being strongly nonlinear, the estimation of parameters characterizing its progress is extremely delicate. This difficulty is amplified by the *WWTP* operation variability. In fact, the plant is able to adapt to important population (and water to be treated) fluctuations.

- The information provided by the prediction procedure is sufficiently reliable and precise to be exploitable by operators in charge of the plant monitoring and maintenance. Indeed, *COD* and NH_4 on-line knowledge at plant entry and exit allows to estimate the pollution level before and after biological treatment. It makes it possible to check whether the biological process in charge of the pollution removal corresponds (or not) to the fixed objectives for the plant.
  The information precision is satisfactory and makes it possible to know how the system is evolving (possibly towards a critical state requiring an operator action, for example, to avoid a biomass deterioration) and how to react. A highher degree of accuracy for the results would be extremely difficult to obtain and is not necessary.
  Thus, this information effectively completes the process expert knowledge brought by operators and the knowledge provided by various on-line sensors measurements carried out on the site.

- The whole procedure (*K-Means* clustering, principal component analysis and neural networks) uses tools which proved to be efficient and complementary. These tools allowed it to be robust, particularly in case of disturbed data or measurement errors due to on-line sensors faulty operation.



Class	**Class 1**			
Parameter	***COD***		***NH_4***	
Place	Plant entry		Plant entry	
Number of hidden neurones	10		10	
Number of iterations during the learning phase	15		15	
Average relative error	Learning	Validation	Learning	Validation
	8.7 %	9.8 %	11. 7 %	12.8 %

Table 7. Prediction results for parameters *COD* and *NH_4 (influent)* (class 1)

Class	**Class 1**			
Parameter	***COD***		***NH_4***	
Place	Plant exit		Plant exit	
Number of hidden neurones	15		15	
Number of iterations during the learning phase	10		10	
Average relative error	Learning	Validation	Learning	Validation
	8.5 %	13.3 %	12.9 %	14 %

Table 8. Prediction results for parameters *COD* and *NH_4 (effluent)* (class 1)

Class	**Class 2**			
Parameter	***COD***		***NH_4***	
Place	Plant entry		Plant entry	
Number of hidden neurones	15		15	
Number of iterations during the learning phase	15		15	
Average relative error	Learning	Validation	Learning	Validation
	8.2 %	13.9 %	11 %	15.7 %

Table 9. Prediction results for parameters *COD* and *NH_4 (influent)* (class 2)

Class	**Class 2**			
Parameter	***COD***		***NH_4***	
Place	Plant exit		Plant exit	
Number of hidden neurones	20		20	
Number of iterations during the learning phase	15		15	
Average relative error	Learning	Validation	Learning	Validation
	10.9 %	15.2 %	5.8 %	6.6 %

Table 10. Prediction results for parameters *COD* and *NH_4 (effluent)* (class 2)

