VTT INFORMATION TECHNOLOGY

RESEARCH REPORT TTE1-2000-28

MODUS-project
Case Study WasteWater

Prediction of quality indicators based on measurement history

Version 1.1-1

16.10.2000

Mikko Hiirsalmi

Version history

Version	Date	Author(s)	Reviewer	Description
1.0-1	29.9.2000	Mikko Hiirsalmi		First version
1.0-2	2.10.2000	Mikko Hiirsalmi	Esa Rinta-Runsala	Delivered to steering group 5.10.2000
1.1-1	13.10.2000	Mikko Hiirsalmi	Ilkka Karanta, Esa Rinta-Runsala, Antoni Wolski	Corrected version, still confidential.

Contact information

Mikko Hiirsalmi
VTT Information Technology
P.O. Box 1200, FIN-02044 VTT, Finland
Street Address: Tekniikantie 4 B, Espoo
Tel. +358 9 4561, fax +358 9 456 6027
Email: Mikko.Hiirsalmi@vtt.fi
Web: http://www.vtt.fi/tte

Last modified on 16 October, 2000
T:\Modus\Cases\MetsaSerla\Raportti\Prediction\rr2k29-c-ww-pred2.doc

Abstract

We describe work aimed at applying prediction methods for detecting abnormal conditions in quality measurements. The methods predict the future values of a few quality indicators of a process based on multivariable histories of past values of sensor readings. The models have been created empirically on the basis of a temporal history of both online and laboratory measurement time series. The test environment for the methods has been an industrial waste water cleaning plant using an activated sludge method for cleaning the water.

The aim has been to develop methods that may be implemented as online analysis tools at the process control environments and are capable of being interfaced with existing data sources. By creating a demonstration application using the prediction capabilities, we have shown how the measurements can be stored in a RapidBase main memory active database and how the neural network models may be used online for creating predictions of future values.

Various problem formulations have been suggested. They were based on classifying the process states and on predicting future values by way of time series prediction methods. Mostly artificial neural network models have been applied to the problem.

The generated models have been evaluated with various measures evaluating the classification performance and the prediction capability of the system. Classification error, root mean squared error and relative error have been used as evaluation criteria. Results obtained from the best artificial neural network models seemed promising but upon considering the stringent target limits on abnormality the prediction errors were too large. This may be caused by the limited amount of data available, the long time span of the prediction, and the dynamic behaviour and unobservable status (living bacteria) of the underlying process.

Tiivistelmä

Käsittelemme laatukriteerien poikkeamien ennustamiseen tarkoitettujen ennustemallien kehittämistyötä. Käytetyt menetelmät perustuvat prosessin muutaman laatuindikaattorin tulevien arvojen ennustamiseen muiden prosessimittausten ajallisiin historia-arvoihin perustuen. Mallit on rakennettu empiirisesti oppimalla sekä tosiaikaisten että laboratoriomittausaikasarjojen muodostamasta esimerkkiaineistosta. Menetelmien testiympäristönä on ollut teollisuuden biologista puhdistusta käyttävän jätevesien puhdistuslaitoksen mittaushistoria.

Tavoitteena on ollut kehittää menetelmiä, jotka voidaan toteuttaa tosiaikaisina analyysityökaluina prosessin valvontaympäristössä ja jotka voidaan liittää olemassa oleviin tietovarastoihin. Olemme kehittäneet mallisovelluksen, joka osoittaa, miten luodut ennustemallit voidaan liittää RapidBase keskusmuistipohjaisen tietokannan yhteyteen.

Raportissa esitetään vaihtoehtoisia ongelmanasetteluja perustuen sekä epänormaalien tilojen luokitteluun että aikasarjaennustamiseen. Ongelman ratkaisuun on pääasiallisesti käytetty hermoverkkomalleja.

Kehitettyjä ennustemalleja on arvioitu eri mittareilla, kuten niiden kyvyllä erottaa epätoivotut tilat normaaleista ja ennusteiden tarkkuden suhteen. Arviointikriteereinä on käytetty luokitusvirhettä, ennustevirheen varianssia ja suhteellista virhettä. Parhaiden hermoverkkomallientuottamat tulokset vaikuttivat lupaavilta, mutta ottaen huomioon annetut tiukat rajat epänormaaliuuden tunnistamisessa ennustevirheet osoittautuivat liian suuriksi. Aineiston rajallinen määrä, ennusteiden pitkä aikajänne ja prosessin dynaaminen luonne ja näkymätön sisäinen tila (elävän bakteerikannan tila).

Table of Contents

1 Introduction

We consider the problem of predicting the future behaviour of quality indicators of a chemical process by creating prediction models from a historical measurement time series. One of the main aims is to be able to predict abnormal situations in advance so that corrective actions could be taken in time. The prediction task is considered more thoroughly in the case of an activated sludge waste water cleaning system.

1.1 Problem formulation

The problem of detecting abnormal situations may be formulated in different manners motivating alternative approaches to the problem:

? One approach is to try to forecast the continuous-valued outputs of the time series variables a few time steps into the future using time series prediction methods. On top of such prediction capability, one may build decision support tools for simulating the process and for making "What-If"-scenarios of the process behaviour when some of the values have been changed.

? Another approach is to build a classifier assigning each of the process conditions into typical process states based on the historical measurement records available at the time of the classification task. This way the system could warn the user about potential problems before the quality indicators show problems. Examples of good and bad behaviour are needed for building classifiers. The state labels could be normal or abnormal states identified based on the value of the desired quality indicators after the prediction time interval has passed (future values).

? Often in process environments normal behaviour clearly dominates the available process data meaning that examples of abnormal situations is much more rare. Here one may try to model the normal behaviour of the system by clustering the measurement data records into similar prototype vectors, which then describe the typical states of the system. Then one may monitor the distance of each new state from the closest prototype and if the distance is above a given threshold value, an abnormal state has been identified and more detailed analysis methods may be started. [Iiva97]

? Another possibility is to cluster the temporal states into typical prototypes using a topology preserving clustering algorithm like SOM and then to label these prototypes according to the portion of abnormal states in the system. Such a SOM map may be used to visualize the trajectory of the system states while monitoring the system and also an alarm may be issued when a critical state is reached. [Alho99]

? Clustering may be also used to divide the system states into typical ones and then to create a local prediction model for future situations based on the examples assigned to the typical states. Based on the amount of examples available the models may range from simple nearest neighbour averaging methods or linear regression models to nonlinear neural network models.

1.2 Description of the activated sludge cleaning process

Our test case, called WasteWater, concerns two biological subprocesses of an activated sludge waste water cleaning system being used at the Kirkniemi factory of Metsä-Serla, a Finnish wood-processing company. Here we shall briefly describe the relevant aspects of the process studied.

Activated sludge waste water cleaning is a complex process consisting of mechanical, biological and chemical subprocesses. The waste water is continuously fed to the process and then it flows through the process for days. The considered process consists of two partly connected cleaning lines both containing an aeration basin and a sedimentation basin. Before entering these lines, the waste water has been mechanically cleaned, its acidity has been neutralized and additional nutritions have been added to the system. The cleaning is based on allowing the bacteria, protozoans and other micro-organisms to eat organic waste as nutrition.

In the aeration basins additional oxygen is dissolved into the water so as to keep the process aerobic, and thus to allow and accelerate the purifying bacterial consumption process. The micro-organisms and the organic waste form a biosludge, which is extracted from the water in the sedimentation phase. A selected part of the sedimented sludge is circulated through the process back to the aeration basin to be reactivated in order to reach a predefined sludge age (meaning the average amount of time that a particle is kept in the circulation). To maintain stability of the process and good quality of the sludge, excess sludge is periodically removed from the process and dried. Removing the sludge too soon may produce too much waste to be dried and to be transported to a dumping place. Cycling the waste water too long in the system consumes excess cleaning capacity.

The circulation process is slow, a typical delay between the beginning and the end of the circulation process is 2-3 days. Furthermore, the biological aspects of the process have longer delays, typically 10-15 days. An overview of the considered process is presented in Figure 1-1.

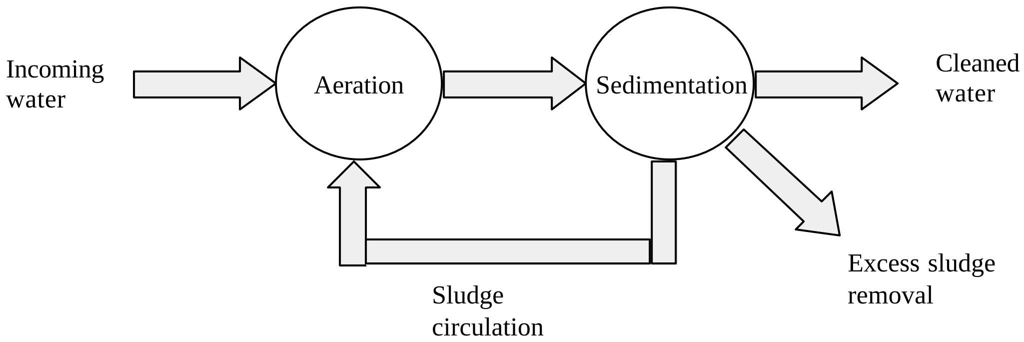

Figure 1-1 Biological waste water cleaning process.

The process operators control the process on the basis of process measurements, such as BOD[1] and COD[2] levels, and prior knowledge. In normal situations, when quantity and quality of the

[1] The biological demand for oxygen (BOD) is a measure of the amount of oxygen used by micro-organisms to decompose the organic matter in the wastewater.

[2] The chemical demand for oxygen (COD) is a measure of the amount of oxygen used to oxidize organic matter and to convert it to carbon dioxide and water.

incoming water is stable, control of the process is easy and well known. In abnormal situations, for example, when chemical concentrations of the input water change rapidly, the process can go out of balance. The biomass reacts slowly to changes in the water quality and quantity, and therefore an out-of-balance situation in the waste water treatment plant occurs several days after the actual cause has occurred. Furthermore, corrective actions are also slow and the results are usually observable only after quite a long delay.

Process measurements are of two basic types: online meterings and laboratory analysis. Online meterings are obtained using sensors and the values from online meterings are available instantly, together with a timestamp. Sometimes online meterings may produce erroneous indications. This is due to fouling of the sensors and drifting of sensor calibration. Typically values from online meterings are available with a time resolution of several seconds, but in this case study averaged values of one hour are used.

Most process quality results are obtained using laboratory analysis. Analysis are performed by taking a sample which is then analyzed in laboratory by chemical experts. Results of the laboratory analysis are entered to the computer system manually with a timestamp coding the time instance the sample was taken. Most of the laboratory analyses are performed in-house, but some analyses are performed in external neutral laboratories. Typically, results of in-house analyses are available within hours (by late afternoon) after the sample period has ended, but external laboratory results can have a delay of several weeks. The sample period may extend for several days.

1.3 Related research

The problem of identifying process conditions in an urban wastewater treatment plant (an activated sludge process) is discussed [West00]. Both parametric (Fisher's discriminant analysis, multivariable regression and two-stage regression) and nonparametric (k-nearest neighbours and kernel density) statistical methods are compared with artificial neural network models (Multi Layered Perceptron (MLP) network, Radial Basis Function (RBF) network and Fuzzy ARTMAP) on the problem of classifying a given process state to a predefined set of normal states. Very good results are reported on the ability to discriminate between abnormal and normal situations with RBF and MLP networks and also with Fisher's discriminant analysis. They investigated the false positive rate (Type I error) and false negative rate (Type II error) and with the best models they achived nearly perfect success (97-100%). The ability to discriminate between 13 typical states was about 90% accurate with the best methods. Their data consisted of 527 daily measurements of 38 sensor readings from one urban wastewater treatment plant. They used 10-fold crossvalidation in order to validate the produced models. However, it remained unclear how the different system states were determined - it seems that they were acquired by some type of manual classification. Also it was not stated what the time span for the forecasts was in the experiments – therefore one would assume that they produced one day forecasts. The authors considered RBF networks as slightly more robust and accurate than the other methods including MLP networks. The other methods were not considered robust enough with the amount of data available.

1.4 Summary of our approach

We have developed time series prediction methods for predicting the future behaviour of quality indicators of a chemical process.

