International Journal of Future Generation Communication and Networking
Vol. 8, No. 6 (2015), pp. 205-214
http://dx.doi.org/10.14257/ijfgcn.2015.8.6.20

Anomaly Detection of Network Traffic Based on Prediction and Self-Adaptive Threshold

Haiyan Wang

Department of Information Engineering, Binzhou University, Binzhou, Shandong, China, 256600
Haiyanwang_631@126.com

Abstract

Security problems with network are significant, such as network failures and malicious attacks. Monitoring network traffic and detect anomalies of network traffic is one of the effective manner to ensure network security. In this paper, we propose a hybrid method for network traffic prediction and anomaly detection. Specifically, the original network traffic data is decomposed into high-frequency components and low-frequency components. Then, non-linear model Relevance Vector Machine (RVM) model and ARMA (Auto Regressive Moving Average) model are employed respectively for prediction. After combining the prediction, a self-adaptive threshold method based on Central Limit Theorem (LCT) is introduced for anomaly detection. Moreover, our extensive experiments evaluate the efficiency of proposed method.

Keywords: *Network traffic prediction, Anomaly detection, Wavelet decomposition, Central Limit Theorem*

1. Introduction

With the development of Internet and the increasing of businesses, security problems with network have been significant nowadays. Network failures and malicious attacks could contribute to anomalies of network traffic [1]. Therefore, how to effectively monitor network traffic and detect anomalies of network traffic has been important in network management.

Network traffic is typically collected as time series, and reveals the statistics characteristics and variations. As an unstable time series data, linear models such as Autoregressive Moving Average (ARMA), Controlled AutoRegressive (CAR) [2], or Autoregressive Integrating Moving Average (ARIMA) [3] cannot comprehensively reflect the characteristics of network traffic, and therefore, the prediction accuracy is relatively poor. Therefore, simple statistical models are not good enough for network traffic prediction.

In this paper, we propose a hybrid method for network traffic prediction and anomaly detection. Specifically, the original network traffic data is decomposed into two components using wavelet analysis [4], that is, high-frequency components and low-frequency components. For high-frequency components, the regularity and periodicity are relatively weak and thus still non-linear. Therefore, non-linear model Relevance Vector Machine (RVM) [5] is applied for prediction. For low-frequency components, the sequence is relatively stable, and therefore, ARMA (Auto Regressive Moving Average) model is employed. Then, combine above two prediction results for both components we get the final prediction for the original network traffic sequence.

Then, we employ a self-adaptive threshold method to detect if a predicted value is an anomaly. Specifically, the threshold is dynamically determined by the Central Limit Theorem (CLT) [6] based on the confidence interval over the network traffic time series sequence data.

ISSN: 2233-7857 IJFGCN

The remain of this paper is organized as follows. Section 2 provides some related work. In Section 3, we present our proposed model for network traffic prediction, and in Section 4, we describe the detection of anomalies. Empirical experiments are conducted in Section 5. Finally, the paper is concluded in Section 6.

2. Related Work

Existing efforts on network traffic anomaly detection include statistics based methods and machine learning based methods.

The first category is statistics based methods. For example, Thottan [7] captured the anomaly through the burst of association patterns of MIB variables. Wang [8] detected time series burst by the nonparametric cumulative summary method. Barford [9] applied wavelet analysis into network traffic anomaly detection. Kim [10] extend wavelet analysis into IP package data anomaly detection. Galeano [11] employed ARMA model for anomaly detection. Then, Asrul [12] introduced ARMIA model for predicting traffic and detect anomaly. Brauckhoff [13] employed PCA analysis for traffic anomaly detection.

The second category is machine learning based methods. For example, Tsai [14] used k-means clustering to group the original data into several clusters, and then find out the objects with maximum deviation. Su [15] introduced KNN (K-Nearest Neighbor) for for online anomaly network traffic identification. Sotiris [16] employed SVM (Support Vector Machines) for classification. Ye [17] applied decision trees to learn a set of classification rules. Intelligence algorithms such as GA (Genetic Algorithm) [18] are also applied for traffic anomaly detection.

In this paper, we propose a hybrid method to solve the network traffic prediction problem.

3. Network Traffic Prediction Model

The basic idea for network traffic prediction is to first perform wavelet decomposition to transform the original network traffic sequence into high-frequency components and low-frequency components. After dealing with each component respectively, the final prediction is combined for the original network traffic sequence.

