A Learning-based Cost Management System for Cloud Computing

Bo Tang, Quan Ding, Prakash Manden, and Jin Ren

Abstract— **Cloud cost management is an emerging and challenging issue as increasing number of business are moving their on-demand computation workloads to the public cloud. Although the business model of "pay-as-you-use" is used by many public cloud providers, the customers still usually pay much more than what they are actually using. To better identify the cost optimization opportunities, in this paper, we introduce four metrics to measure the efficiency of provisioned cloud computing resources and services: resource utilization, instance utilization, cost utilization, and cost saving efficiency, and apply temporal learning algorithms to predict their future values and to detect anomalies. We design a cost management system which can automatically monitor the cloud environment and track the changes of these four efficiency in real time, based on which actionable advisories are provided to the cloud consumers for their cloud cost and performance optimization. Using the Amazon Web Service Cloud as the public cloud provider, our designed system can effectively manage the cloud environment that automatically fit the demands of cloud computing applications.**

I. Introduction

Cloud computing is gaining popularity with the potential to transform the business models and the information technology services. Public cloud computing service providers usually adopt the scheme of "pay-as-you-go" to allow the consumers only pay the resources they require [1], [2]. Since this pricing scheme is similar to how people pay for utilities like water and electricity, cloud computing service is also now considered as another new utility. In recent years, it has been seen that an increasing number of businesses move their on-demand workloads to the public cloud to reduce cost, enhance security, and improve overall system performance. Gartner Research reports that the overall public cloud services market will grow to total $246.8 billion in 2017 and $383.4 billion by 2020 [3].

Currently, three types of services are delivered in the cloud market [4], [5]: infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and software-as-a-service (SaaS). IaaS platforms offer several types of virtualized computing resources that are highly scalable and are easily adjusted on-premise. Taking the Amazon Web Service (AWS) cloud as an example, a wide selection of EC2 instance types are provided for elastic cloud computing with different combinations of CPU, memory, storage, and networking capacity. Since difference instances cost differently, the customers can purchase appropriate computing instances based on their demands, resulting in significant cost reduction. With this new pay as you use model, cost management is becoming a challenge for the customers, as they need to carefully predict their bushiness growths and manage the cloud environment. For example, an e-commerce customer may expect that there would be a high computation or networking burden during the Black Friday holiday, and thus schedule to upgrade their cloud computing instance prior to that day. This kind of pattern may be more complicated and vary from one customer to the other. Notice that, while an over-prediction may increase the cost of provisioned cloud resources, an under-prediction may lower the cloud computing capacity and thus hurt the cloud performance. To optimize the cloud cost and performance, a cost management tool is required to track and predict the changes of cloud computing resource demands and to automatically manage their cloud provisioning.

Moreover, security is still a challenging issue in public cloud [6], [7]. The exposure of identity and access management (IAM) keys may lead intrusions and malicious attacks [8]. Once the attackers are able to access the customer's cloud environment, they are able to create a large number of expensive cloud computing instances, causing financial harm. In this paper, we present a cost management system that monitors a cloud environment, tracks the application needs, and identifies opportunities for cost optimization and security enhancement using machine learning algorithms. More specifically, we first define the concept of cloud efficiency at three levels: resource utilization, instance utilization, and cost utilization, and then apply temporal learning algorithms to predict the demands of cloud computing resource in the future. Abnormal behaviors are further identified with a residual-based method.

The rest of this paper is organized as follows: We provide an overview of our learning-based cost management system in Section II, and introduce four efficiency metrics in Section III. In Section IV, we present regression models to predict future cloud resource demand and to detect anomalies. Conclusions and future works are given in Section V.

