12th IMEKO TC1 & TC7 Joint Symposium on
Man Science & Measurement
September, 3 – 5, 2008, Annecy, France

COMPRESSION OF CYCLIC TIME SERIES DATA

Pekka Kumpulainen[1], *Kimmo Hätönen*[2]

[1] Tampere University of Technology, Tampere, Finland, pekka.kumpulainen@tut.fi
[2] Nokia Siemens Networks, Helsinki, Finland, kimmo.hatonen@nsn.com

Abstract: This paper describes a method to reduce the space of stored indicator time series data collected from a telecommunications network. The method takes advantage of the cyclic nature of the collected time series and behavioral clusters of the network elements. The method separates the shape of the daily activity cycle of a network element from volume of the traffic and clusters the daily patterns to coherent groups in which each sample can be presented with the amount of traffic in the network element and a prototype pattern. Dynamic thresholds are used to determine the samples that deviate from the prototype too much and require being stored separately.

Keywords: data reduction, daily patterns, dynamic thresholds, anomaly detection

1. INTRODUCTION

Collection and storage of performance monitoring data are an ever growing problem for telecommunications network operators. The number and complexity of offered services are increasing and scattering towards the edges of the network. Thus the amounts of collected data as well as the number of data sources multiply. The need for efficient methods to reduce the size of data becomes evident.

Two well known methods for data reduction are aggregation and approximation [1]. Aggregation in both variable and time domain is widely used in telecommunications network management [2, 3]. The variable aggregation is typically done when querying the data from the database. All the variables are stored in the database and aggregated for information presentation purposes. Time domain aggregation is used for data storage, the counters and quality factors are summed or averaged over given time period. For monitoring purposes the time period lengths typically range from 5 minutes to an hour. For long history storage these are further aggregated to daily sums or averages due to huge amounts of data. This kind of aggregation loses all the information about the variation within a day.

Mobile networks are monitored using the data generated in the network elements (NEs). The elements count all the operations they perform to establish a data or voice connection. These operations vary from voice connection or data context reservation attempts to handovers or connection shutdowns. The elements also monitor and record validity of connections by recording detected error rates, signal strengths and other physical quantities describing connection quality. As a result each network element provides a time series of values for each observed indicator.

The usage and traffic in a mobile network varies according to the rhythm of life of surrounding society [4]. The traffic is high when and where there are a lot of active people and low if there are only a few people or they are passive like in the suburban area late in the night or very early in the morning. Each network element resides in a unique physical and geographical environment. The amount of transferred traffic, processed handovers, or the length of calls and their distributions over the day, week and year forms a unique pattern for each element. This pattern includes daily, weekly, and yearly cycles that make the values range between quiet and busy periods.

We introduce a method to reduce the amount of data to store in the database using approximation and anomaly detection. The method is based on the shapes of daily patterns. We distinguish the shape and the volume of traffic and search the daily patterns that best describe the shapes. The data to store are approximated by daily pattern prototypes. Only samples that deviate too much from the pattern are stored. Individual dynamic thresholds are used for each sample. The performance of the method is illustrated. The performance of the compression method using dynamic and the constant thresholds are compared considering the compression ratio and the absolute error introduced in compression..

2. PURPOSE

The main application for the developed method is to reduce space of storage needed to store the data of indicators containing strong periodical cycles. For example, the telecommunication operator database contains usually hundreds of such indicator series, the best known and most important one being the amount of traffic.

Figure 1 illustrates a typical time series of a traffic counter in a mobile network. The value on the y axis is calls per hour: the number of calls handled by one cell in one hour time. Thus there are 24 samples per day. The time period in this example is one week and the numbers of the days on the x axis are located at midnight, starting on Monday.

The daily cycle of life is clearly visible. The night time is very quiet and the amount of calls within an hour in daytime varies between 200 and 400. In this specific cell the weekend does not differ from the rest, in cells located at business areas the weekends usually have significantly less traffic. The shape of each daily pattern changes from day to day.

