DE GRUYTER
OPEN

JACSM 2017, Vol. 9, No. 1, pp. 5 - 22
10.1515/jacsm-2017-0001

ANOMALY DETECTION IN SERVER METRICS WITH USE OF ONE-SIDED MEDIAN ALGORITHM

Szymon Zacher [1], Przemysław Ryba[2]

[1] OVH SAS, Wrocław, Poland
szymon.zacher@corp.ovh.com

[2] Department of Systems and Computer Networks
Wrocław University of Science and Technology, Wrocław, Poland
przemyslaw.ryba@pwr.edu,pl

Abstract

In this paper we consider the problem of anomaly detection over time series metrics data took from one of corporate grade mail service cluster. We propose the algorithm based on one-sided median concept and present some results of experiments showing impact of parameters settings on algorithm performance. In addition we present short description of classes of anomalies discovered in monitored system. Proposed one-sided median based algorithm shows great robustness and good detection rate and can be considered as possible simple production ready solution.

Key words: anomaly detection, time series, one-sided median, server metrics

1 Introduction

Each system has to be monitored in order to gain certain level of stability and robustness. In the world of fast data exchange, behavior of complicated system can change rapidly during very small amount of time. Therefore not only monitoring of system has to be performed, but also constant analysis of collected data in order to detect each signs of instability or undefined behavior. In traditional approach these activities are performed by specialized and trained administrator who keeps track on incoming data and decide whether any changes of system behavior are anomalous or not. Hopefully in modern world of machine learning and data processing we do not have to rely completely on human work, we can craft a mechanism which can learn itself a normal state of system and detect any deviations from such model.

The most demanding part of such mechanism is definitely anomaly detection algorithm which has to be very sensitive for any deviation, but also resistant for random noise coexisting in every data stream. Such algorithm should report anomaly with small rate of false positive alerts, because each alert triggers some human action or at least creates a urge for specialist on duty to check state of the infrastructure manually. Very important is also to choose algorithms with small memory and computation power requirements in order to create a possibility for simultaneous analysis of huge amount of time series data.

2 Available algorithms

Large variety of algorithms for performing anomaly detection over streaming time series data were presented in the literature. These algorithms can be assigned in one of 4 groups: cluster based methods, prediction models, density based methods and profile description methods.

Cluster based methods such as k-means [1] or k-medoids [2] works by discovering in historical data some clusters of probes and verifying whether current value of metric can be fitted in one of discovered set. For creation of cluster some parameters derived from probe has to be used. Example parameter can be day of week or hour of measurement. Similar approach is presented in works based on Support Vector Machine (SVM) [3] where a hyperplane separating different clusters of probes is created. There was also successful attempts to use Expectation-Maximization algorithm for profiling behavior of Hadoop clusters [4].

Profile description methods relies on learning by algorithm normal shape of analyzed time series and comparing collected data points with learned shape. Good example is Tiresias algorithm [5] used for monitoring performance of operating systems. Other solution can use specialized neural network for maintaining shape of checked series [6].

Another approach is to take into consideration density or distance between samples. In such methods all samples with number of neighbors below some threshold are taken as an anomalies. To use such method in streaming data, time sliding window for constant removal of old outdated samples, need to be introduced. The main work related to this subject makes use of ISB structure for storing neighborhood relation [7] or Yang algorithms which utilize some properties of sliding window [8].

Prediction approach is based on creating a model which will predict future value of data point and compare predicted value with real one. This comparison can be used to decide whether data point should be considered as anomaly or not. There are multiple different ways for creating a predicting model. Some examples of elementary ones are ARIMA model [9] and linear neural

network. There are also some works comparing use of multilayer perceptron with linear neural network, naïve predictor and nearest cluster [10]. Other solution can be Gibbs probes [11] or weighted maximal likelihood estimation [12].

3 Data streams source

All metrics series used in this work comes from one cluster of 70 mail servers processing around 26 million messages per day. In this paper as server metrics we will understand every time series containing some system specific variables such as local mail queue, number of concurrent connections or other simple system measurements like load average or size of free system memory. All data was collected by collectd system monitor and was stored in whisper datafiles. Samples was collected during five months period with granularity of one point per minute.

4 Data streams classification

Based on characteristic of time series data gathered in monitored system we distinguished following classes of time series:

4.1 Type I series – uniform series with rapid changes

In all Type I series seasonal effect does not exist or is negligibly small, additionally all existed anomaly have characteristic of rapid change with huge amplitude. Those kind of series describe systems witch workload is time independent or metrics whose should have always have the same value in correctly working system. As example we can use time series describing number of deferred e-mails on server inside mail infrastructure. Due to mail system specification such queue should be fixed at zero and each difference from this value can be considered as anomaly.

