Binary classifiers applied to detect DO sensors faults during washing events

Tatiana Chistiakova[1], Jesús Zambrano[1], Oscar Samuelsson[1,2], Bengt Carlsson[1]

[1]Division of System and Control, Department of Information Technology, Uppsala University, P.O. Box 337, 75105 Uppsala, Sweden. (e-mail: tatiana.chistiakova@it.uu.se; jesus.zambrano@it.uu.se; bengt.carlsson@it.uu.se).
[2]IVL Swedish Environmental Research Institute, P.O. Box 210 60, 10031 Stockholm, Sweden. (e-mail: oscar.samuelsson@ivl.se).

Abstract
In this paper, three classification techniques are applied for monitoring the status of DO sensors in a wastewater treatment plant. In particular, the use of DO sensors readings during washing events are proposed and indication parameters from these events are used for deciding if the sensor is faulty or not. The methods considered are the following: *k*-Nearest Neighbours, Radial Basis Function and Random Forest classifiers. The experimental result indicates that data from washing events could be feasible to use for fault detection of DO sensors and that the Random Forest classifier gives the best classification accuracy.

Keywords
Classification, DO sensors, Fault detection, *k*-Nearest Neighbours, Radial Basis Function, Random Forest

INTRODUCTION

Many modern systems, including wastewater treatment plants (WWTPs) are complex and use an increasing number of on-line sensors for supervision and automatic control, see Chiang et al. (2001). A successful automation in, for example, wastewater treatment plants depends on reliable sensors. The harsh environment where sensors are deployed may, however, cause sensor faults (e.g. bias or drift) or complete failure (no signal). Hence, sensor fault detection is becoming increasingly important in WWTPs given the stringent effluent standards and the goal of resource efficient operation.

The aeration in the biological treatment step is an essential task in a WWTP. It supplies microorganisms with oxygen needed for oxidation of organic matter as well as for nitrification. The aeration intensity is normally governed by a feedback control from dissolved oxygen (DO) sensors, see Åmand et al. (2013). In order to maintain an effective process and to restrict the air consumption, correct DO measurements are required.

A large number of fault detection algorithms have been studied on real and simulated data in case of WWTPs, see Corominas et al. (2011), or Carlsson and Zambrano (2013). One of the common methods to use for fault detection in WWTPs is Principal component analysis (PCA), see Baggiani and Marsili-Libelli (2009) or Garcia-Alvarez et al. (2009). Needless to say, many other machine learning methods are applicable to the problem of fault detection and isolation. In this study, we will apply three binary classification algorithms from the machine learning field which may not have previously been studied in a WWTP application.

The goal of this paper is to identify possible faulty DO sensors using data from the periods when the sensors are cleaned by washing. The assumption is that data from the washing events could be

used to distinguish between faulty and non-faulty sensors. The idea is then to use clustering techniques to group the data in two sets corresponding to the sensor condition.
The paper is organised as follows. First, the data preprocessing is described followed by a detailed description of the methods studied: *k*-Nearest Neighbours, Radial Basis Function and Random Forest classifiers. Then, the results are demonstrated in case of real data from DO sensors washing events. Finally, some conclusion is presented.

THEORY

Preprocessing

The data provided by DO sensors is a sequence of measurements of indication parameters (the amplitude, the time constant, the rise time and the maximum slope). However, all four parameters are measured in different measurement units. In order to get reliable results of the experiment it is advisable to normalize the data to zero mean and unit variance.

Since no labelling is known, i.e. there is no prior knowledge whether the measurements are faulty or not, a clustering procedure is performed. The *k*-means clustering technique results in a set of classes within the data set. Basically, the outcome is binary classified data sets of faulty and non-faulty measurements. It will allow to assign a new measurements with one of the existing classes using binary classifiers later.