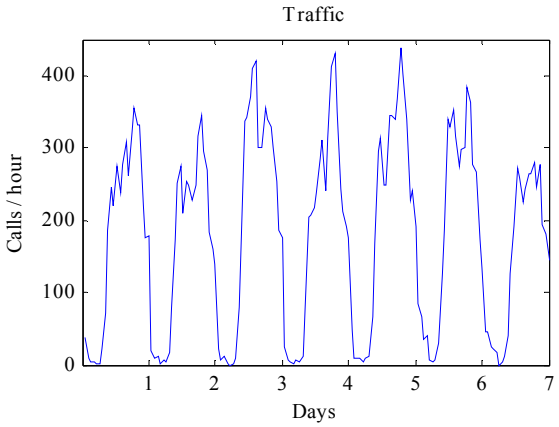

Fig. 1. One week period of network traffic

The primary objective of this work has been to develop a method to compress vast amounts of cyclic measurement data in operator data storage. A common storage solution has been to keep only few weeks of the detailed time series data in the database. The older data has been aggregated to a long term data warehouse and the detailed data is compressed and archived to the backup system. If the details are needed, the data has to be searched for, uncompressed and uploaded from the backups to the database management system.

Our solution aims at prolonging the period of time, which the detailed data can be kept accessible in the database.

3. CYCLIC DATA COMPRESSION METHOD

In this section we present the method using the amount of traffic as an example. We assume the traffic data to be aggregated to one sample per hour and cyclic patterns of one day long. The method can be used for variety of sample rates and pattern lengths.

The data storage minimization is based on replacing each daily pattern by a prototype pattern and a scale factor. This reduces the required storage from 24 samples per day to only one scale factor and the identifier of the descriptive prototype pattern. In addition the prototype patterns have to be stored but the number prototypes in our applications is typically between 3 and 10. This is significantly less than the total number of daily patterns, the product of the number of days and the number of network elements.

The true daily profiles naturally differ from the prototypes and thus an error is introduced when the daily data are approximated by the prototypes. In order to reduce this error we detect and separately store samples that deviate from the prototype more than a threshold. Any anomaly detection method could be used to find the samples. We introduce dynamic thresholds to minimize the error while still keeping the number of individually stored samples low.

The method divides to three phases: training, storage and restoring. The steps in these phases are listed below and the following subsections describe the steps in detail.

Training

- Preprocess the samples
- Identify the prototype patterns
- Give identifiers to prototypes (optional)

Storage

- Preprocess the samples
- Find corresponding prototype patterns for each sample
- Compute dynamic thresholds
- For each daily pattern store the prototype ID, daily mean and the samples outside the dynamic thresholds

Restoring

- To restore the pattern, scale the prototype with the daily average and replace individual samples if they exist

3.1. Data

The indicator time series of each NE are split to daily patterns. Each pattern covers 24 hour period of time. The patterns are concatenated to form a data matrix that has 24 columns, one for every hour of the day. The number of rows in the data matrix equals the product of the number of days covered by the measurement period and the number of network elements, $N = N_{days} * N_{NE}$.

Thus the traffic data matrix $\mathbf{T} = [\mathbf{t}_1\ \mathbf{t}_2\ \ldots\ \mathbf{t}_N]^T$ where each row $\mathbf{t}_i = [t_{i0}\ t_{i1}\ \ldots\ t_{i23}]$ contains hourly traffic of one cell for one day and is called is a *daily pattern*.

3.2. Preprocessing

Preprocessing is an essential part in data analysis [5]. The preprocessing phase is similar both in training and application of the method. The main objective is to distinguish the shape from the volume by normalizing all the elements of the sample. The scaled matrix is $\mathbf{T}^s = [\mathbf{t}^s_1\ \mathbf{t}^s_2\ \ldots\ \mathbf{t}^s_N]^T$, where each row is scaled by dividing it by its mean value μ_i, the daily average traffic

$$\mathbf{t}_i^s = \frac{\mathbf{t}_i}{\mu_i} \tag{1}$$

The mean of each scaled pattern equals 1. This levels out the differences in total traffic and makes it possible to use distance measures required by clustering methods.

Figure 2 depicts an example of the scaling on two cells that have distinct levels of activity. The sample at midnight has been discarded in our data due to large amount of missing values. This is most probably due to the data collection that has been scheduled to occur at midnight in the network segment, where the data has been collected from. This has made the data collection to miss simultaneously added traffic values.