Both traditional time series prediction methods (ARIMA) and artificial neural network models have been applied. The process data considered has been obtained from an industrial waste water cleaning plant using activated sludge cleaning process. Based on our experience it was found that the ARIMA models were not promising in this domain as the data contained no clear seasonality nor trends. Results obtained from the best artificial neural network models seemed promising but upon considering the stringent target limits on abnormality the prediction errors were too large. This may be caused by the limited amount of data available, the long time span of the prediction, and the dynamic behaviour and unobservable status (living bacteria) of the underlying process.

We have demonstrated how the RapidBase platform may be enhanced by adding new analysis modules to it. In this case, we have integrated a neural network execution environment to make time series predictions based on the currently available measurements. The same tool could be used for other types of modeling as well covering classification, function approximation (regression), and clustering tasks. The system could be applied to other time series prediction tasks where one has no exact model of the process but has good process measurement data available.

2 Review of relevant modeling approaches

2.1 Time series prediction methods

Many methods have been developed for performing time series prediction. Statistical methods range from exponential smoothing and ARIMA models to state space models. In data intensive tasks neural network models are often used in time series prediction tasks and also memory based reasoning may be used.

ARIMA (Auto Regressive Integrated Moving Average) methods have been developed as methods capable of dealing with univariate time series that contain seasonal behaviour or trends. Some research systems also have ARIMA models for multivariable time series models called Vector ARIMA models.

When example data is available and there is not enough explicit process knowledge to build a simulation model one may use data mining and machine learning methods to learn a model of the data. Artificial neural network modeling tools are often used to make predictions in such data intensive cases.

The methods may be divided to global and local ones. Global methods try to describe the data with one model of the whole data range. Local models try to divide the learning data into similar subsets and then apply local modeling to each such subset. While using the models the local model best matching the current situation is selected for prediction. The division to local models may be performed e.g. by using topology preserving clustering with SOM (Self-Organizing Map) or by using memory-based reasoning by searching for the k nearest neighbours which then form the local subset. Local models are typically considered more appropriate for situations where the time series are nonstationary [3].

SOM based clustering and local linear models have been applied in [Vesa97] in predicting a univariate chaotic time series. Their method is based on first embedding the temporal dimension to feature vector using a time window of historical values as a delay line. Then SOM clustering is applied to identify a representable prototype set for the delay line feature vectors. Next each of the data records is assigned to their BMUs (best matching units) and a local linear model is calculated for each prototype using the assigned data records or by expanding the context by considering the data records of the topologically neighbouring units as well. Also more complex local models, like MLP networks or splines, could be developed if the amount of local data records is large enough. Good results were reported for the selected chaotic time series test case.

In [Kosk97] a similar approach is described but in this case a temporal version of SOM, called Recurrect SOM (RSOM), was used for the clustering and linear regression for the local models. The method was applied to three test cases including a Mackey-Glass chaotic time series test problem, a laser pulse time series and electricity consumption time series forecasting. The method was compared to an MLP network solution and to AR model. The test results indicate that in this case the RSOM method was not as successful as ordinary SOM in [Vesa97] and MLP was typically the best performing method is these test cases. AR was only better than RSOM with the electricity consumption forecasting case. These problems may be due to implementation problems in the RSOM algorithm.

In [Reic00] next neighbor prediction methods have been generalized to multivariate data in order to predict zooplankton growth in German North Sea. It is claimed that the use of additional variables is useful as they can provide additional information especially while only short time series are available. In their approach the time series is embedded and nearest neighbour methods are used to select the appropriate context for the local model. Center-of-mass (COM) prediction and local linear prediction are used. Good results are reported in their test problem.

2.2 Classification methods

Classification deals with the task of assigning given feature vectors to the best matching predefined label. Teaching may be used to create classification models from given prelabeled example data vectors using supervised learning techniques. In creating classification models the minimized error criterium is typically the overall classification error[4]. Many methods are available for creating classification methods ranging from neural network models to decision trees.

---

[3] Stationarity is the property of a time series in which its statistical properties do not change with time

[4] Classification error = ? $f(y_i, x_i)$ * Cost (y_i, x_i) where $f(y_i, x_i) = 1$ if $y_i <> x_i$ and = 0 otherwise

2.3 Methods for state identification and monitoring

In [Simu99] various ways of using the Self-Organizing Map (SOM) technology are described. It is indicated that SOMs may be used to visualize the operating point (the BMU of the current measurement) or the trajectory (formed as the set of BMUs of the latest measurements). If the SOM map has been created using all measurement data the map units may be labeled according to the abnormality ratio of the measurements whose BMU it is. In this way the visualisation will indicate if one gets to an abnormal state (assuming that some of the prototypes show clear abnormality levels).

In order to detect abnormalities in the process behaviour (fault detection) one may teach the SOM using only measurement vectors describing the normal behaviour , thus creating a mapping of the "normal operation". A faulty situation may be detected by monitoring the quantization error (distance between the input vector and the BMU). Large errors indicate that the process behaviour is different from the normal data that was used for training the map. By defining a suitable threshold we can select the sensitivity for the failure (novelty) detector.

3 Tools and methods applied

Here we shall first introduce the relevant neural network models applied in this case study. After that the commercial neural network tool we have used is described. It takes care of implementing the described neural network architectures. Next we outline the process of building models with the selected tool (NGO). Finally we briefly introduce the VALO prediction software that has also been tested in this project.

3.1 Introduction to neural network models

Neural networks are computational models for distributed processing of input signals. They consist of a network of interconnected simple processing elements mimicking the way brain is thought to be working. Various types of neural networks are available and they are applicable to different tasks. Early applications ranged from autoassociative memories to self-organizing clustering networks. In practice, function approximating and classification networks have become popular in data intensive areas. In the sequel, we discuss several relevant models more precisely.

3.1.1 Multi-Layer Perceptron Networks (MLP)

Multilayer feed forward neural network architecture accepts continuous or discrete inputs and emits continuous valued outputs. Data is fed into the input neurons which fan out the data to one or more hidden layers (internal layers of neurons). Each of the signals from the input neurons is multiplied by a weighting factor and these weighted signals are accumulated by the hidden neurons. A transfer function is applied to these accumulated signals and the hidden neurons distribute their signal to output neurons, where again a weighting factor is applied to each hidden neuron signal. The output neurons take the accumulated signals from the hidden neurons, apply a transfer function and emit the final outputs. There may be multiple hidden layers in the network. An example of the MLP network is presented in Figure 3-1. Theoretically networks with two hidden layers and an unlimited number of hidden nodes have been proven to be a universal function approximator which means they are capable of approximating any reasonable function.

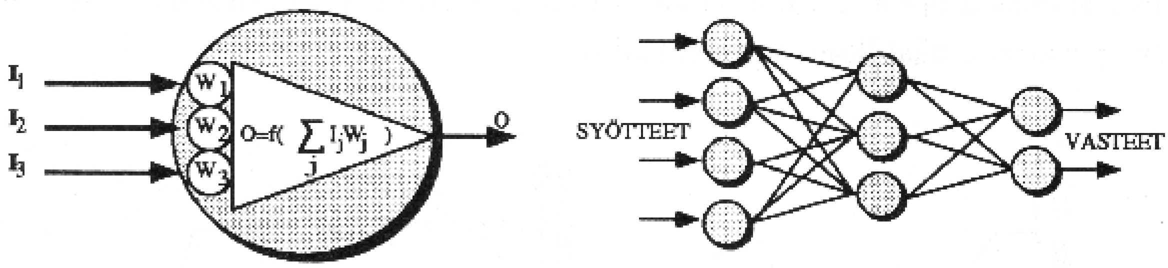

Figure 3-1 The MultiLayer Perceptron network structure on the right and the computational structure of each node in the network on the left.

MLP networks are trained by providing them input values and desired output values. An error function such as the mean squared error is calculated between the outputs of the neural network and the desired output values. This error is back-propagated through the network, and errors are proportioned to individual weights. The connection specific error is then used to determine how to adjust the weights to reduce the error. This process is repeated over and over again until the weights converge to a final value.

One shortcoming of basic MLP networks is that they contain no concept of time. This may be compensated by lagging desired historical lines as additional features of the input vector but the dimensionality of the problem space grows fast.

3.1.2 Time Delay Neural Network (TDNN)

A TDNN network is like an MLP network where there are multiple connections from the input neurons to the hidden neurons and from the hidden neurons to the output neurons. Each of these connections looks back over time and sets its weights for each connection to minimize Mean Squared Error (MSE) of the overall network. It is exactly like lagging the inputs by N periods of time but TDNN does this with hidden neuron outputs too, thus using patterns in the hidden neuron outputs created by the data over time.

In a TDNN, each connection is set to a specific time interval back in time with the first connection set at the current time (current record) and the second connection set to 1 period ago (third connection to 2 periods ago, and so on). This look-back is performed in the NGO tool by providing each neuron with memory, so that it can remember previous layer outputs for N periods of time.

3.1.3 Continuous Adaptive Time Neural Network (CATNN)

CATNN networks are similar to TDNN but their look-back intervals are adaptive. This means that CATNN connections change their look back intervals as learning progresses, seeking phase relationships that produce higher correlation over history.

In NGO look-back is interpolated smoothly across records adding extra flexibility.

3.2 NeuroGenetic Optimizer tool (NGO)

In our work, we have used a tool called NeuroGenetic Optimizer 2.6 (NGO) provided by BioComp Systems, Inc (see http://www.biocompsystems.com). This tool provides an environment for developing such relevant neural network architectures as the MLP architecture using error Back-Propagation (BP) learning algorithm, and two time-extensions to the basic MLP model called Time Delay Neural Network (TDNN) and Continuous Adaptive Time Neural Network (CATNN). NGO also supports SOM (Self Organizing Map), PNN (Probabilistic Neural Network) and GRNN (Generalized Regression Neural Network). These have not been used in this case study.

The tool loads the data from specific data files, takes care of normalizing the input variables appropriately (typical range from -1 to 1) and provides a wizard interface for selecting an appropriate task (regression, classification, time series prediction or clustering). It also provides defaults for the parameters available. The tool provides a genetic algorithm based search method for automatically selecting the best fitting model for data. The models are ranked according to the accuracy of the model on a test set and according to model parsimony (simpler model is better).

The best models may be saved into proprietary model specification files. These models are not open and may only be used with NGO tools like Predictor (an application for making predictions for user-supplied input data or for file data), Penney Excel extension or a user application created with an ActiveX programming interface called MicroPredictor. Both Penney and MicroPredictor are extension tools to the basic modeling tool and must be licensed separately. We have tested the BETA release of MicroPredictor. It allows the programmer to load and unload the desired network model into memory and apply new data into it and gather the results back. One network may be active at one time but multiple applications may use the same DLL at the same time. One must be careful to feed the system the right data in the correct columns as the data is all numbers. It is possible to examine the network structure to some detail but the connection weight parameters are not available. Therefore the actual effects of the connections can not be examined.

3.3 The NGO model building process

At first, data has to be organized into a feature vector format where each record contains mutually relevant feature values for items whose dependencies one wants to model. One should try to select relevant features and possibly calculate new ones based on the raw data. These steps are performed at the preprocessing stage.

Then one has to divide the data into training, testing and validation data sets. Training data is used to teach the network parameters, in NGO testing data is used to select the model with best fitness and validation data is used to evaluate the selected best model for predicting unseen situations. The training sets for time series prediction must be consecutive records in order to preserve the time continuity.

The desired types of models are specified via the manual user interface of NGO (there is no API for the teaching part). NGO tries different architectures to the problem and alters the parameters (which inputs to use, how many nodes per hidden layer to use, what type of transfer functions to use and how far back to look when using the temporal networks) of the models by genetic search. First results are available within minutes but progress may still occur after days depending on the problem size. When adequate models are available they are saved into appropriate files.

The predictions produced by the models may be tested with the help of Predictor tool or via an Excel interface called Penney (which we have not tested). Another possibility is to embed the neural network application into user applications by using an ActiveX API (called MicroPredictor) and provided separately.

3.4 The VALO prediction software

VALO is an internal prediction software developed for making forecasts based on various statistical models. Its model classes range from regression and ARIMA models to exponential smoothing. It is developed with JAVA.

The VALO software contains a model selection part that looks for characteristic features in the history data such as trend or seasonality or outliers. The system is suitable for prediction based on data where there is a clear trend and/or seasonality. Also the ARIMA models adapt quickly to sudden changes so that solitary nonlinear phenomena do not disturb the system. If no characteristic features are found, the system provides a flat estimate that agrees with the level of the last part of the history data.

