International Environmental Modelling and Software Society (iEMSs)
2012 International Congress on Environmental Modelling and Software
Managing Resources of a Limited Planet, Sixth Biennial Meeting, Leipzig, Germany
R. Seppelt, A.A. Voinov, S. Lange, D. Bankamp (Eds.)
http://www.iemss.org/society/index.php/iemss-2012-proceedings

Monitoring the performance of Soft Sensors used in WWTPs by means of Formal Verification

L. Luccarini [a], **D. Pulcini**[a], **D. Sottara**[b], **S. Bragaglia**[b] **and P. Mello**[b]

[a] *ENEA – UTVALAMB–IDR, Via Martiri di Monte Sole 4, 40129 Bologna (BO), Italy (luca.luccarini@enea.it, dalila.pulcini@enea.it)*

[b] *DEIS, University of Bologna, Viale Risorgimento 2, 40136 Bologna (BO), Italy (davide.sottara2@unibo.it, paola.mello@unibo.it, stefano.bragaglia@unibo.it*

Abstract: In Waste-Water Treatment Plant (WWTP) automation, “soft” sensors might be used in conjunction with “hard” sensors to improve the reliability of the measurements, or even to replace the latter when they would be too expensive or difficult to maintain. Unfortunately, many soft sensors are created using black-box data mining techniques such as neural networks or Bayesian networks. These algorithms approximate the relation between simpler, more easily available data and the desired “sensed” quantity. However, they are usually dependent on the training data and cannot always generalise correctly when processing completely different inputs. Like their hardware counterparts, then, soft sensors may have input validity ranges. Moreover, they may be subject to “failures” when analysing inputs for which the training algorithm could not capture the input-output relation correctly. Due to their black-box nature, it is quite difficult to obtain a 100% accurate soft sensor and even more to debug it. So, in our approach, we propose to deploy a soft sensor together with a dedicated monitoring sub-system that processes the inputs and the outputs of the sensor itself. This monitor, created using a specific type of rules supporting the concept of “expectation”, applies some logic criteria to define whether a particular sensing is acceptable or not for the purpose of the application using the soft sensor. We will discuss different types of criteria, both qualitative and quantitative, and how they impact the confidence in the estimated measurements. As a use case, we will present a soft sensor for the estimation of the nitrogen compounds in the aeration tank of a 500 litres pilot scale WWTP. Its performance, both in presence and in absence of the monitoring system, will be compared to a real nitrogen sensor placed in the same tank.

Keywords: Soft sensors, WWTP, Monitoring

1 Introduction and Related Works

Wastewater treatment plants are complex systems and the measurement of nutrients and chemical-physical parameters is very important to understand the behaviour of the biological processes involved and to implement an automatic management of the plant. Automation is often considered costly and it is rarely included in the initial design, but an adequate use of the ICA (Instruments, Control and Automation) can make a plant to run consistently and economically. However, an automatic control system requires reliable measurements to produce reliable control. Today, the instrumentation technology is mature and complex instruments – like on-line in-situ nutrient sensors – are now regularly mounted on larger plants, where the costs/benefits ratio is sustainable. Many of those sensors, however, are still too expensive and need too much maintenance to be used in smaller WWTPs. Moreover, in many cases the plant operators are not sufficiently qualified to use and maintain instrumentation systems Olsson et al. [2005]. In such cases, software sensors are an economical alternative to hardware sensors. The problem, then, is to build an accurate model to estimate measures, with an adequate confidence interval and compatible with any additional requirements of the specific deployment domain.

Software sensors can be divided into three classes according to the model used by the sensor for estimation: mechanistic, black-box and hybrid or gray-box models (James et al. [2000]). The mechanistic model – such as the ASM family (Henze et al. [2000]) – often lack statistical identifiability. The gray-box approach is preferred to a mechanistic one, because it often consists of models with a reduced number of parameters needing estimation than the statistical methods. The black-box models are likewise popular, because they do not require detailed prior knowledge about the system. For example, Cecil and Kozlowska (Cecil and Kozlowska [2010]) propose an *Observer-Based Estimator* (*OBE*) for a Biodenipho process. This sensor combines the prediction of a measured state variable and the correction of that prediction, based on the difference between the measurement and the prediction itself. Using this technique, it allows to estimate the ammonium plus ammonia concentration (NH4), the maximum nitrification rate, the nitrate plus nitrite concentration (NO) and the maximum denitrification rate in the tank.