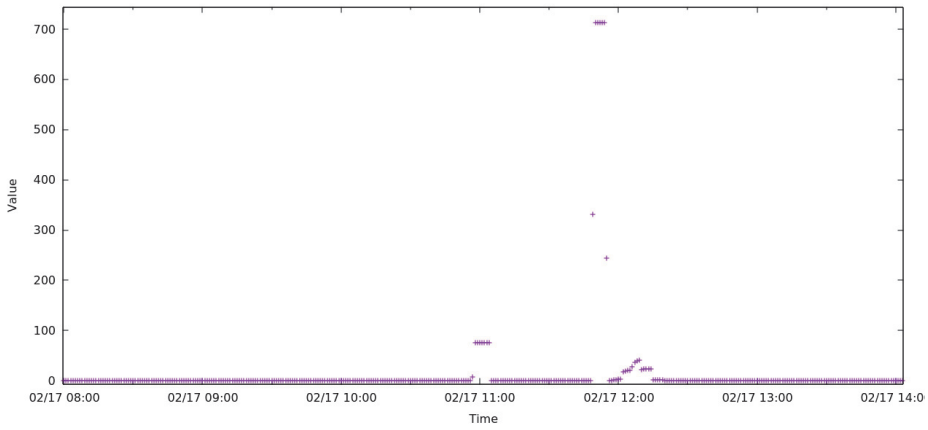

Figure 1. Type I series with stable fixed value

Other example of slightly more noisy series can be metric of active processed messages. Seasonality of such data is definitely small, therefore it can be considered as constant series with some random noise.

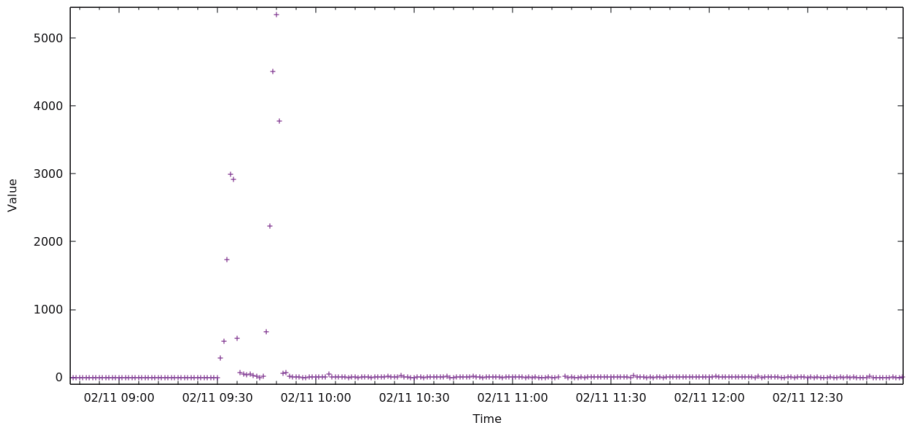

Figure 2. Type I series with very small seasonal effect

Those type of time series are characterized by high local consistency and complete time independency. Because of that there is no need to consider wider context of data. Each algorithm can just work on most recent examples and detect only local outliers.

4.2 Type II series – uniform series with slow changes

Very interesting example of time series is uniform and time independent series with very slow growing abnormal value. Such series can for example describe deferred message queue on output nodes caused generally by failure on the remote service providers side. Traffic to single provider is too small to cause rapid grow of deferred queue, but because of the long term characteristic of such failure, constant grow of queue on output nodes, can be possible.

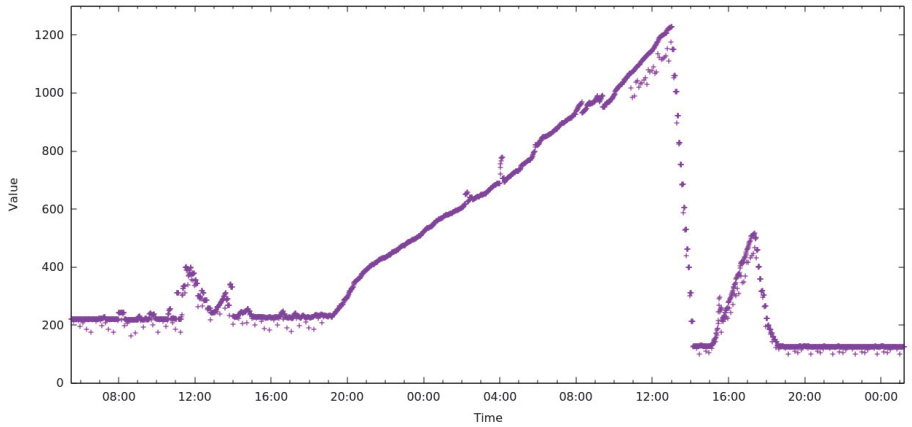

Figure 3. Type II series with slow growing anomaly

In production environment we cannot find any constant normal value for such metrics. This fact makes analysis much harder. Hopefully all series of this type have high level of local consistency.