4 Modeling in the WasteWater case

It was decided to gather relevant online measurements and results from laboratory experiments for two partly connected aeration lines. The task was set to develop prediction models for forecasting the future values of some quality indicators a few days ahead of time. Three days was considered the shortest useful time span but longer range would be useful (7-14 days). In the ilmas3 aeration line there are five online measurements (such as temperature, water flow, oxygen contents) and eight laboratory measurements (measured 2-5 times per week) and also four descriptive calculated values (available after the laboratory values are available). It was decided to gather values for these measurements on a daily basis from the history database. About 390 records were gathered.

The raw measurements required some preprocessing as some of the values were considered clear outliers and many values were missing from the database. Several methods were applied to preprocessing. Outliers were detected and removed by using confidence interval screening. Missing values were interpolated using linear interpolation. In order to reduce random variation in the data it was smoothed by moving average or median filtering. Each variable is preprocessed separately with some combination of these methods. The appropriate method selection and preprocessing results were performed together with a domain expert.

The cleaned data was then extended by adding three new columns to it denoting the future quality indicators. Neural network models were then generated by allowing MLP, TDNN and CATNN architectures compete against each other and allowing NGO to select, which inputs to include to the network and to select the transfer functions and the number of hidden layers and the number of nodes on each hidden layer.

We have created separate models for ILMAS2 and ILMAS3 aeration lines to predict the values of three quality indicators a few days (3 days and 14 days) in advance. The indicators were LIUK_P3 (dissolved phosphorus in the cleaned waste water), CODOUT3 (chemical oxygen consumption of the cleaned waste water) and JSSAK3 (thickness of the cleaned waste water).

The prediction models for all the cases were developed in a similar fashion. At first, the data was preprocessed by smoothing it and by computing some new indicators. We also tested some models with precalculated features such as differences between the current and earlier values of the same variable called D1 (=D_n – D_{n-1}), D3 (=D_n – D_{n-3}) and D7 (=D_n – D_{n-7}) and by computing Haar wavelet coefficients for 4 window sizes (W1, W2, W3 and W4). These features were meant to embed the temporal behaviour of the time series into the same feature vector. In experiments it was found that the differences produced some of the best prediction models – they were even comparable with the specialized time series models that NGO supports. These time series models consider the historical values within a short time window. The wavelets did not produce as good results, and it seems that much more training data would be needed for them to be useful in this case.

Regression and time series prediction methods performed typically best in producing regression results for the quality indicators. The optimized error function was the RMS (Root Mean Squared) error criterium. We also tried to develop some specialized classifiers to predict whether the future values will be normal or abnormal. Here the optimized error function was the classification error. The results were not better than those produced by producing regression predictions at first and then by classifying based on the status of the predicted value (below or above the normal limit).

We also tested traditional ARIMA models on the data using the VALO software. Here the idea was that all the potential explaining variables were individually forecasted into the future and then the software attempted to forecast the quality indicators by forming regression models from the predicted variables to forecast the quality indicators. Few linear dependencies were found and the results were not convincing. As neither seasonality nor trends could be detected from the data the predictions tended to reflect the latest history. Typically when using ARIMA models the prediction models would be always reestimated just before making forecasts (like in consumption forecasts during the budjeting cycle). We estimated that the model updating could take a few hours on modern workstations. Therefore, this updating could be possible to do automatically also in the waste water prediction case as new data is only made available once a day so there is ample time to update the models. However, the results were not good enough to justify further work.

Local prediction models were not tested.

5 Assessment of the produced predictions

In this chapter, we first summarize the characteristics of the quality indicators to be predicted. Then we introduce the evaluation criteria used and finally report on the results acquired. Overall it can be said that the current results tend to predict average values and are not good enough to detect abnormally high values. Some of the phenomena are also faster than the prediction period and therefore such phenomena can not be forecasted.

5.1 Characteristics for the predicted time series

5.1.1 Quality indicators of the ILMAS3 process line

In the following figures, we view time series of the quality indicators in the gathered process data. There are around 390 daily observations of the values. The first 256 were used as training data, the next 64 as test data while building the models and the last 70 as validation data to objectively assess the quality. Ideally the values in all these sets should reflect the same range of values and behaviour; they should be from the same distribution.

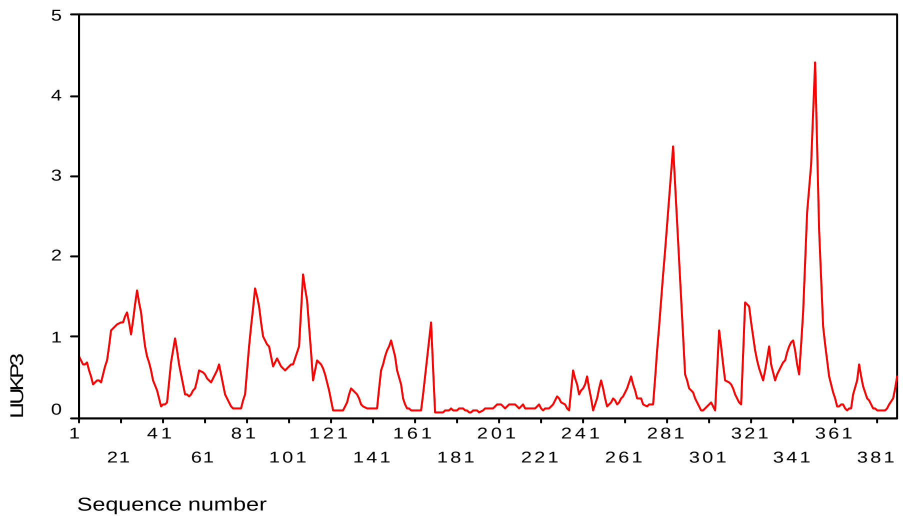

Figure 5-1 The time series of the LIUK_P values

The normal values for LIUK_P3 are from 0.07 to 0.35 mg/l (targeted area is 0.1-0.2). In Figure 5-1, we can see that there are many problematic periods in the data set and they seem to be fairly evenly distributed along the time series so that the distribution of the values is fairly stationary. So the division of the data into training/testing/validation data sets can be justified.

The normal values for JSSAK3 are from 5 to 50 mg/l (the smaller the better). In Figure 5-2, we can see that there are three huge peaks in the training data which may cause some scaling problems in the preprocessing. There are also a few smaller peaks in the data.

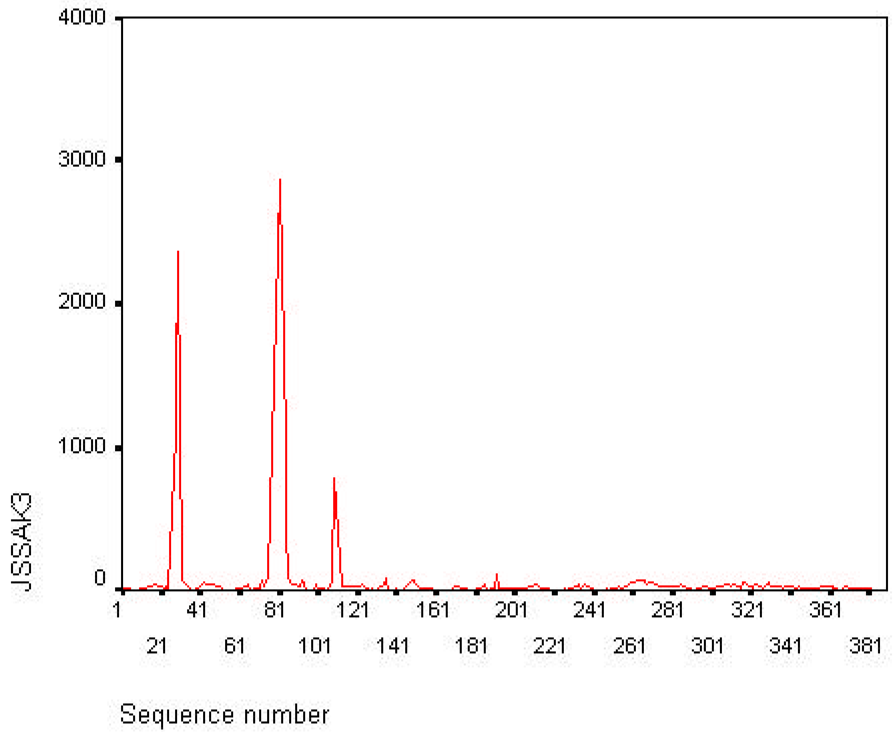

Figure 5-2 The time series of the JSSAK3 values.

The normal values for CODOUT3 are from 130 to 200 mg/l (the smaller the better). In Figure 5-3, we can see that in the beginning of the training data the values are all above normal and there seems to be a decreasing trend. In the testing data, the values are all normal and during the validation period the data contains mostly normal values but some slightly above normal ones. The data is not stationary along the whole sequence, which causes problems with the selected division of the data sets. A better selection of training/test/validation sets could improve the situation.

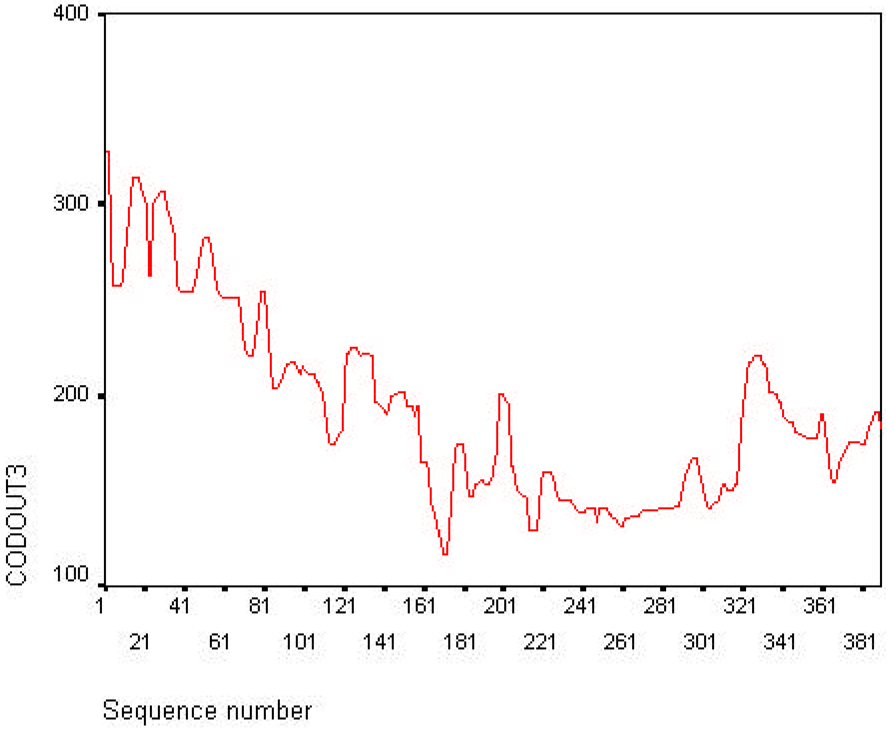

Figure 5-3 The time series of the CODOUT3 values.

5.1.2 Quality indicators of the ILMAS2 process line

The normal values for LIUK_P2 are from 0.07 to 0.35 mg/l (targeted area is 0.1-0.2). In Figure 5.4, we see that the values are fairly stationary along the time. There is only one big peak in the end of the data set.

The normal values for JSSAK12 are from 5 to 50 mg/l (the smaller the better). In Figure 5.5, we see that except for two large peaks in the data the values are fairly stationary.

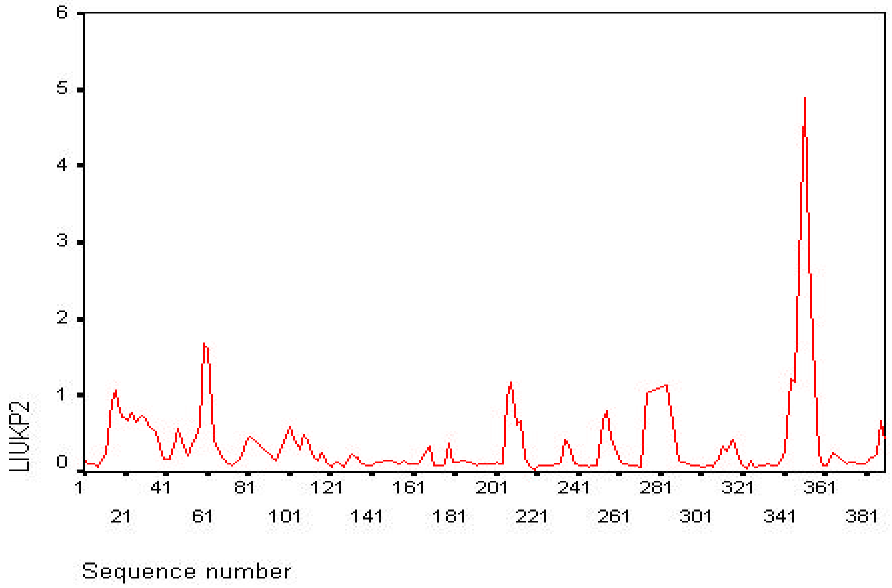

Figure 5-4 The time series of the LIUK_P2 values.