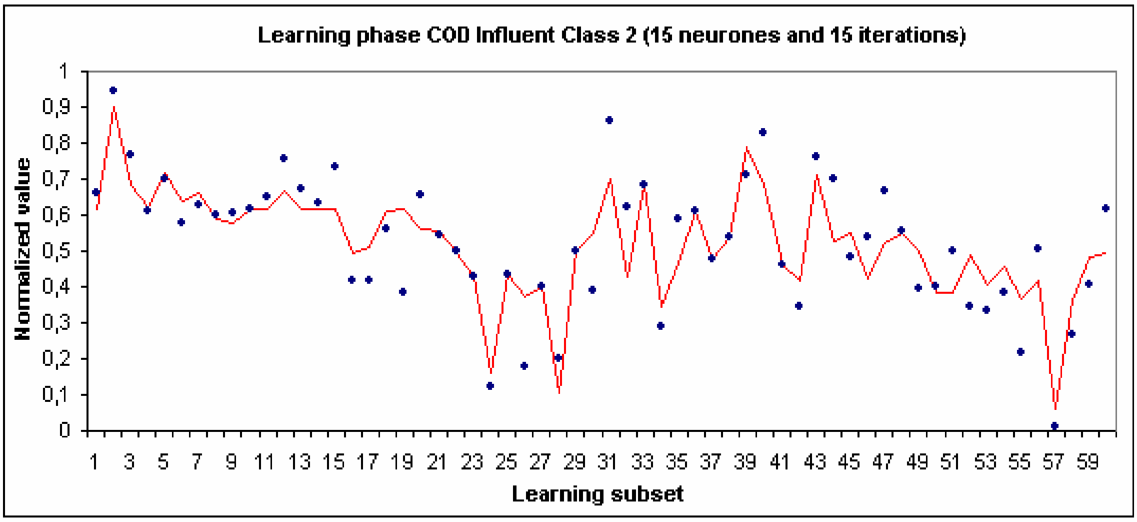


Figure 11. Learning phase for parameter *COD (influent)* (class 2)

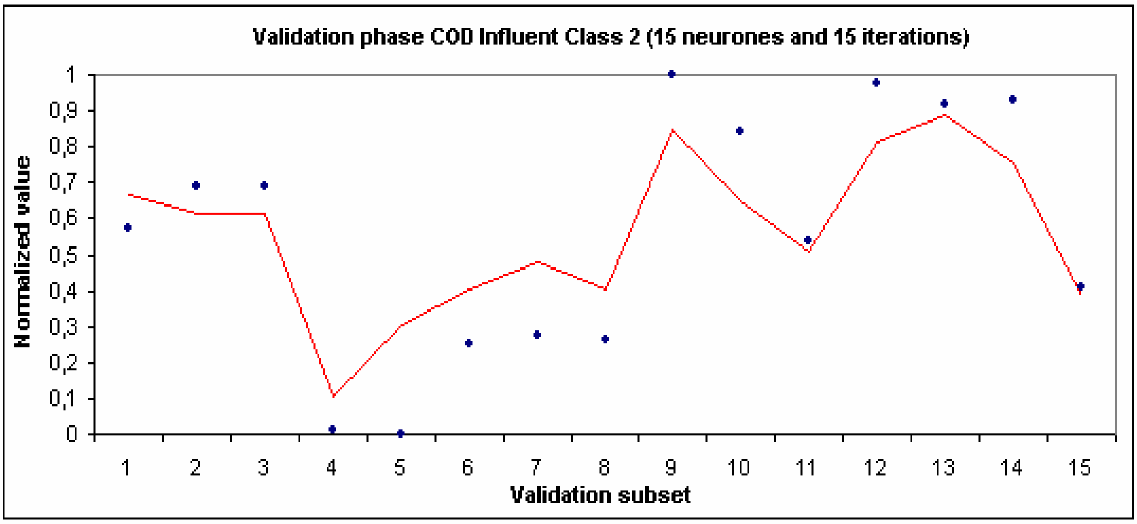


Figure 12. Validation phase for parameter *COD (influent)* (class 2)

