An Efficient Anomaly Detection Framework for Cloud Computing Environment

Mingwei Lin[1*], Shuyu Chen[2]

[1] Faculty of Software, Fujian Normal University, Fuzhou 350108, Fujian, China.
[2] School of Software Engineering, Chongqing University, Chongqing 400044, China.

* Corresponding author. Tel.: 13657617291; email: linmwcs@163.com
Manuscript submitted April 22, 2014; accepted May 5, 2015.
doi: 10.17706/jcp.10.3.155-165

Abstract: Infrastructure as a Service (IaaS) is an important service type provided by cloud computing. Infrastructure resources are encapsulated into services and they are provided to users over the Internet in the form of virtual machines. A malicious user can upload malicious software into the virtual machine allocated by a cloud computing service provider and launch the side channel attacks to other virtual machines located in the same physical node by operating his own virtual machine. In order to address the above problem, this paper proposes an efficient anomaly detection framework for cloud computing environment to detect the virtual machines that present abnormal behaviors. A new feature extraction algorithm is designed to reduce the dimensionality of the collected data and a new anomaly detection algorithm is also designed to detect the abnormal virtual machines. A series of experiments are conducted on a cloud computing environment that is deployed using the open source project OpenStack to evaluate the proposed framework. Experimental results show that the proposed framework is better than other anomaly detection methods designed for cloud computing environment in terms of precision, recall, false alarm rate, and runtime.

Key words: Anomaly detection, cloud computing, principle components analysis, locality preserving projection, feature extraction.

1. Introduction

In recent years, due to the development of parallel computing, distributed computing, and grid computing, cloud computing develops quickly. Many famous information technology companies develop the related technologies about cloud computing and promote the development of industrialization of cloud computing [1].

Cloud computing is a new computing paradigm that is based on the Internet and improves the utilization of hardware resources [2]. Currently, the mainstream IT companies implement the cloud computing in different ways. Therefore, they proposed different concepts about cloud computing and there is not a unified definition about cloud computing. The National Institute Standards and Technology that is a standard organization for the U.S. federal government undertakes the task by providing the technologies and standard supports for U.S. government. This organization got together with the mainstream IT manufacturers in cloud computing and they proposed a widely accepted definition of cloud computing [3]. The core of cloud computing is providing the users over the Internet with on-demand services. Cloud computing provides three types of services, which are the Infrastructure-as-a-service (IaaS), Platform-as-a-service (PaaS), and

Software-as-a-service (SaaS). IaaS is the most basic cloud computing service model. Cloud computing service providers of IaaS often offer the virtual machines to the users over the Internet. Users can access the allocated virtual machines through the Internet and upload any software into the virtual machines. In this case, malicious users often upload malicious software into their virtual machines and launch side channel attacks to other virtual machines located in the same physical node [4]. In order to address the above problem, a number of anomaly detection methods have been proposed for cloud computing.

C. Mazzariello *et al.* proposed a network-based anomaly detection system for an open source cloud computing environment [5]. They defined a series of rules that determine if the behaviors are anomalous or not. This proposed system has a high detection rate. However, all the rules about attacks must be defined in advance and this kind of system can not detect unknown attacks.

C. N. Modi *et al.* proposed an anomaly detection framework that integrates the Bayesian classifier and Snort-based network anomaly detection system for cloud computing environment [6]. The proposed framework uses the Snort tool for collecting the network data from the cloud computing environment and then the Bayesian classifier is introduced to classify the collected data. Experimental results show that the proposed framework could not only reduce the false positive rate and the computational cost, but also detect unknown attacks.

P. Kumar *et al.* improved the security of cloud computing environment with the help of the Hidden Markov Model (HMM) and the clustering technique [7]. The clustering technique is introduced to provide only those data that are actually required by users. The normal system call sequences are used to build the HMM of normal system behaviors, and then the probability of an observed sequence under the normal model is calculated. The observed sequence can be determined if it is an anomaly behavior or not according to the value of its probability. The clustering technique can reduce the computational cost and the HMM can effectively detect any known attack from network.

A. Kannan *et al.* proposed a new anomaly detection model that combines the genetic algorithm-based feature selection algorithm and the fuzzy support vector machine [8]. The genetic algorithm is used for feature selection and then the fuzzy support vector machine is used for anomaly detection. Experimental results show that the proposed model reduces the runtime and improves the detection precision.

