Research of Anomaly Detection Based on Time Series

Guilan Wang, Zhenqi Wang, Xianjin Luo
Information and Network Management Center, North China Electric Power University, Bao ding 071003, China. E-MAIL:yu_bing_2000 @163.com

Abstract:

With the continuous deterioration of the network environment, a variety of viruses, Trojans continue to affect the security of the network. Through the network traffic anomaly detection and analysis can efficiently find problems existing in the network. This paper discusses the network traffic flow data predict and network anomaly detection, network traffic prediction using ARMA model, network anomaly detection using the exponential smoothing model. ARMA model supplies the expectation value to abnormal detection, at the same time exponential smoothing model can restoration historical flow data, making the following traffic forecast more accurate. A network traffic predict and network anomaly detection system has been developed, with which can find network anomaly and send alarms, thus improve network stability and robustness.

1. Introduction

With the network size's continued expanding and increasing network complexity, the Internet network, based on the TCP/IP protocol cluster, becomes into a global information infrastructure and is indispensable in modern life; its collapse may lead to huge economic losses. Take network attack for example, in July 2001, Code-Red worm in less than 9 hours of time, infected 250,000 computers, the direct economic loss is 2.6 billion U.S. dollars; In January 2003, SQL Slammer worm outbreak, in the first 5 minutes which led 9.5 to 1.2 billions dollars in losses. The network ingestion, caused by worms, denial of service attacks and inappropriate network allocation, has seriously affected the normal operation of the network. How effectively manages the modern large scale networks, makes them more efficient, reliable and safely, is a pressing issue faced in network maintenance.

In recent years, anomaly detection, as a branch of data mining, is being more and more attention and study. These studies generally can be divided into five categories: methods based on distribution [1], methods based on depth [2], methods based on cluster [3], methods based on the distance [4] and methods based on the density [5]. Methods based on the distribution and depth is the main methods used in statistical area. Method of distribution assumes the data set is known, according to the data distribution take "inconsistent" test for each object, and if the object is not in line with the distribution, it is considered abnormal. This is the approach taken in this paper.

In this paper, history network traffic time series is the known distribution, only the current moment is forecast expectation value, if the actual value fails to pass "inconsistent" test, then it is considered to be abnormal. At the same time, the abnormal value in history data time series will be replaced by a reasonable alternative, so that the later forecast will be more accurate. Here, a practical system is implemented to accomplish the network traffic forecast and anomaly detection, in which the network traffic expectation uses the ARMA forecasting model, consistency analysis uses exponential smoothing model. Once a data is identified an abnormal value, the system will report to the administrator, then can take measures to guarantee the normal operation of network.

This paper is structured as follows, Section 2 describes the ARMA model and its application, Section 3 establishes the ARMA model for a campus network load system, Section 4 gives an exponential smoothing model used to find anomaly and improve forecast accuracy, Section 5 gives the system structure, Section 6 draws the conclusions and our future work.

2. Arma Model Forecasting Procedure

For a stationary time series of network load data, because the properties of historical data and future data are mutually similar, the historical data can be used as the

978-0-7695-3570-8/09 $25.00 © 2009 IEEE
DOI 10.1109/WCSE.2009.382

reference to formulate an adequate ARMA model [6]. In this section introduce the general apply of ARMA model, which is the linear combination of autoregressive (AR) and moving average (MA). Figure 1 depicts the identification process of this model, which can be largely divided into three steps, including system modeling, parameter estimation, and adequacy validation.

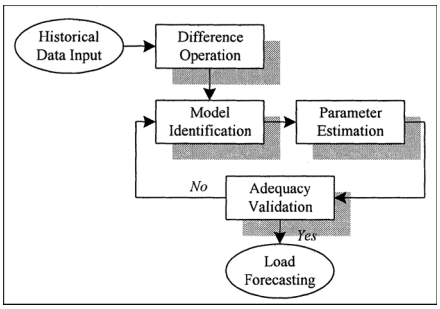

Figure 1. ARMA Model identification process

2.1. Modeling Process

In statistics, autoregressive moving average (ARMA) models, sometimes called Box-Jen-kins models after the iterative Box-Jenkins methodology usually used to estimate them, are typically applied to time series data.

A system load expressed as the following ARMA form:

$$\phi_{\mathrm{p}}(B)y_t = \theta_q(B)a_t \qquad \text{(Eq. 1)}$$

where y_t is the observed time series of load at time t, a_t is the white noise at time, and B is the back-shift operator such as that $By_t = y_{t-1}$, $B^m y_t = y_{t-m}$. In the left-hand side of (Eq.1), $\phi_{\mathrm{p}}(B) = 1 - \phi_1 B - \phi_2 B - \ldots \phi_P B^p$, where $\phi_1,\ldots,\phi_p$ are parameters of the AR part, and is the AR order. For the right-hand side of (Eq.1), $\theta_q(B) = 1 - \theta_1 B - \ldots - \theta_q B^q$ where $\theta_1,\ldots\theta_q$ are parameters of the MA part, and is the MA order. In the study, the information on the sample autocorrelation function and the sample partial autocorrelation function are used as references to conjecture the appropriate model order.