4.3 Type III series – seasonal

Seasonal series are the most common ones, because they can describe typical standard working system with time dependent workload. Generally series of this kind have different average depending of the time of the day and day of the week. Let the example be number of messages originating from one of the input nodes.

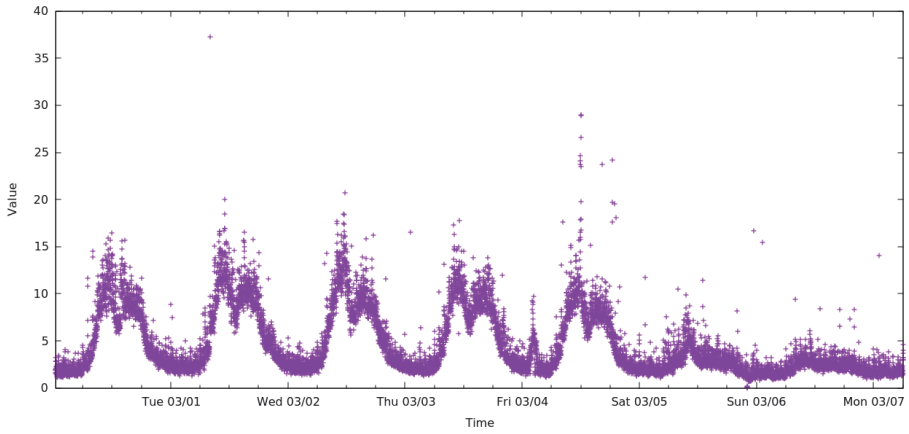

Figure 4. Type III seasonal series

Characteristic daily repeating pattern with two maximums around 11 and 16 o`clock can be seen. In addition to that there is possibility to observe general decrease of samples values during weekend or free days. It's worth to stressed out that the value of anomalous points at midnight 4th of the march is lower than normal workload during rush hours. The best algorithm should take time of measurement into consideration. Existence of local consistency in all series of this type is very promising. This means that all outliers points are anomalies. Thanks to that property there is possibility to use local consistency algorithm for anomaly detection.

4.4 Type IV series – seasonal not uniform series

Type IV of time series is a simple derivation of Type III series without local consistency. Loss of consistency can be caused by some repeating event which resulted in creation of peaks in series data. It is very important to note that such events are not anomalous and there is no need to generate alert for each occurrence. Good example of Type IV series is number of concurrent connections gathered from infrastructure input nodes.

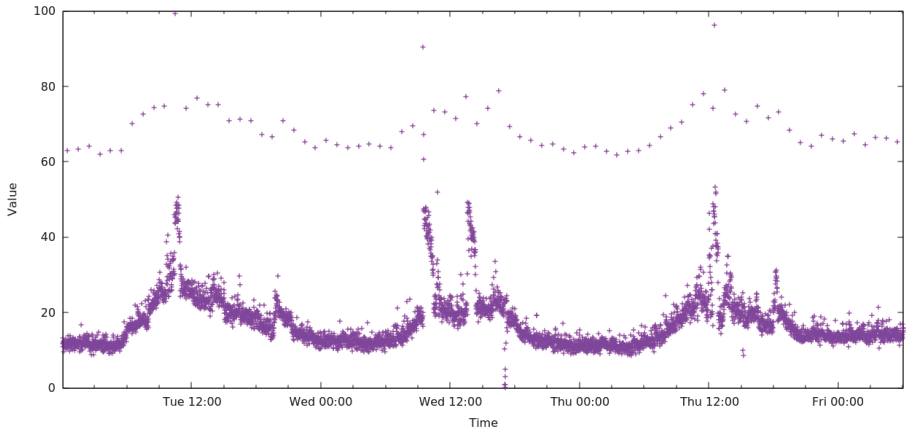

Figure 5. Type IV seasonal series with series of peaks

There is an easily distinguishable seasonal trend known from Type III series. Main difference is periodical rapid increase of value in every 15 minutes block, such points will be for sure marked as outlier but they are not an anomalies. Loss of local uniformity forces algorithm to consider wider context of data points in order to detect places where outlying point should occur. All of that makes this type of metric harder to analysis than Type III.

5 One sided median

The one sided median algorithm is based on quite simple prediction model, main assumption is that the median of series is almost always constant during single chosen time window. Described algorithm is based on work of S. Basu and M. Meckesheimer [13].