Black-box estimation techniques provide a description of the process behaviour through input-output data mapping and are commonly used for nonlinear modelling. Typical application are based on *Artificial Neural Network*s (*ANN*s), multivariate statistical methods and *Partial Least-Square* (*PLS*) regression. ANNs have the ability to accurately model complex non-linear systems without the need for mechanistic modelling of complex behaviour. The accuracy of neural network modelling has also made it possible to effectively use *Modelling Predictive Control* techniques for the control of biological process (James et al. [2000]). For example, an ANN-oriented approach has been used to estimate the trend of the nitrogen and phosphorus removal in a *Sequencing Batch Reactor* (Luccarini et al. [2002]), to detect the effective duration of the phases in a SBR process, (Aguado et al. [2009]). Multivariate statistical methods, instead, may include component analysis, such as *PCA* (*Principal Component Analysis*) and *ICA* (*Independent Component Analysis*), or Partial Least Squares regression (Ferrer et al. [2008]).

A key assumption in supervised machine learning is that the data used to train a predictor is representative of the data that the predictor itself will later encounter. However, the data gathered from real processes can vary over time. Examples of this include seasonal or weekly changes in inputs to a WWTP. Even more importantly, it is assumed that any relation to be approximated will not change significantly over time, so that similar inputs will roughly correspond to similar outputs. This principle does not always hold, especially in biological plants where the dynamics of interest depend on bacterial populations which are subject to fluctuations over time, not to mention catastrophic events (such as a toxic influent injection) which can alter the population abruptly. Using a static model in such domains is inadequate as the data is exhibiting a phenomenon known as "concept drift" (Tsymbal [2004]). When the characteristics of the inputs and/or the system change significantly, there is no way to guarantee that any model will always generalise correctly. Moreover, it is often assumed that enough data has been given to the training algorithm to obtain a reliable model. Wastewater treatments plants, instead, are a good example of domains where data are hard and/or expensive to acquire (e.g. through chemical analysis): it usually takes a few years to get the training set necessary to create a *good* model. However, it might take less to acquire a partial training set, sufficient to create a *decent* model. A soft sensor trained using a partial dataset, then, might be deployed and used to some extent much earlier than a fully validated one, even if it is more likely to suffer from concept drifts. The only assumption is that the missing data are not so different from the others to require a completely different model (an even more radical phenomenon known as concept *generation* - Tsymbal [2004]).

In literature, there exist two main approaches to deal with concept drift. The first does not attempt to identify when drift is occurring but continuously and regularly updates the classifier, assuming that this will allow the classifier to handle the drift whenever it occurs. This is the most common approach and can be considered a "continuous rebuild" approach. The second explicitly detects when a change in concept is occurring and only then adjusts the classifier. This is achieved by monitoring the value of some indicators and rebuilding the classifier only when the values change significantly. This can be considered a "triggered rebuild" approach (Lindstrom et al. [2011]). While the former is more reactive and transparent to the sensor users, it requires additional care to avoid including noise and temporary disturbances in the rebuild process; the latter instead decouples the detection of the anomaly condition from the policy chosen to deal with it, but may delay the decision to a moment when it becomes no longer effective.

