Souvik Chowdhury and Shibakali Gupta

3 Anomaly detection in cloud big database metric

Abstract: After cloudification of various big data sources or big databases, there is a need to monitor health and security metrics of these big databases. Now there is already a monitoring setup provided by various monitoring suites. The monitoring software collects various metrics with the help of custom codes, plugins, and so on. Here we are proposing a novel approach of modifying the normal metric thresholding to anomaly detection. Every system administrator has a common problem to deal with some intelligent alarm methods, which can produce predictive warnings, that is, the system can detect any anomalies or problem before it occurs.

Here we are proposing an approach, which is basically a modification on standard monitoring where it will detect any anomalies by analyzing previous metric data and indicate any problem. We are planning to harness the power exponential moving average and exponential moving standard deviation method to implement the solution.

Keywords: cloud, big data, statistics, monitoring, moving average, anomaly

3.1 Introduction

Anomaly occurs when a system deviates from its normal running operation. Normally in any anomaly situations, except system overload, system malfunction, brutal network attack, defect, or error during a program arises [1]. Anomaly is also being termed as malware, intrusion, or outlier sometimes [2]. Any system, for example, monitoring software generates data continuously, that is, in time series manner. The volume of data is huge and due to its time series data, this is constantly changing; hence, processing of these data and tracking any kind of problem are very difficult by using a traditional threshold-based monitoring approach.

Let us start with an example. Let's consider a big database hosted in a server, which is running various complex queries, data loading activities, and others. Now we put a threshold of 70% for CPU utilization in that server. This may happen during weekends, month ends, quarter ends as per business needs. A havoc of data loading activity happens and due to that we saw a spike in CPU utilization till 99%. Does that

Souvik Chowdhury, Oracle India, Bangalore, India.

Shibakali Gupta, Department of Computer Science, University Institute of Technology, The University of Burdwan, Burdwan, West Bengal, India.

mean there is a problem in CPU that time? No, the CPU spike happened due to extra load. Now all the traditional monitoring software collects metrics, projects them in graph, and triggers alarm in case if it breaches the predefined threshold. The anomalies in various health and security metric could be due to bug in latest patch/upgrade happened or random hardware failures. Simple metrics can support majority cases. But this is not a good approach since it does not indicate any problem.

This is becoming tougher when the data is complex in nature. Because complex data has various features to monitor. Anomaly detection for such a system is a tough job [3]. That is why it is so important to do anomaly detection [4]. In real-time anomaly detection setup, it constantly monitors the time series data and automatically detects any anomaly and executes corrective actions based on the situation. Hence, this kind of proactive prevention not only saves from a drastic system crash but also saves lot of time and money.

Infrastructure as service is a private cloud-based infrastructure service available at a data center. End users can deploy their workloads using this secure on-demand system. Resource management for information centers needs examining user necessities and, it's on the market resources to satisfy its client demands [5]. The system should have dynamic scalability in terms of infrastructure resource because of varying user demand to stay the pace of all these, an information center should have a period observance system.

The system tracks the resource performance info that comes ceaselessly and detects the abnormal behavior. It then reschedules its resources to remain the system in balance. As an example, an information center allocates a tough and quick amount of CPU and memory for sort of users [6]. After a certain time, if the users run extra computationally expensive programs, the electronic equipment and memory usage are high, which they have extra resources to try and do their job [7]. The amount the system monitors the data and detects these abnormal resource statistics then allocates extra resources dynamically.

Stream process could also be a time-sensitive operation. It needs info preprocessing before doing any analysis. It converts continuous high-volume, high-rate, structured, and unstructured Brobdingnagian info into amount price. Moreover, these Brobdingnagian info volumes must be compelled to be processed with ease, and delivered with low latency, even once info rate is high. To create such a system, Brobdingnagian's ascendible amount operational capability is needed. Recently, we have got seen several massive info frameworks (e.g., Hadoop, Map Reduce, HBase, Mahout, and Google Bigtable) that address the quantifiability issue [8]. However, they surpass batch-based process. The amount the system demands put together the streaming process. Apache Storm could also be a distributed framework that gives streaming integration with time-based in-memory analytics from the live machine info as they're accessible in stream. It generates low latency, amount results. It has instant productivity and no hidden obstacles. When put together contains an easy cluster setup procedure [9].