C. Wang *et al.* proposed the EbAT (Entropy-based Anomaly Testing), which offers novel methods that detect anomalies for utility cloud computing by analyzing the metric distributions rather than individual metric thresholds [9]. Entropy is used as a measurement to capture the degree of dispersal or concentration of such distributions. Experimental results show that the proposed method is better than threshold-based anomaly detection methods.

M. Gupta *et al.* proposed an efficient context-aware time series anomaly detection framework for cloud computing environment [10], which is called CATS. The proposed framework integrates the information from system logs and time series measurement data to improve the precision of anomaly detection. Experimental results show that the proposed framework is more effective for anomaly detection compared with existing time series-based methods.

K. Bhaduri proposed a technique for automated anomaly detection using the machine performance data from the cloud computing [11]. The proposed technique uses a distance-based anomaly definition to identify if a machine is faulty or not. Experimental results show that the proposed technique has a low overhead for tracking anomalous machines in a cloud infrastructure.

The above researches only introduce existing anomaly detection methods in the cloud computing environment and don not consider the influence of high dimensional data on the performance of anomaly detection methods. If the dimensionality of data is very high, existing anomaly detection methods will show high computational cost and degrade the efficiency of the performance for anomaly detection. In order to

address this problem, this paper proposes an efficient anomaly detection framework for the cloud computing environment.

Our contributions can be summarized as follows:

1) Because the high dimensional data have an impact on the performance of anomaly detection, a new feature extraction algorithm is designed to reduce the high dimensionality of the collected data. The proposed feature extraction algorithm combines the advantages of the principle components analysis (PCA) and the locality preserving projections (LPP), and effectively reduces the computational cost.
2) A new anomaly detection algorithm is designed to detect abnormal virtual machines. The proposed anomaly detection algorithm introduces a Mahalanobis distance-based clustering algorithm for clustering the collected data into clusters. If the new data does not belong to any cluster, this new data will be considered as an outlier and the virtual machine that produces this data will be considered as an abnormal virtual machine.

A series of experiments are conducted on a cloud computing environment that is deployed using the OpenStack to evaluate the performance of the proposed anomaly detection framework and experimental results show that our proposed framework performs better than existing anomaly detection methods.

The remainder of this paper is organized as follows. Section 2 briefly reviews the commonly used feature extraction methods. Section 3 presents the detailed implementation of our proposed anomaly detection framework. Performance evaluation is reported in Section 4. Section 5 draws the conclusions.

2. Commonly Used Feature Extraction Methods

The data collected from the real cloud computing environment often have no class labels. Therefore, two unsupervised feature extraction methods are discussed in this section.

2.1. Principle Components Analysis

Principle Components Analysis (PCA) that is also called Karhunen-Loeve Transform is an important technique used to process the high dimensional data [12].

There is a sample data set $X=[x_1,x_2,\cdots,x_n]$. PCA aims to obtain d projection vectors (or eigenvectors) to form a transformation matrix W and maximizes the variance of the projected data space $Y=[y_1,y_2,\cdots,y_n]$ by the transformation $y_i=W^T x_i$. Namely, PCA aims to maximize the following equation:

$$\begin{aligned}&\max_W \sum_{i=1}^{n}\left\|y_i-\bar{y}\right\|^2\\&=\max_W \sum_{i=1}^{n} W^T\left(x_i-\bar{x}\right)\left(x_i-\bar{x}\right)^T W\\&=\max_W W^T C W\end{aligned} \tag{1}$$

where

$$\bar{y}=\left(\sum_{i=1}^{n} y_i\right)/n \tag{2}$$

$$\bar{x}=\left(\sum_{i=1}^{n} x_i\right)/n \tag{3}$$

$$C = \left(\left(x_i - \bar{x} \right) \left(x_i - \bar{x} \right)^T \right) / n \tag{4}$$

$$W = \left[w_1, w_2, \cdots, w_d \right] \tag{5}$$

2.2. Locality Preserving Projection

There is a sample data set $X = \left[x_1, x_2, \cdots, x_n \right]$. LPP aims to obtain *d* projection vectors (or eigenvectors) to form an optimal transformation matrix *W* and projects the original sample data set in the following way [13]:

$$y_i = W^T x_i \tag{6}$$

The idea of LPP is that the projected data space $Y = \left[y_1, y_2, \cdots, y_n \right]$ should preserve the neighborhood structures from the original data space. So the projected data space $Y = \left[y_1, y_2, \cdots, y_n \right]$ should satisfy the following equation:

$$\min \sum_{i,j} \left\| y_i - y_j \right\|^2 S_{ij} \tag{7}$$

y_i is the projection value of x_i and S_{ij} is a similarity matrix that captures the relations between data points.

