Local Anomaly Detection for Network System Log Monitoring

Pekka Kumpulainen
Tampere University of Technology
pekka.kumpulainen@tut.fi

Kimmo Hätönen
Nokia Siemens Networks
kimmo.hatonen@nsn.com

Abstract

Huge amounts of operation data, including system logs, are being collected from communication networks. System operators and developers need easy to use and robust decision support tools based on these data. One of their key applications is to detect anomalous phenomena of the network. We present an anomaly detection method that describes the normal states of the system with a self organizing map (SOM) identified from the data. Anomalous behavior is specified as one in which the data deviate from clustered SOM nodes by more than a threshold. Instead of one global threshold, we use local thresholds. Threshold to use depends on the set of close by nodes according to how much variation the identification data have around the nodes. Our anomaly detection method can be applied both in on-line monitoring and in analysis of history data.

1. Introduction

The number of security tools and their complexity is rapidly growing. They produce a growing number of logs that are collected and stored. It is impossible to analyze all the data manually. Therefore, automatic methods are needed to scan the data sets and detect the most interesting or suspicious parts of the data. These potentially interesting data are then to be examined by human expert.

Detection of anomalies or outliers is important in log data analysis. Locating rare or suspicious parts of the data can reveal new valuable information from the system. As Kruskal wrote in 1960 [1]: *An apparently wild (or otherwise anomalous) observation is a signal that says: "Here is something from which we may learn a lesson, perhaps of a kind not anticipated beforehand, and perhaps more important than the main object of the study.".*

Outliers can be errors or signs of otherwise undesired performance and, they should be detected as soon as possible. Logging data are available from a period of time in history. The data usually include samples from normal states as well as abnormal situations. With such data we have to assume that the vast majority of the data are from normal functionality and the rare states present some sort of anomalies or errors.

A general definition for an outlier was given by Hawkins [2]: *An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism.* This definition is very extensive but it gives no guidelines how to determine whether an observation is an outlier or not.

Various statistical methods have been used in outlier detection [7]. For online monitoring purposes there are specific tests in statistical process control (SPC) [8]. These are mostly univariate methods and rely on the knowledge of the underlying distribution. Multivariate SPC methods [9] also assume multinormal distributions. Log activity counters, however, do not usually follow any known distribution. Therefore, these methods can not be applied to our problem.

ICA (Independent Component Analysis) has been applied for MSPC in order to perform better with nonnormally distributed variables [10]. Knorr et al. present notion of distance based outliers [11]. This requires no assumptions of distribution, but is based on the distances between the data samples.

All the methods mentioned above are global in a sense that they treat all the data set as one group. If the data are clustered, they may fail completely. Definition of local outliers by Breunig et al. [3] takes the clustering structure of the data into account. Their method detects the degree of a sample being an outlier in the local neighborhood.

Höglund et al. [4] presents an anomaly detection method based on quantization errors of self organizing map (SOM) [12]. This method is independent of the distribution and cluster structure of the data since the

SOM approximates the data. Due to effective SOM algorithm, it can be used also with large data sets. This method works well in most cases. However it has shortcomings if the data are not homogenous and doesn't have constant variance. The global threshold is likely to be too low in those parts of the data space, where the natural variation is large, thus producing false alarms. On the other hand, where the natural variance is low, the global threshold tends to be to high, leaving locally anomalous samples undetected.

We have developed an improvement to that method to emphasize the local characteristics of the data. We cluster the rest of the SOM nodes to groups and calculate thresholds for each group separately. This way we have more sensitivity locally where the data have less variance. At the same time number of false positives can be reduced in those parts of the data space where there is larger variance. In addition to detecting individual samples, this method can reveal new states that are rare, but contain too many samples to be detected as outliers. Such cases can be a new unseen behavior in the process to learn about, or caused by a longer lasting error and require attention.

In this paper we present the original and the improved algorithms. Main features of the methods are shown with a synthetic example. Finally use cases are presented. The first case is to analyze history data to find anomalous samples and suspicious groups. The second case applies the identified anomaly detection model to a new data to find out whether it complies with the previous behavior of the system. The performance of the original and our improved method are compared in this use case. The second use case is then repeated with another data set and the performance of the methods is discussed.

2. Method description

Our goal was to improve the anomaly detection method by Höglund et. al. [4]. Our improved method detects local distance based outliers. In this section/chapter we first describe the original method and after that, we introduce the improved version.

2.1 Original

The original method is based on quantization errors of a SOM. Reference data are used to train 1 dimensional SOM, which is more flexible than 2-D. Quantization error of the data samples are compared to a threshold that is a predefined quantile of the quntization error in training data. Samples exceeding the threshold are labeled as anomalous to be further examined by human experts. The basic idea of the algorithm is listed step by step.

1 Fit a SOM to a reference data set of n samples, drop out nodes with no hits

2 Calculate distances D_1 … D_n to BMUs (quantization errors) for the reference data. A predefined percentile is used as anomaly threshold

3 For a sample to test calculate distance D_{n+1} from its BMU, considered as anomaly if it exceeds the threshold

An example was generated to visualize the method. Figure 1 shows a 2 dimensional example case. The blue dots are the data. The data set consists of three groups. Each group contains gaussian random samples with similar variances in a group and the means close to each others. Group 1 has small variance, group 2 had medium variance and group 3 has largest variance. The small red circles, connected by line, are the SOM nodes after training. The larger black circles present the anomaly threshold, which was selected to be 95th percentile of the quantization errors. Any data points outside these circles are assumed to be anomalies, marked with pentagrams.