The workflow of network traffic prediction is illustrated in Figure 1. The subsequent sections will discuss each step in details.

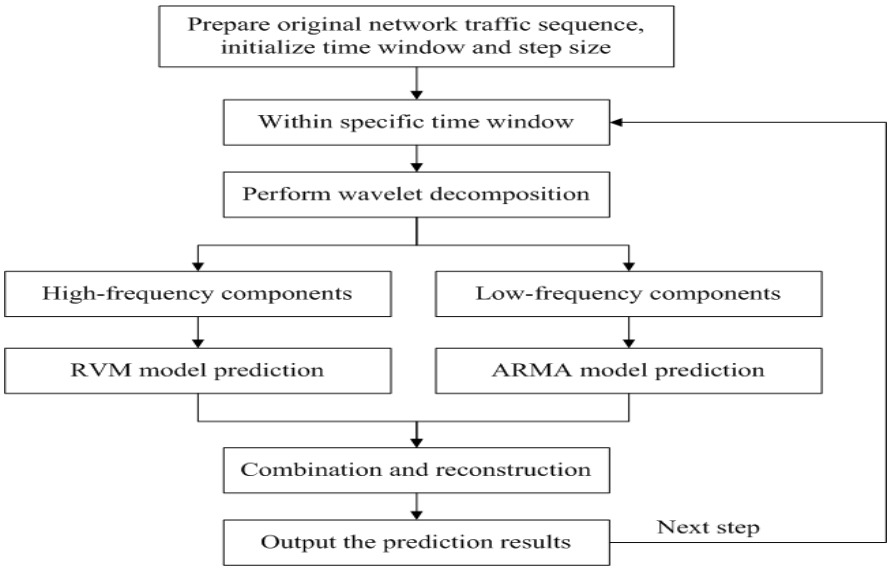

Figure 1. Workflow of Network Traffic Prediction

3.1. Wavelet Decomposition

Transform the original network traffic sequence S using wavelet decomposition into multi-scale sequences, that is, high-frequency components and low-frequency components. Wavelet transformation is the inner product of a square integrable function $f(t)$ and a wavelet function $\varphi(t)$:

$$W_f(a,b) = \left\langle f, \varphi_{a,b} \right\rangle = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t) \varphi^* \left(\frac{t-b}{a} \right) dt \ , \tag{1}$$

where $\langle \cdot \rangle$ denotes inner product, $a > 0$ is the scaling factor, b is the shifting factor, $*$ denotes complex conjugate, and $\varphi_{a,b}(t)$ is the wavelet, and

$$\varphi_{a,b}(t) = \frac{1}{\sqrt{a}} \varphi \left(\frac{t-b}{a} \right). \tag{2}$$

Adjusting the value of a could either extend ($a > 1$) or shrink ($a < 1$) $\varphi_{a,b}(t)$, and adjusting b would affect the analysis results of $f(t)$ around b. The values of a, b are related to the forms of $\varphi(t)$. $\varphi(t)$ is the mother wavelet function, and satisfies following conditions:

$$\begin{aligned} &\int_{-\infty}^{+\infty} \varphi(t) dt = 0, \ or \\ &\int_{-\infty}^{+\infty} \frac{|\psi(\omega)|}{|\omega|} d\omega = C_\varphi < \infty, \end{aligned} \tag{3}$$

Where $\varphi(t)$ is the Fourier transformation of $\psi(\omega)$.

Typically, discretization is performed on a, b:

$$a = a_m^0, b = n b_0 a_m^0 , \tag{4}$$

Where m, n are integers, $b_0 > 1$ is a constant.

Therefore, the discrete wavelet function is represented as:

$$\varphi_{m,n}(t) = \frac{1}{\sqrt{a_0^m}} \varphi \left(\frac{t - n b_0 a_0^m}{a_0^m} \right) = \frac{1}{\sqrt{a_0^m}} \varphi \left(a_0^{-m} t - n b_0 \right). \tag{5}$$

And the corresponding wavelet transformation is:

$$W_f(m,n) = \left\langle f, \varphi_{m,n} \right\rangle = \int_{-\infty}^{+\infty} f(t) \varphi_{m,n}(t) dt \ . \tag{6}$$

Specifically, when $a_0^m = 2, b_0 = 1$, it is called binary discrete wavelet transformation. Apply Mallat algorithm to decompose the original sequence into two components.

$$\begin{cases} a_{j+1} = h_0 a_j \\ d_{j+1} = h_1 d_j \end{cases}, j = 0,1,..., m \ , \tag{7}$$

where h_0 is the low-pass decomposition filter, h_1 is the high-pass decomposition filter, a_j is low frequency coefficient, and d_j is high frequency coefficient. When $j = 0$, it denotes the original sequence S.