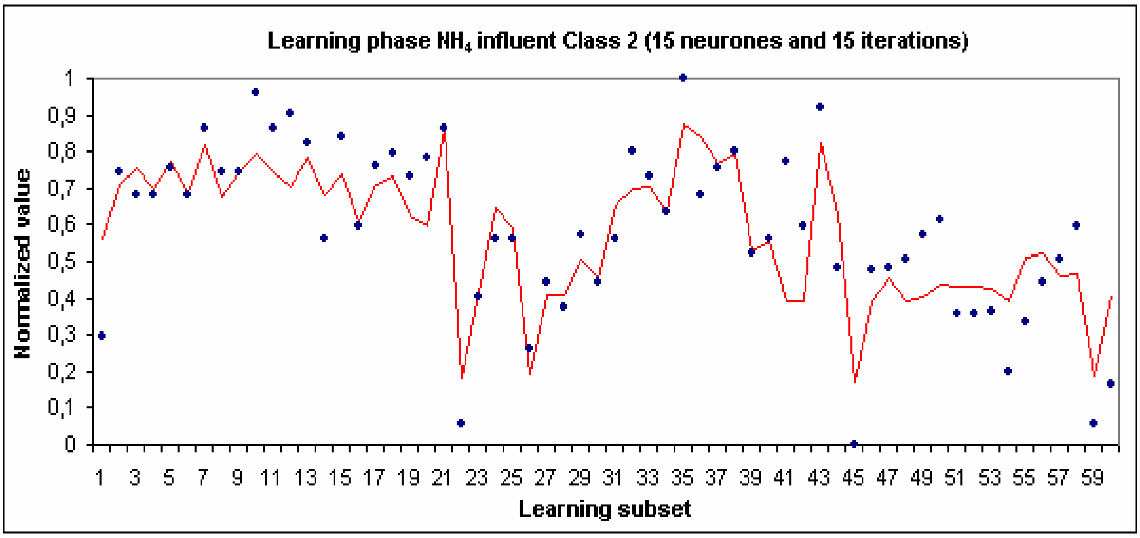


Figure 13. Learning phase for parameter *NH_4 (influent)* (class 2)

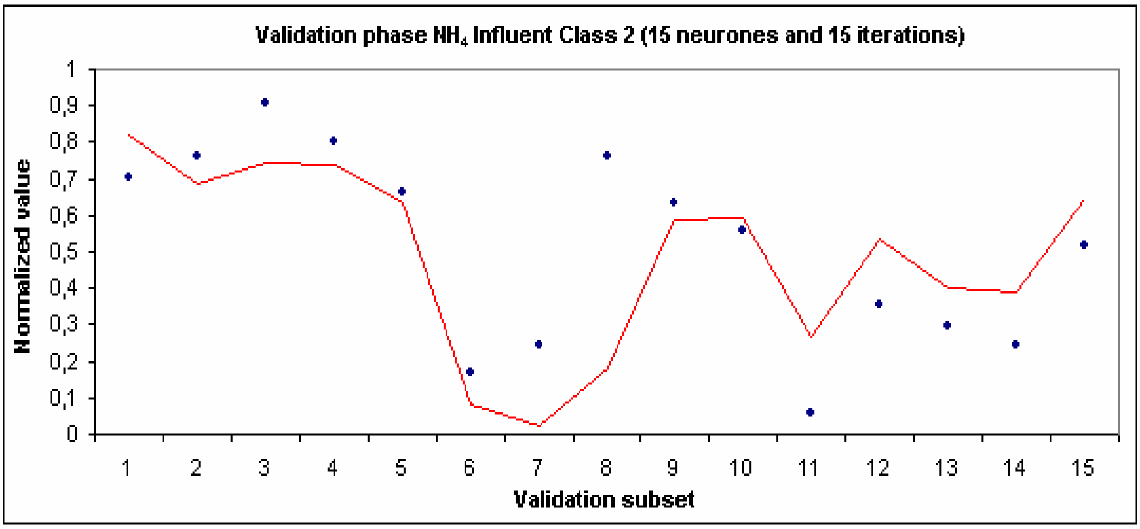


Figure 14. Validation phase for parameter *NH_4 (influent)* (class 2)