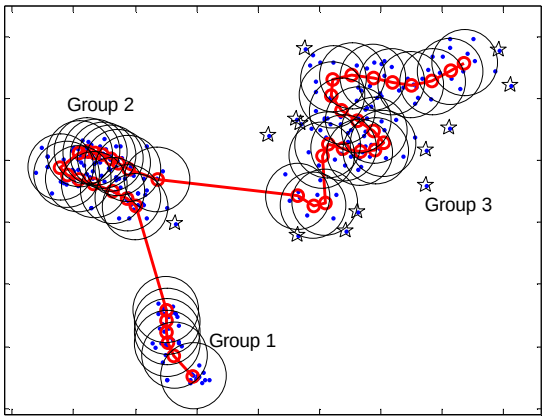

Figure 1 Scatter plot of the generated example data with anomaly thresholds

Figure 1 demonstrates the ability of the SOM to approximate the data space regardless of the distribution or the clustering structure. Especially 1-dimensional line, instead of the more used 2-dimensional grid, is very flexible and useful for this purpose. The original method uses one global threshold for anomaly detection. That makes the anomaly threshold circles equal in size around the data space. Therefore in this example, most of the detected anomalies are located in group 3, which has the largest variance. Only one sample close to group 2 is considered an anomaly.

2.2. Improved

In order to further emphasize the local structure of the data, we introduce local thresholds. This takes into account also the local variance of the data.

We form groups of SOM nodes and calculate a threshold for each group. Groups are created by clustering the SOM code vectors. The algorithm:

1 Train SOM and drop out nodes that have less than a specified number of hits

2 Cluster the code vectors, these clusters will be called *reference groups* later on.

3 Set anomaly threshold to predefined percentile of the distances from BMUs (quantization error) in each reference group.

4 For a sample to test calculate distance D_{n+1} from its BMU, considered as anomaly if it exceeds the local threshold

This way we have higher threshold in those parts of the space, where samples are deviated more from BMUs.

When new data are available, the BMU and the distance from BMU are calculated for each data sample. The distance is compared to the threshold in that reference group which the BMU belong in. If the threshold is exceeded, the sample is labeled as anomaly.

Results of this method for the same generated example in 2-D are presented in Figure 2. The anomaly threshold is calculated for each group. The threshold is larger where the data have more variation and smaller in the parts of the data space, where the there are more dense clusters.

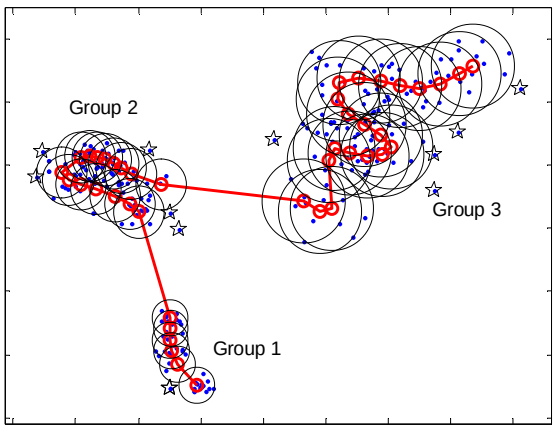

Figure 2 Anomaly thresholds for each reference group

Now the sizes of the threshold circles vary according to the variance of the data. There are fewer anomalies detected in group 3 than with the original method. Thus the risk of false positive detections is reduced in those parts of the data space, where the natural variance of the data is higher. At the same time, there are more detected anomalies in groups 1 and 2. The sensitivity of the method is locally adapted to the variation in the data.

Figure 3 shows the histograms of the quantization errors in different groups. The detection thresholds in each group are marked with a vertical line. As can be seen, the threshold varies from 0.57 in the group 1 to 1.33 in the group 3.

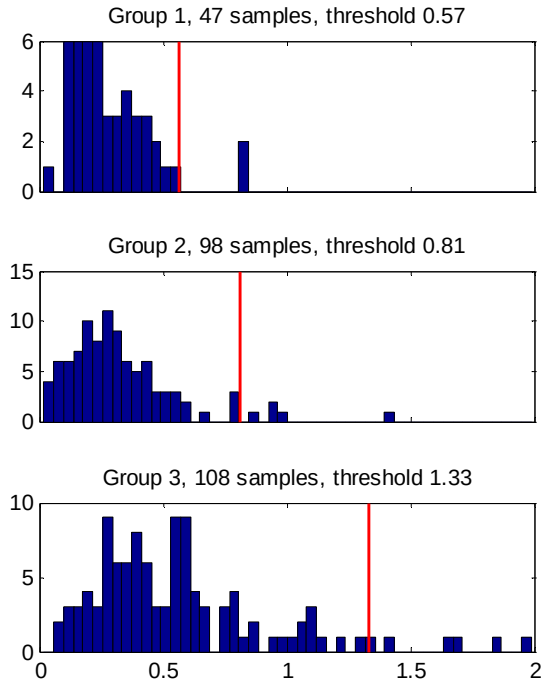

Figure 3 Histograms of the quantization errors

3. Results in use cases

The method has been tested with two data sets from a network used for new mobile technology field tests. The data have been collected from network management system servers. The collected data contain information about the automatic system processes and manual operator activity. Thus, it can be used for monitoring system behavior for maintenance purposes and for searching for security incidents in security monitoring.

In networks a major system malfunction causes a clear sign to system logs. Some processes can either begin to produce huge amounts log entries or they can stop logging. In both cases the resulting logging activity is anomalous when compared to the normal logging behavior that changes in quite static cycles. Also in security monitoring all anomalies in system component behavior or user activity or appearance of new software may be signs of an incident.