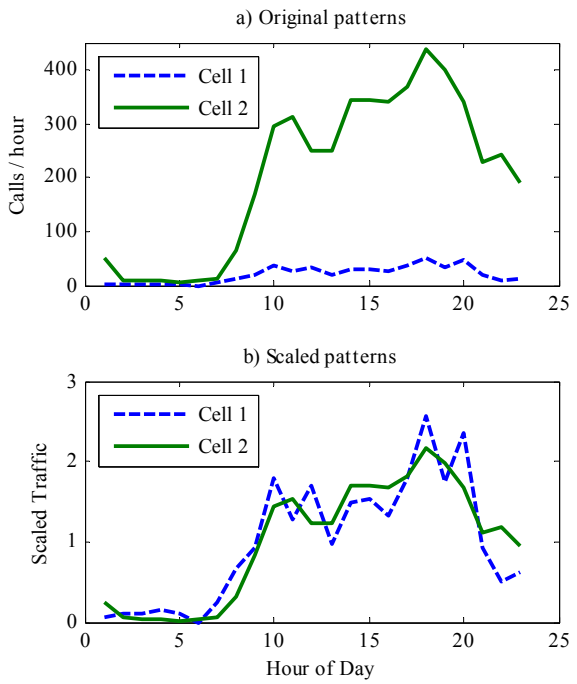

Fig. 2. Examples of two daily patterns from cells that have distinct levels of traffic but similar shapes

The original traffic patterns are presented in Figure 2 a). Cell 2 has significantly more traffic than Cell 1. The scaled patterns in Figure 2 b) show that the daily patterns of these two cells on these specific days are relatively similar when the difference in the total traffic has been compensated by scaling.

This similarity is the basis for the compression method introduced in this paper. Network elements share a small number of behavior profiles, which reflect the rhythm of people using mobile networks. The scale of the profile changes from an element to another, but the rhythm of the behavior keeps the same.

Figure 2 b) also shows that when the patterns are scaled the one that has lower traffic levels has also more deviation. The same difference in the number of calls will yield larger relative deviation when the absolute value is lower.

This observation is another corner stone of the introduced method. When we search for prototype for daily pattern, we have to consider not only the shape, but also the volume of the traffic that the scaled pattern represents. If the traffic was low compared to the average, we have to allow larger deviations from the prototype than if the traffic volume was high.

3.3. Identification of the prototype patterns

The training of the system continues with finding the prototypes that describe the most essential ones of the daily patterns.

In order to find the prototypes the data is clustered. The training data is typically a set of samples recently collected from the network. Usually the objective is to find a normal or problem free description of the behavior of the cells and therefore the data set should be cleaned from abnormal or problematic samples.

In principle the method used for clustering can be any unsupervised method that is able to find cluster centroid from multidimensional data. We have chosen to use the hierarchical clustering with Euclidean distance and Ward linkage [6, 7].

The cluster centroids are used as prototype patterns in the application phase. The centroids $\boldsymbol{m}_C$ denote the averages of daily patterns being assigned to each cluster.

3.4 Dynamic thresholds

The most basic threshold to detect anomalous samples for example in statistical process control [8], is k*σ where σ denotes the standard deviation of the variable and k the coverage factor, which tells the required confidence level. For normal distribution k=1.96 means 95 % level. Coverage factors 2 and 3 are often used regardless of the underlying distribution. In this study the standard deviation σ is calculated separately within each cluster and for each time instant.

To take the volume of the traffic into consideration, we scale the confidence limits so that for higher traffic, i.e., for a sample with larger daily mean, we get tighter limits. In addition to the coverage factor we take into account the daily average traffic of each day. Thus we get a unique coverage factor for each day as a function of the daily average traffic and the cluster centroid.

$$k_i^{Dyn} = \frac{k}{\sqrt{\mu_i / \mu_C}} \tag{2}$$

μ_i is the mean traffic of the ith day and μ_C is the mean of all the patterns in the cluster in which the ith day was assigned to.

This is analogous to standard deviation of mean estimate, which is proportional to the inverse of the square root of the number of samples used in calculation.