$$S_{ij} = \begin{cases} e^{-\left(\left\| x_i - x_j \right\|^2 / t \right)}, & if\ x_i \quad \text{is among} \quad \text{nearest neighbors} \\ 0, & otherwise \end{cases} \tag{8}$$

The *K*-nearest neighbor algorithm is used to determine the value of S_{ij}. The term *t* is an empirical parameter that controls the value of S_{ij}. Due to $Y = W^T X$, (7) can be calculated as follows:

$$\begin{aligned} & \frac{1}{2} \sum_{i,j} \left\| y_i - y_j \right\|^2 S_{ij} \\ & = \frac{1}{2} \sum_{i,j} \left(W^T x_i - W^T x_j \right)^2 S_{ij} \\ & = \sum_{i,j} W^T x_i S_{ij} x_i^T W - \sum_{i,j} W^T x_i S_{ij} x_j^T W \\ & = W^T X D X^T W - W^T X S X^T W \\ & = W^T X (D - S) X^T W \\ & = W^T X L X^T W \end{aligned} \tag{9}$$

where D is a diagonal matrix and $D_{ii} = \sum_j S_{ij}$. $L = D - S$ is a Laplacian matrix. $W = \left[w_1, w_2, \cdots, w_d \right]$ is a transformation matrix and $w_i \left(i = 1,2,3, \cdots, d \right)$ is a eigenvector.

Then, (7) is changed to be an optimization problem as follows.

$$\arg\min_{W} W^T XLX^T W \tag{10}$$

3. Proposed Framework

In this section, the detailed implementation of the proposed framework is described.

3.1. System Architecture

Fig. 1 shows the system architecture of the proposed anomaly detection framework for cloud computing environment. Because the normalized data set shows a high dimensionality and contains many redundant attributes, the normalized data set is processed by using a feature extraction algorithm in order to improve the performance of the anomaly detection framework in terms of the computational complexity. After dimensionality reduction, an anomaly detection algorithm is used to detection the abnormal data instances.

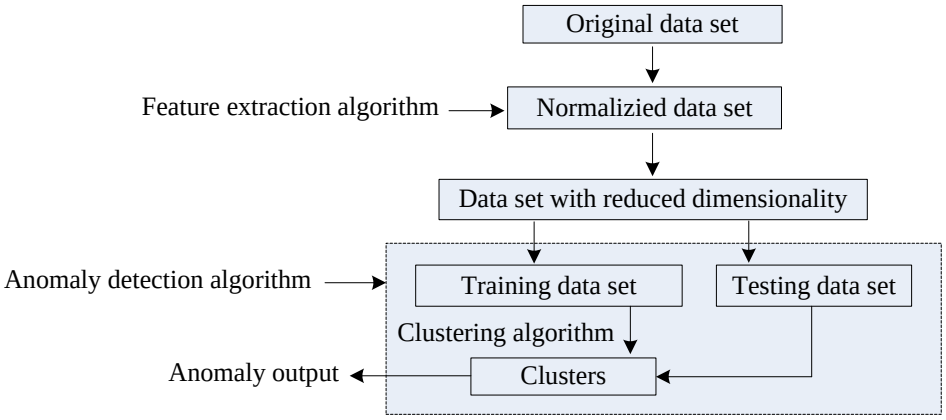

Fig. 1. System architecture.

3.2. Feature Extraction Algorithm

PCA preserves the global structure of the original data space during the dimensionality reduction process and it has been widely used in pattern recognition. However, it can not identify the local manifold structure embedded in the original data space. LPP can address the disadvantage of PCA and effectively preserves the local manifold structure of the original data set in the low dimensional space. So, we combine the advantages of PCA and LPP and propose a new feature extraction algorithm that preserves the global structure and local manifold structure of the original data set in the low dimensional space.