Therefore, Mallat algorithm decomposes the original sequence into low frequency component and high frequency component. The former is the approximate component, which reflects the outline and trend features. The latter is the detail component, which reflects the dynamic factors influences such as random perturbation.

After wavelet decomposition, we get the coefficients of low frequency and high frequency. The reconstruction is performed as follows:

$$\begin{cases} A_j = g_0 a_j \\ D_j = g_1 d_j \end{cases}, j = 0,1,..., m , \tag{8}$$

where g_0 is the low-pass decomposition filter, g_1 is the high-pass decomposition filter, A_j is the low frequency component, and D_j is the high frequency component.

Therefore, Given a time window length w and step size k, for the network traffic sequence within w, notated as $S_t : \{x_1, x_2,..., x_w\}$, we decompose S_t into low frequency and high frequency components. Next, we employ different prediction methods for each component.

3.2. RVM Based Prediction

We use RVM model to predict the network traffic for high frequency components. The reason is that the high frequency component is indeed non-linear. As a machine learning technique, RVM is a popular non-linear regression model. Compared to SVM, RVM can avoid over learning, reduce the computation of kernel function, and is more suitable for online analysis. Indeed, RVM has been widely applied in failure detection [19] and network traffic analysis [20].

3.2.1. RVM Basics

The basic idea of RVM is to calculate the weights for Relevance Vectors by maximizing the posteriori probability. If the training sample is $\{x_i, t_i\}, i = 1,2,..., N$, where x_i is the input sample eigenvalues, and y_i is the target variable, then

$$y(x, \boldsymbol{\omega}) = \sum_{i=1}^{N} \omega_i K(x, x_i) + \omega_0 + \varepsilon_n , \tag{9}$$

where $K(x, x_i)$ is the kernel function, $\boldsymbol{\omega} = (\omega_1, \omega_2,..., \omega_N)^T$ are the weights, N is the size of training samples, and ε_n is the noise.

If $\varepsilon_n \sim N(0, \sigma^2)$, then the likelihood function is

$$p(t \mid \boldsymbol{\omega}, \sigma^2) = (2\pi\sigma^2)^{-\frac{N}{2}} \exp\left\{-\frac{\| t - \boldsymbol{\Phi}\boldsymbol{\omega} \|^2}{2\sigma^2}\right\}, \tag{10}$$

where $\boldsymbol{\Phi} = [\varphi(x_1), \varphi(x_2),..., \varphi(x_N)]^T$, $\varphi(x_i) = [K(x_i, x_1),..., K(x_i, x_N)]^T$, and to avoid over-learning, ω satisfies:

$$p(\boldsymbol{\omega} \mid \alpha) = \prod_{i=0}^{N} \frac{\alpha_i}{\sqrt{2\pi}} \exp\left\{-\frac{\alpha_i \omega_i^2}{2}\right\}, \tag{11}$$

where α_i is super parameter.

Suppose the probability distribution of α_i, σ^2 is Gamma distribution, that is, $p(\alpha) = Gamma\ (a,b)$, $p(\sigma^2) = Gamma\ (c,d)$, and $Gamma\ (a,b) = \Gamma(a)^{-1} b^a a^{a-1} e^{-ab}$, where $\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt$, a, c are the shape parameters of Gamma distribution, b, d are the scaling parameters. Then the posterior probability is:

$$p(\boldsymbol{\omega} \mid t, \alpha, \sigma^2) = \frac{p(t \mid \boldsymbol{\omega}, \sigma^2)\, p(\boldsymbol{\omega}, \alpha, \sigma^2)}{p(t \mid \alpha, \sigma^2)} = N(t \mid \mu, \sum), \tag{12}$$

where $\sum$ is the posterior covariance, μ is the mean, and

$$\begin{cases} \sum = (\sigma^2 \Phi^T + A) - 1 \\ \mu = \sigma^{-2} \sum \Phi^T t \end{cases}, \tag{13}$$