In statistics, smoothing of information set is to make necessary approximation which helps to capture important patterns within the data and removal of noise or alternative fine-scale structures/rapid phenomena. In smoothing is basically indication of change in square measure of info points i.e. individual points (mostly due to noise) square measure reduced and points that square measure not up to the adjacent points' square measure multiplied resulting in a Sander signal [10]. Smoothing is also utilized in two vital ways, which aids in information analysis:

1. The setup must be able to get more information out of the data, provided the smoothing assumption is reasonable enough.
 2. This should provide both robust and flexible analyses. A variety of algorithms are applied in smoothing technique.
- Smoothing technique can be differentiated from the partially and related overlapping concept of curve fitting in the below ways:
 - Many times, an explicit function is being used in curve fitting to produce result and if a function form is ever being used, then the immediate results from smoothing are basically the smoothed values and which have no further user.
 - Smoothing mainly generalizes the idea of relative slow movement of values, where a small attention is being paid to the close matching of data, whereas curve fitting focuses on matching to the nearest possible data values.
 - A tuning parameter is often available with the smoothing method to control the behavior of smoothing. To get the “best fit,” the data curve fitting can modify any number of parameters.

Exponential smoothing may be a rule-of-thumb technique for smoothing statistic knowledge victimization the exponential window operates. Whereas within the easy moving average, the historical observations assigned equal weights, exponential functions would not allocate exponentially decreasing weights over time. Its associate in nursing simply learned and applied the procedure for creating some determination that supported previous assumptions by the user, like seasonality. Exponential smoothing is usually used for the analysis of time series knowledge [11].

3.2 Time series modeling

Before understanding the smoothing technique, we will have a quick look into time series data and time series modeling to understand various smoothing techniques well.

A time series could be an assortment of observations created consecutively over an amount. In different words, the information on any characteristic collected with relation to time over a span of your time is named as statistic. Normally, we tend to assume that observations square measure offered at equal intervals of your time,

for example, on associate hourly, daily, quarterly, or yearly basis. The ways of analyzing statistic represent a crucial space of study in statistics [12]. However, before we tend to discuss the statistical analysis, we would prefer to show the plots of its slow series from completely different fields. Within the next three sections, we glance at the plots of three time series, namely, statistic with trend impact, statistic with seasonal impact, and statistic with cyclic impact. These plots of square measure are known as time plots.

3.2.1 Trend effect

A trend could be a future swish variation (increase or decrease) within the statistic. Once values in a very statistic area unit aforethought in a very graph and, on a median, these values show associate degree increasing or decreasing trend over an extended amount of your time, the statistic is named as the statistic with trend impact. We should always note that incomparable series don't show associate degree increasing or decreasing trend [13]. In some cases, the values of the statistic fluctuate around a continuing reading and don't show any trend about time. We should always additionally remember that an increase or decrease might not essentially be within the same direction throughout the given amount. Statistic could show associate degree upward trend, a downward trend, or haven't any trend in any respect, which allow us to make a case for all three cases with the assistance of examples [14].

3.2.1.1 Time series with upward trend

If a time series exhibits upward trend, that is, the metric values that get increased as time progresses is termed as time series with upward trend [15]. Figure 3.1 shows an upward trend, which depicts the profit of a company plotted for the period 1981–2012.

Figure 3.1: Profit of a company from 1981 to 2012.

3.2.1.2 Time series with downward trend

Metric values of a time series when plotted in a graph with respect to time showing a downward trend are termed as time series with downward trend. Figure 3.2 shows a downward trend in the time plot, which describes the values of mortality rate for a developing country from 1991 to 2009.

