THE 25th EUROPEAN MODELING & SIMULATION SYMPOSIUM

SEPTEMBER 25-27 2013

ATHENS, GREECE

EMSS
2013

EDITED BY

AGOSTINO G. BRUZZONE
EMILIO JIMÉNEZ
FRANCESCO LONGO
YURI MERKURYEV

PRINTED IN RENDE (CS), ITALY, SEPTEMBER 2013

ISBN 978-88-97999-16-4 (Paperback)
ISBN 978-88-97999-22-5 (PDF)

© 2013 DIME UNIVERSITÀ DI GENOVA

RESPONSIBILITY FOR THE ACCURACY OF ALL STATEMENTS IN EACH PAPER RESTS SOLELY WITH THE AUTHOR(S). STATEMENTS ARE NOT NECESSARILY REPRESENTATIVE OF NOR ENDORSED BY THE DIME, UNIVERSITY OF GENOA. PERMISSION IS GRANTED TO PHOTOCOPY PORTIONS OF THE PUBLICATION FOR PERSONAL USE AND FOR THE USE OF STUDENTS PROVIDING CREDIT IS GIVEN TO THE CONFERENCES AND PUBLICATION. PERMISSION DOES NOT EXTEND TO OTHER TYPES OF REPRODUCTION NOR TO COPYING FOR INCORPORATION INTO COMMERCIAL ADVERTISING NOR FOR ANY OTHER PROFIT – MAKING PURPOSE. OTHER PUBLICATIONS ARE ENCOURAGED TO INCLUDE 300 TO 500 WORD ABSTRACTS OR EXCERPTS FROM ANY PAPER CONTAINED IN THIS BOOK, PROVIDED CREDITS ARE GIVEN TO THE AUTHOR(S) AND THE CONFERENCE.

FOR PERMISSION TO PUBLISH A COMPLETE PAPER WRITE TO: DIME UNIVERSITY OF GENOA, PROF. AGOSTINO BRUZZONE, VIA OPERA PIA 15, 16145 GENOVA, ITALY. ADDITIONAL COPIES OF THE PROCEEDINGS OF THE EMSS ARE AVAILABLE FROM DIME UNIVERSITY OF GENOA, PROF. AGOSTINO BRUZZONE, VIA OPERA PIA 15, 16145 GENOVA, ITALY.

ISBN 978-88-97999-16-4 (Paperback)
ISBN 978-88-97999-22-5 (PDF)

USING DATA MINING AND MACHINE LEARNING METHODS FOR SERVER OUTAGE DETECTION – MODELLING NORMALITY AND ANOMALIES

Matthias Wastian[(a)], Dr. Felix Breitenecker[(b)], Michael Landsiedl[(c)]

[(a)]dwh GmbH, Neustiftgasse 57-59, 1070 Vienna, Austria
[(b)]Vienna University of Technology, Institute for Analysis and Scientific Computing, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria
[(c)] dwh GmbH, Neustiftgasse 57-59, 1070 Vienna, Austria

[(a)]matthias.wastian@dwh.at, [(b)]felix.breitenecker@tuwien.ac.at, [(c)]michael.landsiedl@dwh.at

ABSTRACT

This paper will discuss several approaches to detect abnormal events, which are considered to be worth further investigation by the modeler, in a time series of frequently collected data as early as possible and – wherever applicable – to predict them. The approaches to this task use various methods originating in the field of data mining, machine learning and soft computing in a hybrid manner. After a basic introduction including several areas of application, the paper will focus on the modular parts of the proposed methodology, starting with a discussion about different approaches to predict time series. After the presentation of several algorithms for outlier detection, which are applicable not only for time series, but also a chain of events, the results of the simulation gained in a project to detect server outages as early as possible are put up for discussion. The text ends with an outlook for possible future work.

Keywords: abnormal event detection, prediction, data mining, machine learning

1. INTRODUCTION, APPLICATIONS AND STATE OF THE ART

Definition 1 (Event): An event shall be defined as an occurrence happening at a determinable time and place with a certain duration. It may be a part of a chain of occurrences as an effect of a preceding occurrence and as the cause of a succeeding occurrence. It is possible that more than one event occurs at the same time and/or place.