The feature extraction algorithm should satisfy (1) and (10) at the same time, so the objective function of the proposed feature extraction algorithm is:

$$\max_{W} \frac{W^T CW}{W^T XLX^T W} \tag{11}$$

Because WTW=1, the Lagrangian multiplier method can be used to solve (11) and a Lagrange equation is constructed as follows.

$$\psi(W) = W^T CW - \lambda(W^T XLX^T W - 1) \tag{12}$$

Derivation operation is performed to (12) and the derivation process is operated as follows:

$$\frac{\delta\psi(W)}{\delta W} = CW - \lambda XLX^T W \tag{13}$$

If (13) is equal to 0, the solution is the optimal one. So, (13) can be changed to

$$CW = \lambda XLX^T W \tag{14}$$

Let $C = XLX^T$, and then $C^{-1}CW = \lambda W$.

All the eigenvalues of $C^{-1}C$ are worked out and the largest *k* eigenvalues are selected. The *k* eigenvectors corresponding to the *k* eigenvalues are used to form the transformation matrix $W = [w_1, w_2, \cdots, w_d]$, where w_i is a eigenvector.

Finally, the projected data set can be obtained as follows:

$$Y = W^T X \tag{15}$$

where *X* is the original data set and *Y* is the projected data set. *Y* has a lower dimensional than *X*.

3.3. Anomaly Detection Algorithm

After the dimensionality reduction of the original data set, a Mahalanobis distance-based clustering algorithm is designed for clustering the data set with reduced dimensionality into clusters. If the new data does not belong to any cluster, this new data will be considered as an outlier and the virtual machine that produces this data will be considered as an abnormal virtual machine.

The Mahalanobis distance-based clustering algorithm is described in Fig. 2.

Algorithm 1 Clustering algorithm for clustering data set into clusters

Input: (1) $Y=[y_1, y_2, \cdots, y_n]$ representing the data set processed by the proposed feature extraction algorithm; (2) *PT* is the threshold of point distance

Output: Clusters $C_1, C_2, C_3, \cdots, C_k$.

```
1: Get a data instance y_t from the cloud computing environment at time;
2: IF clusters==NULL
3:         Initialize a cluster with y_t;
4: ELSE
5:         Select a cluster with the minimum distance MD to y_t, and get the cluster ID, j;
6:         IF MD[j]>PT //MD[j] is the distance between the cluster j and data instance y_t;
7:                 Initialize a cluster with y_t;
8:         ELSE
9:                 Insert the data instance y_t into the cluster j;
```

Fig. 2. Clustering algorithm for clustering data set into clusters.

During the clustering process, the distance between a data instance and the cluster centroid is calculated to capture the similarity between the data instance and the cluster. Because the Mahalanobis distance considers the distribution characteristic of the data instance and captures the associated relations between each attribute of the data instance, the Mahalanobis distance is used to calculate the distance between a data instance and the cluster centroid [14].

Therefore, the distance between a data instance and a cluster centroid is defined as:

$$P_{ij} = \sqrt{\left(y_i - C_j\right) A^{-1} \left(y_i - C_j\right)^T} \tag{16}$$

where P_{ij} is the distance between the data instance y_i and the centroid of the cluster *j* denoted as C_j. C_j can be calculated as $C_j = \frac{1}{l}\sum_{k=1}^{l} y_k$. y_k is the *k*th data instance in the cluster *j* and l is the size of the cluster *j*.

A is a covariance matrix of each attribute variable a_i of the data instances in the cluster *j*. This covariance matrix captures the associated relation among each attribute variable and it can be calculated as:

$$A = \begin{bmatrix} cov(a_1,a_1) & cov(a_1,a_2) & \cdots & cov(a_1,a_d) \\ cov(a_2,a_1) & cov(a_2,a_2) & \cdots & cov(a_2,a_d) \\ \vdots & \vdots & \ddots & \vdots \\ cov(a_d,a_1) & cov(a_d,a_1) & \cdots & cov(a_d,a_d) \end{bmatrix} \tag{17}$$

where

$$cov(a_i,a_j) = \frac{1}{n}\sum_{m=1}^{n}\left(a_i - \bar{a_i}\right)\left(a_j - \bar{a_j}\right) \tag{18}$$

Our anomaly detection algorithm relies on assumption 1.

Assumption 1: A normal data instance belongs to a cluster in the data set, while an abnormal data instance does not belong to any cluster [15].