Figure 3.2: Mortality rates of a developing country from 1991 to 2009.

3.2.1.3 Time series with no trend

Metric values in a time series when plotted in a graph with respect to time if it does not show any trend or to be precise random behavior, thus not showing upward or downward trend then the time series is described as time series with no trend. Figure 3.3 shows the time plot of the commodity production in tons of a factory from 1988 to 2012 where the time series shows no trend.

Figure 3.3: Commodity production in tons from 1988 to 2012.

3.2.2 Time series with seasonal effect

Metric values in a time series when plotted at a graph with respect to time if it reflects variation seasonally with respect to any period such as half yearly, quarterly, monthly, or a yearly, then the time series is termed as time series with seasonal effect. Figure 3.4 describes time series with seasonal effect, which plots the data of weekly sales of air.

Figure 3.4: Weekly sales of air coolers.

3.2.3 Time series with cyclic effect

Metric values in a time series when plotted in a graph with respect of time if shows a cyclic trend then the time series is termed as time series with cyclic effect. Figure 3.5 shows an example of time series with cyclic effect, which gives employees attrition rate in software industries for last 25 years.

Figure 3.5: Employees attrition rate in software industries for the last 25 years.

3.2.4 Time series components

The variations within the time series statistic values square measure of various varieties arise because of a spread of things. These differing kinds of variations within the values of the information in a very statistic are referred to as elements of the statistic [16].

Various components of time series that involved in variations are (i) trend, (ii) seasonal, (iii) cyclic, and (iv) remaining variations attributed to random fluctuations.

3.2.4.1 Trend component

Usually statistic information shows random variations; however, over a protracted amount of your time, there is also a steady shift within the mean level to the next or a lower level. This steady shift within the level of your time series is thought because of the trend. In other words, the final tendency of values of the information to extend or decrease during a protracted amount of your time is named as the trend.

3.2.4.2 Seasonal component

In a time series, variations that occur because of regular or natural forces/factors and operate in an exceedingly regular and periodic manner over a span of bus or adequate 1-year area unit termed as differences due to the season [17]. Though we tend to typically think about seasonal movement in statistic as occurring over one year, it may also represent any frequent continuance pattern that's but one year in period. As an example, daily traffic volume knowledge shows seasonal behavior at intervals a similar day, with peak level occurring throughout rush hours, moderate flow throughout the remainder of the day, and lightweight ensue hour to early morning. Thus, in an exceedingly statistic, differences due to the season could exist if knowledge area unit recorded on a yearly, monthly, daily, or hourly basis.

3.2.4.3 Cyclic component

Sometimes time series show variation for a fixed period due to certain physical reasons and this is not part of seasonal effects. For example, sometimes economic data are prone to be affected by business cycles with a period varying from few to several years (see Figure 3.5). A period of moderate inflation followed by high inflation is a primary reason for these cyclic variations. Hence, the existence of these business cycles causes some intermittent bias about various cyclic, trend, and seasonal effects [18]. To overcome this problem, we will consider a cyclic pattern or trend in time series only when the duration is more than a year.

3.2.4.4 Irregular component

The long variations, that is, the trend part, and short-run variations, that is, the seasonal and cyclic parts, are referred to as regular variations. Except for these regular variations, random or irregular variations, that don't seem to be accounted for by trend, seasonal, or cyclic parts, exist in virtually incomparable series [19].

3.3 Smoothing of time series

Smoothing of time series is basically filtering out the effect of irregular fluctuations from any time series so that the trend and seasonal effect could be estimated easily and error freely for any time series [20].

Smoothing can be done using three methods: simple moving average method (equal weight), the weighted moving average (unequal) method, and the exponential moving average (EMA) method.

3.3.1 Simple moving average method (equal weight)

Here we calculate the simple moving average of time series data over n periods of time called n -period moving averages. Below is the step to calculate simple moving average.

1. The average of the first m values of the time series is calculated.
2. First value has been discarded, and the average of the next n values has been measured again.
3. Steps 1 and 2 are repeated till all data are used.