The dynamic threshold is

$$\mathbf{DT}(h) = \mathbf{m}_C \pm k^{Dyn} \boldsymbol{\sigma}_C(h) \tag{3}$$

$\boldsymbol{m}_C$ is the cluster centroid, and $\boldsymbol{\sigma}_C(h)$ is the standard deviation of the data within the cluster for each hour, h, of the day.

The constant threshold for daily patterns in our case is

$$\mathbf{T}(h) = \mathbf{m}_C \pm k\boldsymbol{\sigma}_C(h) \tag{4}$$

An example of the constant and dynamic thresholds is given in Figure 3. The results are presented for the same two patterns which were shown in Figure 2. They are assigned to the same cluster, thus represented with the same prototype. Cell 1 which has lower average traffic will have the dynamic thresholds wider apart as seen in the upper part of Figure 3. The dynamic thresholds of Cell 2 are almost identical to the constant ones.

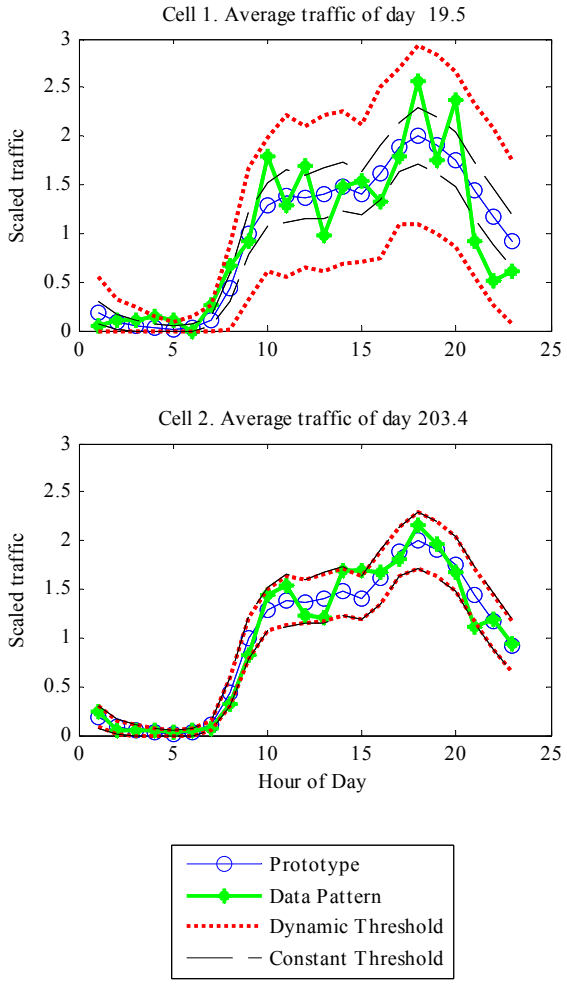

Fig. 3. Example of two patterns including the thresholds

An example of the constant and dynamic thresholds as a function of the average daily traffic is given in Figure 4. The data in the scatter plot are from one cluster and at one specific hour of the day, 12 o'clock in this figure. The dots present the scaled traffic samples as a function of the daily mean of each day.

The dashed lines present the constant thresholds and the dynamic thresholds are depicted by the solid curves. The thick solid line in the middle is the value of the prototype at that hour, which is used to present all the samples except those outside the thresholds.

When the mean traffic of the day is equal to the mean of all the traffic in the cluster, the denominator in equation 3 equals 1 and the constant and dynamic thresholds are equal. In this example that point is at 190 where the thresholds cross. For lower daily traffic the dynamic thresholds allow more variability. When the daily traffic is higher the dynamic thresholds are tighter.