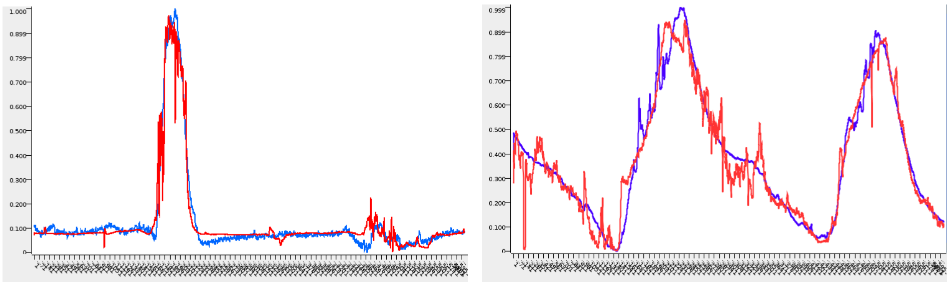
In this paper, we adopt the latter approach and, in particular, focus on a declarative, logic-oriented technique for the monitoring of a soft sensor performance. We take one step back from the recovery policies (not discussed in this work) and focus instead on the criteria and the tools for the constant, real-time monitoring and diagnosis of a soft sensor operational performance. A robust diagnosis, in fact, is recommended to decide whether the sensor's outputs are reliable or not and what should be done when the sensor's performance degrades. Moreover, one should be able to discriminate reliable responses from accidental failures (e.g. due to noise or glitches in the inputs) from chronic situations due to shifts in the input or state data. While in all cases the sensor's output should be given low confidence, if not discarded at all, only the latter situations would require an intervention on the sensor itself.

Section 2 will introduce a concrete use case, where an ANN-based soft sensor is used to estimate the trend of ammonium concentration in the aerobic tank of a continuous flow, pilot-scale WWTP. This sensor exploits indirect signals, such as pH, redox potential (ORP) and dissolved oxygen concentration (DO). Since the plant has been set up only recently, the sensor has been trained using a limited set of data. However, it had to be deployed as soon as possible since the real nitrogen probe had to be relocated for other uses. This problem led to the setup of a monitoring and validation module, whose theoretical background will be presented briefly in section **??**. The module itself, instead, will eventually be discussed in section 3.

2 USE CASE: PILOT-SCALE WWTP

Our studies are being carried out on a pilot plant located close to the municipal WWTP of Trebbo di Reno (Bologna, Italy). The pilot plant is fed with the real nearby plant's municipal wastewater, drawn after the full-scale screening process to remove debris. It has an overall volume of 342 litres and it is equipped with a mechanical stirrer, a variable-flow blower (connected to a membrane diffuser), three peristaltic pumps for influent loading, internal and sludge recycle. The plant is equipped with pH, ORP and temperature probes in the anoxic tank, and pH, ORP, DO, nitrogen ($NH_4 - N$ and $NO_3 - N$) and suspended solids (TSS) probes in the aerobic one. All probe data are sampled and acquired using a stand-alone data logger, at the rate of 1 sample/min. The data is then packaged and sent to a remote node where an *Environmental Decision Support System* (*EDSS*) Cortes et al. [2001] processes and validates it. In our implementation, the EDSS is a hybrid, knowledge-intensive component based on the hybrid rule engine Drools[1].

[1]`http://www.jboss.org/drools`



(a) NH_4: actual vs trained (b) NO_3: actual vs trained

Figure 1: Soft Sensor Validation.

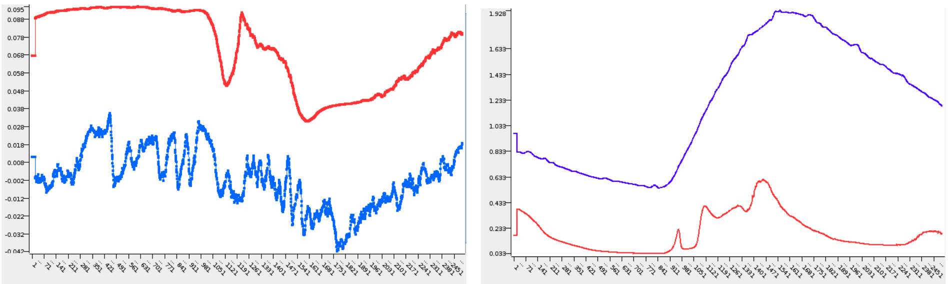
Given the high initial and maintenance costs, we wanted to replace the nitrogen probe with a much cheaper soft sensor. For our first attempt, we have chosen to implement a simple back-propagation, feed-forward artificial neural network Haykin [1999]. The training and validation sets for the ANN have been extracted out of the database of the sensor probe data acquired during the plant's operating periods. Being an experimental pilot plant, it has operated intermittently from late August 2010 to December 2010. Out of this period, we extracted data from the week between November 1st, 2010 and November 6th, 2010, for a total of about 8000 tuples. This period has been characterised by an ideal overall operating condition, with no anomalies in the influent nor in the process, and with no malfunctionings in the probes nor in the data collection system.