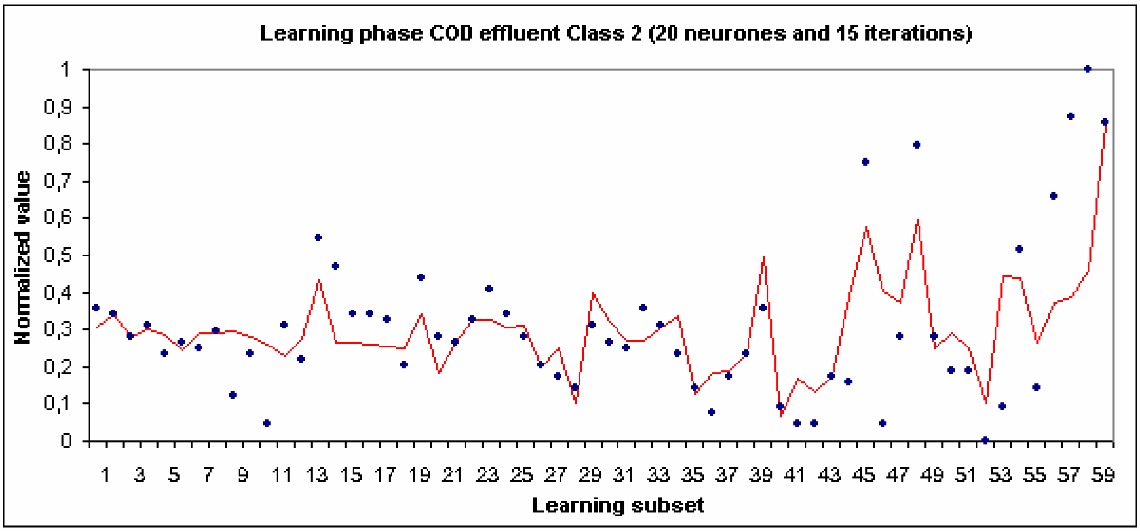


Figure 15. Learning phase for parameter *COD (effluent)* (class 2)

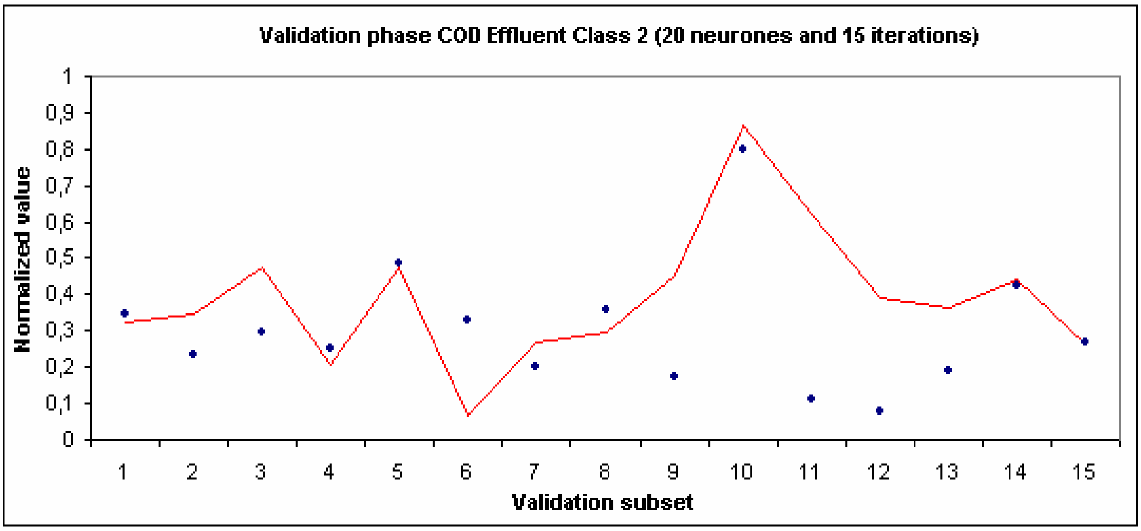


Figure 16. Validation phase for parameter *COD (effluent)* (class 2)