The above steps generate a new time series of n -period moving averages [21].

3.3.2 Weighted moving average method (unequal)

The simple moving average methodology delineated in Section 4.1 isn't typically well compatible for activity trend though we are able to take away seasonal variation victimization. It should additionally not lie about to the most recent values. Therefore, the weighted (unequal) moving average methodology is employed. During this methodology, rather than giving equal weights to any or all values, unequal weights are given in such a way that each one of the weights is positive, and there add is capable one. If w_i depicts i th observation weight, the weighted moving average value y_t can be given as follows:

$$Y_t = \sum_{i=-q}^q W X_i t_i, \quad w_i \geq 0, \quad \sum_{i=-q}^q W_i = 1 \quad (3.1)$$

where x_t is the actual series.

Simple moving average is nothing but a weighted moving average for

$$m = 2q + 1 \text{ and } w_i = \frac{1}{(2q + 1)}$$

3.3.3 Exponential smoothing

Exponential smoothing could be a golden rule technique for smoothing statistic information victimization the exponential window operates. Whereas within the easy moving average the historical observations square measure weighted equally, exponential functions square measure would not assign exponentially decreasing weights over time. Its associate in nursing simply learned and applied the procedure for creating some determination that supported previous assumptions by the user, like seasonality. Exponential smoothing is usually used for the analysis of time series information.

3.3.3.1 Exponential moving average

Below are the steps to calculate exponential moving average (EMA) method.

1. If $y_1, y_2, \dots, y_t$ is a time series, then the first element of EMA is equal to first element of time series, that is, $y_{1'} = y_1$.
2. The second smoothed value based on EMA, $y_{2'} = w*y_2 + (1 - w)*y_{1'}$, where $y_{1'}$ is the first element of smoothed series and y_2 is the second element of original time series and w is the smoothing factor and it ranges from $0 < w < 1$.

In this way, t th smoothed value can be calculated by $y_{t'} = w*y_t + (1 - w)*y_{(t-1)'}$.

3.3.3.2 Exponential moving standard deviate

We have introduced the modified concept of exponential moving standard deviation (EMSD), which is basically refined as smoothing technique over EMA method.

Steps to get EMSD as follows:

1. First element of EMSD is calculated as follows: $y_{1'} = \text{sqrt}((1 - w) * (y_1 - y_{1'})^2)$ where $y_{1'}$ is the first element of EMA.
2. Second element of EMSD is calculated by $y_{2'} = \text{sqrt}(w*y_{1'} + ((1 - w) * (y_2 - y_{2'})^2))$.

3.4 Working methods

Let's check out the following graph for normal CPU utilization in a server based on hourly aggregation. The graph suggests normal graphical representation of CPU utilization metric being collected with normal thresholding set.

In Figure 3.6, the fixed threshold mechanism detects a problem well; however, let us think an example where the server is very least loaded and normally during weekends runs some batch jobs, which adds load to the server. In these cases, fixed threshold mechanism don't hold good because it will detect the extra load as a problem.

Figure 3.6: Fixed threshold-based mechanism with random spikes.

From Figure 3.7, it is evident that during weekends there is a high growth of CPU, which could be due to some batch job being run on weekends. Hence, this fixed threshold mechanism does not hold good in these cases.

Figure 3.7: Fixed threshold-based mechanism with steady spikes.

Hence, anomaly detection with its adaptive capabilities also can be called as adaptive thresholding, which is more intelligent than the traditional fixed threshold approach. We will see them in Figure 3.7.

In the following example (Figure 3.8), we will see a mix of both unusual growth or spikes (which turn out to be a problem) and steady growth (which could be due to some job) and hence not a problem is successfully being distinguished by this new anomaly detection-based mechanism.

Figure 3.8: Anomaly-based threshold mechanism for unusual spikes and steady spikes.