After a preliminary analysis of the data set, we identified some correlation between the measured NH_4 concentration and the value of indirect signals (pH, ORP and especially DO) in the aerobic tank. Likewise, we found some correlation between the NO_3 signal and the same aerobic tank measurements, in addition to the pH and ORP measured in the anoxic tank. So, we trained two different neural networks to predict the NH_4 and NO_3 concentrations in the aerobic tank, relying on the much cheaper sensors.

The models have been trained using the data mining tool *Knime* [2], an increasingly popular tool with a good selection of algorithms, a friendly license model and the support for the predictive model exchange format PMML[3]. After some experiments, we settled for two networks with a $3-25-1$ and $5-25-1$ architectures, respectively. Both networks have been trained with a maximum of 1000 complete iterations over the normalized training set, with a final average error below 5%. A comparison of the actual (blue) and predicted (red) normalized NH_4 and NO_3 concentrations are shown in Figure 1. The charts show a subset of the training set, relative to the first two days.

Unfortunately, the good performance of the neural networks are misleading. The chosen training set is not a good representative of all possible operating conditions: in fact, we have deliberately chosen not to include other days in order to build an "imperfect" sensor with a limited knowledge, simulating a partial training condition. The resulting estimators have then been tested on other data sampled in different periods: when the environmental characteristics are compatible with the ones in the days chosen for training, the ANN sensors can effectively produce satisfactory estimates of the internal state of the biochemical processes in the aerobic tank. Unfortunately, these conditions have a high variability, significantly affecting the sensor's performance. An example is shown in Figure 2: here the NH_4 concentration trend is correctly estimated, if not for an absolute 10% offset error. The NO_3 concentration, instead, is erroneous: as one can see from

[2] http://www.knime.org/
[3] http://www.dmg.org/



(a) NH_4: actual vs predicted (b) NO_3: actual vs predicted

Figure 2: Soft Sensor Prediction.

the chart, the real value is well above the ranges typical of the training set (the maximum level of the normalized signal is around 2), making it harder for the sensor to predict.

Rather than deciding whether the sensor should be deployed or discarded, we argue that it's possibleto deploy the sensor, provided that it's paired with a monitor, which has the task of evaluating expectations on the sensor's inputs and outputs. The monitor's evaluations, in turn, are used by the DSS to validate the outputs and to decide *if* and *when* the sensor should be deactivated, incrementally retrained or replaced altogether.

3 Soft Sensor Monitoring

The knowledge-based monitor we propose is implemented using *Event-Condition--Expectations rules* (*ECE rules*, Bragaglia et al. [2011a]). ECE rules are an extension of the common *EventConditionAction rules* (*ECA rules*) used in event driven architectures and active database systems. ECA rules typically consists of three parts: (i) an *event*, (ii) a *condition* and an *action*. The event denotes the signal that triggered the invocation of the rule, the condition is a logical test that, when satisfied, causes the execution of the action (e.g. updates on data). Expectations extend ECA in the sense that they allow to express the expected behaviour of a system in presence of certain triggering event and contextual preconditions. Actions, then, are defined to be performed in case of fulfilment or failure of expectations triggered by events. Notice that expectations may be nested to cope with complex behaviours and it is also possible to define corrective actions to recover the status of the system from an inconsistent state.

The integration between an ECE-based monitor and a PMML soft sensor is seamless within a Drools engine instance. Following the approach described in Sottara et al. [2011], the model can be exported in PMML format and deployed within a Drools knowledge base, where it will be converted automatically into a set of facts and rules which, together, emulate the behaviour of the predictive model. The model's inputs and outputs are inserted and retracted from the working memory at runtime and so can be used to trigger rules, including the monitor's. For each input or output data field defined in the model's `MiningSchema`, a java bean extending the system class `DataField` is created to store the value and some additional metadata, including confidence and validity status. `DataField` are also events, so they can be used with temporal reasoning. Moreover, the particular PMML implementation also allows to intercept and reason over the internal model's data structures: in the case of neural networks, for example, the output of all hidden neurones is also available as data. The model evaluation actually results from a join between the input data and the model's parameters and exploits the truth-maintenance capabilities of the production rule engine. Thanks to this feature, any fine-grained update of the model will be reflected immediately on its outputs, without the need to redeploy the sensor nor to halt the system.