The *k*-means clustering technique is the simplest to implement and to run, it divides the data set into a number of clusters, based on samples similarity. In *k*-means, Euclidian distance *D* is used as measure of a similarity, where the distance between two points *p* and *q* is

$$D(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \quad (1)$$

The basic concept of the algorithm is to construct a partition of a set of objects into a set of clusters, where each object belongs to exactly one cluster and the number of clusters is given in advance.

Suppose we have a data set $\{x_1, \ldots, x_N\}$, the idea is to split it into a number of *k* clusters $\{C_1, \ldots, C_K\}$, such that objects within a cluster are similar (or related) to one another and different from (or unrelated to) the objects in other cluster, see Bishop (2006).

In order to define the number of *k* clusters, the Silhouette index is used. The index estimates how similar samples are in one cluster to samples in another cluster, applying distance similarity measures, see Rousseeuw (1986). The algorithm for index calculation is the following:
For *k* number of clusters and for each data point *x:*
Step 1: Find the average distance between *x* and all other points in the same cluster – the measure of cohesion, *a*
Step 2: Find the average distance between *x* and all other points in the nearest cluster – the measure of separation from the other cluster, *b*
Step 3: Calculate the difference between the measure of separation and the measure of cohesion as a Silhouette index:

$$S = \frac{b-a}{\max(a,b)} \quad (2)$$

The larger the value of the index, the better quality of a cluster analysis is obtained. The value varies between 0 and 1. A reasonable clustering structure is considered for indices ≥ 0.51.

When the number of clusters is known, we can proceed with the *k*-means algorithm. The procedure is described as the following:
For finite data set *X* and the number of clusters *k:*
Step 1: Initialize the *k* centroids to be random artificial points
Step 2: Calculate *D* between the centroids and each point in the dataset
Step 3: Calculate the mean of *D* for all points
Step 4: Form clusters
Step 5: Set the new centroids to be the calculated means of step 3
Step 6: Repeat 2-5 until convergence

The method is considered to converge quickly to a local optimum. After the convergence, the data set is separated into *k* clusters with different centroids.

Methods used in the experiment
Given a data set from DO-sensor washing events and the label for each data point (fault or non-faulty) after the preprocessing, we want to apply and to compare several classification methods in order to be able to estimate new incoming measurements correctly within the time.

k-Nearest Neighbours (k-NN)
The k-NN algorithm is one of the fundamental algorithm to implement for classification and has been shown to often work well in practice. It is considered as one of the top 10 data mining algorithm covering the classification, clustering, statistical learning and other areas, see Wu et al. (2007).

k-NN algorithm is a technique for density estimation that can be updated to the classification problem. It is a non-parametric lazy learning algorithm, meaning that no assumptions are made on the underlying data distribution. The lazy characteristic implies that there is no generalization performed in training part, providing a fast training time. The algorithm is distance based, so that the classification of an unknown instance is performed with respect to the class of known instances using similarity, or distance, function. In other words, the procedure consists of three main parts, see Cover et al. (1967):

1. Measure a minimum distance between the unknown instance and a number of its neighbours
2. Take the most frequent class represented among neighbours
3. Assign the unknown instance with a class obtained in Step 2

For the distance metric, the Euclidian distance is commonly applied in case of continuous variables, see Kuncheva (2004).

The choice of *k* is a crucial part of the algorithm. If the chosen value is too small, the probability that the noise presented in the data will cause false results is high. On the other hand, if the value is too big, the computational property of the algorithm becomes too expansive. In case, of binary classification, one of the simplest approaches is to choose *k* to be an odd number to avoid a controversial outcome.

Radial Basis Function (RBF) network
The RBF network corresponds to a neural network approach. It is a popular alternative to a multilayer perceptron network which has a simpler structure and learning methods. The method is based on combinations of fixed basis functions and performs a non-linear classification of the data. The basis functions depend only on the radial distance from a centre of a class, see Bors (2001). The

network aims to measure the similarity between the unknown instance and each known instance and to take the class of the most similar known instance.
In other words, given a number of input vectors $\{x_1 \dots x_N\}$ and the corresponding label vector $\{l_1 \dots l_N\}$, the goal is to uncover a function $f(x)$, that will fit exactly each instance and will be presented as a linear combination of radial basis functions.