Definition 2 (Abnormal Event): An abnormal event shall be defined as an outlier in a chain of events, an event that deviates so much from the other events as to arouse suspicion that it was caused by something that does not follow the usual behavior of the considered system and that it could change the entire system behavior.

Applications of abnormal event detection can be found in a broad variety of areas, almost all of them following the idea to guarantee a certain level of safety for the system considered. Examples are the prediction or detection of server outages, of natural catastrophes like flooding, hurricanes or earthquakes, of stock market breakdowns and of network intrusions. In the area of audio and video surveillance crowd behavior or traffic might be analyzed, but abnormal event detection also plays an important role in ambient assisted living.

Various approaches have been suggested for abnormal event detection. This paper is going to focus on time series forecasting with artificial neural networks (ANN) and outlier detection of the prediction errors with one-class support vector machines (OC-SVM). OC-SVMs were proposed (among others) by Heller, Svore, Keromytis and Stolfo (2003), by Evangelista, Bonnisone, Embrechts and Szymanski (2005) who additionally propose the use of fuzzy ROC curves, by Zhang, Zhang, Lan and Jiang (2008), Dreiseitl, Osl, Scheibböck and Binder (2010) as well as by Lecomte, Lengellé, Richard, Capman and Ravera (2011). Not all of them take into account the factor time. Other applied methods in the field of abnormal event detection are listed below:

- sparse reconstruction cost (Cong, Yuan and Liu 2011)
- wavelet decomposition (Suzuki and Ihara 2008)
- clustering based abnormal event detection (Jiang, Wu and Katsaggelos 2008)
- statistical methods
  - change point detection (Guralnik and Srivastava 1999)
  - explicit descriptors statistical model
  - bayes estimation
  - maximum likelihood
  - correlation analysis
  - principal component analysis (PCA).

2. DATA GENERATION AND DATA PREPROCESSING

2.1. Data Generation

Server monitoring is rampant nowadays. Server monitoring software allows to measure lots of features of a server that somehow describe its status. For our simulations, we had a total of up to 1439 features per server which were measured at a sampling rate from about one per fifteen minutes up to one per minute.

Besides historic data sets of several servers that were logged in the past, a software tool was used to generate artificial data sets. The capacity-planning tool was used to run tests, also called scripts and workloads, against a targeted server to measure its server capacity and response metrics. During these tests, each client generated a simulated user load of transactions against the server under test, which reported server statistics back to the client.

2.2. Data Preprocessing

First of all, the size of the recorded data set is rather large. All the simulations for a rapid server alert system have to be carried out at least nearly online. Thus a reduction of the original data set is indispensable. We used expert knowledge and did a feature selection by categorizing the features into four groups of different priorities, resulting in up to 14 features of the highest priority 0 and up to 73 features of the two most important priorities 0 and 1. Most simulation runs were implemented using the data labelled with these two priorities.

As the model intends to recognize the actual and future status of a server, those features that accumulate values (e.g., number of mails sent since the start of the server monitoring) were transformed into their differences.

Wrong measurements are also an issue that has to be dealt with for the server outage detection model. Especially features that have something to do with the queue lengths of hard disks delivers impossible values in a few cases. These values were substituted by their predecessors (if those were possible values) during the learning process. Of course, this substitution is also possible during on-line simulation runs. Another possibility is to delete those wrong values like it needs to be done, when a measurement cannot be carried out correctly due to any reason and the feature at this time is NaN. The distribution of these NaNs can be investigated separately, the algorithms proposed in the following sections are not able to deal with NaNs.

The ranges of the features considered in the model differ a lot. To make them comparable, the whole data set needs to be normalized. When using the neuro-predictor for the rapid server alert model, is seems best to use the following minmax-mapping to normalize the data:

$$f(x) = y_{min} + \frac{(y_{max}-y_{min})(x-x_{min})}{(x_{max}-x_{min})} \qquad (1)$$

This is an affine transformation from $[x_{min}, x_{max}]$ to $[y_{min}, y_{max}]$.