2.2. Parameter Estimation

Following the model formulation, the related parameters are required to estimate. This work can be done with the aid of the gradient-based method, where parameters are estimated in order to have zero gradient of mean squared sum of fitting errors to the historical load data.

2.3. Adequacy Validation

When the parameters of the model are well estimated, the adequacy of the model should be validated. Three items listed below are required to confirm in this procedure:

1) Weather the estimated parameters are significantly different from zero, in other words the sequence requirements mean zero.

2) Residuals are the realization of white noise process.

3) The fitted model is adequate.

After the confirmation of the above items, the formulated model is ready to perform the load forecast. Yet, if the model fails to pass these tests, the aforementioned process should be repeated again until an adequate model is achieved.

3. System Modeling

Every one hour collects the total exports flow data of a campus network forming into a time series. Figure 2 is a graphics description of the load data time series. The horizontal axis is time; the vertical axis is flow data, in units of Meta Bytes.

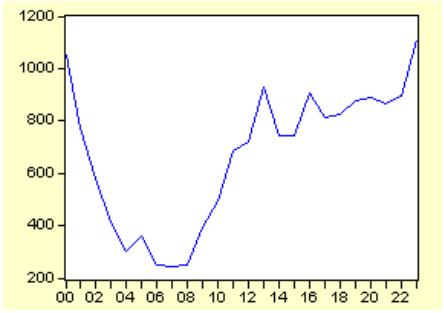

Figure 2. The time series plot of network load

3.1. Autocorrelation and Partial Autocorrelation Analysis

By autocorrelation and partial autocorrelation analysis, using the software Eviews[7], the results is shown in Figure 3(in the next page).

Through Figure 3 we can see that the sequence of the correlation coefficient quickly falls into the random interval and with a tailing, showing that sequence is stable. Only a smooth time series can be directly used to form an ARMA model, or it must be properly dealt with so that a smooth sequence could meet the requirements. Here the sequence is stable, so can directly establish the ARMA model.

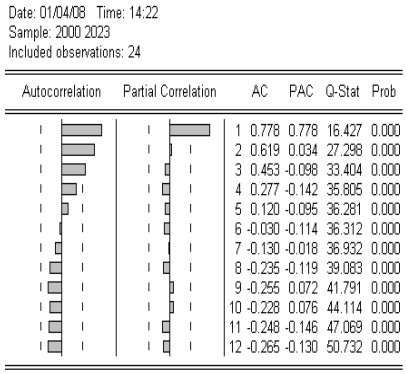

Figure 3. The autocorrelation and partial autocorrelation plot

3.2. ARMA Model Estimation

To apply the ARMA(p, q) model we need to identify parameters p and q. And by the autocorrelation and partial autocorrelation plot we can determine them. Figure 2 indicates that partial autocorrelation in the future will soon become zero, p can be admitted 1[8]. Since the correlation coefficient, when k=1, 2 or 3, is significant non-zero, or can take q=1 or 2[9]. System can be used ARMA(1,1) model or ARMA(1,2) model. In order to make the problem as simple as possible, we take the ARMA(1, 1) model.

3.3. Adequacy Validation

Now we check the adequacy of above ARMA model, it can be completed through checking in if it meets the 3 rules put out in 2.3.

3.3.1. Estimated parameters fitness. In the application of ARMA model, the sequence requirements mean zero, if the series does not mean zero, through a transformation, the sequence could be transformed into mean zero.

In this case the first to take the logarithm sequence, and then have their backward difference sequence. Concrete steps are as follows.

$$y_k = \log(y_k) \quad \text{(Eq. 2)}$$

$$y_k = y_k - y_{k-1} \quad \text{(Eq. 3)}$$

Analysis the transformed sequence, we get the result shown in Fig.4.

As shown in the Figure 4., the two roots are both outside the unit circle, meet the regressive and smooth requirement.

3.3.2. White noise testing. There we should confirm in this procedure if the residuals e^t are the realization of white noise process. If the residual sequence is not white noise sequence, means that still some useful information exist in the residual sequence and have not been extracted. Then we need to further improve the model. The testing usually focus on the residual sequence of random testing varies, that is when the lag k≥1, the sample sequence's autocorrelation coefficient should be approximate 0.