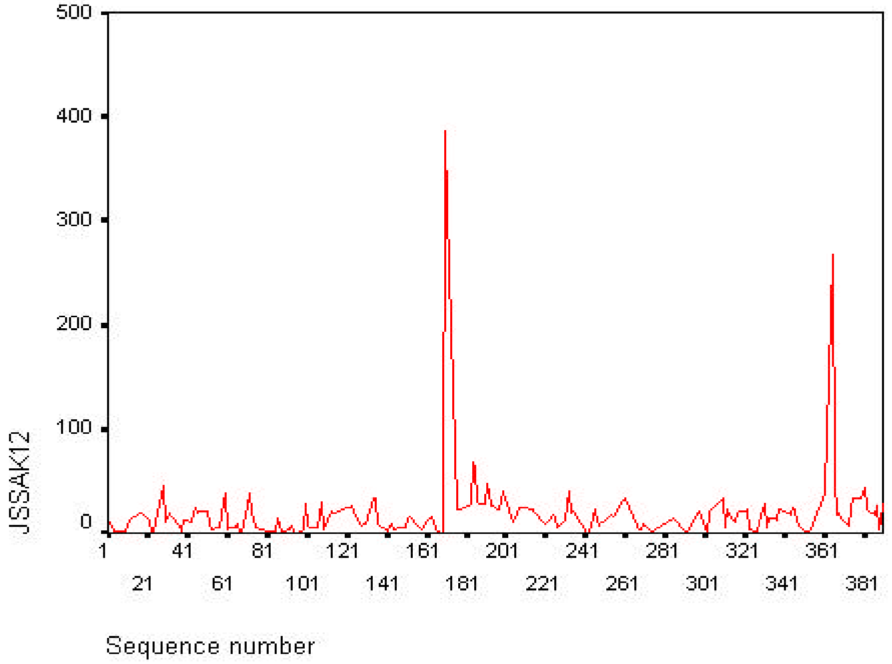

Figure 5-5 The time series of the JSSAK12 values

The normal values for CODOUT12 are from 130 to 200 mg/l (the smaller the better). In Figure 5-6, we see that the data is fairly stationary along the whole sequence, which justifies the selection of training/test/validation sets.

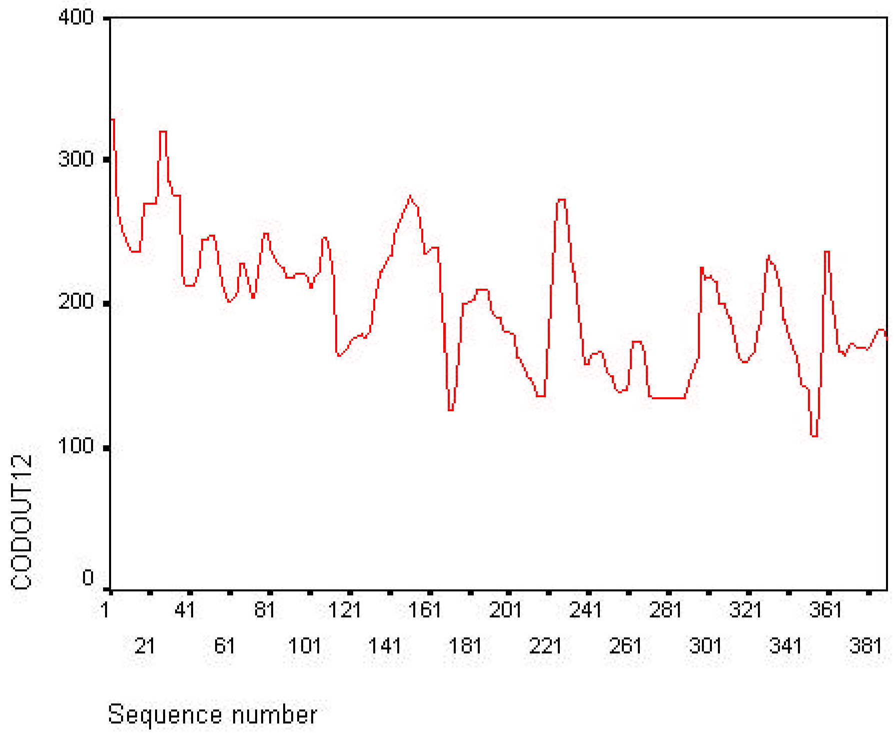

Figure 5-6 The time series of the CODOUT12 values.

5.2 Description of the selected evaluation criteria

The predictions have been evaluated with the following criteria:

Average Absolute Error (AAE) measures the average error of the predictions. The scale is variable dependent.

Relative mean absolute error is AAE that has been scaled by the value range (MAX-MIN) of this variable. (A better measure would be produced by replacing MAX with $MAX_{0.95}$ and MIN with $MIN_{0.05}$ so that the extreme values of the ranges are ignored.) This gives a clear indicator of the inaccuracy level of the predictions. The measure should be as small as possible.

Normalized Root Mean Squared (NRMS) error[5] measures the average squared error scaled by the standard deviation of the values therefore indicating how large the root mean squared error is in standard deviations.

Percent of trends incorrectly predicted measures how large fraction of the 1-step trends have been incorrectly predicted. This tries to determine how well the predictions follow the real values. Just by random guessing 50 percent error is reachable.

---

5 NRMS = RMS / $?_x$

Correlation [6] measures the similarity of the shapes of the original and predicted time series. The values range from –1 to 1 and for perfect predictions correlation is 1.

Classification error measures the fraction of incorrectly predicted instances in the validation set. False positive rate (FP) measures the fraction of incorrect predictions in the set of predicted positives. False negative rate (FN) measures the fraction of incorrect predictions in the set of predicted negatives. Missclassified negatives rate measures how large portion of the negative records have been incorrectly classified. Missclassified positives rate measures how large portion of the positive records have been incorrectly classified.

5.3 Prediction results

We have created models for ILMAS3 aeration line to predict the values of three quality indicators some days in advance. The indicators were LIUK_P3 (dissolved phosphorus in the cleaned waste water), CODOUT3 (chemical oxygen consumption of the cleaned waste water) and JSSAK3 (thickness of the cleaned waste water).

We have evaluated the generated predictions by considering separately results for the two aeration lines (Ilmas3 and Ilmas2). We considered 3-day and 14-day regression forecasts. The models have been created using training data from the beginning period (around 320 records) of the available data set and they have been evaluated by using the ending period (60-70 records) of the data set as validation set.

The prediction capability has been evaluated visually and with the previously defined quality measures. The classification capability has also been addressed.

5.3.1 Predictions of the variables of the ILMAS3 process line

Here we shall only report the results from 14-day and 3-day . 5.3.1.1 Predictions 14 days into the future

See Appendix A for the detailed indicators for the quality of the predictions.

In Figure 5-7 we see how well the training set was learned by the reported prediction model. One sees that the match is almost perfect possibly indicating overlearning (overfitting the training data) which usually means worse performance with unseen data samples.

[6] corr = aver((x_t – aver(x_t)) ($x_{t'}$ – aver($x_{t'}$))) / ? ? '

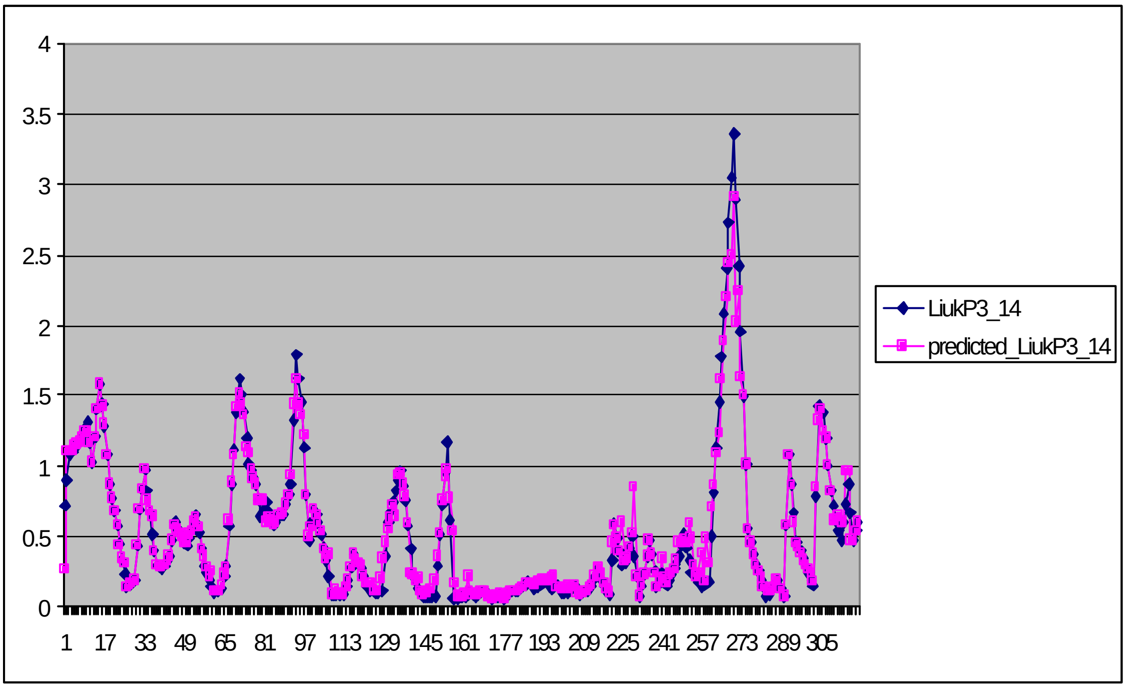

Figure 5-7 The training (256 first records)+test set (64 last records) predictions for Liuk_P3

Next we show the predicted and real future values for the desired quality indicators of the records in the validation data set. Visually it seems that the predicted values follow fairly well the actual values.

In the LiukP3 time series (in Figure 5-8) one sees that a very large peak has occurred for a two week period. The predicted values do not foresee that phenomena. Anyhow, the predicted values are above the high limit during that period. While discussing with the domain expert, it was found that the peak was caused by Midsummer break in the production, which causes the waste water input to temporally end. This causes in a couple of days an increase in the LiukP level. Obviously this can not be determined from raw measurement data at least 14 days in advance. Some of the phenomena are also faster than the prediction period and therefore such phenomena can not be forecasted.

Towards the end of the set of samples the actual values fell well below the high limit except for two small peaks. Most of the time the predicted values are much higher than the actual values during the validation period.

From the quality measures (see Appendix A) calculated, one sees that the correlation between the time series is fairly high (0.65). The NRMS measure is around one standard deviation (?=0.95) and the average absolute error (AAE) is 0.55 which are fairly high considering the alarm limit (0.5). Relative mean absolute error is fairly good (14% from the value range of the actual measurements) because of the large peak period. Obviously, the peak period affects these measures a lot.

If one evaluates the prediction results in terms of their capability to predict abnormally high values, one gets an overall classification error of 28.8% with a 33% false positive rate. In this case, the numbers of values above and within normal values are rather evenly distributed.