3. PREDICTOR

Given any process that is checked for abnormal events, usually some features of this process can be measured at a constant sampling rate. Let m be the number of observed features. This results in m univariate time series. Given some past values and the actual value x_n of a certain feature, it is possible to predict the next observation x_{n+1} with a predictor and to calculate the prediction error as soon as the true new value x_{n+1} is measured.

Besides the classic ARIMA models that can be used for time series prediction, a certain kind of ANNs has proven to be an efficient predictor. Both models are going to be introduced in the following subsections. A multivariate approach is not recommended based on the simulation results for the server outage prediction as well as based on the results of various other authors. If a multivariate approach is desired nevertheless, we suggest to cluster the features first into several groups and to use an own multivariate predictor for each group.

The basic idea for any predictor of the abnormal event detection model is that the predictions are very good, if there are no abnormal events, i.e., the system's status is normal. The predictions become worse and do not originate from the usual distribution at least at the beginning of an abnormal event.

From a time series point of view, the most difficult task for the predictor is to consider the seasonality of the time series of some features. For example, the number of logged in users of a company on a certain Monday at 9:00 a.m. will probably strongly depend on the number of logged in users on Monday one week before at the same time. Feasts and holidays can cause problems for such models.

3.1. Neuro-Predictor

ANNs are non-linear and data-driven by nature and therefore at least theoretically very well suited to model seasonality interacting with other components.

Palit and Popovic (2005) refer to Simon Haykin, who suggests choosing the number of training patterns based on

$$N = \frac{W}{\varepsilon}. \qquad (2)$$

W shall be the number of weights used in the ANN, ε shall be the error the training examples should be classified with and N shall be the number of patterns in the training set in this context.

When using ANNs to forecast time series, data normalization is a key issue. Various normalization methods can be applied; logarithmic or exponential scaling can be used if problems with non-linearities are expected during the network training. Linear normalizations like (1) can be used to meet the requirements of the network input layer, as the input range must not be too wide.

Significant patterns as seasonality and trends should be removed, if possible, to make the ANN time series model easier. To be able to use the concept of cross-validation, appropriate training, test and validation data sets need to be chosen. For our simulations the training data includes 70%, the test and the validation set includes 15% of the preprocessed data each.

The tasks of structuring the data and choosing the number of input nodes n_i of the ANN predominantly depend on the number d of lagged values to be used for forecasting of the next value in the standard case of a one-step-ahead prediction. Thus the function to be modeled by the ANN is of the type

$$x_{n+1} = f(x_n, x_{n-1}, \dots, x_{n-d+1}) \tag{3}$$

This function can also be alternated to

$$x_{n+1} = f(x_n, x_{n-1}, \dots, x_{n-d+1}, x_{n-s}, \dots, x_{n-2s}, \dots) \tag{4}$$

for a seasonality s. If the seasonality was not removed and the data preprocessing produces suitable input data blocks, seasonality can thus be modeled in an explicit way by the neuro-predictor.

The number of output neurons n_o directly corresponds to the forecasting horizon, i.e. in the case of a one-step-ahead forecast there is only one output neuron. Usually only one hidden layer is used. The number of the neurons in the hidden layer n_h was chosen according to the geometric pyramid rule:

$$n_h = \alpha\sqrt{n_i n_o}, \;\; \alpha \in [0.5, 2] \tag{5}$$

Choosing the number of hidden neurons as well as the data normalization involves trial-and-error experimentation.

We used the hyperbolic tangent as activation function in the hidden layer (the sigmoid function is also possible) and the linear activation function for the output layer. According to Zhang and Kline (2007), a non-linear activation function in the output layer is only needed, if the time series shows a significant trend even after the data preprocessing.

For the training of such neuro-predictors we use the Levenberg-Marquardt algorithm. The training sets are presented to the ANNs in several epochs. The supervised learning stops as soon as one of the following three break conditions is met:

1. The number of training epochs exceeds the value of a chosen tuning parameter.
2. The number of back-to-back epochs, which the error function of the validation set increases in, exceeds the value of a chosen tuning parameter.
3. The error value of the test data set falls below some minimal error value (e.g. 10^{-6}).