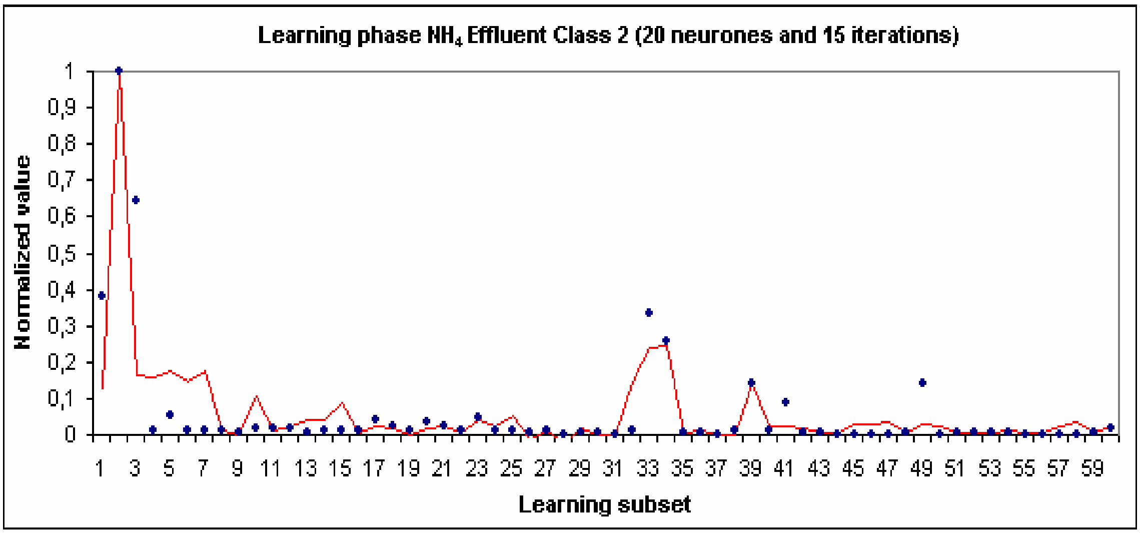


Figure 17. Learning phase for parameter *NH_4 (effluent)* (class 2)

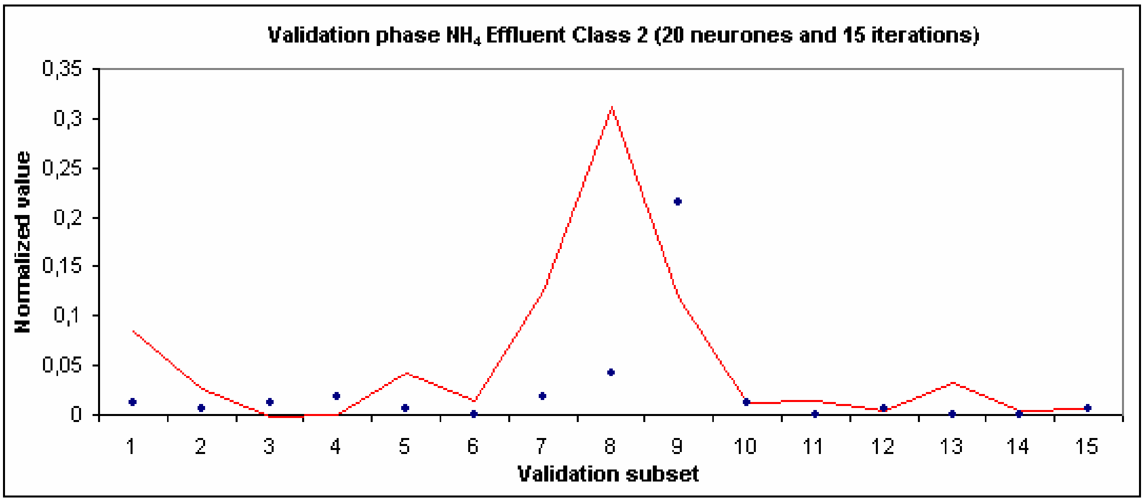


Figure 18. Validation phase for parameter NH_4 *(effluent)* (class 2)

Three important remarks can be made on the results before concluding this work :

- The error levels show that NH_4 prediction is more difficult than *COD* one. The predictive character of the variables used as neural network inputs is more pronounced and thus more efficient with the *COD* parameter.
- The influent parameters prediction is more difficult than the effluent one. The error rate is generally more significant because the biological process efficiency has an important influence on the effluent parameters prediction.
- For parameter NH_4 *(effluent)* (class 2), the average relative error made during learning and validation phases is very weak. In fact, this parameter has weak variations which make easier its estimation.

IV. Conclusion

In this paper, a prediction procedure was presented, based on a multi-layer neural network, to obtain a *COD* and NH_4 on-line estimation for *WWTP* influent and effluent. These parameters are not directly measurable and can only be evaluated by laboratory analyses which can be relatively long (a few days for the *COD*). Knowledge of these parameters, at *WWTP* entry and exit, is fundamental, and makes it possible to estimate the correct working of the organic pollution removal process. Their estimation by neural networks was carried out using real data obtained from the Saint Cyprien *WWTP*.

A preprocessing phase by *K-Means* clustering and principal components analysis completed the neuronal prediction phase in order to improve the results obtained from real data characterizing a strongly nonlinear process. A classification by *K-Means* clustering allows us to highlight significant differences in plant operation and thus to carry out fractional training which facilitates the phenomena comprehension by the neural network.