where $A = (\alpha_1, \alpha_2, ..., \alpha_N)$ is the diagonal matrix, and

$$\begin{cases} p(t \mid \alpha, \sigma^2) = \int p(t \mid \boldsymbol{\omega}, \sigma^2)\, p(\boldsymbol{\omega} \mid \alpha) d\boldsymbol{\omega} \\ N(0, \psi) = 2\pi - \frac{N}{2} \mid \psi \mid^{-1/2} \exp\left\{ -\frac{t^T \psi^{-1} t}{2} \right\} \end{cases}, \tag{14}$$

where $\psi = \sigma^2 I + \Phi A^{-1} \Phi^T$.

We iteratively estimate the maximum approximate solution of Equation (14). Let the partial derivatives on α, σ^2 be 0, we get

$$\alpha_i^{new} = \frac{\gamma_i}{\mu_i^2}, \tag{15}$$

Where $\gamma_i = 1 - \alpha_i \Sigma_{ii}$, and Σ_{ii} is the i-th diagonal element, and

$$(\sigma^2)^{new} = \frac{\| t - \Phi \mu \|^2}{N - \sum_{i=0}^{N} \gamma_i}. \tag{16}$$

After many iterations of Equations (15) and (16), we get the convergence value of α_i, and the corresponding x_i is called Relevance Vector.

3.2.2. RVM Prediction Model

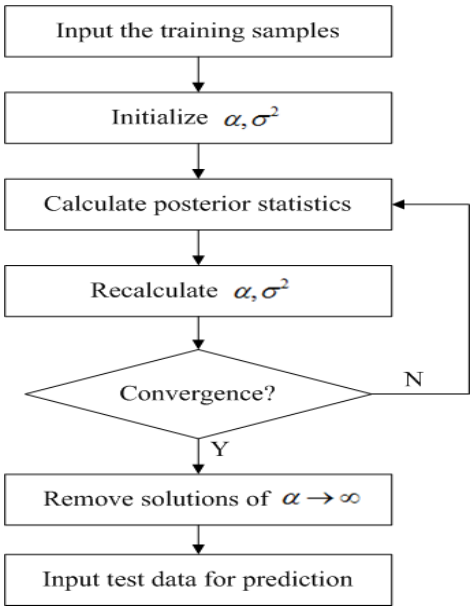

Figure 2. Flow Chat of RVM Prediction Model

Let the high frequency components $\{D_1, D_2, ..., D_w\}$ as training sample, and we need to predict $\{D_{w+1}, D_{w+2}, ..., D_{w+k}\}$ given the network traffic sequence $S_t : \{x_1, x_2, ..., x_w\}$. Figure 2 gives the flow chat of RVM prediction model. The steps are as follows.

Step 1: perform data normalization for $\{D_1, D_2, ..., D_w\}$ as follows:

$$\hat{D}_i = 2 \frac{D_i - \min(D_i)}{\max(D_i) - \min(D_i)} - 1, \tag{17}$$

And $\hat{D}_i \in [-1,1]$.

Step 2: prepare the input vector $X_{m,l+1}$

$$X_{m,l+1} = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \\ x_{1+k} & x_{2+k} & \cdots & x_{m+k} \\ \vdots & \vdots & \cdots & \vdots \\ x_{1+lk} & x_{2+lk} & \cdots & x_{m+lk} \end{bmatrix}, \tag{18}$$

and the output vector $Y_{k,l+1}$

$$Y_{k,l+1} = \begin{bmatrix} x_{m+1} & x_{m+2} & \cdots & x_{m+k} \\ x_{m+k+1} & x_{m+k+2} & \cdots & x_{m+2k} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m+lk+1} & x_{m+lk+2} & \cdots & x_{m+k(l+1)} \end{bmatrix}, \tag{19}$$

where $m + k(l+1) \le w$, and train the RVM model.

Step 3: feed $X_p = [x_{w-m+1} \quad x_{w-m+2} \quad \cdots \quad x_w]$ into the learned RVM model, and get the predicted value $Y_p = [x_{w+1} \quad x_{w+2} \quad \cdots \quad x_{w+k}]$, and perform inverse normalization on that.