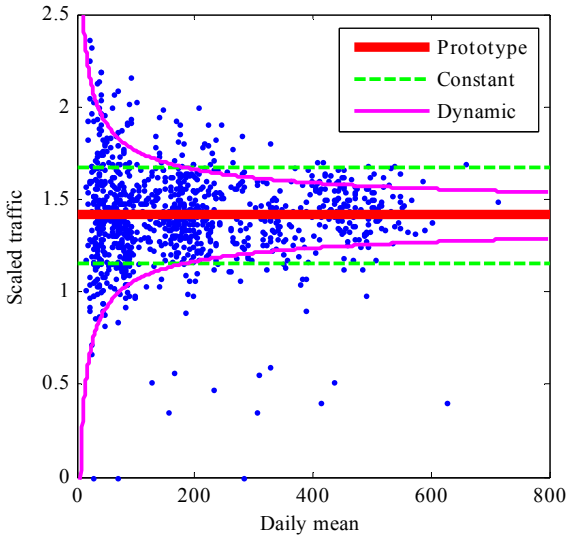

Fig. 4. Example of samples and confidence limits

4. RESULTS

In this section we present results using data collected from a mobile network. The data contains the traffic measurement one sample per hour for a period of 42 days from 100 network elements. Due to large amount of missing values at midnight that hour was ignored. Therefore the daily patterns consist of 23 samples. The data was also cleaned by leaving out daily patterns that contained many missing values or otherwise abnormal behavior. We ended up having 3865 daily patterns of length 23 samples. The data were scaled using equation 1.

We used hierarchical clustering to find the prototype patterns. We used Euclidean distance measure and Ward linkage. The final selection of the number of the prototypes to use is case-specific. Here we present results for various numbers of clusters. We used data reduction and the approximation error to measure the efficiency of this method.

The reduction of the data is represented by the percentage of the samples outside the thresholds, which is the amount of the samples that need to be stored separately. This is depicted in Figure 5 as a function of the number of clusters. The number of samples outside the thresholds increases when more clusters are used. The dynamic thresholds will result in fewer samples exceeding the thresholds than the constant ones.

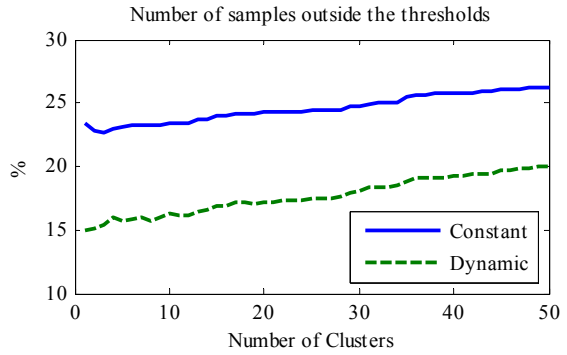

Fig. 5. Constant and dynamic thresholds compared. Percentage of samples outside the thresholds as function of the number of clusters

The compressed presentation of each daily pattern requires two figures: the reference to the prototype and the average daily traffic. In addition the samples exceeding the thresholds are stored. When N_{out} is the number of the samples outside the thresholds and N_s is the number of samples in each pattern, the compression ratio for each daily pattern will be

$$CR = 100 * \left(1 - \frac{2 + N_{out}}{N_s}\right) \quad (5)$$

In our dataset we have discarded the samples at midnight, thus $N_s = 23$.

Figure 6 depicts the total compress ratio for the whole data set using constant and dynamic thresholds. Increasing the number of clusters i.e. the number of prototypes will decrease the compression ratio. However, the dynamic thresholds will result in better compress ratio in data storage.

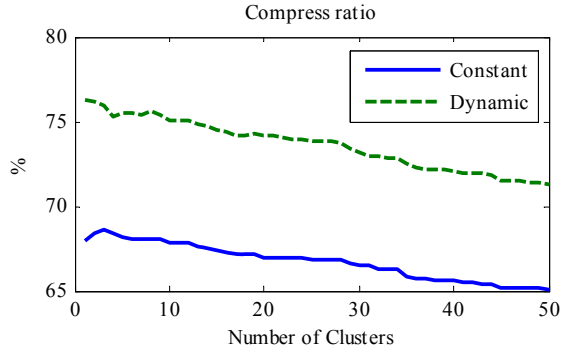

Fig. 6. Compress ratio as function of the number of clusters

The approximation error is presented by the average of the error: the difference between the prototype and the actual data samples within the thresholds. These samples are the ones represented by the prototype. Figure 7 illustrates the error for both dynamic and constant thresholds as a function of the number of clusters, i.e. the prototype patterns used.