We have used the method for two types of analysis on two data sets. First we have analyzed the reference data by training a SOM with it and by clustering the SOM nodes. There have been 7 groups of SOM nodes in both data sets. The groups represent common system states that can be analyzed, described and named. The SOM nodes represent 95% of the data. The rare states are presented as anomalies that differ from all the groups. Their number is quite low, which enables us to analyze them manually. This kind of analysis can be used, for example, for periodical audits of the systems or on-demand analysis of the system history, where we suspect that, for example, a security incident has taken place.

The other type of analysis that we have done, is the comparison of test data set against the previously identified model. This has resulted in a set of anomalies that are analyzed further. This type of analysis can be used to do continuous monitoring of a system. All anomalous data points might or might not be signs of something interesting that must be analyzed further.

In analysis we have used variables that record logging activity level of system components and functions. 7 variables were available in the data set 1 and 8 variables in the data set 2.

These examples simulate real world cases. We have the previously collected reference data to calculate the scaling parameters and to tune the anomaly detection model. The model is then used to examine the performance of the system and to detect potential problems later on.

The examples represent a real life situation also in a sense that there is no information available about whether a sample is truly anomalous or not. Thus there are no “right results”. The method is aimed to find the most suspicious parts of the data for the user for further study.

3.1 Preprocessing

In anomaly detection, the scaling plays very important part. Scaling defines more or less what kind of anomalies will be detected [5, 6]. Proper scaling should make the variables equal in their importance in the problem they are used in.

All multivariate methods based on distances or variance, are very sensitive to scaling. Variables with larger variance dominate in the methods. It is common to scale all the variables to zero mean and unit variance. This is rarely the best possible choice. Variables with small variance are amplified and therefore could be overvalued in the analysis. Also the noise is amplified. At the same time the variables with high variance are attenuated and possibly underrated in the analysis.

Log data consist of event counts. Significance of a difference in counts depends on the total number, the value of a variable. A difference of 2 events, for example is more significant if there are 10 events altogether, than 2 events of total 1000 events. We use a robust logarithmic scaling for log data that preserves the importance of the variables.

The basic normalization is done by subtracting the mean and dividing by the standard deviation. Our scaling first takes a natural logarithm of the variable plus one and then divides by a robust standard deviation.

$$x_{\log s} = \frac{\ln(x+1)}{s},$$

where $s = std\{\ln(x+1) \mid x > 0, x < q99\}$

and $q99$ refers to 0.99 quantile of the variable x.

Adding one to the variable eliminates the need to separately handle the zeros in the data. Now zeros will remain zeros instead of minus infinity. This also separates ones and zeros in original scale, since values of one will be scaled to ln(2) instead of ln(1), which equals zero. The standard deviation is calculated ignoring zeros and 1% from the upper tail.

Finally the mean is subtracted.

$$x_{scaled} = x_{\log s} - mean\{x_{\log s}\}$$

The following figure shows an example of the difference between normalization and the logarithmic scaling applied to a variable that has peaks and smaller scale variation. The normalized scaling leaves high peaks and attenuates the smaller scale variation so that any changes within that are hard to detect. The logarithmic scaling attenuates the peaks, which still remain detectable, but leaves more variation to the smaller scale.

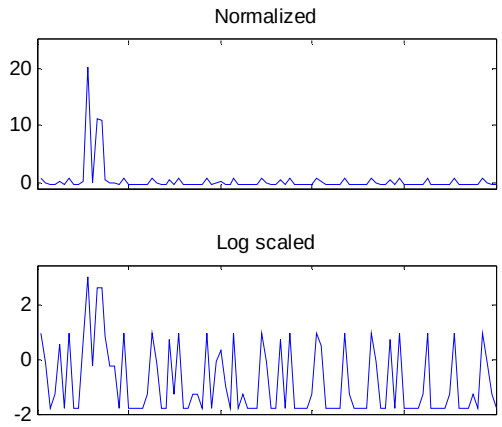

Figure 4 Example of scaling, normalized and log scaled

3.2 Use case 1, the reference data on data set 1

Reference data set 1 contain 7 variables, 33 days hourly samples, total of 792 samples.

First we train a 1 dimensional SOM using the SOM Toolbox for Matlab [15]. We use N/5 nodes, where N is the number of samples in the reference data. Nodes that have less than 3 hits are dropped out. In the original method nodes with no hits are dropped. The nodes are then clustered to form reference groups. We use hierarchical clustering with ward linkage [16]. The number of clusters is determined by maximum Davies-Bouldin index [14]. We limit the number between 3 and 10. The anomaly threshold for each group is determined as the 95th percentile of the distances from the BMU in each reference group.

In this case we end up with a SOM of 121 nodes and 7 reference groups. The histograms of the distances from BMUs are presented in the following figure. The first histogram at top shows the Distances of the whole data set and the global threshold. The following histograms are for each of the 7 reference groups.