We will see some more graphs, for example, how the anomaly detection-based mechanism works for memory utilization, where it shows a cyclic pattern as demonstrated early.

From Figure 3.9, we could see there has been a cyclic pattern and anomaly detection-based method that successfully smoothed the raw metric data.

Figure 3.9: Anomaly-based threshold mechanism for cyclic pattern (memory utilization in big database servers).

Now we will see how this anomaly detection-based mechanism works good for cyclic spikes as shown in Figure 3.10.

Figure 3.10: Distribution of normal metric data and exponential moving average value.

In Figure 3.10, the blue line depicts normal metric data and red one gives EMA in short EMA values. From the figure we could see for spikes that it is trying to smooth the data which seems to work like normal average. This function works better if the spike stays a bit, which could be due to some load, thus distinguishing a problem with normal loaded situation.

Now we will see the difference between the normal standard deviation and EMSD.

From Figure 3.11 it is evident that EMSD is even a far more refined version of standard deviation, where it is trying to smooth the data and track any real problem. We could see only the extreme variations the standard deviation value also spiked up; however, the case is not the same for EMSD which contains historical knowledge and tries to smooth the data and detect any anomalies.

Figure 3.11: Standard deviation (SD) versus exponential moving standard deviation (EMSD).

Figure 3.12 depicts the alert mechanism being set with the help of EMA and EMSD. The orange dots on top are the cases when the alerts got triggered. Here we could see in the highlighted portion that the spikes seem to be regular and this mechanism did not trigger alert for this case.

Figure 3.12: Alert mechanism setup using exponential moving average and exponential moving standard deviation.

3.5 Methodology section

We can use the below sample query to get the EMA value and this is specific to Oracle Big database SQL.

```

with t1 as (
select <column names>, row_number() over (partition by <partition column> order by
<ordered column>) r_n,
--2 / (1 + row_number() over (partition by <partition column> order by <ordered col-
umn>)) k_i
0.5 k_i
from <metric table>
), t2 as (
select <column names>, (case when r_n = 1 then 1 else k_i end * c_i) a_i, case when
r_n = 1 then 1 else (1 - k_i) end b_i from t1
), t3 as (
SELECT <column names>,
ai,
xmlquery(REPLACE(wm_concat(bi) over(PARTITION BY <partition column> ORDER BY <ordered
column> rows BETWEEN unbounded preceding AND CURRENT ROW), ',', '*') RETURNING con-
tent).getnumberval() mi
FROM t2
), t4 as (
select <column names>, m_i, (a_i / m_i) x_i from t3
)
SELECT <column names>,
round(m_i * SUM(x_i) over(PARTITION BY <partition column> ORDER BY <ordered column>
rows BETWEEN unbounded preceding AND CURRENT ROW), 3) ema
FROM t4;

```

We can use the following sample to query EMA using simplified method. Here we have two options to select the weight factor w mentioned in Section 3.3.3.1.

Special case $w = 0.5$. Here the weight factor has been fixed to 0.5, that is, giving equal weightage to current metric data and past metric data, thus leaving a trail of historical metric data in calculation:

```

with t1 as (
select <column names>, row_number() over (partition by <partition column> order by
<ordered column>) r_n, amount * power(2, nvl(nullif(row_number() over (partition by
<partition column> order by <ordered column>) - 1, 0), 1)) c_i
from <metric table>
)
select <column names>, round(sum(c_i) over (partition by <partition column> order by
<ordered column> rows between unbounded preceding and current row) / power(2, r_n),
3) ema

```

```
from t1;
```

Special case $w = 2/(1 + i)$. Here it is dynamic; hence, for various levels it would have different behavior:

```
with t1 as (
select <column names>, row_number() over (partition by <partition column> order by
<ordered column>) r_n, amount * row_number() over (partition by <partition column>
order by <ordered column>) c_i
from <metric table>
)
select <column names>, round(sum(c_i) over (partition by <partition column> order by
<ordered column> rows between unbounded preceding and current row) * 2 / (r_n * (r_n
+ 1)), 3) ema
from t1;
```