For the low-frequency components, we employ ARMA model for prediction. Combine the predictions of high-frequency and low frequency components together, we get the predicted value of network traffic, denoted as $\hat{Y}$.

4. Anomaly Detection of Network Traffic

Now we have the predicted value of network traffic, and we need to determine if it is an anomaly. We use an adaptive threshold based method in this paper.

Basically, the threshold is determined by the Central Limit Theorem based on the confidence interval over the network traffic sequence $S_t : \{x_1, x_2, ..., x_w\}$. The confidence interval of S_t is:

$$\left(\bar{x} - t_{\frac{\alpha}{2}}(n-1)\frac{S}{\sqrt{n}}, \bar{x} + t_{\frac{\alpha}{2}}(n-1)\frac{S}{\sqrt{n}} \right), \tag{20}$$

where $\bar{x}$ is the mean of x_i , $n \le N$, and S is the mean square deviation.

Therefore, the empirical threshold is defined as the range:

$$\left(\bar{x} - 3 \times t_{\frac{\alpha}{2}}(n-1)\frac{S}{\sqrt{n}}, \bar{x} + 3 \times t_{\frac{\alpha}{2}}(n-1)\frac{S}{\sqrt{n}} \right). \tag{21}$$

Therefore, the upper bound is $U = \bar{x} + 3 \times t_{\frac{\alpha}{2}}(n-1)\frac{S}{\sqrt{n}}$, and lower bound is $L = \bar{x} - 3 \times t_{\frac{\alpha}{2}}(n-1)\frac{S}{\sqrt{n}}$. The rules of anomaly detection are as follows:

(1) If $L < \hat{Y} < U$, the predicted network traffic is normal;
(2) If $L < Y < U$, the observed network traffic is normal;
(3) If $\hat{Y} > U$ or $\hat{Y} < L$, the anomaly of predicted network traffic is detected;
(4) If $Y > U$ or $Y < L$, the anomaly of observed network traffic is detected.

5. Experiment

5.1. Dataset Description

The dataset we use in this experiment is achieved from the network traffic library (http://newsfeed.ntcu.net/-news/2006), which collected 300 network traffic data records per hour from August 1st to November 10th, 2011, notated as $\{x_t, t = 1,2,...,300\}$. Former 250 records are used as training sample, and the latter 50 are for testing. Figure 3 shows the data.

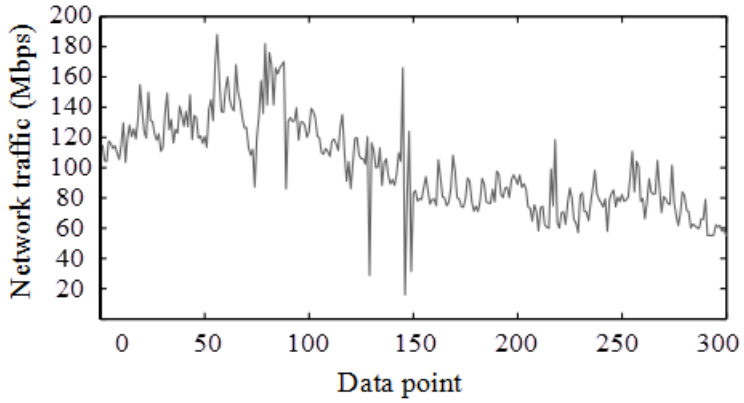

Figure 3. Data Sample of Network Traffic

5.2. Self-Similarity Analysis

Self-similarity can be measured by Hurst exponent H . If $H = 0.5$, the network traffic sequence is random, and there exist no relations between events. If $H \in [0,0.5)$, the network traffic sequence is anti-persistent. If $H \in (0.5,1)$, network traffic sequence is persistent with self-similarity, and larger H means more self-similarity.

We use rescaled range analysis method to calculate H :

$$(R/S)_n = A \cdot n^H , \tag{22}$$

Where n is the scale of sample data, R is the rescaled range, S is the standard deviation, A is a constant.

Figure 4 gives the log-log plot of $(R/S)_n$ and n . The slope of the fitting line for all data points are the value of Hurst exponent H . With our dataset, we have $H = 0.761$, which satisfies $H \in (0.5,1)$. Therefore, the network traffic sequence we use in the experiment has self-similarity characteristic. Accordingly, non-linear model should be employed for network traffic prediction to reduce the prediction error.