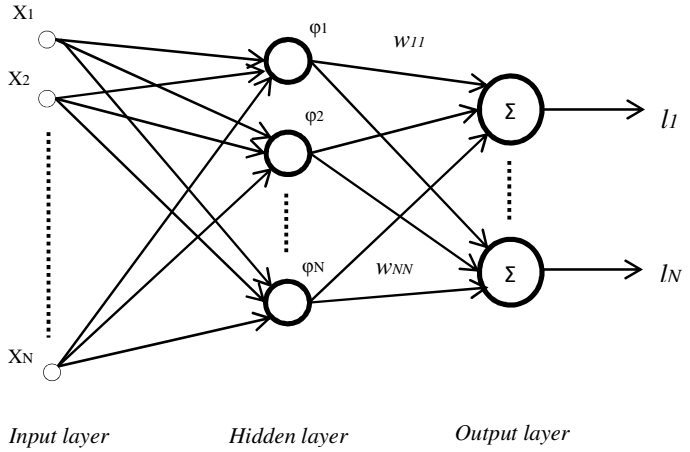

Figure 1. The typical architecture of radial basis function network.

In the Figure 1, the typical structure of RBF network is shown. It consists of three layers, see Ham et al. (2000):

1. An input layer, where the data is introduced to the algorithm
2. A single hidden layer of nonlinear processing neurons
3. A linear output layer

Then, the output of the network is calculated according to

$$f_i(x) = \sum_{k=1}^{N} \omega_{ik} \varphi_k(x, c_k) = \sum_{k=1}^{N} \omega_{ik} \varphi_k(\|x - c_k\|_2) \quad (3)$$

Where $\varphi_k(\cdot)$ is the k[th] scalar-valued radial basis function and ω_{ik} is the weight of the output layer. Therefore, for each neuron k in the hidden layer, the Euclidian distance $\|x - c_k\|_2$ between its associated centre vector c_k and the input vector x to RBF network is computed. Finally, the output of the RBF network is calculated as a weighted linear sum of the hidden layer outputs.

The properties of RBF network are defined by the weights ω_{ik} of the output layer and the centres c_k of the radial basis functions, see Powell (1987). The most common choice is to choose fixed centres, which are randomly selected from the input vector.

Centres are stored in the hidden layer neurons and often called “the prototype vectors”. Each prototype vector is obtained from the training set and is compared to every input vector from the input layer. The measure of similarity (usually, the Euclidian distance) is then calculated. The neuron produces the value between 0 and 1 for each of the input vectors. The higher the value, the more similar the input vector to the prototype, or the centre. This value is also called the activation value as the result of the activation function.

The nodes of the output layer represent the categories that are needed to be defined. Hence, each output node calculates a score for every category, which is the weighted sum of the activation

values, gained in the previous level. A classification decision is taken based on the assigning the input vector to the category with the highest value.

Random Forest (RF) algorithm
The RF algorithm is presented as an ensemble learning procedure, where several learning algorithms are applied to get better predictions. RF algorithm is robust to outliers in training data and accurate in the output. Particularly, a set of decision tree algorithms is used and the mode of the label output by each individual tree is given as an output label, see Breiman (2001).

The aim of a decision tree learning is to classify observations to a certain target value. Each node in the tree represents a set of attributes, or parameters, while each branch corresponds to one of the possible classes for this attribute. The algorithm starts with all attributes contained in the root node. A new sample is classified by starting at the root node, where a set of attributes is stored, then moving down the branches to next node with a new set of attributes. The procedure is repeated until a leaf node is reached. The classification decision is made by pushing a new sample down the tree, and assigning it with the label of the leaf node, see Rokach and Maimon (2008).