Now we will discuss about various jobs and procedure setup to achieve this task. The challenge was to achieve good performance because the real-time calculation of EMA and EMSD is a difficult one and with enormous number of targets and its metric for a cloud big database, cloud datacenter is even getting worse. We concentrated on SQL tuning part and tried to make the query as fast as possible. We included parallelism in query. But still that was not good enough. We took another option to parallelize the job itself. Hence, the current model has been depicted as follows:

Figure 3.13 depicts the job flow. Here, a master job is responsible for all job allocations. It was being tested to use parallel 20 jobs. The main job ANOMALY_COUNT_REFRESH analyzes currently running jobs and forks a new session and attaches the job to the session (Figure 3.14).

Figure 3.13: Job flow.

Figure 3.14: Job flow(1).

```

BEGIN
<Scheduler job create module> (
  job_name      => 'ANOMALY_COUNT_REFRESH',
  job_type      => <Object Type>,
  job_action    => 'ANOMALY_COUNT',
  start_date    => <Job start date>,
  repeat_interval => 'FREQ=<minutely, hourly etc>;INTERVAL=<interval duration 5
mins, 1 hour etc>;',
  end_date      => NULL,
  enabled       => TRUE,
  comments      => <Job Description>;
END;
/

```

Once this job is assigned to a session, the main job comes in which it collects all necessary data and stores the output (Figure 3.15).

Figure 3.15: Job flow(2).

We have used the following sample commands to create pool of jobs. At present, only two jobs have been shown.

```
BEGIN
  <Scheduler job create module> (
    job_name      => 'ANOMALY_COUNT_JOB1',
    job_type      => => <Object Type>,
    job_action    => 'ANOMALY_COUNT1',
    end_date      => NULL,
    enabled       => FALSE,
    number_of_arguments => 2);
END;
/
BEGIN
  <Scheduler job create module> (
    job_name      => 'ANOMALY_COUNT_JOB2',
    job_type      => => <Object Type>,
    job_action    => 'ANOMALY_COUNT1',
    end_date      => NULL,
    enabled       => FALSE,
    number_of_arguments => 2);
END;
/
```

3.6 Future work

This chapter tested on CPU utilization and memory utilization of big database servers. This approach can be extended to other types of hosts, servers, VM, and so on. This method of new monitoring mechanism should be tested for network devices as well. For network device monitoring, there lies special challenge because network devices have lot of interlinked metrics to detect a problem, for example, for a network device the reduction in traffic does not necessarily mean reduction in connected users. It could also be due to error, discards, packet loss, and so on. The method can be extended to datacenter monitoring for complete stack, say from security appliances to bare metal server. This kind of anomaly detection-based mechanism can be helpful for capacity planning as well. Capacity planning for network devices of a complete cloud datacenter footprint is a tedious job. Because it consists of lot of network devices, and a cisco switch or juniper device can have 1,000 or 2,000 network interfaces attached to it. Hence, doing capacity planning or pinpointing problem for a network interface is a difficult task.