When a new data instance is collected, the proposed feature extraction algorithm reduces the dimensionality of the new data instance. Then, the distance between the data instance and the centroid of each cluster is calculated. If the minimum distance is larger than the threshold of point distance denoted as *PT*, this data instance is considered as an anomaly point. If the minimum distance is smaller than or equal to the threshold of point distance, it belongs to one of the clusters and it is considered as a normal point.

Our proposed anomaly detection framework works well for large-scale data such as the data collected from the cloud computing environment. However, it performs badly for small-scale data.

4. Performance Evaluation

As a proof concept, a prototype of our proposed anomaly detection framework has been implemented. In order to evaluate the effectiveness of the proposed anomaly detection framework, a series of experiments were conducted on a cloud computing environment that was deployed using OpenStack [16]. The experiment setup and experimental results are described in this section.

4.1. Experiment Setup

We test our proposed anomaly detection framework in a cloud computing environment in which the open source cloud computing platform called OpenStack is used. The cloud computing environment consists of 2 server clusters in our lab. Each server cluster contains 2 servers. The cluster controller of each cluster is equipped with Intel Pentium4 3.0GHz, 2GB of RAM, and 80GB of storage capacity. One server in the cluster is equipped with Intel Core2 P8400 2.26GHz (supporting Intel-VT), 6GB of RAM, and 320GB of storage capacity, while the other one is equipped with AMD X4 640 3.0GHz (supporting AMD-V), 4GB of RAM, and 500G of storage capacity.

The data are collected from each server in the cluster and sent to the cluster controller in which our

proposed anomaly detection framework is deployed. If an anomaly occurs, the cluster controller will report the results to the cloud controller.

4.2. Experimental Results

Four performance metrics are used to evaluate the effectiveness of the proposed anomaly detection framework, which are the precision, recall, false alarm rate, and runtime.

The first three performance metrics are defined as follow [9]:

$$Precision = \frac{\#\ of\ \ successful\ \ detections}{\#\ of\ \ total\ \ alarms} \tag{19}$$

$$Recall = \frac{\#\ of\ \ successful\ \ detections}{\#\ of\ \ total\ \ anomalies} \tag{20}$$

$$False\ \ Alarm\ \ Rate = 1 - Precision \tag{21}$$

Our proposed anomaly detection framework called PCA-LPP-C is compared with EbAT, CATS, and FDCS that are described in the introduction.

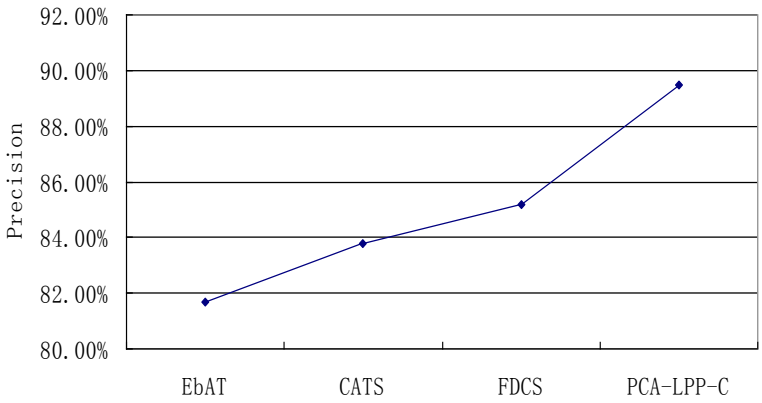

Fig. 3. Precisions for anomaly detection methods.

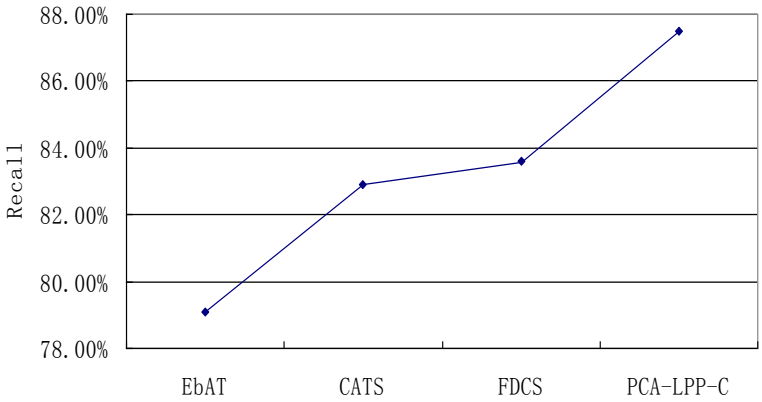

Fig. 4. Recalls for anomaly detection methods.