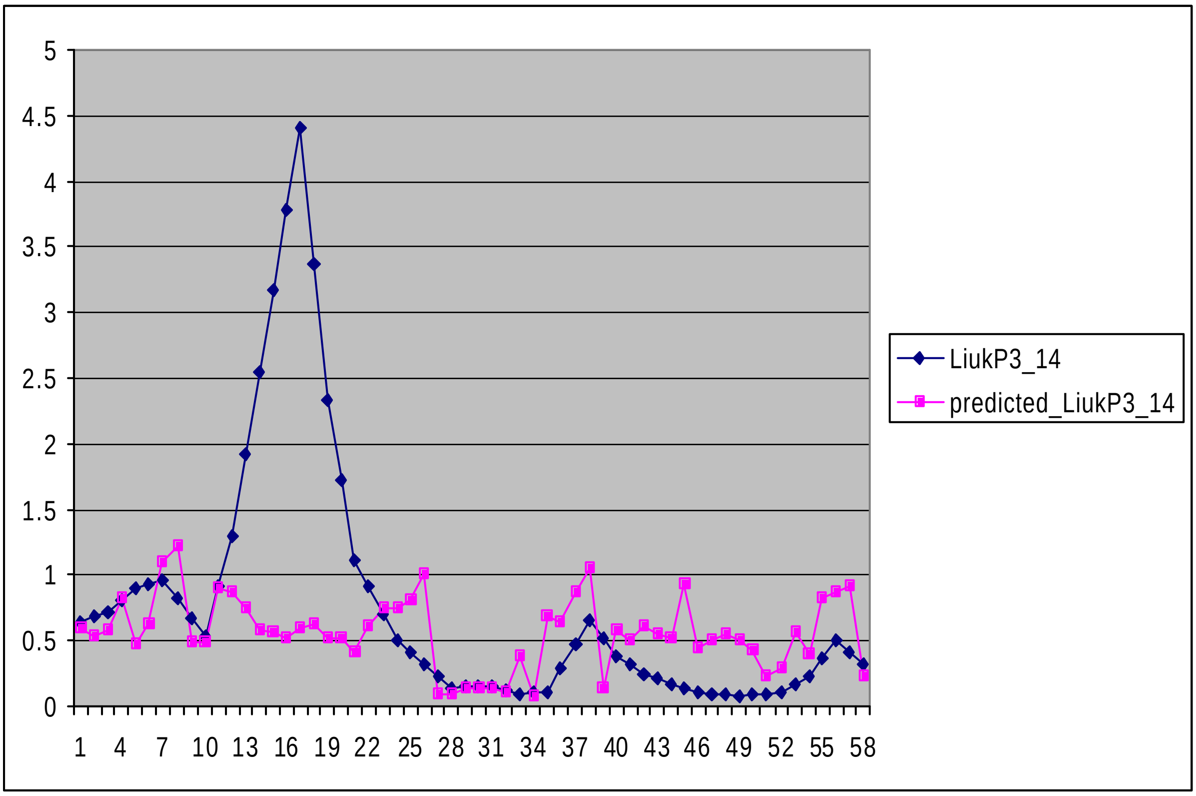

Figure 5-8 Predicted (14 days) and real values for variable LiukP3_14 in the validation set

In the JSSak3 time series (see Figure 5-9) one sees that the measured values are all within the normal band. The predicted values are mostly higher than the actual values but typically within the normal band. However, there are 3 large sharp peaks. These may be caused by the properties of the training data used. The predictions seem to be rather far from the actual values. Some abnormal data would be needed to test the classification performance properly.

From the quality measures (see Appendix A) calculated, one sees that the correlation between the time series is very low and the NRMS measure is around ten standard deviations (?=8.9) and the average absolute error (AAE) is 28.9, which are fairly high considering the alarm limit (50). Relative mean absolute error is very large (98% of the value range) because the prediction errors are large compared to the value range of the measured values.

If one evaluates the prediction results in terms of their capability of predicting abnormally high values one gets an overall classification error of 5.1% with a 100% false positive rate. However, here one must note that the validation data contained no above normal situations and therefore the results are not very reliable for evaluating the classification capabilities.

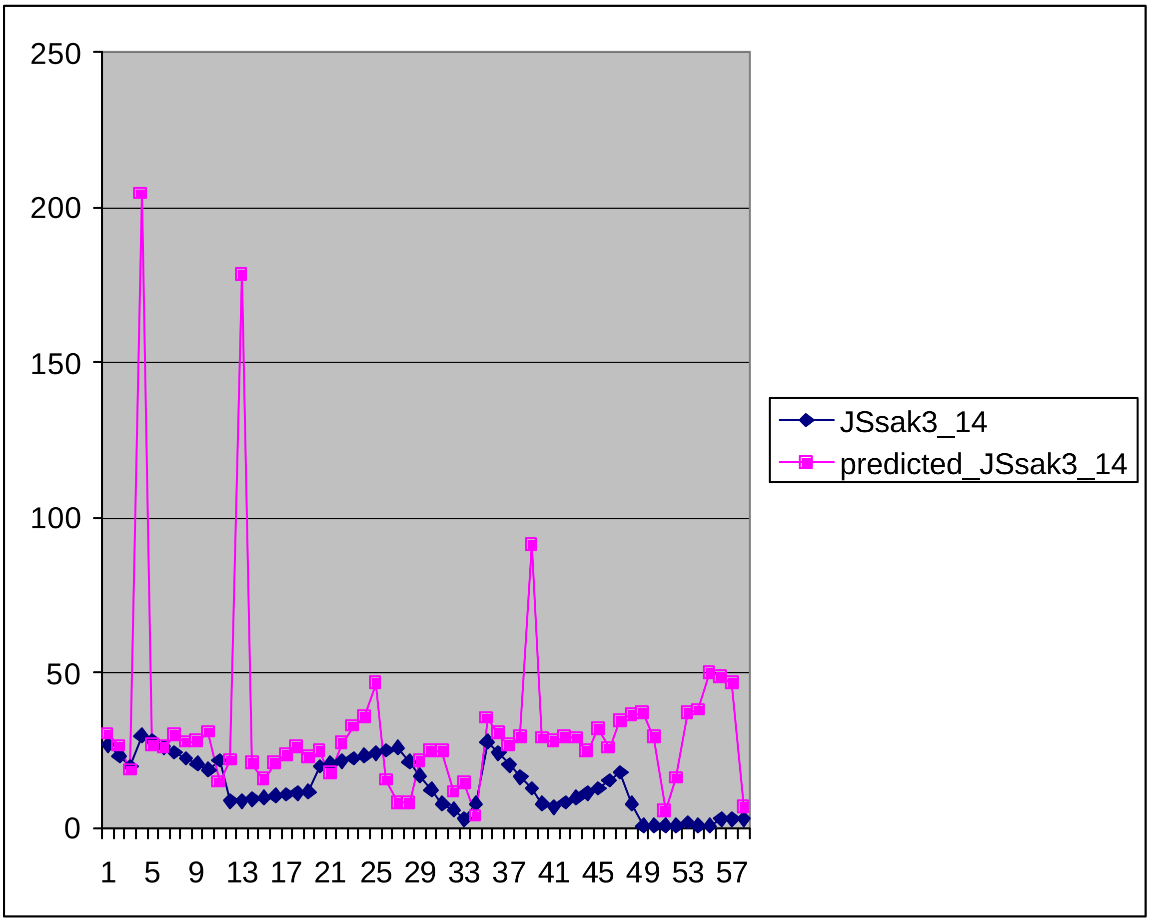

Figure 5-9 Predicted (14 days) and real values for variable JSSak3_14 in the validation set

In the CODout3 time series (see Figure 5-10) one sees that the measured values are all within the normal band. The predicted values seem to reflect the actual values fairly well but there is much more variation in the predictions which might indicate overlearning. The validation set contains only few above normal values.

From the quality measures calculated (see Appendix A), one sees that correlation between the time series is very low and the NRMS measure is within two (NRMS=1.7) standard deviations (?=11) and the average absolute error (AAE) is 14.5, which are fairly good considering the alarm limit (200). Relative mean absolute error is fairly large (31% of the value range).

If one evaluates the prediction results in terms of their capability to predict abnormally high values one gets an overall classification error of 11.9% with a 60% false positive rate. However, here one must note that the validation data contained only few above normal situations, and therefore the results are not very reliable for evaluating the classification capabilities.

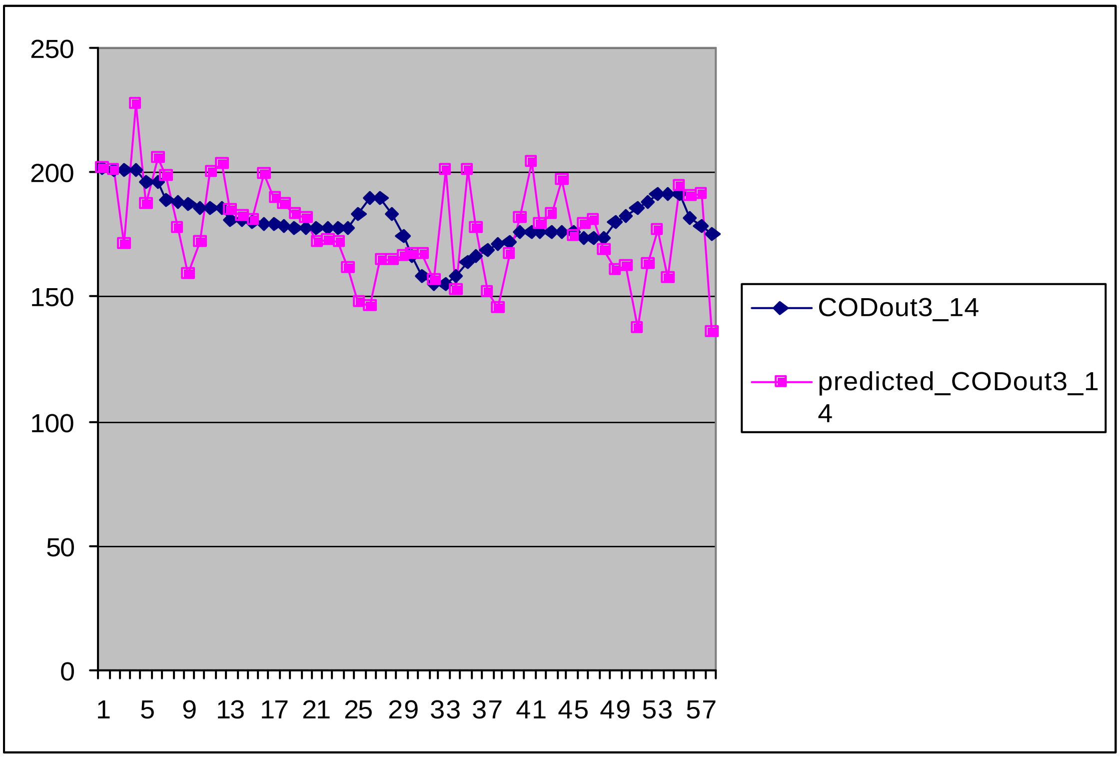

Figure 5-10 Predicted (14 days) and real values for variable CODout3_14 in the validation set

5.3.1.2 Predictions 3 days into the future

Here we show the predicted and real future values for the desired quality indicators of the records in the validation data set. Visually it seems that the predicted values follow fairly well the actual values. In the LiukP3 time series (see Figure 5-11), one sees that a very large peak has occurred for a two week period. The predicted values reflected that phenomena partly: The rising levels were detected only after three days (the prediction interval) but the peak was correctly placed and the end correctly predicted.

From the quality measures calculated (see Appendix B), one sees that NRMS measure is 0.74 standard deviations (?=0.88) and the average absolute error (AAE) is 0.36, which are fairly high considering the alarm limit (0.5). Relative mean absolute error is fairly good (8.3% from the value range of the actual measurements) because of the large peak period. Obviously, the peak period affects these measures a lot.

If one evaluates the prediction results in terms of their capability of predicting abnormally high values, one gets an overall classification error of 24.3% with a 19% false positive rate. In this case, the numbers of above and within normal values are rather evenly distributed.