If there are several ANN models that we can finally choose from, an adapted version of the AIC can be applied:

$$AIC = Nn_o \ln(\sigma^2) + 2k \tag{6}$$

The model with the smallest AIC shall be preferred.

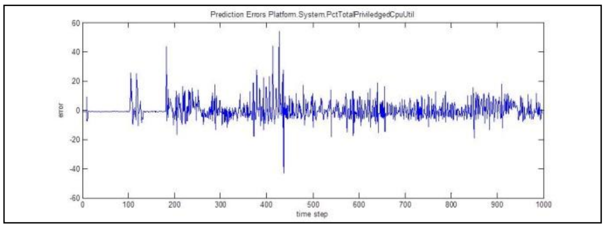

Figure 1: Prediction errors of a certain server feature, using a neuro-predictor

3.2. SARIMA Models

B being the backshift operator, autoregressive integrated moving average models with parameters $p, d$and q for a time series $\{x_t\}$ with error terms $\{\varepsilon_t\}$ are given by

$$\phi(B)x_t = \theta(B)\varepsilon_t \tag{7}$$

with

$$\phi(B) = \left(1 - \sum_{i=1}^{p} \phi_i B^i\right)(1 - B)^d \tag{8}$$

and

$$\theta(B) = 1 - \sum_{i=1}^{q} \theta_i B^i. \tag{9}$$

If the time series exhibits a strong seasonality, the model is adapted to a seasonal autoregressive integrated moving average model with parameters $(p, d, q) \times (P, D, Q)_s$, which is given by

$$\Phi(B^s)\varphi(B)\nabla_s^D \nabla^d x_t = \Theta(B^s)\theta(B)\varepsilon_t \tag{10}$$

with ∇ being the differencing operator, D the number of seasonal differences, Φ a polynomial of degree P, Θ a polynomial of degree Q and

$$\varphi(B) = \left(1 - \sum_{i=1}^{p} \phi_i B^i\right). \tag{11}$$

First of all the orders of differencing have to be identified to attain a stationary time series, several transformations like the logarithmic one might be useful. By looking at the plots of the autocorrelation function (ACF) and the partial autocorrelation function (PACF) - they are in fact bar charts - of the differenced series, the numbers of AR and/or MA terms that are needed can tentatively be identified, for example following the advices that can be found at the course of Nau (2005).

3.3. Comparison Between Neuro-Predictors and SARIMA Models

When using ANNs for prediction, the results obtained by various authors differ widely in quality: Some suggest that ANNs are better than other forecasting models, others contradict them. Some have seemed to obtain better results with seasonally adjusted data, others think that ANNs are able to directly model seasonality in an implicit way, without any seasonal adjustments on the input data. Detailed research results are presented in Zhang and Kline (2007).

In 1991 Sharda, Patil and Tang identified a number of facts that determine which method is superior, by experiments:

- For time series with long memory, both approaches deliver similar results.
- For time series with short memory, ANNs outperform the traditional Box-Jenkins approach in some experiments by more than 100%.
- For time series of various complexities, the optimally tuned neural network topologies are of higher efficiency than the corresponding traditional algorithms. (Palit and Popovic 2005)

A hybrid combination of neural networks and traditional approaches – maybe also including GARCH models – seems very promising.

For the server outage detection model, some time series involved might have a long memory, others a short one. All in all, it seems reasonable that it is less inexact to choose the same parameters for all the feature predictors, if the neuro-predictors are used. Choosing the same parameters for all the predictors simplifies the model a lot.

4. ANOMALY DETECTOR

An analysis of prediction errors is the basis for the anomaly detector. The anomaly detector decides in a multivariate way, whether the prediction errors of all the features belong to the class ‚normal‘ or not. We did not only let the anomaly detector decide upon the most recent prediction error, but we also made him judge upon a moving average of the prediction errors, which increases the tolerance against weaknesses within the prediction models.

Depending on the number of features predicted, the dimension of the prediction error vector is a key issue for choosing a good anomaly detector. For increasing dimension the relevance of distance converges against 0.