The error will decrease when the number of prototypes increases. The average errors are below 18 Calls per hour even in the worst case. Considering that the actual amount of calls in daytime in this data set has a range from 10 up to 1500 calls per hour, this error level is well acceptable. The usage of dynamic thresholds performs better on all numbers of clusters. The error is about 20% smaller when the dynamic thresholds are used.

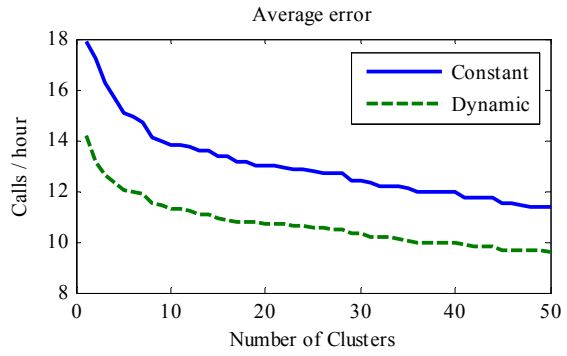

Fig. 7. Constant and dynamic thresholds compared. Average absolute error as function of the number of clusters

The performance of the constant and dynamic thresholds is presented numerically in Table 1.

Using the dynamic thresholds gives better performance at every number of clusters, the number of prototypes used to approximate the data. There are fever samples outside the thresholds, producing better data reduction. At the same time the mean error introduced by the approximation remains smaller.

Table 1 Percentage of samples outside thresholds and mean of error for samples inside the limits.

	Dynamic threshold		**Constant threshold**	
Clusters	Compress ratio	Mean error	Compress ratio	Mean error
1	76.3	14.2	67.9	17.9
5	75.5	12.1	68.2	15.0
10	75.0	11.3	67.9	13.8
25	73.8	10.5	66.9	12.7
50	71.3	9.6	65.1	11.4

Increasing the number of prototypes will reduce the error but also increase the number of samples outside the thresholds and the data reduction will suffer.

The number of the prototypes to use could be selected by some clustering criteria, such as a commonly used Davies-Boulding index [9]. However, while such criteria do optimize the number of clusters, the result may not be optimal for the data compression application. A more appropriate method for the selection of the number of prototypes is to choose a compromise between the compression ratio and the total error allowed using the curves in figures 6 and 7.

The thresholds are directly proportional to the coverage factor k. The effect of the k on the compress ratio and the average absolute error are presented in Figures 8 and 9. These results are calculated using 5 prototypes. As suggested above, it was selected as a compromise between the compress ratio and the average error.

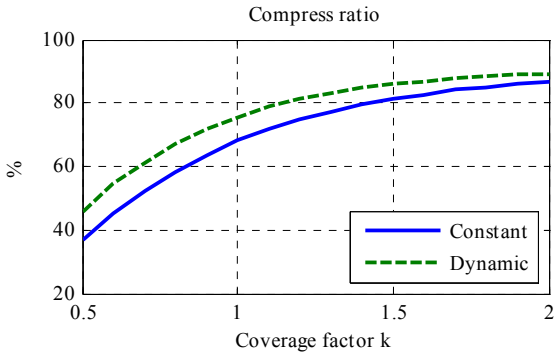

Fig. 8. Compress ratio as a function of the coverage factor k

The compress ratio depicted in Figure 8 increases for larger values of the coverage factor *k*. The compress ratio seems to have logarithmic relation to the *k*. Increasing the *k* for values larger than 1.5 does not increase the compress ratio significantly. Using the dynamic thresholds will give better performance than the constant ones throughout the whole range of the coverage factor.

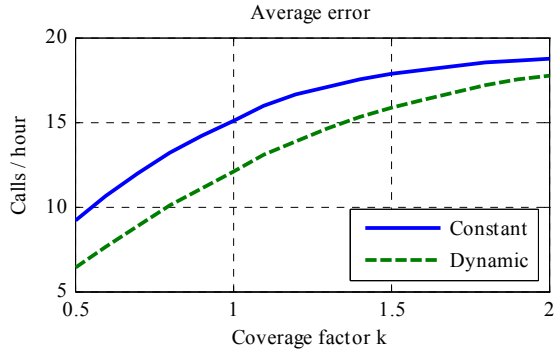

Fig. 9. Average error as a function of the coverage factor k

For higher values of coverage factor *k* the average error will increase as presented in Figure 9. The relation of the average error to the *k* is more linear than the one of the compress ratio. The usage of the dynamic thresholds will yield in smaller error for all values of *k*.