Consider time series Y containing ordered data points $y_1, y_2 \dots y_{n-1}, y_n$. For every time t we can define a k-length time window which will contains points

$$W_t^{(k)} = \{y_{t-k}, y_{t-k-1} \dots y_{t-1}\}, \tag{1}$$

We can also define series of differences between following points in time window $W_t^{(k)}$ let's denote

$$z_t = y_t - y_{t-1}, \tag{2}$$

than such series can be written as

$$Z_t^{(k)} = \{z_{t-k}, z_{t-k-1} \dots z_{t-1}\}, \tag{3}$$

Let's denote the median of $W_t^{(k)}$ as $\widetilde{W}$ and the median of $Z_t^{(k)}$ as $\tilde{Z}$. Finally in one sided median prediction value of data point at time t is estimated as

$$\tilde{y}_t^{(k)} = \widetilde{W} + \frac{k}{2}\tilde{Z}, \tag{4}$$

and the error of estimation is calculated using equation

$$\epsilon = \left| y_t^{(k)} - \tilde{y}_t^{(k)} \right|, \tag{5}$$

and compared to some threshold value τ. If the value of error ϵ is greater than threshold algorithm assume that the point is an anomaly and should be marked. Figure 6 shows concept of the current algorithm.

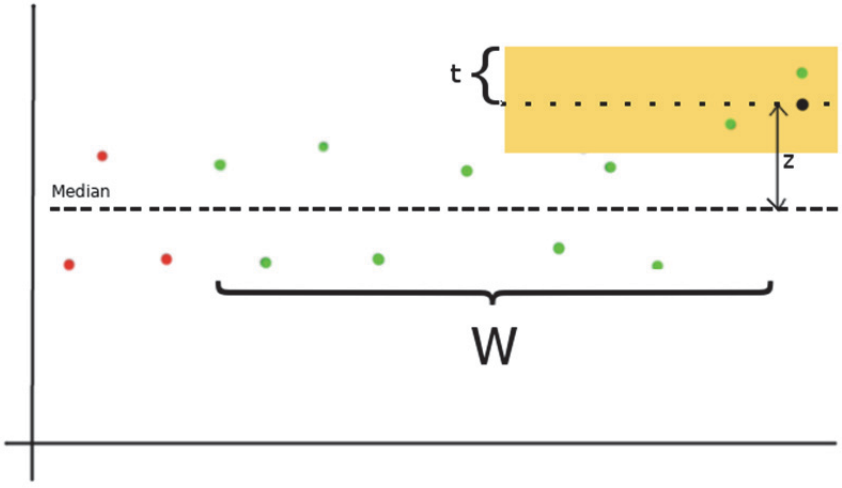

Figure 6. One sided median algorithm

There is also variant of algorithm which does not utilize differential module $Z_t^{(k)}$. In such approach value of data point can be estimated simply as median value:

$$\tilde{y}_t^{(k)} = \widetilde{W}. \tag{6}$$

Threshold value of τ has to be chosen very carefully and needs to match to the variability level of each series. Therefore we try to make it independent of series characteristic by using median absolute deviation parameter of time window. In our approach we assume that

$$\tau = MAD\left(W_t^{(k)}\right) * n, \tag{7}$$

where n is the value of allowed multiplication of MAD. Median absolute deviation parameters is robust measurement of variability of given sample and can be defined as shown in equation (4).

$$MAD\left(W_t^{(k)}\right) = median(y_t^{(k)} - median\left(W_t^{(k)}\right)), \tag{8}$$

6 Performance evaluation

6.1 Test procedure

To check efficiency and accuracy of proposed anomaly detection method we performed several experiments. Impact of presence of differential module, window length and threshold value on anomaly detection system performance were tested. Experiments were performed with all types of data series. We used number of detected anomalies as a metric of sensitivity. Each test result was manually reviewed and assessed against anomaly detection rate and number false positive alerts.

6.2 Presence of differential module

Evaluation of both models with (4) and without differential module (6) on the same sample series has shown that use of differential module results in huge increase in number of points detected as anomalies. Results of experiments are shown in Table 1. For the evaluation we used the threshold value as three times of the MAD parameter, and window length of 60 probes.

Table 1. Influence of differential module presence on number of points detected as anomalies

Series Type	Differential module present	Differential module absent
Type I	332	332
Type II	34582	34490
Type III	56256	9518
Type IV	68422	12043

However when we rely on manual assessment of results, introduction of differential module resulted in extraordinary increase of false positive rate. Example results are shown below on Figure 7 and Figure 8.