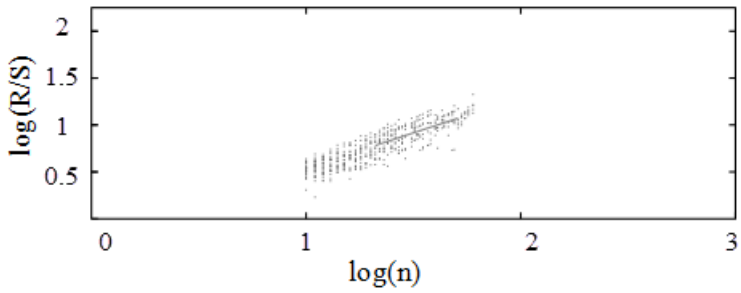

Figure 4. Hurst Exponent of Network Traffic Data Sample

5.3. Prediction Results

In order to measure the prediction results, we introduce Root Mean Square Error (RMSE) and Relative Root Mean Square Error (RRMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \hat{x}_i)^2}{N}} , \tag{23}$$

$$RRMSE = \sqrt{\frac{\sum_{i=1}^{N} (\frac{x_i - \hat{x}_i}{N})^2}{N}}, \quad (24)$$

where x_i is the observed value, and $\hat{x}_i$ is the predicted value.

Table 1 lists the prediction error of proposed method compared with wavelet analysis only method and ARMA only model. We can observe that our hybrid method outperforms other two with better accuracy, and therefore can effectively detect anomalies.

Table 1. Prediction Error of Network Traffic

Method	RMSE	RRMSE
Proposed	0.0213	0.8996
Wavelet analysis	0.2844	1.3542
ARMA	0.7652	2.6883

Figure 5 shows prediction results for an intercept of the network traffic sequence. We can observe that prediction becomes precise when the enough number of samples are collected. In this case, an anomaly is detected around data 36~38.

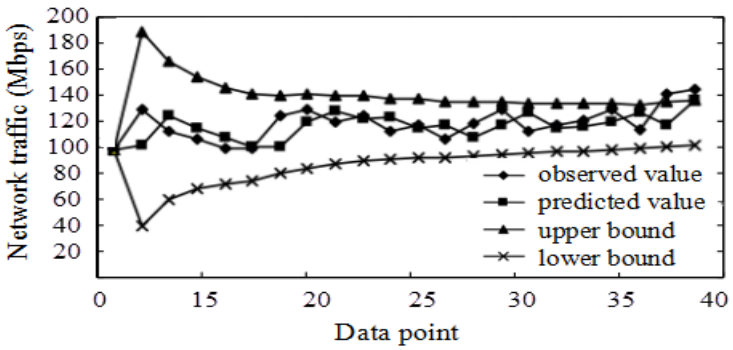

Figure 5. Prediction Results of Network Traffic

Besides, Figure 6 gives the prediction error of RVM model for high-frequency components. Basically, the prediction error is small enough, and the prediction model can fit the network traffic data pretty well.

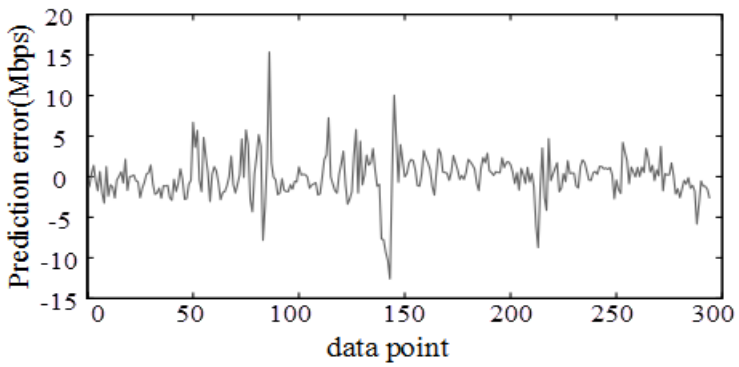

Figure 6. Prediction Error of RVM Model

6. Conclusion

In this paper, we study on the problem of predicting network traffic and detecting network traffic anomalies. Specifically, we propose a hybrid method based on wavelet analysis and RVM, and then employ a threshold based method for anomaly detection. Besides, our experiments evaluate the efficiency of our model.

This work indicates the feasibility of combining statistical methods and machine learning methods together. In future works, we'll try to explore the possibility of combining others for more interesting applications.