Hodge and Austin (2004) distinct three fundamental approaches to detect outliers:

1. Model neither normality nor abnormality. Determine the outliers with no prior knowledge of the data. This is essentially a learning approach analogous to unsupervised clustering.
2. Model both normality and abnormality. This approach is analogous to supervised classification and requires pre-labeled data, tagged as normal or abnormal.
3. Model only normality; maybe tolerate abnormality in very few cases. Authors generally name this technique novelty detection or novelty recognition, especially if only normal data is given. It is analogous to a semi-supervised recognition or detection task. Only the normal class is taught but the algorithm learns to recognize abnormality. The approach needs pre-classified data but only learns data marked normal.

4.1. Threshold

For lower dimensions a simple threshold for a prediction error norm like the Euclidean norm can be sufficient to detect anomalies (assuming that all the features have been transformed to similar ranges during the preprocessing). If the predictions of several features are as bad as the ones on the outside margin of the Gaussian bell of figure 2, they will be detected by simple threshold.

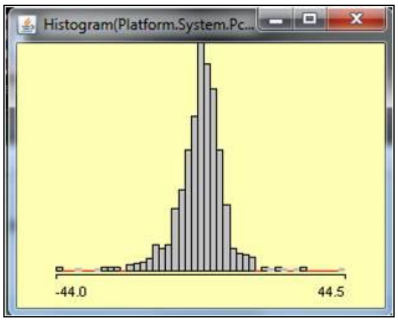

Figure 2: A Typical Histogram of the Prediction Errors of a Single Server Feature: A Gaussian Bell and a Few Outliers Clearly Visible on the Outside Margin

4.2. Angle-Based Outlier Detection

Angles are more stable than distances in high-dimensional spaces, which suggests the use of angles instead of distances for high-dimensional data. In fact, the situation is contrary for low-dimensional data. The angle-based outlier detection (ABOD) method alleviates the effects of the notorious curse of dimensionality compared to purely distance-based methods.

Following the idea of the algorithm developed by Kriegel, Schubert and Zimek (2008), a point is considered as an outlier, if most other points are located in a similar direction, and a point is considered as an inlier, if many other points are located in varying directions. The broadness of the spectrum of the angles between a certain point A and all pairs of the other points is a score for the outlierness of A: The smaller the score, the greater is the point’s outlierness. The idea of the algorithm is illustrated for two dimensions in figure 3.

The angles in the so-called angle-based outlier factor are weighted by the squared inverse of the corresponding distances to avoid bigger problems with low-dimensional data sets.

$$ABOF(A) = VAR_{B,C \in D}\left(\frac{\langle AB,AC\rangle}{\|AB\|^2\|AC\|^2}\right) \qquad (12)$$

A possibility to approximate the computationally expensive ABOF is to calculate the variance of the angles only of the pairs of points which belong to the k nearest neighbors of A, since these are the ones with the largest weights in the formula (12). Pham and Pagh (2012) provide further details on this issue.

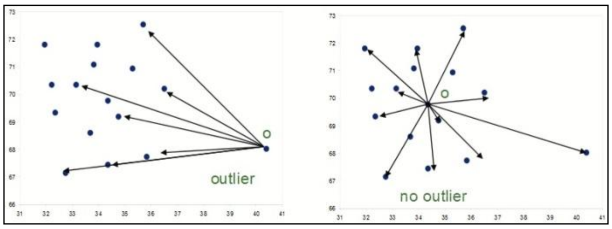

Figure 3: Idea of Angle-Based Outlier Detection

4.3. One-Class Support Vector Machine

In general, one-class support vector machines (OC-SVMs) are designed for the certain type of a $(1 + x)$-class learning task. This is a model with an unknown number of classes, but the modeler is only interested in one specific class. Typical examples for these kinds of tasks are content-based image retrieval or document-retrieval in general. Making research for this paper on the internet can be seen as such a task: Papers which treat relevant topics are alike, they represent the class the modeler is interested in. These are the positive examples and it is easy to find some good representatives of this class. The negative examples are simply the rest of the web pages or papers, and they originate from an unknown number of different negative classes.