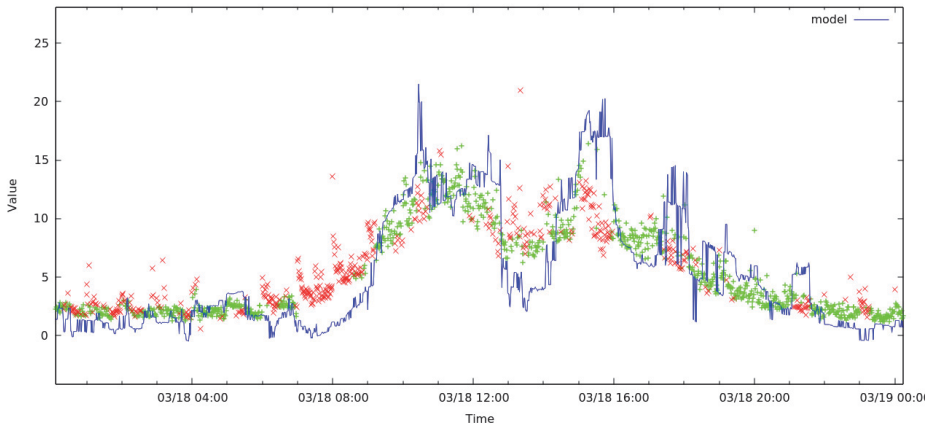

Figure 7. Estimation made by model with differential module enabled

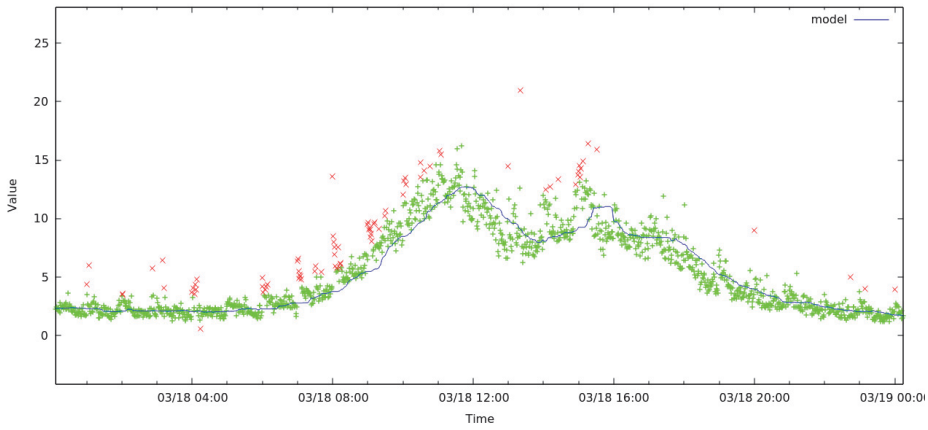

Figure 8. Estimation made by model with differential module disabled

Adding difference of following data points made median based estimation very variable, which lead to decrease of prediction accuracy. To show effect of this loss, test over a single generated series was performed. Evaluated series consists of stable value of zero with some random noise without any anomaly. Therefore estimated median should be fixed at the level of 0. As we may see in the Figure 9 estimated median has huge amplitude and vary a lot from the real median value.

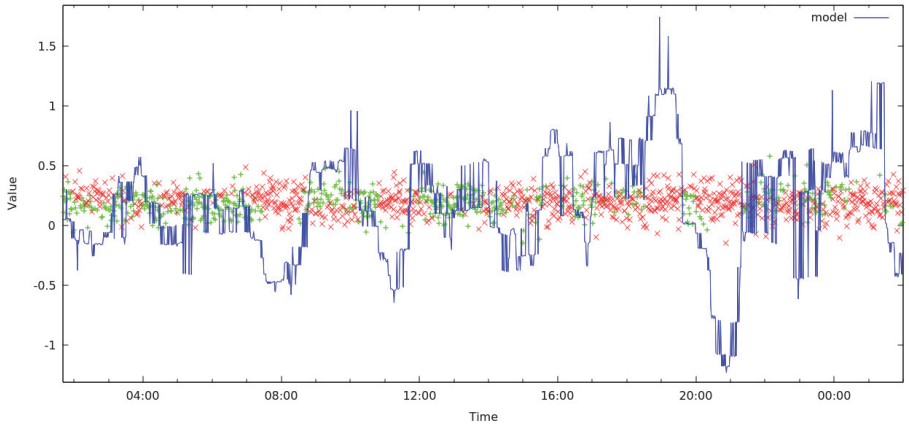

Figure 9. Effect of differential module on linear series with random noise

6.3 Sliding window length

Length of sliding window is key factor for gaining good prediction rate. Usage of small window can increase adaptivity of algorithm but will effect in increase of false positive rate. In the Table 2 number of points described as anomalies in function of window length and series type is presented. The threshold value was fixed at the level of three times of MAD value.