References

[1] A.S. Ratner and P. Kelly, "Anomalies in network traffic" Intelligence and Security Informatics (ISI), 2013 IEEE International Conference on. IEEE, **(2013)**, pp. 206-208.
[2] Y. Xiao, G. Song and Y. Liao, "Multi-innovation stochastic gradient parameter estimation for input nonlinear controlled autoregressive models", International Journal of Control, Automation and Systems, vol. 10, no. 3, **(2012)**, pp. 639-643.
[3] M. Valipour, M. E. Banihabib and S. M. R. Behbahani, "Parameters Estimate of Autoregressive Moving Average and Autoregressive Integrated Moving Average Models and Compare Their Ability for Inflow Forecasting" Journal of Mathematics and Statistics, no. 3, **(2012)**.
[4] C. Torrence and P. Gilbert Compo, "A practical guide to wavelet analysis." Bulletin of the American Meteorological society vol. 79, no. 1, **(1998)**, pp. 61-78.
[5] M.E. Tipping, "Sparse Bayesian learning and the relevance vector machine." The journal of machine learning research 1 **(2001)**, pp. 211-244.
[6] J. Davidson, "Establishing conditions for the functional central limit theorem in nonlinear and semiparametric time series processes" Journal of Econometrics vol. 106, no. 2, **(2002)**, pp. 243-269.
[7] M. Thottan and C. Ji, "Proactive anomaly detection using distributed intelligent agents [J]" Network, IEEE, vol. 12, no. 5, **(1998)**, pp. 21-27.
[8] H. W. Danlu, "Detecting SYN Flooding Attacks", Infocom, Twenty-first Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings, IEEE, **(2002)**, pp. 1530-1539.
[9] P. Barford, J. Kline and D. Plonka, "A Signal Analysis of Network Traffic Anomalies" In Internet Measurement Workshop, **(2002)**.
[10] S. S. Kim and A. L. N. Reddy, "Statistical Techniques for Detecting Traffic Anomalies Through Packet Header Data" Networking, IEEE/ACM Transactions on, vol. 16, no. 3, **(2008)**, pp. 562-575.
[11] P. Galeano, D. Peña and R. S. Tsay, "Outlier Detection in Multivariate Time Series by Projection Pursuit" Journal of the American Statistical Association, vol. 101, no. 474, **(2006)**, pp. 654-669.
[12] A.H. Yaacob, I. K. T. Tan and S. F. Chien, "ARIMA Based Network Anomaly Detection", Communication Software and Networks, 2010. ICCSN '10, Second International Conference, **(2010)**, pp. 205-209.
[13] D. Brauckhoff, K. Salamatian and M. May, "Applying PCA for Traffic Anomaly Detection: Problems and Solutions", INFOCOM 2009, IEEE. IEEE, **(2009)**, pp. 2866-2870.
[14] C. Tsai, Y. Hsu and C. Lin, "Intrusion detection by machine learning: A review", Expert Systems with Applications, vol. 36, no. 10, **(2009**), pp. 11994-12000.
[15] M. Su, "Using clustering to improve the KNN-based classifiers for online anomaly network traffic identification", Journal of Network and Computer Applications, vol. 34, no. 2, **(2011)**, pp. 722-730.
[16] V. A. Sotiris, P. W. Tse and M. G. Pecht, "Anomaly Detection Through a Bayesian Support Vector Machine", Reliability, IEEE Transactions on, vol. 59, no. 2, **(2010)**, pp. 277-286.
[17] Y. Xiaolong, L. Julong and G. Tong, "Network anomaly detection method based on principle component analysis and tabu search and decision tree classification", Journal of Computer Applications, vol. 33, no. 10, **(2013)**, pp. 2846-2845.
[18] S. Anil and R. Remya, "A hybrid method based on genetic algorithm, self-organized feature map, and support vector machine for better network anomaly detection", Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on. IEEE, **(2013)**, pp. 1-5.
[19] C. He, Y. Li and Y. Huang, "Relevance Vector Machine Based Gear Fault Detection", Pattern Recognition, 2009. CCPR 2009. Chinese Conference on IEEE, **(2009)**, pp. 1-5.
[20] Z. Qunhui, "Online Network Traffic Classification Algorithm Based on RVM", Journal of Networks, **(2013)**.