References

- [1] Zhu, Qingsheng., & Liu, Renyu. "A Network Anomaly Detection Algorithm based on Natural Neighborhood Graph", "International Joint Conference on Neural Networks 2018".
- [2] Wang, Lin., Xue, Bai., Wang, Yulei., Li, Hsiao-Chi., Lee, Li-Chien., Song, Meiping., Yu, Chunyan., & Li, Sen. et al. "Iterative anomaly detection", "IEEE International Geoscience and Remote Sensing Symposium 2017".
- [3] Edisanter Lo. "Hyperspectral anomaly detection based on a generalization of the maximized subspace model," "2013 5th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing".
- [4] Yao, Danfeng., Shu, Xiaokui., Cheng, Long., Stolfo, Salvatore J., & Bertino, Elisa., Ravi et al. "Anomaly Detection as a Service: Challenges, Advances, and Opportunities, Synthesis Lectures on Information Security, Privacy, and Trust, Electronic".
- [5] Zhao, Rui., Du, Bo., & Zhang, Liangpei. "GSEAD: Graphical score estimation for hyperspectral anomaly detection, 2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote".
- [6] Chen, Mingqi., Jiang, Ting., & Zou, Weixia. "Differential physical layer secret key generation based on weighted exponential moving average, 2015 9th International Conference on Signal Processing and Communication".
- [7] Alexander, Belyaev., Ivan, Tutov., & Denis, Butuzov. "Analysis of noisy signal restoration quality with exponential moving average filter, International Siberian Conference on Control and Communications, 2016".
- [8] Md. Emdadul, Haque., Md. Nasmus, Sakib Khan., & Md. Rafiqul, Islam Sheikh. "Smoothing control of wind farm output fluctuations by proposed Low Pass Filter, and Moving Averages, International Conference on Electrical & Electronic Engineering 2015".
- [9] Lutfi Al-Sharif, Ahmad Hammoudeh. "Estimating the elevator traffic system arrival rate using exponentially weighted moving average(EWMA), IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies 2017".
- [10] Tajiri, Hiroki.. "Input filtering of MPPT control by exponential moving average in photovoltaic system, Teruhisa Kumano IEEE International Conference on Power and Energy 2012".
- [11] Xie, Y. J., Xie, M., & Goh, T. N. "A MEWMA chart for a bivariate exponential distribution, IEEE International Conference on Industrial Engineering and Engineering Management 2009".
- [12] Hansun, Seng., & Kristanda, Marcel Bonar. "Performance analysis of conventional moving average methods in forex forecasting, International Conference on Smart Cities, Automation & Intelligent Computing Systems 2017".
- [13] Guo Feng Liu, Chen-Yu., Bin, Zhou., & Su-Qin, Zhang. "Spares Consumption Combination Forecasting Based on Genetic Algorithm and Exponential Smoothing Method, Fifth International Symposium on Computational Intelligence and Design 2012".
- [14] Akpınar, Mustafa., & Yumusak, Nejat. IEEE International Conference on Environment and Electrical Engineering and IEEE Industrial and Commercial Power Systems Europe 2017".
- [15] Wang, Shuo., & Li, Ning. "MYCAT Shard Key Selection Strategy Based on Exponential Smoothing, 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference 2018".
- [16] Morimoto, J., Kasamatsu, H., Higuchi, A., Yoshida, T., & Tabuchi, T. "Setting method of smoothing constant in exponential smoothing, SICE 2004".
- [17] Yanwei, Du.. "Research on Requirement Forecasting of Raw Materials for Boiler Manufacturing Enterprise Based on Exponential Smoothing Method, Second International Conference on Computer Modeling and Simulation, Electronic 2010".

- [18] Setiawan, Wawan., Juniati, Enjun., & Farida, Ida.. "The use of Triple Exponential Smoothing Method (Winter) in forecasting passenger of PT Kereta Api Indonesia with optimization alpha, beta, and gamma parameters, 2nd International Conference on Science in Information Technology 2016".
- [19] Li, Jing., Xu, Danning., Zhang, Jinfeng., Xiao, Jianhua., & Wang, Hongbin.. "The comparison of ARMA, exponential smoothing and seasonal index model for predicting incidence of Newcastle disease, World Automation Congress, Electronic 2010".
- [20] Wi, Young-Min., Kim, Ji-Hui., Joo, Sung-Kwan., Park, Jong-Bae., & Oh, Jae-Chul. "Customer baseline load (CBL) calculation using exponential smoothing model with weather adjustment, Transmission & Distribution Conference & Exposition 2009".
- [21] Chu, Chao-Ting., Chiang, Huann-Keng., Chang, Chao-Hsi., Hong-Wei, Li., & Chang, Tsung-Jui.. "A exponential smoothing gray prediction fall detection signal analysis in wearable device, 6th International Symposium on Next Generation Electronics 2017".