II. Overview of Learning-based Cost Management System

With the use of virtualization technologies, cloud computing is able to effectively manage distributed computing infrastructures as a whole. The resource utilization and flexibility can also be greatly improved, as it is possible to run multiple applications or operation systems in one single hardware. Many types of computation, storage, and networking resources and services are now delivered by the cloud computing providers to many independent consumers. Cloud consumers can customize their own cloud environment with different types of resources to fulfill the demands of

Bo Tang, Quan Ding, Prakash Manden, and Jin Ren are with FittedCloud, Inc. E-mail: {bo, quan, prakash, jin@fittedcloud.com}.

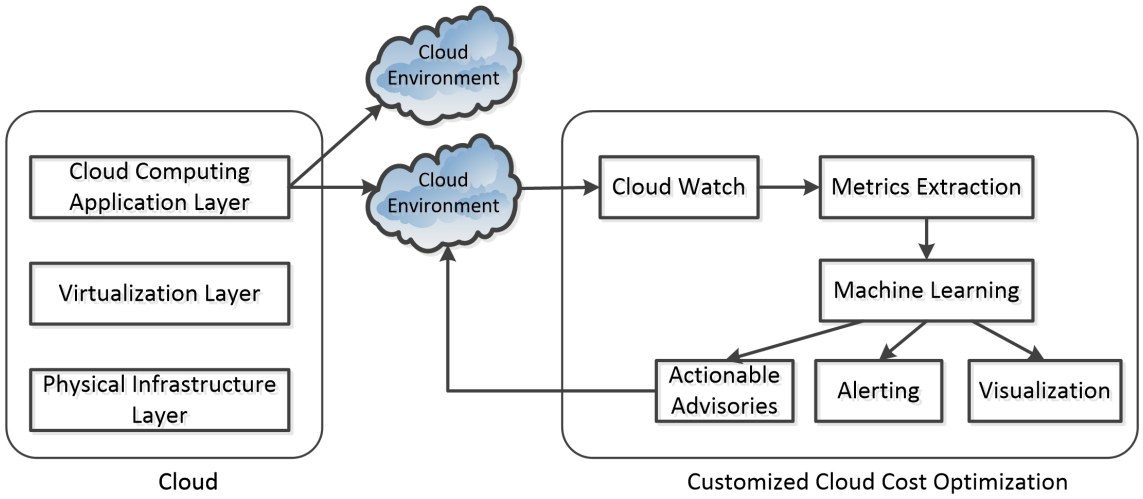

Fig. 1: Architecture of our learning-based cost optimization system

workloads. As a commodity, these different types on-demand cloud resources are usually charged differently by the hour. Taking AWS EC2 General Purpose Computing Family as an example, while the cheapest computing instance (t2.nano) charges less than 1 cent per hour, the most expensive one (m4.16xlarge) costs more than 3 dollars per hour. Considering a cloud environment for a middle business, hundreds of instances are provisioned 24 hours 7 days, and a fitted cloud can save millions of dollars every year.

Fig. 1 shows the architecture of our learning-based cost optimization system which can be customized for each consumer. For a given cloud environment, we use a Cloud Watch module to monitor the usage of each provisioned cloud computing resource and service in real time, as well as their cost charged by the cloud provider. Metric extraction module is used to extract useful metrics from the collected usage data streams, including resource utilization, instance utilization, cost efficiency, and cost saving efficiency. These four metrics are defined in detail in the following section. We use two utilization metrics to measure the demand of cloud computing resources and services and track their patterns over time, and use the efficiency metric to measure the total efficiency of investment in the cloud. All these four metrics can help us to identify all possible opportunities to adjust the on-demand provisioning for cost optimization. We then apply online learning algorithms to learn the patterns of cloud computing demand from the collected metric data and identify any significant changes from normal patterns. Since the collected metric data are time-series data, we use autoregressive (AR) models with polynomial coefficients in the system. An AR model is a stochastic process which assumes that the current value is a weighted sum of past values. By learning the weights for a group of historical data, one can use the AR model to predict future values.