Table 2. Influence of time window length on number of points detected as anomalies

Series Type	**Window length**					
	15	30	45	60	75	**90**
Type I	385	328	364	332	327	329
Type II	34903	31584	33208	34490	35627	36233
Type III	10779	10140	9824	9518	9480	9687
Type IV	12214	11946	12040	12043	12210	12349
Series Type	**Window length**					
	105	120	135	150	165	**180**
Type I	333	312	312	312	312	312
Type II	36861	37036	37378	38077	38696	39515
Type III	9841	9997	10233	10825	11528	12349
Type IV	12566	12708	12905	13249	13590	13993
Series Type	**Window length**					
	210	240	270	300		
Type I	312	312	312	312		
Type II	40493	41284	41931	42647		
Type III	14078	15828	17289	18414		
Type IV	14986	16022	17193	18263		

In the Figures 10 and 11 algorithm reaction on anomaly is presented, chosen time window sizes are 15 and 60 respectively.

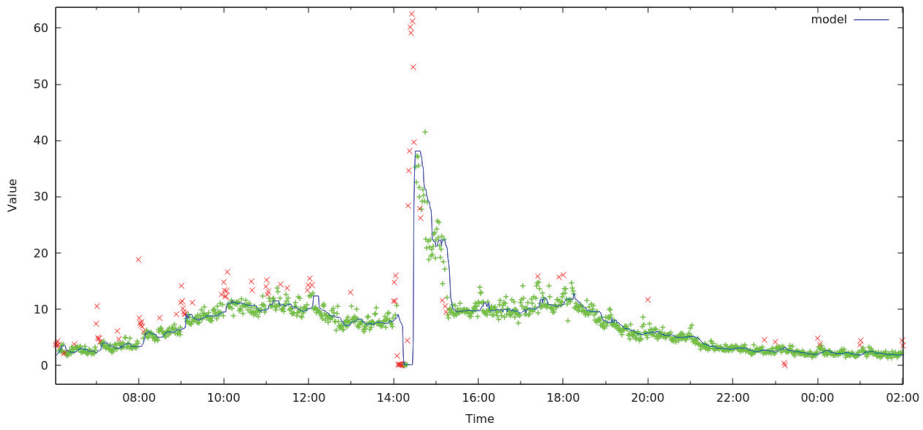

Figure 10. Anomaly detection with time window of 15 probes

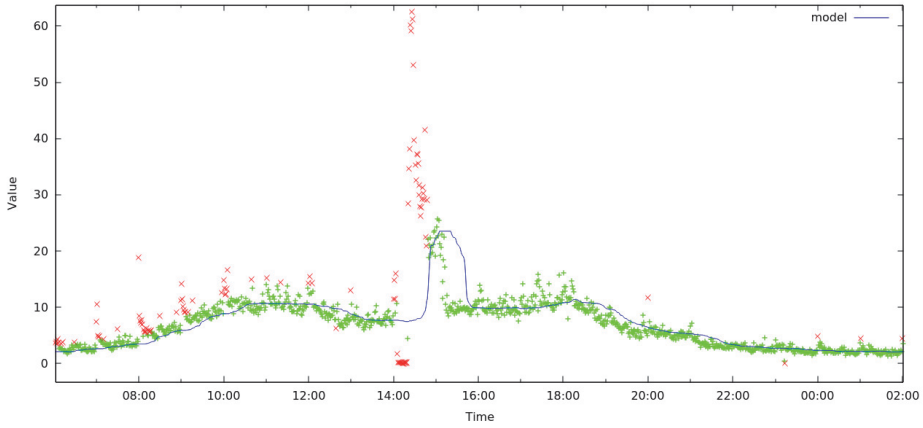

Figure 11. Anomaly detection with time window of 60 probes

For the small time window very fast adaptation of algorithm to anomalous level of metrics can be observed. Using larger size time window results in increasing of model stability.

However, time window can't be increased too much because at some point it will decrease a model accuracy, especially in time dependent time series such as type III and IV. On Figure 12 the same fragment of time series is presented, but this time window length is set to 360 points.