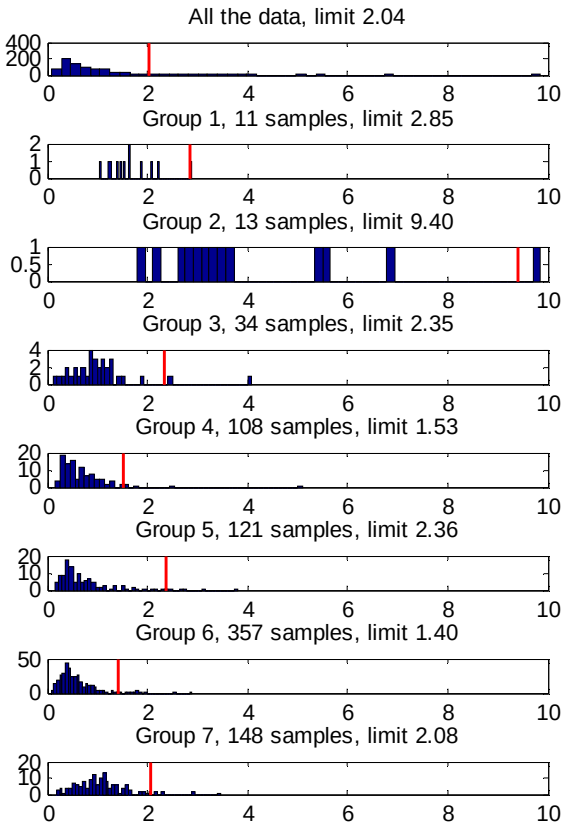

Figure 5 Histograms of the quantization errors in all the data on top and in each reference group below.

Here we find the small groups 1, 2 and 3, where the size of a reference group refers to the number of samples in the reference data associated to the group. They should be studied in case they present a repeating error or otherwise undesired state.

Centroids of the reference groups are presented in Figure 6. Groups 1 and 2 are clearly distinguished from others. Variable 5 has very high values in both groups. There groups are separated from each others by variable 2, which is high in group 2 and variable 6, which is low in group 2.

In group 3, variable 3 has high values, which distinguishes it from all the other groups.

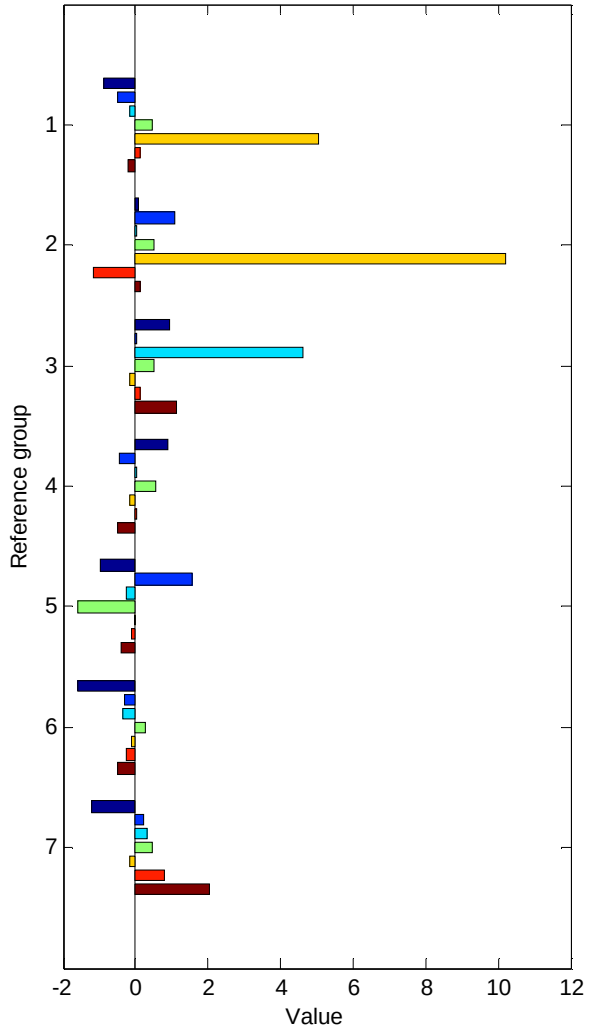

Figure 6 Bar plot of reference group centroids.

Component planes are a common way to visualize 2-dimesional SOM. A component plane presents the SOM grid for each variable and the values of the variables in each node of the grid are color coded on the plane. In this case, when we have 1-dimensional SOM, the component planes can be replaced by line plots as in Figure 7. Each line plot presents the values of the SOM code vectors corresponding to each variable. The x axis is the number of the SOM node along the one dimensional "grid".

Clustering of the code vectors in 2-D SOM is usually presented color coded or with markers on another plane [17]. In this 1-D case the clustering information can be integrated into the line plots. The clusters are separated by vertical lines. The numbers of the clusters from 1 to 7 are below variable 1 at the bottom of the figure.

Groups 1 and 2 are located at the end of the SOM line. They are clearly the only ones having high values in variable 5. These states of the system are common enough not to be detected as anomalies. But still rare enough, so that they should be examined further.

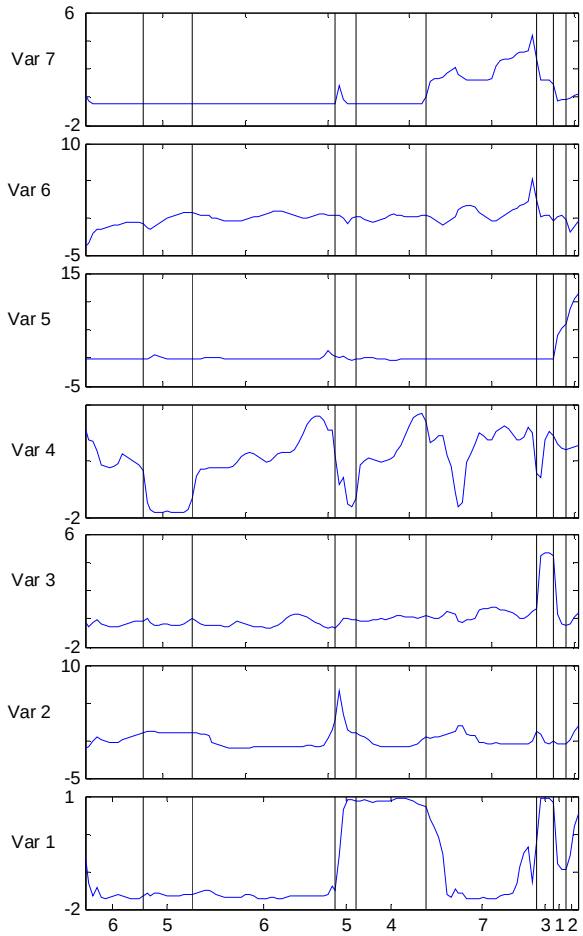

Figure 7 Code vectors of the SOM with reference groups numbered below.