Fig. 3 shows the precisions for four anomaly detection methods. It can be seen that the proposed PCA-LPP-C anomaly detection framework shows the highest precision since an efficient anomaly detection

algorithm is designed for anomaly detection and a Mahalanobis distance-based clustering algorithm is used to cluster the collected data into clusters.

Fig. 4 shows the recalls for anomaly detection methods. It can be seen that the proposed PCA-LPP-C anomaly detection framework shows the highest recall since an efficient anomaly detection algorithm is designed for anomaly detection.

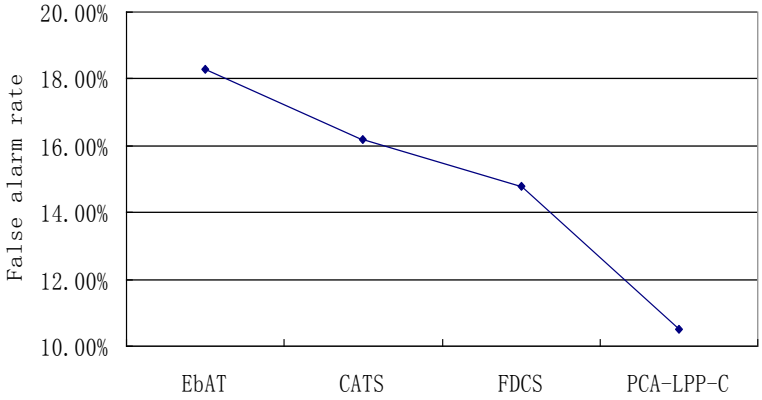

Fig. 5. False alarm rates for anomaly detection methods.

Fig. 5 shows the false alarm rates for the anomaly detection methods. It can be seen that the proposed PCA-LPP-C anomaly detection method shows the lowest false alarm rate since the false alarm rate is calculated as (1-precision) and the proposed PCA-LPP-C anomaly detection method shows the highest precision.

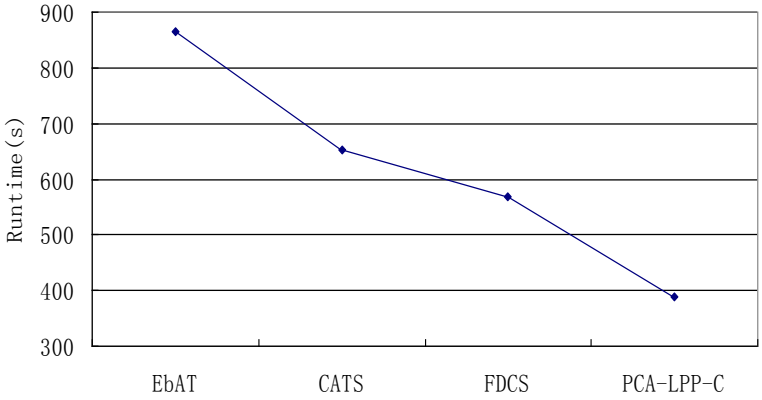

Fig. 6. Runtimes for anomaly detection methods.

Fig. 6 shows the runtimes for three anomaly detection methods. It can be seen that the proposed PCA-LPP-C anomaly detection framework shows the least runtime since an efficient feature extraction algorithm is designed in the proposed PCA-LPP-C anomaly detection framework for dimensionality reduction of the collected data and the computational cost is reduced. The EbAT, CATS, and FDCS do not reduce the high dimensionality of the collected data. They only focus on improving the precision of anomaly detection.

5. Conclusions

In this paper, an efficient anomaly detection framework is proposed for cloud computing environment. Because the high dimensional data could increase the computational cost of anomaly detection, a feature extraction algorithm is designed for dimensionality reduction of the collected data. The proposed feature extraction algorithm combines the advantages of PCA and LPP to preserve the global structure and local

manifold structure of the original data set. And then, an anomaly detection algorithm is designed to cluster the data set with reduced dimensionality into clusters and consider the data instance that does not belong to any cluster as an anomaly point. A series of experiments are conducted on a cloud computing environment that is deployed using OpenStack and experimental results show that the proposed anomaly detection framework is better than other anomaly detection methods that are designed for cloud computing environment in terms of precision, recall, false alarm rate, and runtime.