Dependent Variable: DY
Method: Least Squares
Date: 01/04/08 Time: 15:15
Sample (adjusted): 2002 2023
Included observations: 22 after adjustments
Convergence achieved after 13 iterations
Backcast: 2001

Variable	Coefficient	Std. Error	t-Statistic	Prob.
AR(1)	0.698745	0.221956	3.148126	0.0051
MA(1)	-0.565834	0.307623	-1.839379	0.0808

R-squared	0.217754	Mean dependent var	0.016207
Adjusted R-squared	0.178642	S.D. dependent var	0.222692
S.E. of regression	0.201823	Akaike info criterion	-0.276344
Sum squared resid	0.814650	Schwarz criterion	-0.177158
Log likelihood	5.039782	Durbin-Watson stat	2.085573

Inverted AR Roots	.70
Inverted MA Roots	.57

Figure 4. ARMA (1, 1) model parameter estimate

The residual sequence's autocorrelation function is:

$$r_k(e) = \frac{\sum_{t=k+1}^{n} e_t * e_{t-k}}{\sum_{i=1}^{n} e_t^2} \qquad k = 1,2,\cdots,m \quad \text{(Eq. 4)}$$

where n is the number of the observations used to calculate r_k, m is the greatest lag. If the sample size n is great, n/10 may be desirable for m. When n is small, m desirable n/4.

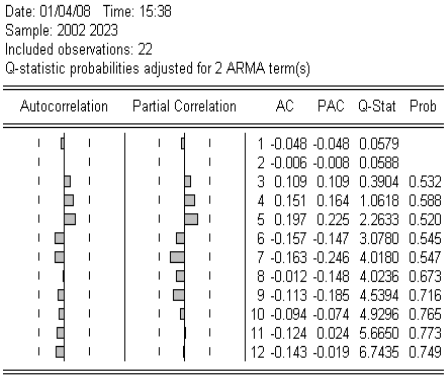

Figure 5. Residual sequence's autocorrelation plot

Testing the statistics:

$$Q = n(n+2)\sum_{k=1}^{m} \frac{r_k^2(e)}{n-k} \quad \text{(Eq. 5)}$$

Under the assumptions of zero, Q subject to distribution $\chi^2(m-p-q)$. Given confidence 1-α (usually take

α =0.05 or 0.10), if $Q \le \chi^2(m-p-q)$ can't refuse the original assumption that the residual sequence is independent of each other, then pass the test; or the model fails to pass the test.

In Figure 5 the sample size of residual sequence is n, so the greatest lag m may take 12/4=3. From the line k=3 we get the test's statistics Q's value 0.3904, from the row Prob read out the probability of the first error of refusing the original assumption is 0.532, in other words, the residual sequence is independent or being white noise's probability is great. So can't refuse the sequence's mutual independence, our process passes the test.

4. An exponential smoothing model

Here is the exponential smoothing model, which can identify large swings in the network and remove them from an observed time series, so that network location volume can be more accurately predicted.

For simplicity, a single hour is utilized, and hj used to denote the observed impressions during that hour in week j (i.e., time point i in the time series).

To determine the expected value for each point, denoted by E(hj), the exponential smoothing model allows accumulated expectations to be maintained without requiring additional resources (e.g., memory) to store prior values. It also adjusts quickly to learn when a spike is in fact an increase rather than an anomaly. This model requires at least two points in the time series, and is defined recursively as follows:

$$E(h_2)=h_1, \; E(h_{j+1})=\alpha h_{j+}(1-\alpha)E(h_j) \text{ for } i>1 \quad \text{(Eq. 6)}$$

In other words, the expected value for a point is a linear combination of the previous value and the previous expected value. In practice the (Eq. 6) is used to calculate the expectation "on the fly:" for each new data point and the expectation for the next point is also updated by applying (Eq. 6).

The variance of (of any random variable) hi, which is denoted by Var(hi), is defined as:

$$Var(h_i)=E([h_{i-E}(h_i)]^2) \quad \text{(Eq. 7)}.$$

In words, the variance is the expected squared difference between the observation and the expectation. Expanding the squared term, and noting that E(E(X))=E(X) yields:

$$Var(h_i)=E(h_i^2)-E(h_i)^2 \quad \text{(Eq. 8)}.$$

Thus, given a model for the expected squared values in the sequence, it is easy to derive the variance. The expected squared values are modeled using another exponential smoothing model:

$$E(h_2^2)=h_1^2 \; E(h_{i+1}^2)=\beta h_i^{2+}(1-\beta)E(h_i^2) \text{ for } i>1 \quad \text{(Eq. 9)}.$$

As in the calculation of E(hj), the expected square values can also be expressed using a weighted sum of observations.

To determine whether or not a value is an anomaly, the difference between that value and the expected value is measured in terms of the standard deviation, denoted SD, which is simply the square root of the variance:

$$SD(h_i)= \{\text{square root over } (Var(h_i))\} \quad \text{(Eq. 10)}.$$

A value is deemed to be an anomaly whenever it is more than 2 standard deviations away from the expected value.