The end users in real life application don't want to see the anything but the essential. Plain time series plot with the detected outliers is often useful for the system expert. Figure 8 gives an example of a time series. The anomalies detected in the reference data are highlighted with a horizontal line and a star. The most interesting anomalies can then be further studied from the original logs.

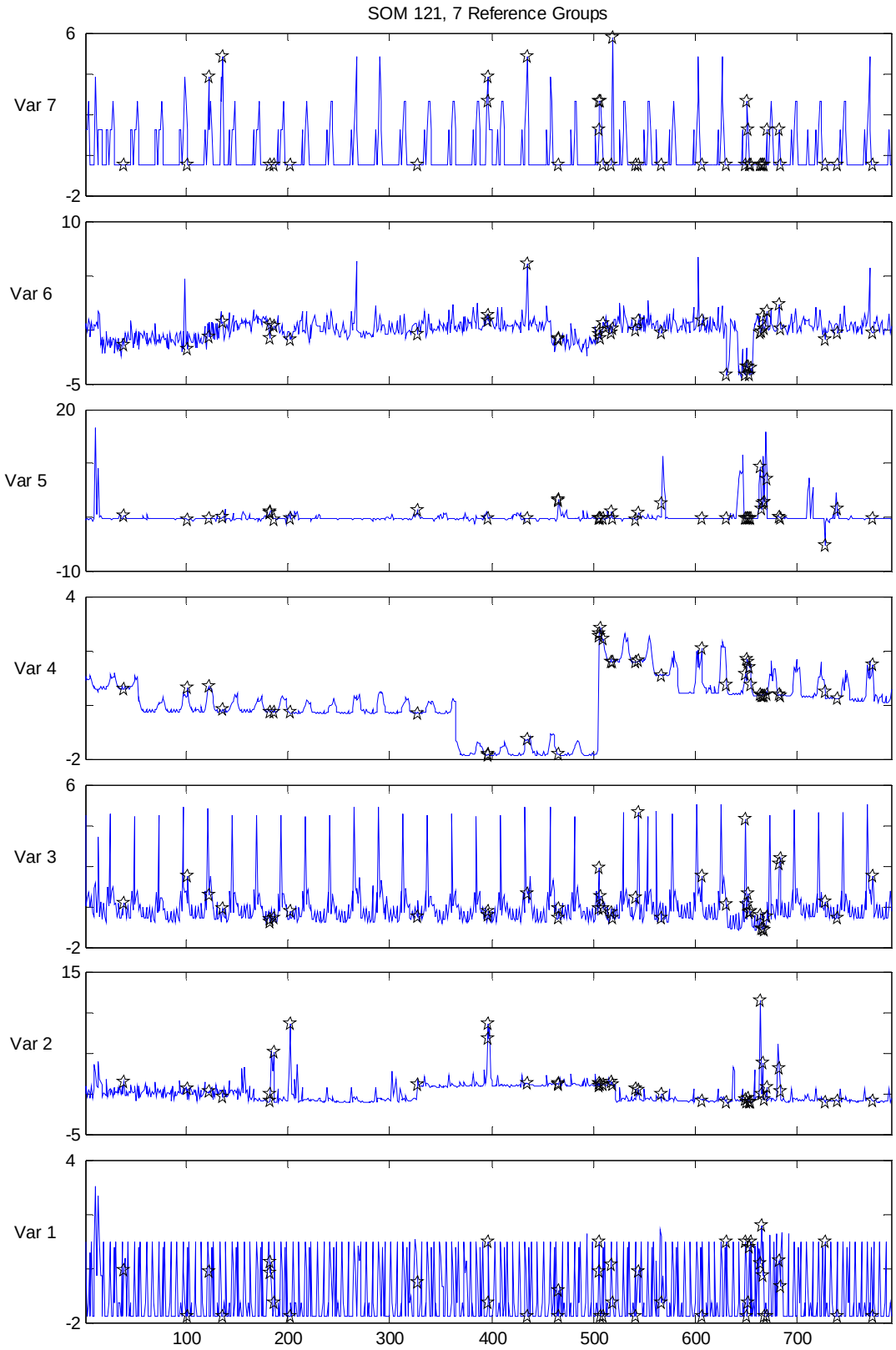

Figure 8 Time series plot of the reference data, outliers marked with stars

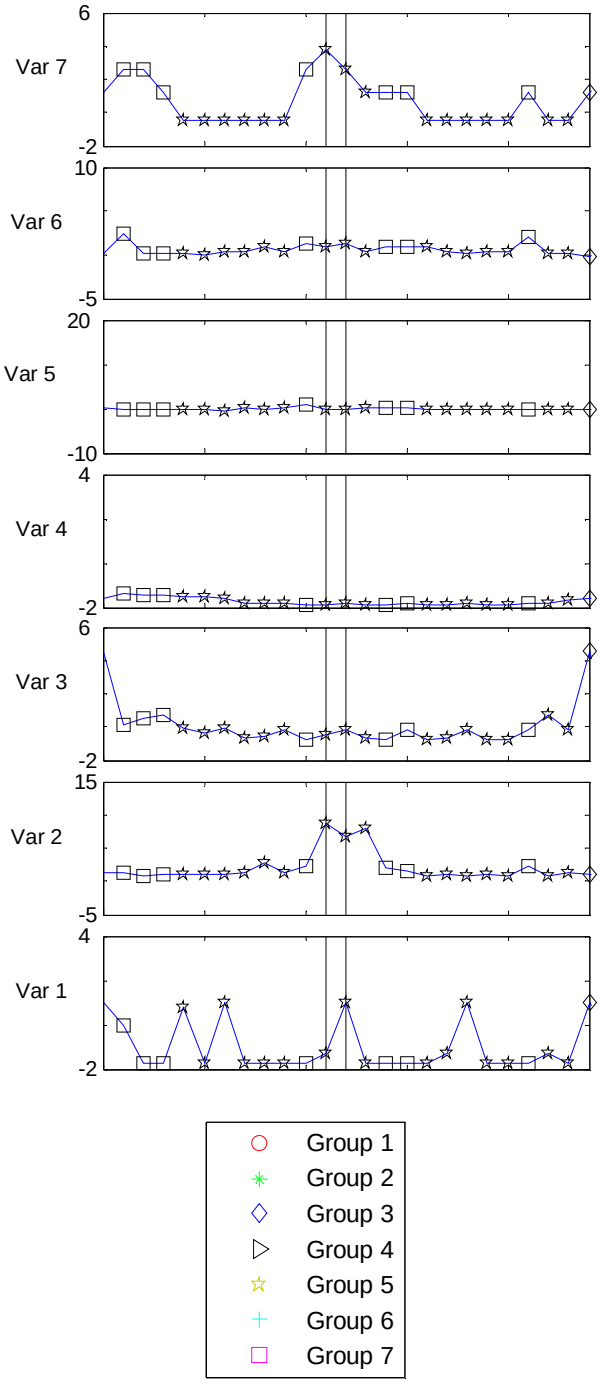

Figure 9 Time series zoomed in for one day, reference groups of samples shown by markers

Figure 9 shows the time series plot zoomed in for a one day period. The reference groups, where each sample belongs to, are coded by markers shown below.

3.3 Use case 2, the test data on data set 1

In the second use case we have a new data set, test data, recorded after the anomaly detection model was identified. It contains 9 days, 216 samples of hourly data right after the reference data set. The test data are scaled the same way than the reference data, using the scale factors calculated from the reference data.

The BMU is searched for each sample of the test data. The BMU defines the reference group and the anomaly threshold to use for each sample.

Table 1 Number of samples and detected anomalies in test data associated to reference groups

Reference Group	Number of samples in test data	Anomalies in test data
1	1	0
2	0	0
3	9	0
4	36	5
5	3	0
6	129	35
7	38	5

Table 1 lists the number of samples of the test data that are assigned to each reference group together with the number of samples detected as anomalies. Majority of the data are assigned to group 6. There are also a lot of anomalies in that group.