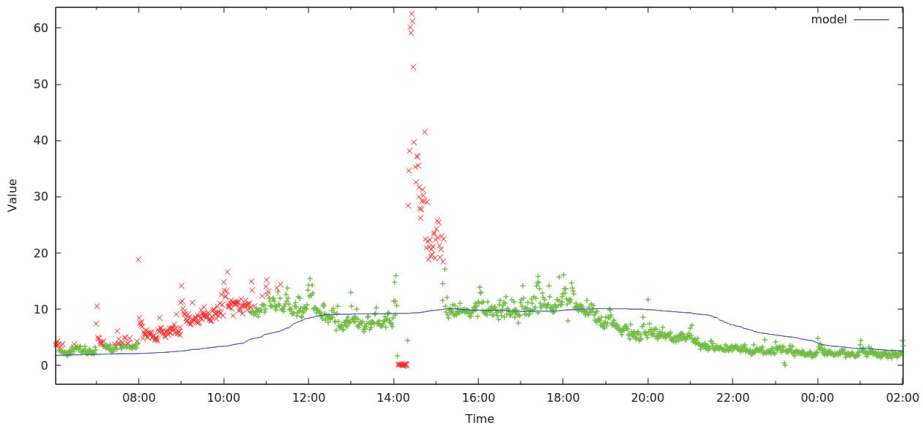

Figure 12. Too big time window

We can see decrease of prediction accuracy caused by not looking up to local trends of series. Estimated median value seems to be delayed in relation with real value. Loss of accuracy resulted in increase of false positive rate.

6.4 Threshold value

With the increase of threshold value used as multiply of MAD value, number of points described as anomalies start to decrease. Results of several test are shown in Table 3. Tests were performed with the window length of 60 probes.

Table 2. Influence of error threshold value on number of points detected as anomalies

Series Type	**Threshold value**					
	1	2	3	4	5	**6**
Type I	360	357	332	318	313	299
Type II	63697	43731	34490	29214	25668	23033
Type III	57995	19559	9518	5163	2950	1868
Type IV	57604	20701	12043	8851	7299	6342
Series Type	**Threshold value**					
	7			**8**		
Type I	290			289		
Type II	21048			19583		
Type III	1200			880		
Type IV	5664			5131		

On figures 13 and 14 two different threshold values are shown (3*MAD and 6*MAD).

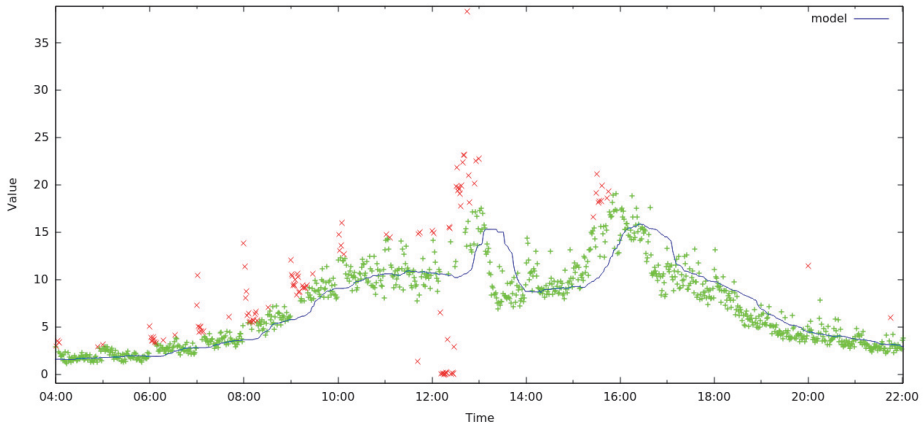

Figure 13. Threshold value set up to 3*MAD

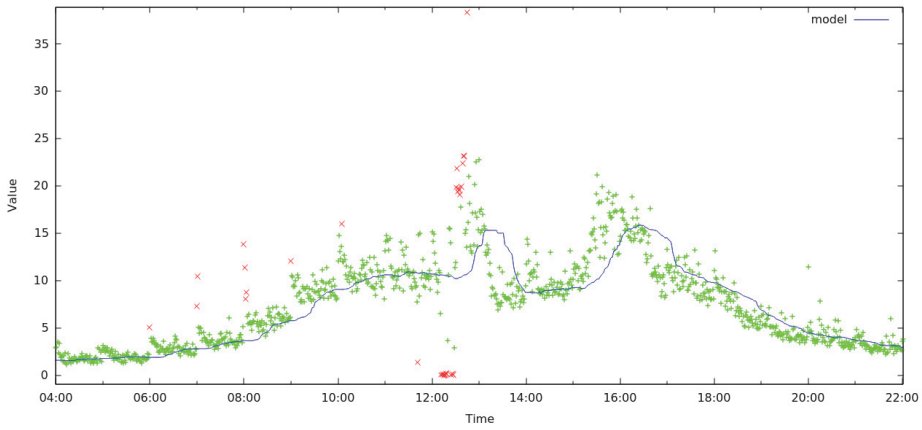

Figure 14. Threshold value set up to 6*MAD

We can see that increase of threshold results in decrease of sensitivity, however even use of 6*MAD level can properly detect critical anomalies.