It is daunting and wrong to try to characterize the distribution of the negatives in such cases; they could belong to any negative class, and the modeler is not even interested which exact negative classes they might belong to. Each negative example is negative in its own way, but as the positive ones are alike, it is possible to model their distribution. According to this the OC-SVM is a typical example of a model of normality, matching the third approach described at the beginning of section 4.

The OC-SVM tries to fit a tight hypersphere W to include most, but not all positive examples. If it is attempted to fit all positive examples, this would lead to overfitting. In fact, the OC-SVM searches for the maximal margin hyperplane

$$\omega x + b = 0 \tag{13}$$

with a normal vector ω and a bias b which separates the training data from the origin in the best way. It may be interpreted as a regular two-class SVM, where almost all the training data lies in the first class and the origin is the only member of the second class.

If the one class the modeler is interested in is considered as the regular data, resulting from normality, the negative examples detected by the OC-SVM can be considered as outliers of a different nature resulting from anomaly. This makes the OC-SVM an effective outlier detection tool.

Let $\{x_1, \dots, x_n\}, x_i \in X \subseteq \mathrm{R}^m$ be a training set of $n \in \mathrm{N}$ observations that belong to a single class. The OC-SVM aims to define the minimum volume region enclosing $(1 - \nu)n$ observations. The parameter $\nu \in [0,1]$ thus controls the fraction of observations that are allowed to be outliers. K shall be a kernel with a mapping function φ. ξ_i shall be the slack variables for observations on the wrong side; non-zero slack variables correspond to the tolerated outliers. The OC-SVM algorithm results in the following minimization problem:

$$\min_{\omega,\xi,b} \frac{1}{2}\|\omega\|^2 - b + \frac{1}{\nu n}\sum_{i=1}^{n} \xi_i \tag{14}$$

subject to

$$\omega^T \varphi(x_i) - b \geq \xi_i \geq 0 \tag{15}$$

Solving the OC-SVM optimization problem is equivalent to a dual quadratic programming problem with Lagrangian multipliers α_i that can be solved with standard methods:

$$\max_{\alpha_i} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j K(x_i, x_j) \tag{16}$$

subject to

$$\sum_{i=1}^{n} \alpha_i = 1, \;\; 0 \leq \alpha_i \leq \frac{1}{\nu n} \tag{17}$$

Those patterns with corresponding $\alpha_i > 0$ are the support vectors. By using the Karush-Kuhn-Tucker conditions ω and b can be obtained as:

$$\omega = \sum_{i=1}^{n} \alpha_i x_i \tag{18}$$

$$b = \sum_{i=1}^{n} \alpha_i x_i^T x_j \tag{19}$$

for any support vector x_j.

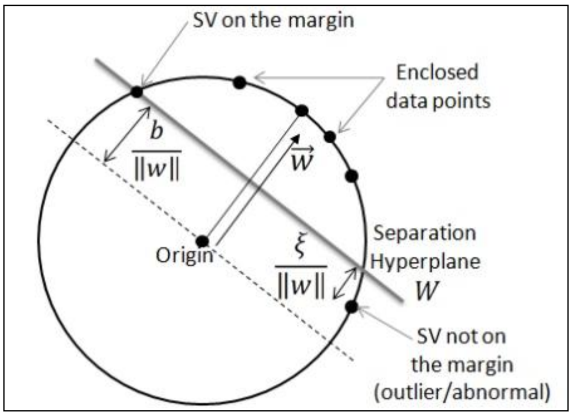

Figure 4: One-Class Support Vector Machine (Lecomte et al. 2011)

A new observation x is labeled by the OC-SVM via the decision function

$$f(x) = \omega^T \varphi(x) - b \tag{20}$$

which is positive for inliers and negative for outliers.

According to Lecomte et al. (2011), it is easily possible to define a family of decision rules introducing a threshold $\gamma \in \mathrm{R}$ by using an adaption of (20) and dividing inliers and outliers along γ, not along 0. This formulation allows controlling the trade-off between the probability to miss outliers and the probability to falsely declare an observation an outlier.