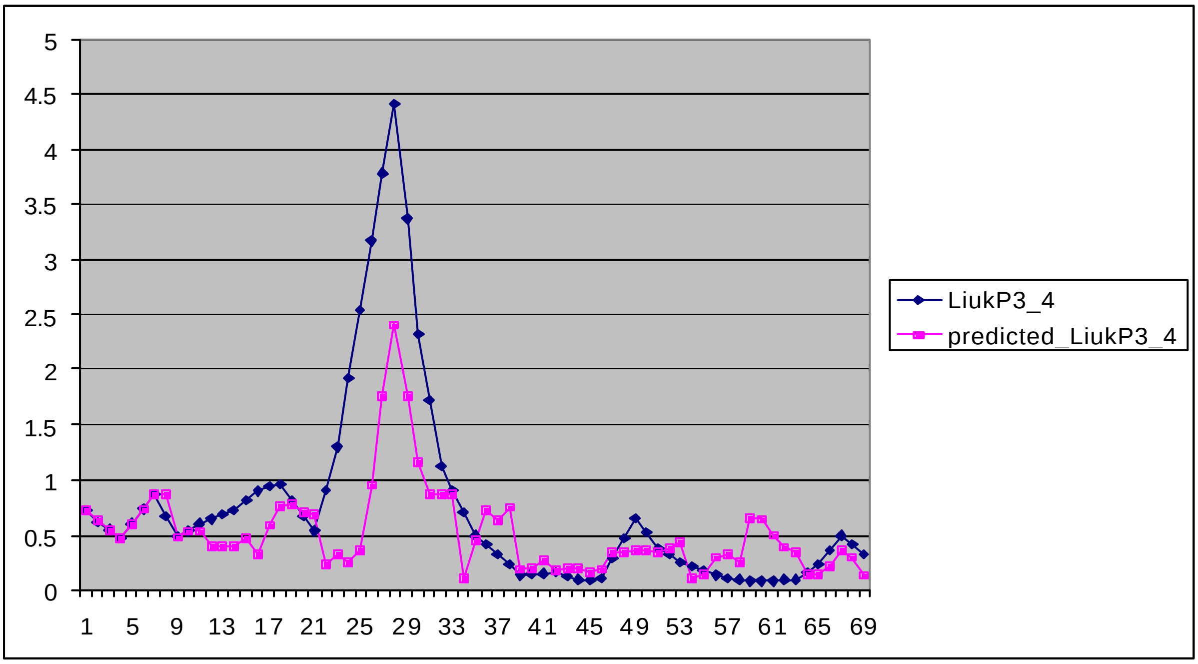

Figure 5-11 Predicted (3 days) and real values for variable LiukP3 in the validation set

In the JSSak3 time series (see Figure 5-12), one sees that the measured values are all (but one) within the normal band. The predicted values do not seem to be able to forecast the real values and seem to be scattered. The prediction capability does not seem to be good.

From the quality measures calculated (see Appendix B),one sees that the correlation between the time series is nonexistent. The NRMS measure is around two standard deviations (?=11.7) and the average absolute error (AAE) is 17.2, which are fairly high considering the alarm limit (50). Relative mean absolute error is fair (29% of the value range) because one of the real measurements is above the normal limits and the range is larger.

If one evaluates the prediction results in terms of their capability of predicting abnormally high values one gets an overall classification error of 21.4% with a 94% false positive rate. Here one must also note that the validation data contained no above normal situations and therefore, the results are not very reliable for evaluating the classification capabilities.

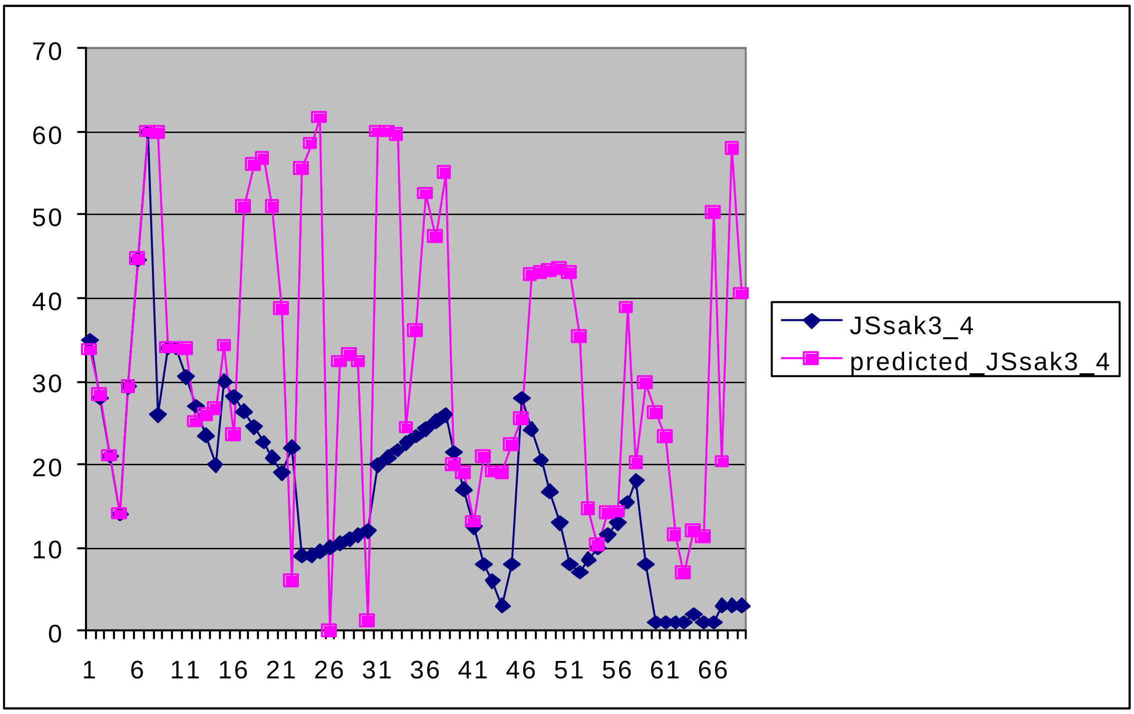

Figure 5-12 Predicted (3 days) and real values for variable JSsak3 in the validation set

In the CODout3 time series (see Figure 5-13), one sees that most of the measured values are within the normal band. The predicted values are not too bad but there is much more variation in the predictions.

From the quality measures calculated (see Appendic B), one sees that correlation between the time series is very low, the NRMS measure is within 2 (NRMS=1.89) standard deviations (?=17), and the average absolute error (AAE) is 26.5, which are fairly good considering the alarm limit (200). Relative mean absolute error is fairly large (40% of the value range).

If one evaluates the prediction results in terms of their capability of predicting abnormally high values, one gets an overall classification error of 12.9% with a 29.4% false positive rate. However, here one must note that the validation data contained only a few above normal situations and therefore, the results are not very reliable for evaluating the classification capabilities.

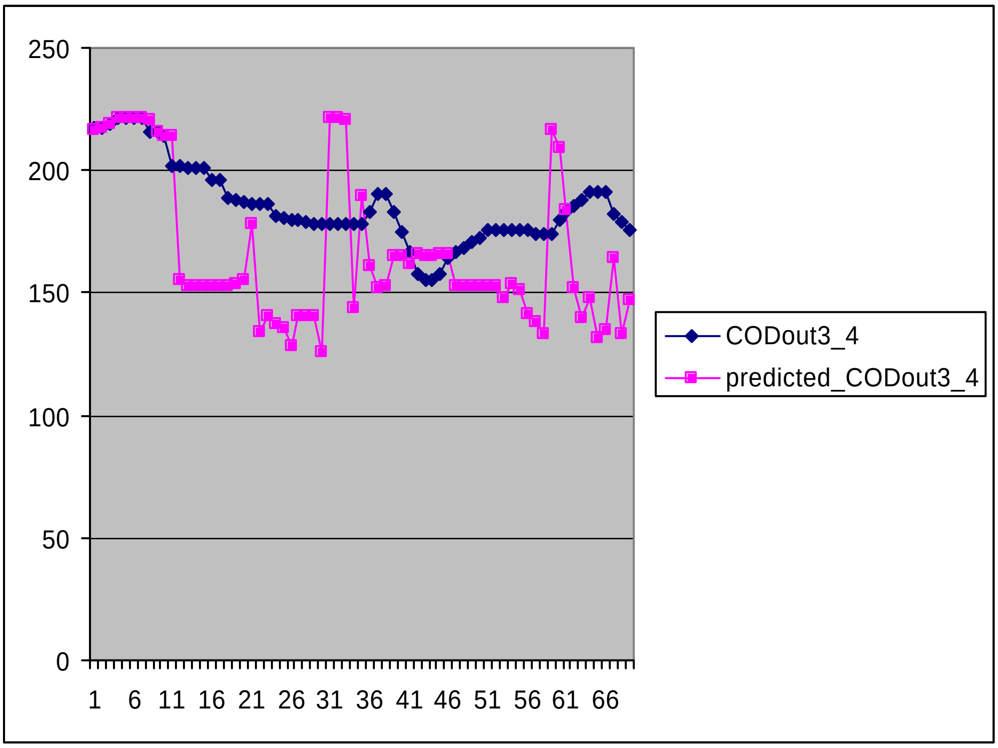

Figure 5-13 Predicted (3 days) and real values for variable CODout3 in the validation set

5.3.2 Predictions of the variables of the ILMAS2 process line

Here we shall only report the results from 3 day predictions as the longer term ones were not significantly different. See Appendix C for the quality indicators for the quality of the predictions.

5.3.2.1 Predictions 3 days into the future

In Figure 5-14, we see how well the training set was learned by the reported prediction model. One sees that even the training set has not been learned perfectly and the peak values are not modeled well. However, there is clear correlation between the peaks.

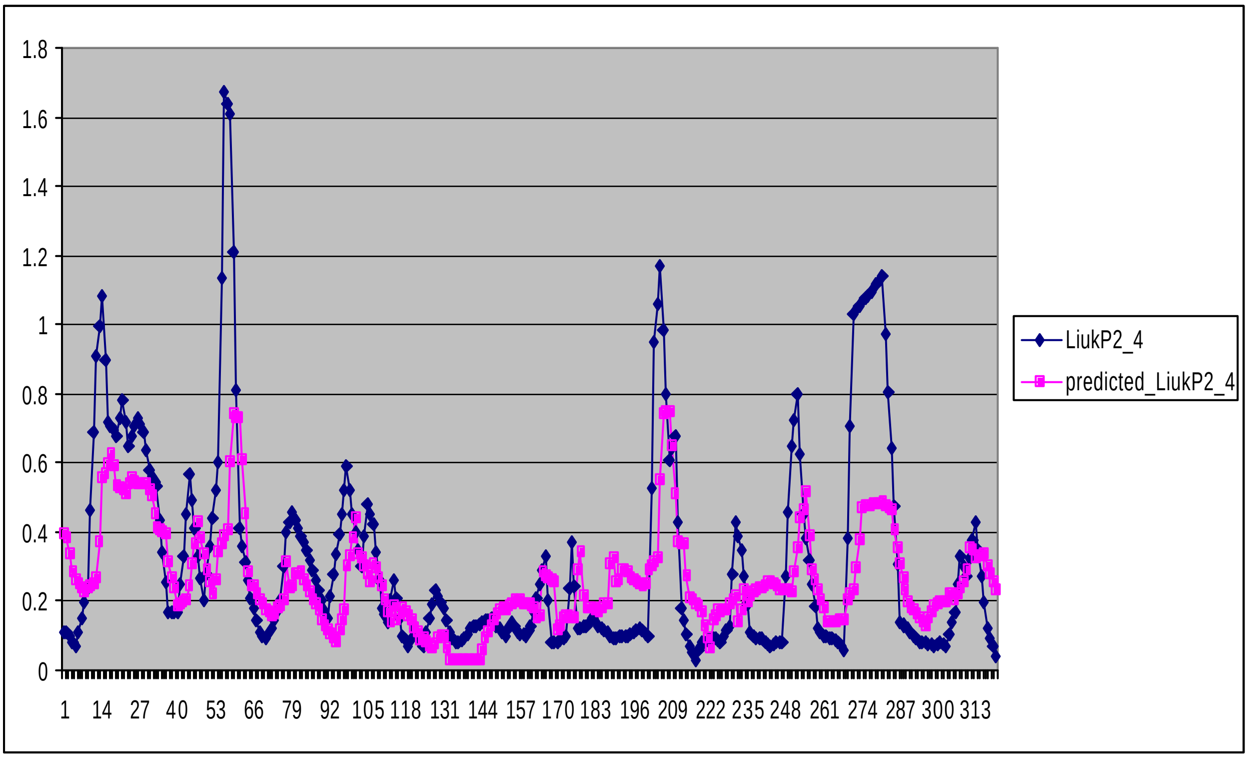

Figure 5-14 The training+test set predictions for Liuk_P2

Next we show the predicted and real future values for the desired quality indicators of the records in the validation data set. Visually it seems that the predicted values tend to follow the mean levels of the actual values and are not as good as the predictions we got with Ilmas3 variables.

In Figure 5-15, showing the LiukP2 time series one sees that a very large peak has occurred for a two week period. The predicted values suggest a much smaller peak but anyhow indicate an abnormal level.

From the quality measures calculated for variable LiukP2 one sees that the NRMS measure is around one (NRMS=0.89) standard deviation (?=1.05) and the average absolute error (AAE) is 0.45, which are fairly high considering the alarm limit (0.5). Relative mean absolute error is fairly good (9.3% from the value range of the actual measurements) because of the large peak period. Obviously the peak period affects these measures a lot.