We will assume that the rules will not be modified: instead, the monitor's rules will only be triggered by the sensor's data. In practice, however, there is no limit to the type and kind of rules which can be written, applying any available knowledge: for this reason, we propose a few basic properties to categorise the various rules.

- **Input vs. Output** The monitor will be triggered by the sensor's input rather than by its output. Such rules can be executed before the sensor's model itself is evaluated.
- **Point-in-time vs. Aggregate** Constraints can be placed on individual or aggregate values (e.g. max, average), accumulated over a sliding window of varying size.
- **Qualitative vs. quantitative** Expectations can be expressed on the measured values, or require a classification process to identify more high-level properties.
- **Single vs. Cross Signal** Some properties can be evaluated using a single signal, others require comparisons between two or more.
- **Context Agnostic vs. Specific** Some rules are invariant w.r.t. the specific application context and can be adapted simply by changing the value some parameters. Others, instead, are valid only under some assumptions which should be verified, statically or at runtime.

The simplest, and context agnostic, monitor rules involve a quantitative check of the point-in-time values of each input and output signal. Rules such as `monitor1` (see 1) guarantee that values are physically admissible (e.g. pH values must be within the range $0 - 14$), so they should be applied to both input and output values. Rule `monitor2`, instead, checks that the actual values also fall within the normal plant operating ranges, so it is typically an output validation rule. The sensor inputs need a similar validation, but the intervals would be dictated by the range of values the sensor has been calibrated on (the model training set).

Listing 1: Validity interval verification

```
rule "monitor1"
when
    $d : DataField( name == "pH", $val : value, valid == true )
then
    expect Number( doubleVal >= 0.0 && <= 14.0 ) from $val
end

rule "monitor2"
when
    $d : DataField( $name : name, $val : value )
    Range( dataField == $name, $min : min, $max : max, valid == true )
then
    expect Number( doubleVal >= $min && <= $max ) from $val
end
```

If the sensor model is capable of self-assessing the confidence range for its output, that value can be checked as well by adding another rule. A more complex monitor pattern, instead, is presented in listing 2. Here, it is assumed that there is a direct correlation between an input and an output signal that the soft sensor is supposed to preserve. The correlation is too weak to be evaluated quantitatively, so it's restated in a weaker qualitative version that simply checks that there is no negative correlation when a positive one is expected (or vice versa). The rule relies on the custom, pluggable accumulate function `trend` which performs a linear regression of the data within the time window and returns the coefficient of the approximating line when the confidence value is above a threshold.

Listing 2: Validity interval verification

```
rule "weak correlation"
when
    $trend1 : Number() from accumulate (
      DataField( name == $x1, $val1 : value, valid == true ),
      trend( $val1 ) ) over window:time[30m]
    $trend2 : Number() from accumulate (
      DataField( name == $x2, $val2 : value, valid == true ),
      trend( $val2 ) ) over window:time[30m]
then
    expect Number( this > 0 ) from $trend1 * $trend2
end
```

Rules such as the one presented in listings 1 and 2 monitor the values presented to and generated by the soft sensor. The monitor, by definition, is passive and does not affect the data themselves. However, it generates events whenever an expectation is `Fulfilled` or `Violated`. In particular, the latter events can be directly used to mark data-fields as invalid, preventing their propagation. Moreover, violations themselves can be reasoned over: a rule such as `chronic problem` in listing 3 can discriminate occasional violations, likely in a noisy environment, from chronic situations requiring further diagnostic actions.