4.4. Combined Detector

As all the proposed outlier detector methods return an outlierness score for a feature vector, they could be used in a hybrid way. Then a weighted sum of the outlierness scores of each method is the final outlierness score of an observation.

5. RESULTS AND OUTLOOK

First of all, it has to be stated that it is almost impossible to precisely define the term server outage, wherefore a definition is not given in this paper. Any limitation to the normal operation of a server is unwanted. Many times only a certain kind of tasks is delayed or cannot be executed at all. The severity of this limitation also depends on the fact whether users can carry out other tasks in the mean time. The only possibilities to give the modeler an idea about the severity of an outage are the total downtime minutes or downtime minutes per user. Thus the basic idea of this model is to be able to provide the administrator of a server with the detection/prediction of irregularities, of anomalies which differ from the usual server operation. A classification of outages would be very useful, but requires outage data to learn from. This data should be labeled with the outage cause by experts. This classification remains future work.

Within the proposed model, the numbers of lagged time series elements that are relevant for the univariate prediction models for each server feature are not very easy to determine and the optimal number probably varies for each variable. Also the seasonality of the feature time series is not easy to diagnose. Nevertheless, the prediction models with global parameters for all the predictors worked very well during a normal operation of servers and seem to be sufficient for an online server outage detection model.

During several test runs, the anomaly detectors easily detected when the servers changed their status from idle to busy and vice versa (see figure 5). They also detected abnormal events within the gas price time series which was used as a benchmark data set (see figure 6). For this time series, an abnormal event is for example the oil crisis of 1979, which was caused by the Islamic revolution in Iran and the first gulf war, i.e. by external events. For the server outage detection model, the verification is rather difficult and there will be done further research on this topic: Besides the difficulty to define a server outage, the model needs to be tested in a real-life scenario which is planned in near future. So far, the detectors worked well with the test data sets.

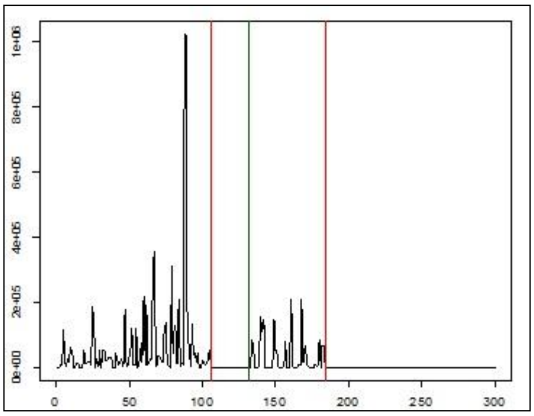

Figure 5: Angle-Based Outlier Detector Detecting the Server Change from Idle to Busy (Green) and Busy to Idle (Red)

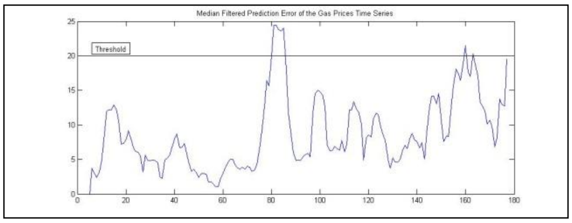

Figure 6: Median-Filtered Prediction Error of the Gas Prices Time Series Using a Neuro-Predictor with a Delay of 3 Months, 10 Hidden Neurons and a Threshold for Abnormal Event Detection. The Median Was Calculated Over 6 Months. The First Peak Above the Threshold 20 Corresponds to the 1979 Oil Crisis.

Of course, a server outage prediction software has a cold start: During the training some internal model parameters that are required to run the model need to be adjusted, before an expert can adjust several tuning parameters to control the alert sensitivity of the software. The most important tuning parameters are part of the anomaly detector. One could say that the server outage detection model needs to get to know the server that the outages shall be predicted of. As parts of the model are able to learn from the past, the software will highly improve its performance after several days.

An important question that still remains unanswered is when the neuro-predictors should be retrained or when the ARIMA models should be updated. Certainly, if the way the server is used changes considerably, a re-start of the model is necessary.