In order to make a correct decision which branch to follow, the best split is calculated. The best split is based on the information gain calculation. The information gain *G(A,L)* of an attribute *A* from a set *L* is obtained as following:

$$G(A,L) = E(A) - \sum_{v \in labels(L)} \frac{|A_v|}{|A|} E(A) \quad (4)$$

where *labels (L)* is the set of possible values for labels *L* and A_v is the subset of all attributes *A* for which the label *L* has the value *v* and *E(A)* is an entropy, that characterizes the impurity of an label collection of instances:

$$E(A) = \sum_{i=1}^{r} - p_i \log_2 p_i \quad (5)$$

where p_i is the proportion of *A* characterizing the class *i.*

Therefore, the information gain *G* is the knowledge about the target function value, given the value of some other label, see Mitchell (1997).

In case of RF algorithm, the above described procedure is repeated over all trees in the ensemble with different set of attributes used, and the average vote of all trees is taken as a final decision.

DATA
The data used for the experiment is taken from DO sensors installed in one of the aerated zones of a treatment line in a full scale WWTP in Bromma, Sweden. The DO sensors are automatically and periodically cleaned.

The data consists of DO readings from a number of washing occurrences. From each washing occurrence a number of indication parameters are estimated, in order to characterize each washing event. The washing gives an impulse response of the measured DO concentration.

The estimated indication parameters are the amplitude, the time constant (time to 63% of the step response), the rise time (time from 10% to 90% of the step response) and the maximum slope of the rise. A typical response of the DO sensor during a washing event is shown in the Figure 2.

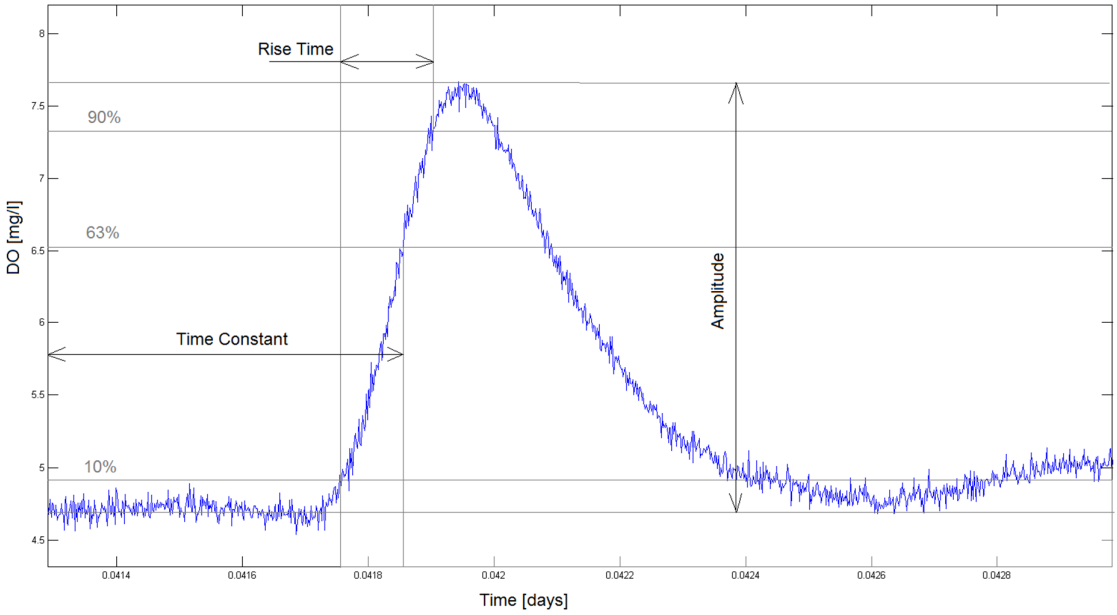

Figure 2. A typical DO sensor response to a washing event

RESULTS AND DISCUSSION

After the preprocessing with *k*-means clustering, two clusters in the data set are obtained. Hence, the labels for each of the data value are provided. The data set is split in the training and the test data sets for performance estimation of the classification methods.

The classifiers use the label information from the training data set to predict the class for data values from the test data set. In case of k-NN classifier, the predictions are made using the number of neighbours equals to 19.