If one evaluates the prediction results in terms of their capability to predict abnormally high values, one gets an overall classification error of 12.9% with a 16.7% false positive rate, which is not too bad for this validation set.

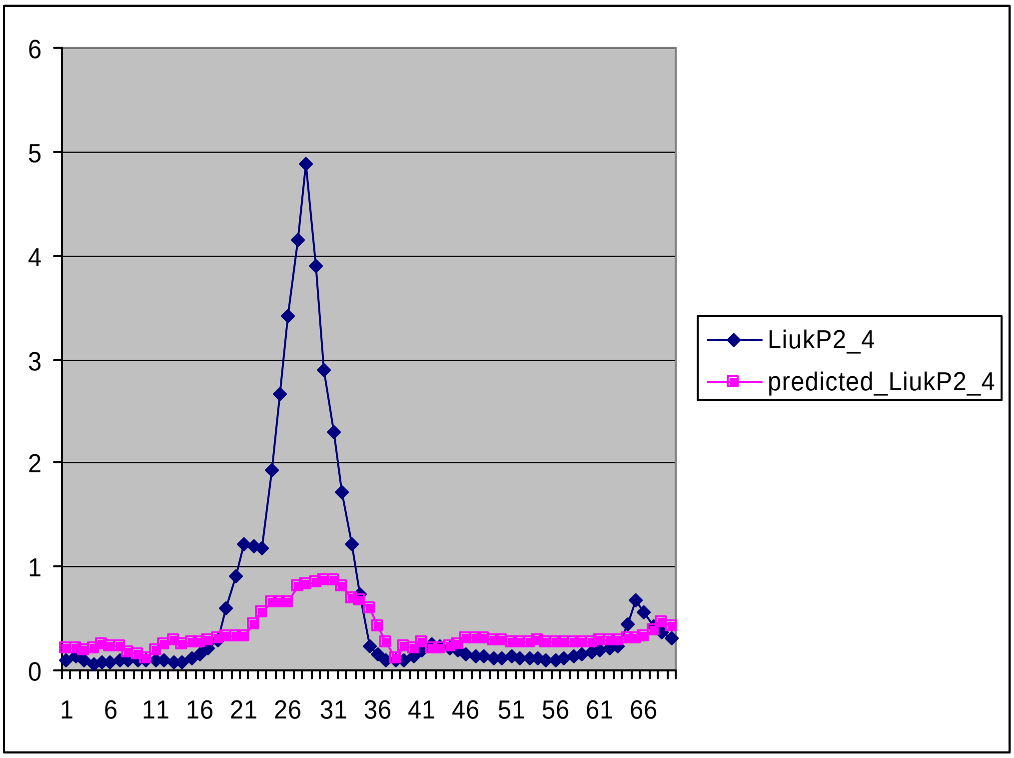

Figure 5-15 Predicted (3 days) and real values for variable LiukP2 in validation set

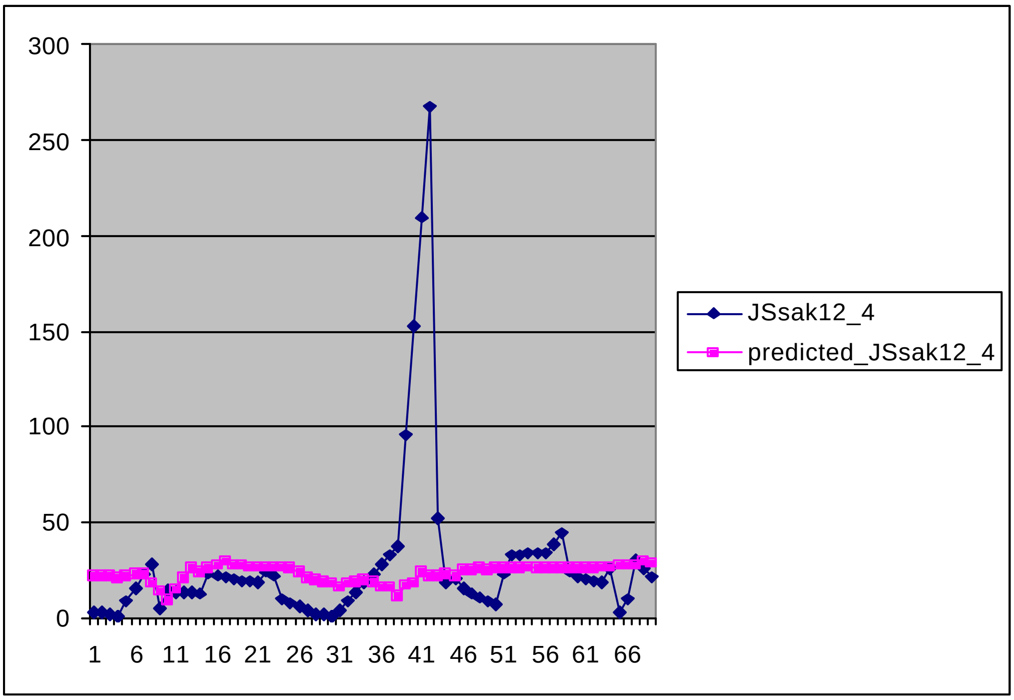

Figure 5-16 Predicted (3 days) and real values for variable JSSak12 in validation set

In the JSSak3 time series (see Figure 5.16), one sees that the measured values contain one 5 day peak which the predicted values do not forecast. Overall the predictions follow the mean levels of the actual values.

From the quality measures calculated, one sees that the correlation between the time series is very low, and the NRMS measure is around 1 standard deviations (?=42.3), and the average absolute error (AAE) is 19.4, which are fairly low considering the alarm limit (50). Relative mean absolute error is fair (9.3% of the value range.

If one evaluates the prediction results in terms of their capability of predicting abnormally high values, one gets an overall classification error of 7.1% with a 0% false positive rate (none forecasted so no errors).

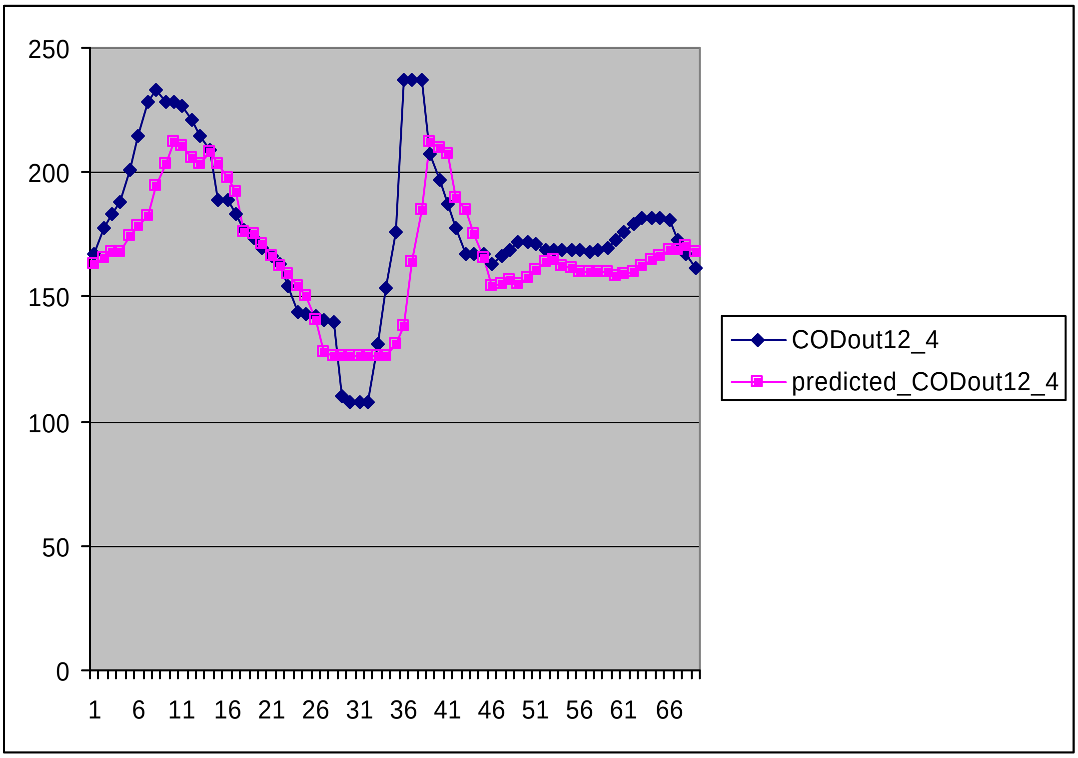

Figure 5-17 Predicted (3 days) and real values for variable CODout12 in validation set

In the CODout3 time series (see Figure 5.17), one sees that predicted values seem to reflect the actual values fairly well but there is a significant delay in the predictions considering the 3 day prediction period.

From the quality measures calculated, one sees that correlation between the time series is very low, the NRMS measure is 0.75 standard deviations (?=30), and the average absolute error (AAE) is 15.5, which are fairly good considering the alarm limit (200). Relative mean absolute error is fair (15% of the value range).

If one evaluates the prediction results in terms of their capability of predicting abnormally high values, one gets an overall classification error of 14.3% with a 20% false positive rate.

6 Integrating prediction capability with the Rapidbase platform

We describe a simple application developed for demonstrating how an external modeling tool may be integrated to the RapidBase platform. In this case, we are using Rapidbase as the storage place for the measurement time series of a waste water cleaning plant. The measurements have been stored in a normalized record format for the training and also into RapidBase database tables. Preprocessing has been already done with MATLAB functions by syncronizing the time stamps into a 24 hour period and interpolating missing values to form complete time series. An architecture of the demonstration system is presented in Figure 6-1.

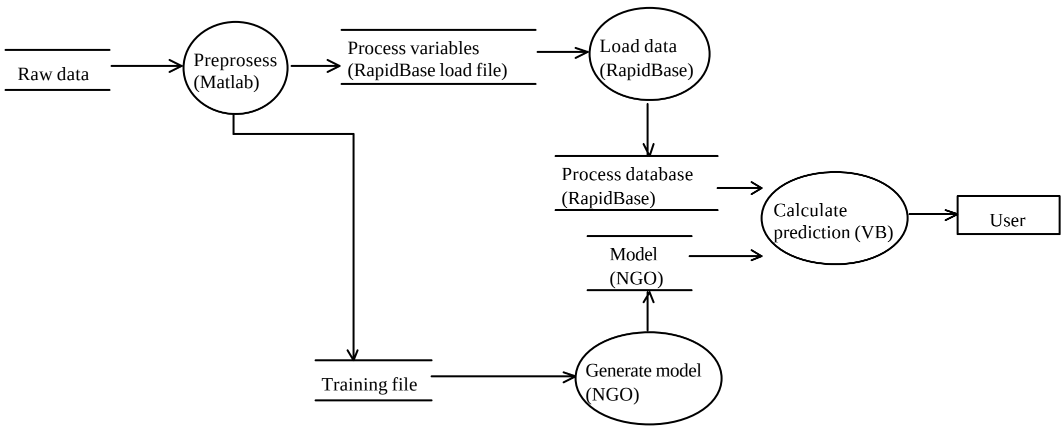

Figure 6-1 The architecture of the demonstration system for the time series prediction engine

Once the demonstration program is started, the measurements are retrieved from the database using the ODBC interface and the RQL language. (Access to an ordinary relational database would be handled similarly with SQL and ODBC.) The application has been developed using Visual Basic and predictions are generated by executing a neural network time series prediction model. The neural networks are evaluated using a commercial neural network tool (NGO) and its ActiveX programming interface (the MicroPredictor library).

In this demonstration, all the available measurements are retrieved at once and predictions are produced for all the situations. We predict 3 quality indicators 3 days into the future. These measurements are drawn on the screen as a chart display allowing one to view the quality of the results (red/dark color denotes the predicted value and green/lightercolor the actual value later acquired).

In an operational system, this prediction engine would be started every time new measurement records become available and predictions would be made only for the next situation (3 days ahead). For example, the calculation could be triggered with the Rapidbase triggers.

7 Summary

In this document, we have at first outlined various approaches for detecting abnormal conditions in chemical processes: classification, prediction of analog values, and identification of typical states by clustering and abnormality monitoring. Especially we have developed time series prediction methods for predicting the future behaviour of quality indicators of a chemical process.