Acknowledgment

This research is partially supported by the National Natural Science Foundation of China (Grant No. 61272399), the Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20110191110038), and the Natural Science Foundation Project of CQ CSTC (Grant No. cstc2013jcyjA40059).

References

[1] AlZain, M. A., Soh, B., & Pardede, E. (2013). A survey on data security issues in cloud computing: From single to multi-clouds. *Journal of Software, 8(5)*, 1068-1078.

[2] Tan, W., Zhao, C., Wu, H., & Wang, X. (2014). An innovative encryption method for agriculture intelligent information system based on cloud computing platform. *Journal of Software, 9(1)*, 1-10.

[3] Grandison, T., Maximilien, E. M., Thorpe, S., & Alba, A. (2010). Towards a formal definition of a computing cloud. *Proceedings of the 6th World Congress on Services* (pp. 191-192).

[4] Arshad, J., Townend, P., Xu, J., & Jie, W. (2012). Cloud computing security: Opportunities and pitfalls. *International Journal of Grid and High Performance Computing, 4(1),* 52-66.

[5] Mazzariello, C., Bifulco, R., & Canonico, R. (2010). Integrating a network IDS into an open source cloud computing environment. *Proceedings of 2010 6th International Conference on Information Assurance and Security* (pp. 265-270).

[6] Modi, C. N., Patel, D. R., Patel, A., & Muttukrishnan, R. (2012). Bayesian classifier and snort based network intrusion detection system in cloud computing. *Proceedings of 2012 3rd International Conference on Computing, Communication and Networking Technologies* (pp. 1-7).

[7] Kumar, P., Shukla, S. S. P., Chauhan, D. S., *et al.* (2011). A novel approach for security in cloud computing using hidden markov model and clustering. *Proceedings of the 2011 World Congress on Information and Communication Technologies* (pp. 810-815).

[8] Kannan, A., Maguire, J. G. Q., Sharma, A., & Schoo, P. (2012). Genetic algorithm based feature selection algorithm for effective intrusion detection in cloud networks. *Proceedings of the 12th IEEE International Conference on Data Mining Workshops* (pp. 416-423).

[9] Wang, C., Talwar, V., Schwan, K., & Ranganathan, P. (2010). Online detection of utility cloud anomalies using metric distributions. *Proceedings of the 2010 IEEE/IFIP Network Operations and Management Symposium* (pp. 96-103).

[10] Gupta, M., Sharma, A. B., Chen, H., & Jiang, G. (2013). Context-aware time series anomaly detection for complex systems. *Proceedings of the SDM Workshop on Data Mining for Service and Maintenance*.

[11] Bhaduri, K., Das, K., & Matthews, B. L. (2011). Detecting abnormal machine characteristics in cloud infrastructures. *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops* (pp. 137-144).

[12] Qu, Q. (2012). Determination of weights for the ultimate cross efficiency: A use of principal component analysis technique. *Journal of Software, 7(10),* 2177-2181.

[13] Zhang, D., Zhao, Y., & Du, M. (2013). A new supervised dimensionality reduction algorithm using linear discriminant analysis and locality preserving projection. *WSEAS Transactions on Information Science*

and Applications, 10(4), 101-115.
[14] Li, P., Huang, J., Ye, L., Wang, Y., Li, Z., & Li, D. (2012). Directional fuzzy data association filter. *Journal of Software, 7(10),* 2286-2293.
[15] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys, 41(3),* 1-72.
[16] Corradi, A., Fanelli, M., & Foschini, L. (2014). VM consolidation: A real case based on OpenStack cloud. *Future Generation Computer Systems, 32(1),* 118-127.

Mingwei Lin received his B.S. degree in software engineering from Chongqing University, China, in July 2009. He has got his Ph.D. degree in computer science and technology at Chongqing University, China, in December 2014. His research interests include NAND flash memory, linux operating system, and cloud computing. He received the CSC-IBM Chinese Excellent Student Scholarship in 2012.

Shuyu Chen received his B.S., M.S., and Ph.D. degrees in computer software and theory from Chongqing University, China, in 1984, 1998, and 2001. From 1995 to 2005, he was with the College of Computer Science, Chongqing University. Since 2005, he has been with the School of Software Engineering, Chongqing University, where he is currently a professor. His current research interests include dependable computing, cloud computing, and Linux operating system. He has published more than 100 papers in international journals and conference proceedings.