Each classification procedure is run a number of times, typically in the order of 100. Results are estimated using the average accuracy, which is the proportion of the total number of predictions being correct according to the *k*-means clustering.

The purpose of the experiment is not only to study the performance of binary classifiers to a specific problem, but also to explore the importance of the number of indication parameters used in this fault detection task.

In the first part of the experiment, all four indication parameters are used: the amplitude, the time constant, the rise time and the maximum slope of the rise. While in the second part, the set of only two parameters, the amplitude and time constant, is processed. The choice of these parameters is justified by their reasonability and efficiency.

Table 1 shows the average accuracy for every classifier obtained during both parts of the experiment.

Table 1. The mean accuracy for each of the classifiers for 4 (amplitude, time constant, rise time, maximum slope of the rise) and 2 (amplitude, time constant) parameters used

Classifier	4 parameters	2 parameters
RF	**0.9937**	**0.9934**
k-NN	0.9781	0.9675
RBF	0.7731	0.7252

The RF classifier gives the best accuracy, probably as it is less sensitive to outliers. Moreover, the tree-based model corresponds to a sequence of binary decisions applied to the individual input indication parameters, which makes it easy to interpret.

Also, the k- NN algorithm provides good results. This nonparametric method is flexible in terms of the distribution forms. In addition, the classifier model can be updated online quickly.

The RBF classifier gives the worst results among selected methods. The RBF network treats all indication parameters with equal importance. However, this might not be the case in the studied data set.

All three classifiers give a similar performance when the number of indication parameters is decreased from four to two. For the illustration of the classification results, the plot of two obtained classes by each of the classifier is shown in the Figure 3 (the two parameters case).

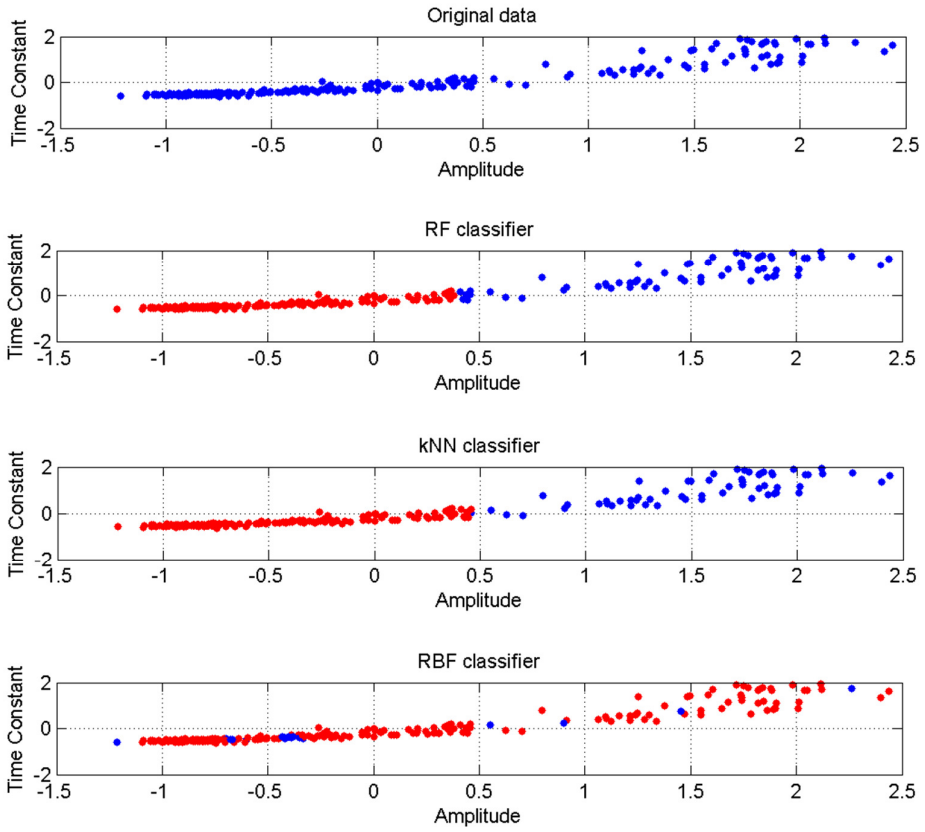

Figure 3. Classification results for 2 parameters (Amplitude vs. Time Constant). Normalized data set

CONCLUSION

Information from washing events was used to classify faulty and non-faulty DO sensor measurements using binary classifiers. In the study, it was found that it is enough to only use two indication parameters which simplifies the classification problem.