Both traditional time series prediction methods (ARIMA) and artificial neural network models have been applied. The process data considered has been obtained from an industrial waste water cleaning plant using activated sludge cleaning process. Based on our experience it was found that the ARIMA models were not promising in this domain as the data contained no clear seasonality nor trends. Results obtained from the best artificial neural network models seemed promising but upon considering the stringent target limits on abnormality the prediction errors were too large. This may be caused by the limited amount of data available, the long time span of the prediction, and the dynamic behaviour and unobservable status (living bacteria) of the underlying process.

We have demonstrated how the RapidBase platform may be enhanced by adding new analysis modules to it. In this case, we have integrated a neural network execution environment to make time series predictions based on the currently available measurements. The same tool could be used for other types of modeling as well covering classification, function approximation (regression), and clustering tasks. The system could be also applied to other time series prediction tasks where one has no exact model of the process but has good process measurement data available.

References

[Alho99]
Alhoniemi, E., Hollmen, J., Simula, O., Vesanto, J., Process Monitoring and Modeling using the Self-Organizing Map, Integrated Computer Aided Engineering, John Wiley & Sons, 1/1999, pp. 3-14.

[Iiva97]
Iivarinen, J., Rauhamaa, J., and Visa, A.. An Adaptive Two-Stage Approach to Classification of Surface Defects. In Proceedings of The 10th Scandinavian Conference on Image Analysis, vol. I, pp. 317-322, Lappeenranta, Finland, June 9-11, 1997.

[Kosk97]
Koskela, T., Varsta, M., Heikkonen, J., and Kaski, K., Time series prediction using RSOM with local linear models. International Journal of Knowledge-Based Intelligent Engineering Systems, 2(1):60-68, 1998. available as Technical Report B15, Helsinki University of Technology, Lab. of Computational Engineering, 1997.

[Reic00]
Reick., C., Bernd, P., Time series prediction by multivariate next neighbor methods with application to zooplankton forecasts, Mathematic and Computers in Simulation, 52, 2000, pp. 289-310.

[Simu99]
Simula, O., Vesanto, J., Alhoniemi, Esa, Hollmen, J., Analysis and Modeling Of Complex Systems Using the Self-Organizing Map, Chapter in "Neuro-Fuzzy Techniques for Intelligent Information Systems, 1999.

[Vesa97]
Vesanto, J., Using the SOM and Local Models in Time-Series Prediction, Proceedings of Workshop on Self-Maps (WSOM'97), 1997, pp. 209-214.

[West00]
West, D., Mangiameli, P., Identifying process conditions in an urban wastewater treatment plant, International Journal of Operations & Production Management, Vol. 20, No. 5, 2000, pp. 573-590.

Appendix A: Ilmas3 quality indicators 14 days in advance

Table 1. Prediction accuracy measurements for forecasting Ilmas3 quality indicators 14 days in advance.

LiukP3_14	0.35, #pos:33, #neg:26		validation set											
	Classification Error	%FN	%FP	#pos	#neg	#FP	#FN	AAE	Rel. mean abs error	NRMS	Correlation	%Incorrect trends	Mean	Stddev
	28.8	9.1	33.3	48	11	16	1	0.55	0.14	1.01	0.65	48.3	0.76	0.96
	44.1	50	41	39	20	16	10	0.6	0.34	1.11	-0.22	51.7	0.76	0.96
	40.7	47	32	34	25	8	16	0.56	0.14	0.93	1.01	47.3	0.76	0.96
JSsak3_14	50 , #pos:0, #neg:59		validation set											
	Classification Error	%FN	%FP	#pos	#neg	#FP	#FN	AAE	Rel. mean abs error	NRMS	Correlation	%Incorrect trends	Mean	Stddev
	5.1	0	100	3	56	3	0	28.9	0.98	10.4	0.0002	47.1	14.5	8.9
	1.7	0	100	1	58	0	1	11.9	0.4	2.04	0.0004	36.8	14.5	8.9
	28.8	0	100	17	42	17	0	76.7	2.57	41.8	0.00006	67	14.5	8.9
CODout3_1	200 , #pos:5, #neg:54		validation set											
	Classification Error	%FN	%FP	#pos	#neg	#FP	#FN	AAE	Rel. mean abs error	NRMS	Correlation	%Incorrect trends	Mean	Stddev
	11.9	2	60	10	49	6	1	14.5	0.31	1.7	0.0014	48.3	180	11
	13.6	4	67	9	50	6	2	23.2	0.49	2.5	0.00026	36.8	180	11
	18.9	0	67	15	44	10	0	24.4	0.52	2.8	0.0016	23.7	180	11

Table 2. The file name data for the previous table.

\Lagged14\rank1_of_264_regr20_all3_GRNN_31in_256h1_4h2_2000_0908.net	\Lagged14\him_esikasitt_uusi14_W&D_2000_0907.txt
\Lagged14\rank2_of_264_regr20_all3_GRNN_28in_256h1_4h2_2000_0908.net	\Lagged14\him_esikasitt_uusi14_W&D_2000_0907.txt
\Lagged14\rank1_of_163_regr49_all3_TDNN_17in_61h1_52h2_2000_0911.net	\Lagged14\him_esikasitt_uusi14_W&D+pca_2000_0908.txt
\Lagged14\rank1_of_264_regr20_all3_GRNN_31in_256h1_4h2_2000_0908.net	\Lagged14\him_esikasitt_uusi14_W&D_2000_0907.txt
\Lagged14\rank2_of_264_regr20_all3_GRNN_28in_256h1_4h2_2000_0908.net	\Lagged14\him_esikasitt_uusi14_W&D_2000_0907.txt
\Lagged14\rank1_of_163_regr49_all3_TDNN_17in_61h1_52h2_2000_0911.net	\Lagged14\him_esikasitt_uusi14_W&D+pca_2000_0908.txt
\Lagged14\rank1_of_264_regr20_all3_GRNN_31in_256h1_4h2_2000_0908.net	\Lagged14\him_esikasitt_uusi14_W&D_2000_0907.txt
\Lagged14\rank2_of_264_regr20_all3_GRNN_28in_256h1_4h2_2000_0908.net	\Lagged14\him_esikasitt_uusi14_W&D_2000_0907.txt
\Lagged14\rank1_of_163_regr49_all3_TDNN_17in_61h1_52h2_2000_0911.net	\Lagged14\him_esikasitt_uusi14_W&D+pca_2000_0908.txt

Appendix B: Ilmas3 quality indicators 3 days in advance

Table 3. Prediction accuracy measurements for forecasting Ilmas3 quality indicators 3 days in advance.

Ilmas3: Predictions 3 days into the future														
LiukP3_3	0.36, #pos:44, #neg:26		validation set											
	Classification Error	%FN	%FP	#pos	#neg	#FP	#FN	AAE	Rel. mean abs error	NRMS	Correlation	%Incorrect trends	Mean	Stddev
	24.3	33	18.6	43	27	8	9	0.36	0.063	0.74	2.27	40.6	0.74	0.88
	45.7	58.8	33.3	36	34	12	20	0.49	0.11	0.93	1.38	50.7	0.74	0.88
JSsak3_3	50, #pos:1, #neg:69		validation set											
	Classification Error	%FN	%FP	#pos	#neg	#FP	#FN	AAE	Rel. mean abs error	NRMS	Correlation	%Incorrect trends	Mean	Stddev
	21.4	0	93.8	16	54	15	0	17.2	0.29	1.96	0.0016	41	17.6	11.7
	35.7	0	96.2	26	44	25	0	37.7	0.64	4.41	0.000067	45.9	17.5	11.7
CODout3_3	200, #pos:16, #neg:54		validation set											
	Classification Error	%FN	%FP	#pos	#neg	#FP	#FN	AAE	Rel. mean abs error	NRMS	Correlation	%Incorrect trends	Mean	Stddev
	12.9	7.5	29.4	17	53	5	4	26.5	0.4	1.69	0.001	43.2	186	17
	20	12.96	43.75	16	54	7	7	18.5	0.28	1.43	0.00091	50	186	17

Table 4. The file name data for the previous table.

\Lagged3\2000_0825\rank1_of_58955_RMS14_out3_GRNN_7in_264h1_4h2_2000_0907.ne	\Lagged3\2000_0825\him_esikasitt_uusi3_W&D_2000_0825.txt
\Lagged3\2000_0825\rank1_of_786_RMS47_out3_GRNN_20in_264h1_4h2_2000_0907.net	\Lagged3\2000_0825\him_esikasitt_uusi3_W&D_2000_0825.txt
\Lagged3\2000_0825\rank1_of_58955_RMS14_out3_GRNN_7in_264h1_4h2_2000_0907.ne	\Lagged3\2000_0825\him_esikasitt_uusi3_W&D_2000_0825.txt
\Lagged3\2000_0825\rank1_of_786_RMS47_out3_GRNN_20in_264h1_4h2_2000_0907.net	\Lagged3\2000_0825\him_esikasitt_uusi3_W&D_2000_0825.txt
\Lagged3\2000_0825\rank1_of_58955_RMS14_out3_GRNN_7in_264h1_4h2_2000_0907.ne	\Lagged3\2000_0825\him_esikasitt_uusi3_W&D_2000_0825.txt
\Lagged3\2000_0825\rank1_of_786_RMS47_out3_GRNN_20in_264h1_4h2_2000_0907.net	\Lagged3\2000_0825\him_esikasitt_uusi3_W&D_2000_0825.txt

Appendix C: Ilmas2 quality indicators 3 days in advance

Table 5. Prediction accuracy measurements for forecasting Ilmas2 quality indicators 3 days in advance

Ilmas3: Predictions 3 days into the future														
LiukP2_3	0.35, #pos:21, #neg:49		validation set											
	Classification Error	%FN	%FP	#pos	#neg	#FP	#FN	AAE	Rel. mean abs error	NRMS	Correlation	%Incorrect trends	Mean	Stddev
	12.9	11.6	16.7	18	52	3	6	0.45	0.093	0.89	4.2	44.6	0.63	1.05
	30	30	0	0	70	0	21	0.55	0.11	1.1	-12.4	43.1	0.63	1.05
JSsak12_3	60, #pos:5, #neg:65		validation set											
	Classification Error	%FN	%FP	#pos	#neg	#FP	#FN	AAE	Rel. mean abs error	NRMS	Correlation	%Incorrect trends	Mean	Stddev
	7.1	7.14	0	0	70	0	5	19.4	0.07	1.02	0.00029	39.1	27.3	42.3
	7.14	7.14	0	0	70	0	5	17.5	0.066	1	0.003	44.9	27.3	42.3
CODout12_3	200, #pos:14, #neg:56		validation set											
	Classification Error	%FN	%FP	#pos	#neg	#FP	#FN	AAE	Rel. mean abs error	NRMS	Correlation	%Incorrect trends	Mean	Stddev
	14.3	11.7	20	10	60	3	7	15.5	0.12	0.75	0.001	31.5	177	30
	21.4	14	54	13	67	7	8	19.8	0.15	0.84	0.0008	29.6	177	30

Table 6. The file name data for the previous table.

\ilmas2\Lagged3_b\rank1_of_3875_ilmas2_RMS9_TDNN_7in_5h1_0h2_2000_0925.net	\ilmas2\Lagged3_b\him_esikasitt_uusi2_W&D_2000_0919.txt
\ilmas2\Lagged3_b\rank2_of_3875_ilmas2_RMS9_CATNN_8in_27h1_6h2_2000_0925.net	\ilmas2\Lagged3_b\him_esikasitt_uusi2_W&D_2000_0919.txt
\ilmas2\Lagged3_b\rank1_of_3875_ilmas2_RMS9_TDNN_7in_5h1_0h2_2000_0925.net	\ilmas2\Lagged3_b\him_esikasitt_uusi2_W&D_2000_0919.txt
\ilmas2\Lagged3_b\rank2_of_3875_ilmas2_RMS9_CATNN_8in_27h1_6h2_2000_0925.net	\ilmas2\Lagged3_b\him_esikasitt_uusi2_W&D_2000_0919.txt
\ilmas2\Lagged3_b\rank1_of_3875_ilmas2_RMS9_TDNN_7in_5h1_0h2_2000_0925.net	\ilmas2\Lagged3_b\him_esikasitt_uusi2_W&D_2000_0919.txt
\ilmas2\Lagged3_b\rank2_of_3875_ilmas2_RMS9_CATNN_8in_27h1_6h2_2000_0925.net	\ilmas2\Lagged3_b\him_esikasitt_uusi2_W&D_2000_0919.txt