Next we want to find out why there are so many anomalies. Figure 10 shows box plots [13] of the error vectors from the anomalous samples in group 6. Error vectors gives distance between data sample and its BMU code vector. The most obvious reason for anomalies is variable 4, which is far on the negative side. This indicates that in all these samples this variable has much lower values than the corresponding code vector in their BMU nodes.

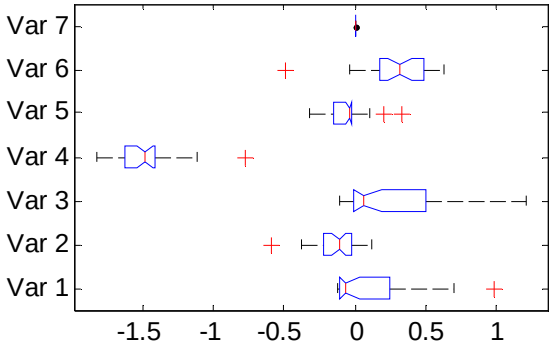

Figure 10 Box plot of the quantization error of the anomalies in test data, which are assigned to reference group 6.

Figure 11 compares the test data that was assigned to the group 6 and the code vectors in the same group. The code vectors present the approximated normal behavior in the group. Values of variable 4 are well below the ones in code vectors. On the other hand, those values of variable 4, close to -2, do exist in the reference data as seen in Figure 8. Such states are assigned to groups 5 and 7; it is the combination of the other variables that makes these samples to be assigned to group 6 in the test data.

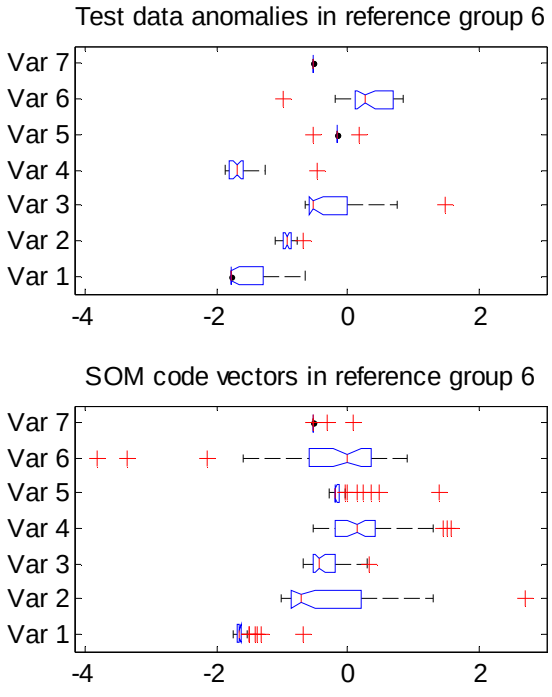

Figure 11 Box plots of the anomalies in test data and SOM code vectors in reference group 6

Variable 2 is also lower in the test data anomalies than in code vectors. This motivated to inspect the combination of variables 2 and 4, which is illustrated in Figure 12. The figure shows a part of the reference data and the test data of variables 2 and 4. They seem to be able to explain the anomalous behavior in the group 6. Only the anomalies detected in that group are presented in the figure by stars over the line. All the data after the gray vertical line are test data.