To conclude, binary classifiers seem to be feasible to detect DO sensors faults during washing events. Three well-known classifiers were studied. For experimental data used in this study, the RF classifier was found to give the most accurate classification result. The k-NN classifier had a slightly lower accuracy than the RF. The RBF classifier gave the lowest accuracy probably because this method treats all indication parameters with equal weights.

ACKNOWLEDGMENTS

The authors acknowledge funding support under the European Union's Seventh Framework Programme managed by the Research Executive Agency (REA) (FP7/2007_2013), Grant Agreement N.315145 (Diamond). Funding from Käppala Association, Syvab and Stockholm Water Company, Foundation for IVL Swedish Environmental Research Institute, and the Swedish Water and Wastewater Association is gratefully acknowledged. We would also like to thank the plant personal at Bromma WWTP for all crucial help in conducting the experiments used in these paper.

REFERENCES

Baggiani, F., Marsili-Libelli, S. (2009). Real-time fault detection and isolation in biological wastewater treatment plants, *Water Science and Technology*, 60(11), 2949–2961

Bishop, C. M. (2006). Pattern Recognition and Machine Learning, Springer-Verlag New York, Inc.

Bors A.G. (2001). Introduction of the Radial Basis Function (RBF) Networks, *Online Symposium for Electronics Engineers,* issue 1, vol. 1, DSP Algorithms: Multimedia, pp. 1-7

Breiman, L. (2001). Random Forests, *Machine Learning,* issue 0885-6125, vol.45 pp. 5-32

Carlsson B., Zambrano J. (2013). Fault detection and isolation of sensors in aeration control systems – the airflow ratio method, Technical report, Uppsala University

Chiang, L.H. and Braatz, R.D. and Russell, E.L. (2001). Fault Detection and Diagnosis in Industrial Systems, Springer London

Corominas, L., Villez, K., Aguado, D., Rieger, L., Rosén, C., Vanrolleghem, P. (2011). Performance evaluation of fault detection methods for wastewater treatment Processes, *Biotechnology and Bioengineering*, 108(2), 333–344

Cover T.M., Hart P.E. (1967). Nearest neighbor pattern classification, *IEEE Transactions on Information Theory* 13 (1): 21–27.

Garcia-Alvarez, D.; Fuente, M. J.; Vega, P.; Sainz, G. (2009). Fault detection and diagnosis using multivariate statistical techniques in a wastewater treatment plant, *7th IFAC International Symposium on Advanced Control of Chemical Processes,* 952-957

Ham, F. M. and Kostanic, I. (2000). Principles of Neurocomputing for Science and Engineering, McGraw-Hill Higher Education

Kuncheva L. I. (2004). Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience

Mitchell, T. (1997). Machine Learning, McGraw Hill

Powell M.J.D. (1987). Algorithms for Approximation, Clarendon Press

Rokach L., Maimon O. (2008). Data Mining with Decision Trees: Theory and Applications, World Scientific Publishing Co., Inc.

Rousseeuw, P. J. (1987). Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis, *Computational and Applied Mathematics* 20: 53–65

Wu X., Kumar V., Quinlan R. J., Joydeep G., Yang Q., Motoda H., McLachlan G., Ng A., Liu B., Yu P. S., Zhou Z., Steinbach M., Hand D. J., Steinberg D. (2007). Top 10 algorithms in data mining, *Knowledge and Information Systems*, 14(1), 1-37