ACKNOWLEDGMENTS

The authors would like to thank the economic partners who helped a lot to acquire tons of data.

REFERENCES

Cong, Y., Yuan, J., Liu, J., 2011. Sparse Reconstruction Cost for Abnormal Event Detection. Proceedings of the 24th *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3449-3456. Colorado Springs, Colorado.

Dreiseitl, S., Osl, M., Scheibböck, C., Binder, M., 2010. Outlier Detection with One-Class SVMs: An

Application to Melanoma Prognosis. Proceedings of the *AMIA Annual Symposium 2010*, pp. 172-176. Washington, D.C.

Evangelista, P., Bonnisone, P., Embrechts, M., Szymanski, B., 2005. Fuzzy ROC Curves for the 1 Class SVM: Application to Intrusion Detection. Proceedings of the *13th European Symposium on Artificial Neural Networks*, pp. 345-350. Bruges, Belgium.

Guralnik, V., Srivastava, J., 1999. Event Detection from Time Series Data, Proceedings of the *5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 33-42. New York, USA.

Heller, K., Svore, K., Keromytis, A., Stolfo, S., 2003. One Class Support Vector Machines for Detecting Anomalous Windows Registry Accesses. Proceedings of the *Workshop on Data Mining for Computer Security in conjunction with the IEEE International Conference on Data Mining 2003*, pp. 2-9. Melbourne, Florida.

Hodge, V., Austin, J., 2004. A Survey of Outlier Detection Methodologies. *The Artificial Intelligence Review* Issue 2 of Volume 22: pp. 85-126.

Jiang, F., Wu, Y., Katsaggelos, A., 2008. Abnormal Event Detection Based on Trajectory Clustering by 2-Depth Greedy Search, Proceedings of the *IEEE International Conference on Acoustics, Speech and Signal Processing 2008*, pp. 2129-2132. Las Vegas, Nevada.

Kriegel, H.-P., Schubert, M., Zimek, A., 2008. Angle-Based Outlier Detection in High-Dimensional Data. Proceedings of the *14th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 444-452. Las Vegas, Nevada.

Lecomte, S., Lengellé, C., Richard, C., Capman, F., Ravera, B., 2011. Abnormal Events Detection using Unsupervised One-Class SVM – Application to Audio Surveillance and Evaluation. Proceedings of the *8th IEEE International Conference on Advanced Video and Signal-Based Surveillance*, pp. 124-129. Klagenfurt, Austria.

Nau, R., 2005, *Forecasting - Decision 411, Online Course*. Available from: http://people.duke.edu/~rnau/Decision411CoursePage.htm [Accessed 13.05.2013]

Palit, A., Popovic, D., 2005. *Computational Intelligence in Time Series Forecasting – Theory and Engineering Applications*. London: Springer Verlag.

Pham, N., Pagh, R., 2012. A Near-Linear Time Approximation Algorithm for Angle-Based Outlier Detection in High-Dimensional Data. Proceedings of the *18th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 877-885. New York, USA.

Suzuki, M., Ihara, H., 2008. Development of Safeguards System Simulator Composed of Multi-Functional Cores. *Journal of Power and Energy Systems* Number 2 of Volume 2: pp. 899-907.

Zhang, G.P., Kline, D., 2007. Quarterly Time-Series Forecasting With Neural Networks. *IEEE Transactions on Neural Networks* Number 6 Volume 8: pp. 1800-1814.

Zhang, R., Zhang, S., Lan, Y., Jiang. J.. Network Anomaly Detection Using One Class Support Vector Machine. Volume 1 of the Proceedings of the *MultiConference of Engineers and Computer Scientists 2008*. Hong Kong.

AUTHORS BIOGRAPHY

Matthias Wastian, born 27.12.1983 in Carinthia, Austria, studies technical mathematics at the Vienna University of Technology and has been writing his diploma thesis about abnormal event detection. He is employed at the dwh, Neustiftgasse 57-59, 1070 Vienna. Apart from that he is the captain of the Austrian wheelchair basketball national team.