Listing 3: Failure handling

```
rule "Invalidate"
when
        Violation( $cause : tuple[0] )
    $d : DataField( valid == true ) from $cause
then
    modify ($d) { setValid( false ); }
end

rule "Chronic problem"
when
    ... // bind $type to a sensor type
    Number( intValue > 20 ) from accumulate (
            Violation( $cause : tuple[0] ) over window:length[30m]
        $d : DataField( name == $type ) from $cause,
        count ($d)
    )
then
        ... //
end
```

4 Conclusions

In this paper, we have presented a knowledge-based monitoring systems which constantly evaluates the performance of a soft sensor. Due to its rule-based implementation, the monitor is completely modular and can be easily extended (or reduced) according to the specific user's requirements. The monitor can observe any of the sensor's inputs, outputs and internal state, applying policies as simple as checking thresholds or as complex as applying domain-specific, diagnostic knowledge. The immediate extension of this approach will be the development of corrective policies to improve the sensor's performance when it starts to degrade. From this perspective, the continuous rebuild and the sensor replacement approaches are simple, non-informed policies. Given that the sensor and its parameters are available as facts, it will be trivial to implement adaptive policies. The other main improvement we plan to introduce involves the switch from a crisp, boolean monitor to a softer set of rules based on fuzzy or many valued logic Bragaglia et al. [2011b]. Many of the monitor criteria would be better expressed by graded degrees of conformance, which would allow to discriminate between minor, occasional failures and chronic failure conditions, admitting any situation in between. The corrective policies, then, could take this degree into account, to modulate the intensity of the corrective actions.

ACKNOWLEDGMENTS

We wish to thank HERA s.r.l. for providing financial and logistic support for this work, which was also partially funded by the Department of Computer Science, Electronics and Systems "DEPICT" project.

REFERENCES

Aguado, D., J. Ribes, T. Montoya, J. Ferrer, and A. Seco. A methodology for sequencing batch reactor identification with artificial neural networks: A case study. *Computers & Chemical Engineering*, 33(2):465–472, 2009.

Bragaglia, S., F. Chesani, E. Fry, P. Mello, M. Montali, and D. Sottara. Event condition expectation (ece-) rules for monitoring observable systems. *Rule-Based Modeling and Computing on the Semantic Web*, pages 267–281, 2011a.

Bragaglia, S., F. Chesani, P. Mello, M. Montali, and D. Sottara. Fuzzy conformance checking of observed behaviour with expectations. *AI* IA 2011: Artificial Intelligence Around Man and Beyond*, pages 80–91, 2011b.

Cecil, D. and M. Kozlowska. Software sensors are a real alternative to true sensors. *Environmental Modelling & Software*, 25(5):622–625, 2010.

Cortes, U., M. Sanchez-Marre, R. Sangueesa, J. Comas, I. R.-Roda, M. Poch, and D. Riano. Knowledge management in environmental decision support systems. *AI Communications*, 14(1):3–12, 2001.

Ferrer, A., D. Aguado, S. Vidal-Puig, J. Prats, and M. Zarzo. Pls: A versatile tool for industrial process improvement and optimization. *Applied Stochastic Models in Business and Industry*, 24(6):551–567, 2008.

Haykin, S. *Neural networks: a comprehensive foundation*. Prentice-Hall, 1999.

Henze, M., W. Gujer, T. Mino, and M. Van Loosdrecht. Activated sludge models asm1, asm2, asm2d and asm3. Scientific and Technical Report 9, 2000.

James, S., R. Legge, and H. Budman. On-line estimation in bioreactors: A review. *Reviews in Chemical Engineering*, 16(4):311–340, 2000.

Lindstrom, P., B. MacNamee, and S. Delany. Drift detection using uncertainty distribution divergence. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 604–608. IEEE, 2011.

Luccarini, L., E. Porra, A. Spagni, P. Ratini, S. Grilli, S. Longhi, and G. Bortone. Soft sensors for control of nitrogen and phosphorus removal from wastewaters by neural networks. *Water science and technology*, (45):101–107, 2002.

Olsson, G., M. Nielsen, Z. Yuan, and A. Lynggaard-Jensen. Instrumentation, control and automation in wastewater systems. Scientific and Technical Report 15, 2005.

Sottara, D., P. Mello, C. Sartori, and E. Fry. Enhancing a production rule engine with predictive models using pmml. In *Proceedings of the 2011 workshop on Predictive markup language modeling*, pages 39–47. ACM, 2011.

Tsymbal, A. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin*, 2004.