6.5 Type of series

Usage of one sided median algorithm on all of analyzed Type I series showed that all existing anomalies can be easily discovered. Unfortunately one sided median algorithm can't be used to evaluate type II series. Two factor makes this kind of analysis really hard:

- Slow, gradual growth of anomaly is treated as a drift in data series values
- Presence of high level noise in the form of peaks determined by the source of this data series type.

High noise level is determined by the source of this series type. In some cases algorithm even mark good data points as anomalies which was caused by stabilized linear growing trend of anomaly.

It is really interesting that it is possible to discover anomalies in most common type III series with very good detection rate and small enough false positive rate by using well fitted sets of parameter. On the other hand use of one sided median algorithm over type IV series is limited due to marking every inconsistency as anomaly.

8 Conclusions

One sided median method of detection anomalies perform very well on metrics gathered from mail cluster. Based on the performed tests we can conclude, that this method is good choice for simple analysis of all kind uniform time series with rapid anomalies. We have shown that median absolute deviation (MAD) can be used as one factor of threshold value and allows model to be independent from series variability.

Results show also that use of differential module is not a good option for production series of metrics, as its introducing additional noise. It is possible that differential module can increase accuracy in series with constant visible trend such as constantly growing linear series.

Size of time window has to be correctly set to contain data points with low variance over median value. In case of Type III series appropriate value of this parameter is approximately 60 data points. We need to emphasis that in series of Type I time window length has no notable influence on the system performance. Number of points described as anomalies was similar between test results.

Threshold value has to be set as a compromise between sensitivity and false positive rate. For typical monitoring usage threshold value as three times the value of the parameter MAD should be enough to gain certain level of accuracy with good level of detection. For detecting only critical anomalies with very small level of false positive multiplication of 6 should be used.

This work shows that automatic anomaly detection over server metrics is possible even by using simple algorithms. One sided median is good enough to be used in production environment with success making administrators job simpler and systems more resistant to unpredicted failures.

References

1. Nairac A., Townsend N., Carr R., King S., Cowley P., Tarassenko L., 1999, *A System for the Analysis of Jet Engine Vibration Data* Integrated Computer Aided Engineering, Boulder, 6, 1, pp. 53–66.
2. Budalakoti S., Srivastava A., Akella R., Turkov E., 2006, *Anomaly Detection in Large Sets of High-dimensional Symbol Sequences,* Tech. Rep. NASA TM-2006-214553, NASA Ames Research Center
3. Eskin E., Arnold A., Prerau M., Portnoy L., Stolfo S., 2002, *A Geometric Framethe Analysis of Jet Engine Vibration Data* Integrated Computer Aided Engineering, Boulder, 6, 1, pp. 53–66.

4. Pan X., Tan J., Kavulya S., Gandhi R., Narasimhan P., 2010, *Ganesha: Black-Box Diagnosis of MapReduce Systems*, SIG-METRICS Performance Evaluation Review, 37, 3, pp. 8–13
5. Williams A.W., Pertet S. M., Narasimhan P., 2007, *Tiresias: Black-box Failure Prediction in Distributed Systems*, 21st Intl. Parallel and Distributed Processing Symposium (IPDPS), pp. 1-8
6. Silvestri G., Verona F., Innocenti M., Napolitano M., 1994*, Fault Detection using Neural Networks*, IEEE Intl. Conf. on Neural Networks, pp. 3796–3799
7. Angiulli F., Fassetti F., 2007, *Detecting Distance-based Outliers in Streams of Data*, 16th ACM Conf. on Information and Knowledge Management (CIKM), pp 811-820
8. Yang D., Rundensteiner E. A., Ward M.O., 2009, *Neighbor based Pattern Detection for Windows over Streaming Data*, 12th Intl. Conf. On Extending Database Technology: Advances in Database Technology (EDBT), pp 529-540
9. Tsay R.S., Pena D., Pankratz A.E., 2000, *Outliers in Multivariate Time Series*, Biometrika, 87, 4, pp. 789-804
10. Hill D. J., Minsker B. S., 2010, *Anomaly Detection in Streaming Environmental Sensor Data: A Data-driven Modeling Approach*, Environmental Modelling and Software, 25, 9, pp. 1014–1022
11. Justel A., Pena D., Tsay R.S, 2001, *Detection of Outlier Patches in Autoregressive Time Series*, Statistica Sinica, 11, 3, pp. 651-674
12. Luceno A., 1998, *Detecting Possibly Non-Consecutive Outliers in Industrial Time Seri*es, Journal of the Royal Statistical Society. Series B (Statistical Methodology), 60, 2, pp. 259-310
13. Basu S., Meckesheimer M., 2007, *Automatic Outlier Detection for Time Series: An Application to Sensor Data*, Knowledge and Information Systems – Special Issue on Mining Low-Quality Data, 11, 2, pp. 137-154