For a typical on-demand cloud environment, consumers are allowed to adjust (upgrade or downgrade) the capacity. Since the consumers pay for the provisioned capacity instead of the usage, the learning-based models can help us to understand how the demands of applications change and identify fitted capacities for the future. We summarize all possible advisories that are actionable for the consumers. If the consumers accept the provided advisories, the corresponding action scripts will be automatically executed to change the capacity. Meanwhile, once a new metric arrives, we compare it with the predicted value following the learned pattern. If it is significantly different from the predicted value, we can consider it as an anomaly, indicating an abnormal behavior occurs, and alert the consumers through email or phone message. The visualization module enables the generation of various kinds of reports for all collected metrics, recommended actionable advisories, identified anomalies, cost saving, etc.

III. Defined Cloud Efficiency Metrics

In this section, we define four efficiency metrics: resource utilization, instance utilization, cost efficiency, and cost saving efficiency. While the first two utilization metrics measure the efficiency of a particular cloud computation resource or service, the cost efficiency measures the efficiency of total investment in the cloud environment. Their detailed definitions are given in the rest of this section.

A. Cloud Computing Resource Utilization

Nowadays, public cloud providers (e.g., AWS, Google, Microsoft, etc.) offer the cloud consumers the flexibility for a broad set of computing solutions in the form of virtual machines, or called computing instances. Each computing instance can be configured with different resource capacities depending on the demands of applications or workloads. The computing instance capacity varies in the combination of four basic resources: CPU, memory, storage, and networking, giving the cloud consumers flexibility to scale the resources to fit the demands of applications. While some applications are computing intensive, other applications are storage or networking intensive. Even one application might have different requirements of one particular resource during different time periods.

To track the demands of applications, we first define a metric of resource utilization from the collected data streams in our Cloud Watch module. The cloud resource utilization would be defined as the ratio of consumed resources to the provisioned resources in percentage. We use the variable X_c to denote the usage of one resource, and X_p to denote the provisioned resource. The formal definition of resource utilization in percentage is given as follows:

$$R = X_c/X_p \times 100 \tag{1}$$

Following the above definition, we have the following utilization definitions for four independent cloud computing resources of CPU, memory, storage, and networking:

$$\begin{aligned} R_{\text{CPU}} &= C_c/C_p \times 100 \\ R_{\text{Memory}} &= M_c/M_p \times 100 \\ R_{\text{Storage}} &= S_c/S_p \times 100 \\ R_{\text{Networking}} &= N_c/N_p \times 100 \end{aligned} \tag{2}$$

where the variables C, M, S, and N with a subscript of c denote the consumed CPU, memory, storage, and networking, respectively, and the variables with a subscript of p denote their provisioned resource capacities. We note that, in addition to these four computing resources, the above definition can also be applied to many other types of resources. The resource utilization indicates that how much resource out of the provisioned capacity is utilized. The understanding of resource utilization is very important to the cloud consumers, as it indicates the demands of particular computing resources in their cloud applications or workloads. Adjusting the provisioning to always fit the demands is a way to reduce the cost in the cloud.

B. Cloud Instance Utilization

Each cloud computing instance type is composed of multiple computing resources with different provisioned capacity. For example, there are seven T2 instance types in the general purpose family of AWS EC2, including t2.nano (1 CPU and 0.5GB memory), t2.micro (1 CPU and 1GB memory), t2.small (1 CPU and 2GB memory), t2.medium (2 CPU and 4GB memory), t2.large (2 CPU and 8GB memory), t2.xlarge (4 CPU and 16GB memory), and t2.2xlarge (8 CPU and 32GB memory).

While Eq. (1) defines the utilization of a particular computing resource, we further define the utilization of individual cloud computing instances to measure how it is utilized as a whole. Given a set of utilization metrics of resources belonging to the instance, e.g., $\mathcal{S} = \{\text{CPU}, \text{Memory}, \text{Storage}, \text{Networking}\}$, we define its instance utilization I as follows:

$$I = \underset{i \in \mathcal{S}}{max} R_i \tag{3}$$

where R_i denotes the resource utilization for i-th resource. The maximum function is used in Eq. (3) due to the fact that the instance is fully utilized if any of its resources is fully utilized. It is noteworthy that one instance has one single instance utilization metric and multiple resource utilization metrics. The instance utilization metric can be considered as the summary of multiple resource utilization metrics.