Increasing k will result in better compress ratio but the error will also increase. A good compromise value for *k* in this data set seems to be between 1 and 1.5.

5. Discussion

The suggested method has proved to be useful in practice. In common network management solutions details are hidden, when the data is aggregated and original time series are removed to the archive. The method enables operators to store longer histories of detailed data in directly accessible database. This prolongs the availability of detailed information.

The presented method does not loose any information as happens in time aggregation, in which the data is summed per day per network element. On the contrary, in the presented method, a daily sample is stored by using daily average and the reference to the matching prototype. The stored daily average defines the daily sum and the referred prototype provides additional information.

From archives it is often very tedious to find and fetch the data back to analysis. With the presented method it is possible also to improve the effectiveness of the archiving solution. If the method is applied inside the database and the compression ratio of, e.g. 75% is achieved, the size of the archived data of certain time period, e.g., one week, will decrease correspondingly. This will mean shorter execution times of archiving processes and smaller physical archives, which are easier to handle.

In addition to the data reduction, the method introduces other useful results as a side effect. For example, the method characterizes network elements based on their daily behavior. This information can be used, e.g., in the network optimization, where parameter profile is created for elements corresponding to certain traffic profile. The information of daily traffic prototypes that the presented method extracts can be used to identify those elements that share the characteristics and to which the created parameter profile can be distributed.

The method serves also as an anomaly detection method. The hourly samples outside the thresholds can be regarded as anomalies. When the dynamic thresholds are used they will be traffic sensitive anomalies.

Another possible application for the method would be the estimation of the missing values. Every now and then it happens that due to some network problems the network management system doesn't manage to collect some part of the data. This will cause error in reporting and difficulties in following the long term trends in the network behavior. The method presented in this paper could be used as a basis for estimating the missing values. This will also be left as a future enhancement of the method.

6. Conclusions

In this paper we have presented a method to identify repeating information from telecommunications network indicator time series and to use that information to compress the analyzed time series. Often the length of the available cycle is 24 hours. The method takes advantage of the shape of the cycle by removing the volume. The shapes of training samples are clustered. In application a reference to a best matching cluster centroid is used to replace the values of the daily sample. If the sample contains deviating values that are outside the dynamic confidence limits that take into account the average volume of the sample, the original value is stored in addition to the cluster reference.

The method is simple and understandable and easy to implement and maintain. It has been demonstrated with indicators describing traffic in cells, but it is general and can be applied to all indicators containing constant daily cycles.

References

[1] A. Deligiannakis, Y. Kotidis, "Data Reduction Techniques in Sensor Networks", IEEE Data Engineering Bulletin, Vol. 28, No. 1, March 2005.

[2] J. Suutarinen, "Performance measurements of gsm base station system", thesis (lic. tech), 1994.

[3] M. Kylväjä, P. Kumpulainen, K. Hätönen, "Information Summarization for Network Performance Management", In: M. Laszlo, J.V. Zsolt, (eds.). Proceedings of the 10th IMEKO TC10 International Conference on Technical Diagnostics, Budapest, Hungary, pp. 167-172, 2005.

[4] H. Khedher, F. Valois, S. Tabbane, "Traffic characterization for mobile networks", Proceedings of the 56th Vehicular Technology Conference, 24-28 Sept. 2002, IEEE, Volume 3, pp. 1485 - 1489, 2002.

[5] K. Hätönen, P. Kumpulainen, P. Vehviläinen, "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] R.A. Johnson, D.W. Wichern, "Applied multivariate statistical analysis 4th Edition", Prentice Hall, 1998.

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

[8] J.S. Milton, J.C. Arnold, "Introduction to Probability and Statistics Principles and Applications for Engineering and the Computing Sciences", Second edition. Singapore, McGraw-Hill. 700 p. 1990.

[9] 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.