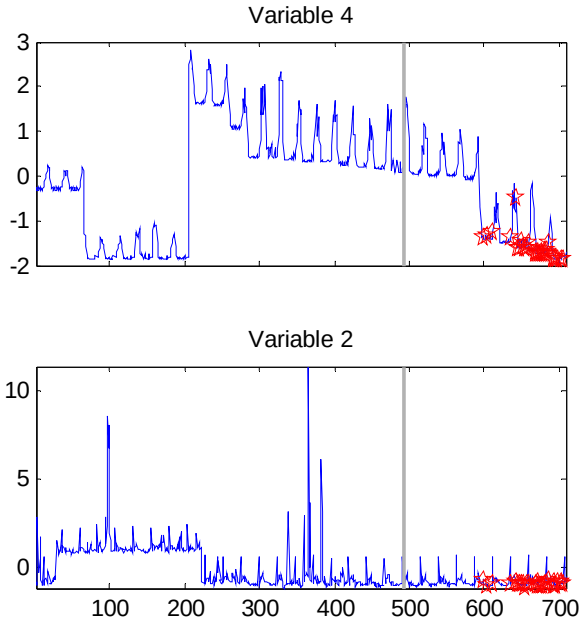

Figure 12 Variables 4 and 2

The samples detected as anomalies in the end of the test data set are from normal operational state. Both variables are in their normal range. But this kind of combination was not present in the reference data. Such groups of similar anomalies are a sign of new unseen behavior and the model should be updated.

When the whole model is identified, the reference groups will change. Reordering the new groups so that the new group number 1 is the one that is the most similar with the old group 1 etc. will help the users in practice.

3.5 Comparison of methods on data set 1

Both, the original global method and the new local one were applied to the data set 1. The anomalies detected by both methods are presented in Figure 13.

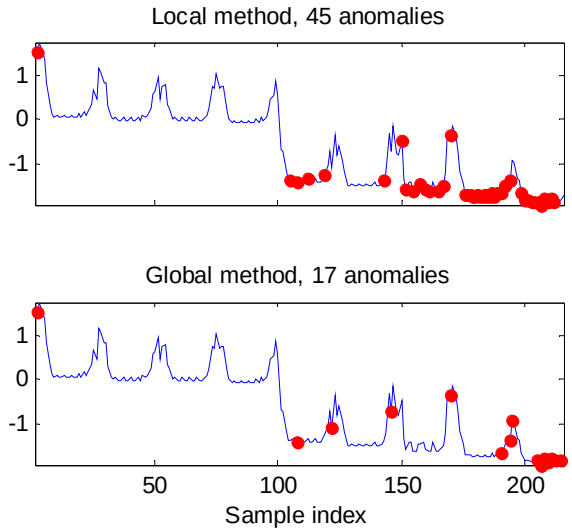

Figure 13 Variable 4 with the anomalies detected by local and global methods

The global method detects 17 anomalies while the local one finds 45. More sensitive detection is not generally desirable. However in this case the system has entered a new state, not present in the reference data as described in previous section. The local thresholds in the improved model allow this state to be detected earlier. The global threshold is too high for this state to be detected to be something new, until at the very end where the variable 4 has very low values.

3.5 Comparison of methods on data set 2

The second data contains 8 variables recorded from a similar system as the first data set. The reference data, which was received fist, contains 13 days of data, 312 samples. 205 samples were received later to be used as test data.

The anomaly models were identified using the same scaling and model parameters as in the previous case. In this case we end up with a SOM of 30 nodes and 7 reference groups in the local model.

The numeric results of the local method are presented in Table 2. The first column is the reference group number. The second column is the number of samples in the reference data associated in each group. The third and fourth volumes are the numbers of samples and the detected anomalies from the test data set.

The test data differs from the reference data. There are no normal samples in two biggest reference groups 1 and 7. Most of the test data falls into groups 4 and 5, which cover only 18% of the reference data. This is a clear indication that the identified model is not valid. The global method doesn't provide this information.

Table 2 Number of samples in reference and test data associated to reference groups followed by the number of anomalies detected from the test set in each group

Reference Group	Samples in reference data	Samples in test data	Anomalies in test data
1	84	11	11
2	11	0	0
3	24	0	0
4	17	57	5
5	40	127	13
6	16	10	0
7	120	0	0

Total number of anomalies detected by the local method is 29. The global method detects 77 anomalies. Time series of one variable and the locations of the anomalies are presented in Figure 14. The third plot at the bottom is the reference group associated for each sample in the test data.