C. Cloud Cost Efficiency

A consumer's cloud environment usually has multiple computing instance with different types depending on the demands of applications. As mentioned earlier, the cloud consumers pay differently for different types of instances according to their provisioned capacities. The total cost efficiency of a cloud environment is very important to consumers since it indicates whether the current cloud environment is over-provisioned or under-provisioned.

Suppose that there are N cloud computing instances in a given cloud environment, and we denote the unit price of the i-th instance as p_i and its instance utilization as I_i. Then, the cloud cost efficiency for this cloud environment is defined as follows:

$$\beta = \frac{\sum_{i=1}^{N} p_i I_i}{\sum_{i=1}^{N} p_i} \times 100 \tag{4}$$

where the denominator is the total cost that the consumer spends for the cloud environment, and the numerator is the sum of all instance prices weighted by the utilization. To illustrate the detailed calculation, consider the cloud environment which contains three on-demand AWS EC2 computing instances: t2.xlarge ($\$0.188$ per hour), m4.4xlarge ($\$0.8$ per hour), and c4.large (($\$0.1$ per hour)), which have the instance utilization of 80%, 50%, and 60%, respectively. Using Eq. (4), it can be easily shown that the cost efficiency of this cloud environment is 56.10%.

It can be easily shown that the cloud cost efficiency ranges from 0 to 100. A cloud cost efficiency of 100 indicates that the cost is fully utilized, but it also indicates the provisioned cloud services may not satisfy the consumer's demands and need to be upgraded, because, for example, the business growths. A very low cloud cost efficiency, e.g., close to 0, indicates that the cost is under utilized, and it also indicates that the provisioned cloud services need to optimize to reduce the cost.

D. Cloud Cost Saving Efficiency

While cloud cost efficiency measures overall utilization of the cost of a consumer's cloud environment, it does not reflect the cost a consumer could have potentially saved. To see this, let's look at the following two examples.

Example 1: consumer has only 1 instance t2.micro (1 CPU and 1GB memory), and consumed CPU utilization is 100% and memory utilization is 0.5GB (50%). Therefore, instance utilization is the maximum of resource utilization which is CPU utilization of 100%. Since there is only 1 instance, according to eq. (4), cost efficiency is 100. However, consumer could have switched to t2.nano (1 CPU and 0.5GB memory) and still covered the consumed CPU and memory utilization. This reduces the cost from $0.012 per

hour (t2.micro) to $0.0059 per hour (t2.nano), and therefore saves $0.0061 per hour.

Example 2: consumer has only 1 instance t2.nano (1 CPU and 0.5GB memory), and consumed CPU utilization is 20% and memory utilization is 0.1GB (20%). According to eq. (4), the cost efficiency is 20%. But t2.nano is already the smallest instance. Even though the instance utilization is only 20%, consumer cannot switch to a cheaper instance so that there is no potential cost saving.

These two toy examples show that 1) a cloud environment with cost efficiency of 100 may still have potential cost savings, and 2) a cloud environment with cost efficiency of less than 100 may not have potential cost savings. To better quantify potential cost savings in a cloud environment, we introduce cost saving efficiency defined as:

$$\gamma = \frac{\sum_{i=1}^{N} p_i^{\min}}{\sum_{i=1}^{N} p_i} \times 100 \tag{5}$$

where p_i is the unit price of the i-th instance and $p_i^{\min}$ is the unit price of the smallest instance to replace the i-th instance while still covering the consumption.