A too great number of variables used as network inputs can further overfitting. The principal components analysis was interesting in order to reduce, without information loss, this number of variables used as *MLP* inputs.

The choice of a single network making it possible to predict both chemical oxygen demand and ammonia content is confirmed by the predictions quality and can be justified by the desire to obtain the most possible compact model.

Compactness and transportability were privileged to the detriment of a possible precision gain for the results. The predictions quality is sufficient to provide exploitable information by operators supervising the Saint Cyprien *WWTP*. The precision is satisfactory because it allows to know how the system is evolving and how to react. The model simplicity and its low cost facilitate its integration in supervision and control systems already existing in the great majority of such plants.

One of this research significant constraints was to facilitate the system installation on various wastewater treatment sites. An easily reproducible methodology allowing to automatically design the software sensor from a new learning base respects this constraint. Such a methodology must include the data preprocessing carried out prior to the prediction phase.

The possibility of installing the system on another plant depends, however, on the on-line measured parameters and on the frequency of laboratory analyses. In case of significant changes in the system state, the model could need a recalibration in order to be always able to provide good predictions. This recalibration is carried out within a relatively short time using a procedure reserved for this purpose.

V. References

Charalambous C., 1992, *Conjugate gradient algorithm for efficient training of artificial neural networks*, IEEE Proceedings, 139, 301-310.

Cibois P., 1983, *L'analyse factorielle : analyse en composantes principales et analyse des correspondances*, PUF, Paris.

Davalo E. and Naim P., 1993, *Neural Networks*, Editorial Macmillan Computer Science Series.

Durand G. and Jacquinot B., 1995, *Les boues industrielles : traitements et valorisation*, Cachan : Technique et documentation.

Degrémont, 1989, *Memento Technique de l'eau*, 9eme édition.

Eckenfelder W. and Musterman J., 1995, *Activated sludge treatment of industrial wastewater*, Technomic.

Grieu S., Traoré A., Polit M. and Colprim J., 2003, *Estimating wastewater BOD_5 and ammonia using simple on-line operational data by applying neural networks - A case study over the simulation benchmark WWTP layout*, ITEE'2003 First International NAISO Symposium on Information Technologies in Environmental Engineering, Gdansk University of Technology, Poland.

Hagan M. T. and Menhaj M., 1994, *Training feedforward networks with the Marquardt algorithm*, IEEE Transactions on Neural Networks, 5, 989-993.

Hamed M. M., Khalafallah M. G. and Hassanien E. A., 2004, *Prediction of wastewater treatment plant performance using artificial neural networks*, Environmental Modelling & Software.

Henze M., Grady Jr C. P. L., Gujer W., Marais G. V. R. and Matsuto T., 1986, *Activated Sludge Model No. 1*, IAWQ Scientific and Technical Report No. 1, IAWQ, London.

Hertz J., Krogh A. and Palmer R. G., 1991, *Introduction to the theory of neural computation, computation and neural systems series*, Addison-Wesley, New York, NY.

Hornik K., Stinchcombe M. and White H., 1989, *Multi-layer Feedforward Networks are Universal Approximation*, Neural Networks, 2, 359-366.

Jain A. K. and Dubes R. C., 1988, *Algorithms for Clustering Data*, Prentice Hall.

Jain A. K., Murty M. N. and Flynn P. J., 1999, *Data clustering : A review*, ACM Computing Surveys, 31, 264-323.

Moller M., 1993, *Efficient training of feed-forward neural networks*, Ph.D thesis, Computer Science Department, Aarhus University, Denmark.

Morineau A. and Aluja-Banet T., 1998, *Analyse en composantes principales*, CISIA-CERESTA.

Philippeau G., 1986, *Comment interpréter les résultats d'une analyse en composantes principales*, Institut technique des Céréales et des Fourages, Paris.

Rosenblatt R., 1957, *The Perceptron : a Perceiving and Recognizing Automaton*, Project PARA, Report 85-460-1, Cornell Aeronautical Lab.

Schittenkopf C., Deco G. and Brauer W., 1997, *Two Strategies to Avoid Overfitting in Feedforward Networks*, Neural networks, 10, 505-516.