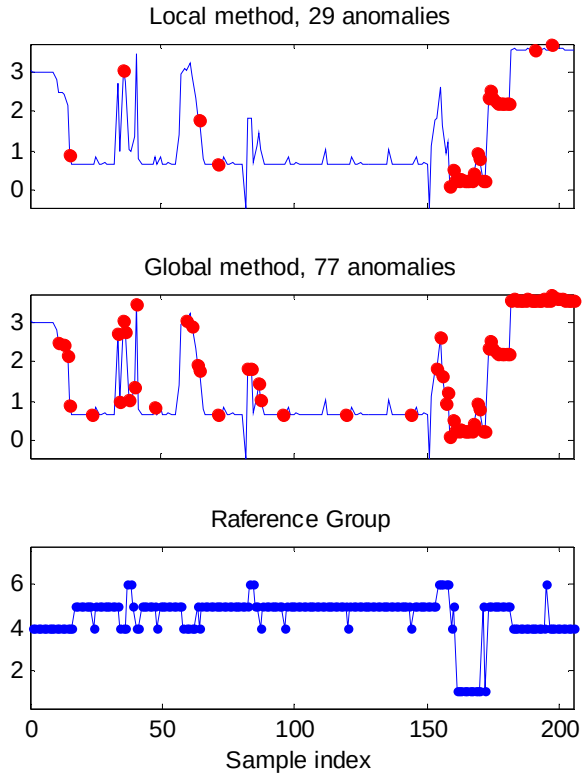

Figure 14 Time series example with anomalies and the reference group in test data set 2.

In this case the global method detects more anomalies, most of which are within normal local variation according to the local method.

In on-line anomaly detection the test data will be analyzed either one sample at a time or 24 samples if used on a daily basis. In that use the global method gives a number of unnecessary detections. Finally towards the end the number of anomalies rises high enough to indicate the need to update the model.

The local method gives less anomaly detections since the samples are from system states, where the variation was larger also in the reference data. However there is still the indication of the need for model update visible in a very early phase. The test data is mostly associated to the reference groups 4 and 5 in the beginning of the test set. Both these are small groups. The test data falling into these groups is indication that the system is in a states which were not sufficiently covered in the reference data.

4. Conclusion

We have introduced an anomaly detection method that uses SOM and clustering. This method is independent of the distribution or clustering structure of the data. It also takes into account the local variations in variance. Therefore it can be used in various systems that produce multivariate data. The scaling of the data is essential, as in all methods using distances or variances. We introduce scaling that is suitable for the system log data. When transferring the method to another environment, the scaling has to be revised to match the importance of the variables used in the analysis. The method scales up to larger systems and it can be used also for large data sets.

We compared the original global method to our local method using two data sets. In one data set the local method gives a warning about the system entering a new state earlier than the global method. In the other data set the local model gives fewer false alarms while the system performs within normal variation in the state it is in, whereas the global threshold given by the original method is lower resulting in a larger amount of detections.

This method has proved in practice to be useful in finding outliers and new phenomena in network system log data. It is applicable to both on-line monitoring and analysis of the history data.

5. References

[1] Kruskal, W. H., “Some Remarks on Wild Observations”, Technometrics, Vol. 2, No. 1 (Feb., 1960), pp. 1-3.

[2] Hawkins D., Identification of outliers. Chapman & HaIl, London, 1980.

[3] Breunig S., Kriegel H.-P., Ng R., Sander J.: 'LOF: Identifying Density-Based Local Outliers', ACM SIGMOD Int. Conf. on Management of Data, Dallas, TX, 2000.

[4] Höglund A. J., Hätönen K. and Sorvari A. S., 2000, “A computer host-based user anomaly detection system using the self-organizing map”, Proc. IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN), vol. 5, pp. 411-416. © 2000 IEEE.

[5] Hätönen, K., Kumpulainen, P., Vehviläinen, P., “Pre and post-processing for mobile network performance data”, In Proceedings of seminar days of Finnish Society of Automation, (Automation 03), Helsinki, Finland, September 2003

[6] Hätönen, K., Laine ,S., Similä, T.,. Using the LogSig-function to integrate expert knowledge to Self-Organising Map (SOM) based analysis, IEEE International Workshop on Soft Computing in Industrial Applications, Birmingham University, New York, June 23-25, 2003

[7] Barnett, V., Outliers in statistical data, Wiley, Chichester 1987

[8] Grant, E.L., Leavenworth, R.S., Statistical quality control. 7th ed. New York, McGraw-Hill. 764 p, 1996.

[9] Fuchs, C., Kenett, R., Multivariate Quality Control. New York, Marcel Dekker, Inc,. 212 p. 1998

[10] Kano, M., Hasebe, S., Hashimoto, I. Evolution of Multivariate Statistical Process Control: Application of Independent Component Analysis and External Analysis. Proceedings of The Foundations of Computer Aided Process Operations Conference (FOCAPO 2003). Coral Springs, US, Jan. 12-15 2003. pp.385-388.

[11] Knorr, E. M., Ng, R. T., Tucakov, V., Distance-Based Outliers: Algorithms and Applications. The VLDB Journal The International Journal on Very Large Data Bases, 8(3-4), Springer Berlin / Heidelberg. pp. 237--253. 2000

[12] Kohonen, T., Self-Organizing Maps, Springer, Berlin,

[13] McGill, R., Tukey, J.W., Larsen, W.A., Variations of Boxplots, The American Statistician, Vol. 32, pp.12-16, 1978.

[14] D.L. Davies and D.W. Bouldin, “A cluster separation measure”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 1, no. 2, pp. 224-227, April 1979.

[15] SOM Toolbox,
http://www.cis.hut.fi/projects/somtoolbox/

[16] Ward, Jr, J. H., Hierarchical Grouping to Optimize an Objective Function, Journal of the American Statistical Association, Vol. 58, No. 301. (Mar., 1963), pp. 236-244.

[17] Vesanto, J., Esa Alhoniemi, E., Clustering of the Self-Organizing Map, IEEE Transactions on Neural Networks, 11(3), May 2000