Once the anomalies are detected, they can be replaced in the time series utilizing the model as well. Intuitively, when a value differs from its expected value by more than some constant (possibly non-integer) number of standard deviations, that value is replaced with its expectation. When removing an anomaly, the squared value within the calculation of the variance is also replaced, but instead of using the expected square value, it is replaced with the squared value of the average of the expected value and the actual anomaly value (described infra); this process ensures that if the time series is actually making a change (i.e., the value was not really an anomaly), that it does not remove too many false-anomaly values.

5. System Implementation

The network traffic forecast and anomaly detection system is comprised of data collection component, network traffic forecast component and anomaly detection. Figure 6 is the system structure diagram. The anomaly detection component receives an input from the data collection component and interacts with the network traffic forecast component, provides an output. The input is typically a historical time series associated with a data set. The anomaly detection component utilizes data slices from the input to facilitate determine anomaly. The output is the anomalies detected from the input.

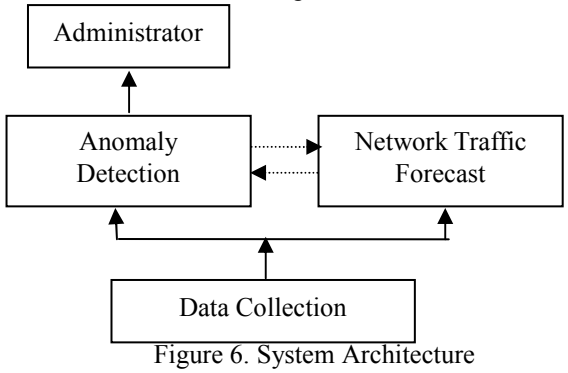

Figure 6. System Architecture

Data collection component: reading campus network load data from the port. By setting a timer, per hour

conducts a data collection. Here the main use is the API interface of related equipments.

The network traffic forecast component provides an expectation value and a standard deviation for each point in the series. Using the ARMA model formulated in chapter 3, forecast the network traffic at that time.

The anomaly detection component receives the historical time series and interacts with the network traffic forecast component to find anomaly. In order to gain the goal, an exponential smoothing model is adopted to analyze a new point in the flow time series, and determine weather it is an anomaly. If a value is deemed to be an anomaly, the system will send administrator an alarm message and in the time series it will be replaced by its expectation value, thus make the next forecast more accurate.

6. Conclusions and the Future Work

Based on the discussion of network traffic forecasting of time series and network anomaly detection, developed a system. The system could network traffic prediction and analyze the similarity between predictive value and the actual value, thus identify anomalies. When anomaly occurs, system will alarm, So that administrators can take measures to protect the normal operation of the network. The forecast for the network traffic uses ARMA forecasting model and the exponential smoothing model is used to determine weather a value is abnormal; what's more, if outlier encountered, exponential smoothing model can also reasonably alternate the history time series point with their expectations, so that the following of a more accurate forecast.

In this paper our model is established manually, and may not feasible to all the cases. In addition abnormal detection is very simple. As we all known, anomaly is variety. In most cases, the network administrator hope to know which kind of anomaly detected. If system can tell administrator anomaly's types and give some advice, it will be better. That is what we are going to do next.

Acknowledge

This work was supported by the Young Teachers Fund for Scientific Research of North China Electric Power University: 200611021.

References

[1] V. Barnett, T. Lewis. Outliers in statistical data [M]. John Wiley, 1994.

[2] T. Johnson, I. Kwok, and R.T. Ng. Fast computation of 2-dimensional depth contours.

[3] Z.He, X.Xu and S.Deng. Discovering cluster based local outliers [J]. Pattern Recognition Letters, 2003, 24(9/10):1641-1650.

[4] Sridhar Ramaswamy, Rajeev Rastogi, Kyuseok Shim, Effiecient Algorithms for Mining Outliers from Large Data Sets, In Proc. of the ACM SIGMOD International Conference on Management of Data, 2000.

[5] W. Jin, A.K.H. Tung, and J. Ha. Mining top-n local outliers in large databases. In Proc. KDD, pages 293–298, 2001.

[6] J. M. Mendel, "Tutorial on higher-order statistics in signal processing and system theory: Theoretical results and some applications", Proc. IEEE, vol. 79, pp. 278-305, Mar. 1991.

[7] Yi-Hui Dan, Data analysis and application of Eviews, China Statistics Press, Beijing, 2002.

[8] D. Aboutajdine, A. Adib, and A. Meziane, "Fast adaptive algorithm for AR parameters estimation using higher order statistics," IEEE Signal Processing Magazine, vol. 44, no. 8, pp. 1998–2009, August 1996.

[9] X. D. Zhang and Y. S. Zhang, "Singular value decomposition-based MA order determination of nongaussian ARMA models," IEEE Trans. Signal Processing, vol. 41, pp. 2657–2664, Aug. 1993.