Cost saving efficiency also ranges from 0 to 100. A cost saving efficiency of 100 indicates that consumer is already using the most appropriate instances and there are no potential cost savings. A small cost saving efficiency implies that consumer is using larger instances than necessary, and could reduce cost by switching to smaller instances.

IV. Temporal Data Analysis

For our four defined metrics, the resource and instance utilization metrics measure the demands of a particular cloud computing resource and instance, and the cost efficiency measures the overall efficiency of a customer's environment. We model the patterns of these four metrics using regression analysis and predict the future demands of applications under normal situation. The prediction of these four utilization metrics can help the cloud consumers to resize the capacity if it is necessary. For example, the cloud consumers can decrease the capacities if the future cloud environment is over-provisioned, thus reducing the cost. Meanwhile, it is possible that the consumer's identity and access management (IAM) keys get exposed, and the attackers use these keys to access the cloud environment and create a large number of computing instances, causing financial harm. We also deploy anomaly detection methods to detect any potential abnormal behaviors.

A. Autoregressive Model for Prediction

The extracted metrics are time-series data denoted by $\{X[0], X[1], \cdots, X[n-1], X[n], \cdots\}$, where $X[n]$ is one efficiency metric at the n-th time step. The AR(p) model [9], [10] assumes that the current value is linearly related to the previous values, i.e.,

$$X[n] = \sum_{i=1}^{p} \alpha_i X[n-i] + w[n] \tag{6}$$

where $w[n]$ is the noise following a Gaussian distribution, and $\alpha_1, \alpha_2, \cdots, \alpha_p$ are model coefficients which are unknown and need to be estimated from historical data.

To model more complicated patterns, we use a generalized AR(p, l) model with polynomial coefficients, which is given as follows:

$$X[n] = \sum_{k=1}^{l} \sum_{i=1}^{p} \alpha_{i,k} X^k[n-i] + w[n] \tag{7}$$

It can be shown that, using this model, the current value is non-linear with the previous values, which makes it a more powerful inference method. Also the model defined in Eq. (6) can be considered as a special case of this model.

Using least-squares estimation methods, we can easily estimate the coefficients $\alpha_{i,k}$, $i = 1, 2, \cdots, p$ and $k = 1, 2, \cdots, l$, as $\hat{\alpha}_{i,k}$. Then the value at the next step can be predicted as follows, when $\{X[n-p+1], X[n-p+2], \cdots, X[n-1], X[n]\}$ are available:

$$\hat{X}[n+1] = \sum_{k=1}^{l} \sum_{i=1}^{p} \hat{\alpha}_{i,k} X[n-i+1] \tag{8}$$

B. Anomaly Detection

Anomaly is one that does not conform to the expected patterns [11]–[13], and its detection is very important for the cloud consumers as it captures valuable information regarding to either the changes of demands or cloud security.

Now that the future metric value $\hat{X}[n+1]$ at the $(n+1)$-th time step is estimated by Eq. (8), one can easily detect if an abnormal patter occurs or not, when the new metric $X[n+1]$ ah the $(n+1)$-th time step is available, by comparing their distance with a threshold:

$$||\hat{X}[n+1] - X[n+1]|| \geq \tau \tag{9}$$

where $||\cdot||$ is the norm function, and τ is the threshold. We call this method a residual-based method, as it considers the residual between the real value and the predicted value. Since the predicted value is assumed to follow the normal pattern, if the real value is far away the predicted value, one can consider it an unexpected pattern or an anomaly. The choose of threshold τ is related to the false alarm rate. A larger threshold leads to a lower false alarm rate, and vice versa.

We note that Eq. (9) is not restricted to the AR(p, l) regression model we use in Eq. (7), and it can be incorporated into any other regression models which are able to predict $\hat{X}[n+1]$.

V. System Implementation

We implement our learning-based cost optimization system in the AWS public cloud. The Amazon CloudWatch is used to automatically monitor cloud resources and applications and collect the data such as resource type, performance, price, etc. The resource performance of each instance is reported every 15 minutes. Instead of sending the raw data streams to the machine learning module, the metric extraction module first extracts four defined efficiency metrics. When

sufficient training data are collected, the machine learning module starts to train our generalized AR models to model the patterns of these four metrics and predict their future values. These predictions are further compared with the current provisioned capacity to check if the current capacity is over-provisioned or under-provisioned. If so, all other available configurations to which the consumers are able to scale are examined, and the best fit one is selected and recommended to the consumers, so that the cloud cost and performance are optimized. The machine learning module is allowed to be re-trained, so that it can keep tracking the change of patterns. Fig. 2 shows the flowchart of our machine learning module.

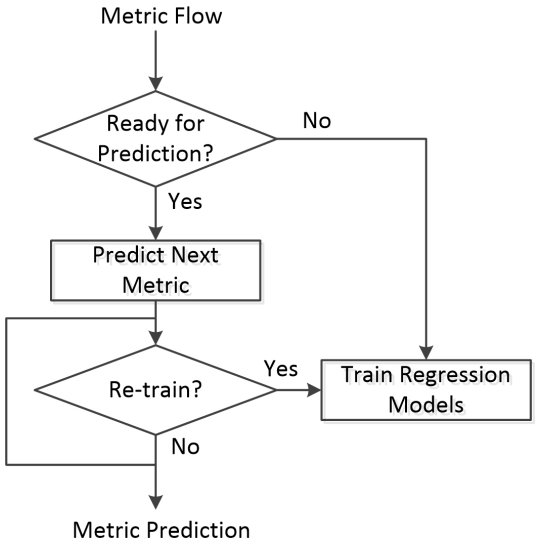

Fig. 2: Flowchart of Training Regression Model

Meanwhile, anomaly detection is performed using a residue-based method, when a new metric arrives. Specifically, we compare the difference between the new observation and its predicted value with a threshold. Once an anomaly is detected by Eq. (9), a message is automatically generated and sent to the consumers for notification. Various reports regarding efficiency metrics, predictions, potential anomalies, and cost, can be generated for visualization.

VI. Experimental Results

In this section, we conduct experiments to show the effectiveness of the proposed cost efficiency metrics: cost efficiency and cost save efficiency in our cost management system using AWS Cloud. The experiment results are shown in Fig. 3. The experiment started on Day 1 with three t2.nano instances (the lowest configuration in EC2 T2 family, see Table I for details), each of which has low baseline usage. In Fig. 3, we can see Cloud Cost Efficiency is low while Cloud Cost Saving Efficiency is 100. This is because t2.nano is already the smallest instance and we cannot switch to cheaper instances to save cost. We next increased usage of these three instances on Day 3, and we can see Cloud Cost Efficiency increases (sometimes hitting 100) and Cloud Cost Saving Efficiency is still 100. When Cloud Cost Efficiency is close to 100, it suggest that it is good time for the consumers to switch to more powerful instances. On Day 6, we upgraded these three instances to t2.micro, t2.medium, m4.large respectively, with the same baseline usage, and both Cloud Cost Efficiency and Cost Saving Efficiency are decreased. On Day 8, we increased the usage of t2.medium instance, and we can see both metrics increased. Increased usage in m4.large on Day 11 leads a growth of both metrics again (close to 100). Again, a high Cost Efficiency and Cost Saving Efficiency indicates that we almost reached full capacity of the instances.

TABLE I: T2 instance family in Amazon EC2

Instance Type	vCPUs	Memory (GB)
t2.nano	1	0.5
t2.micro	1	1
t2.small	1	2
t2.medium	2	4
t2.large	2	8
t2.xlarge	4	16
t2.2xlarge	8	32

On Day 13, we considered a malicious attack scenario in which three m4.10xlarge instances were created, but they had no usage. We can see both metrics dropped to close to 0. This implies something is wrong because the efficiency is so low. On Day 17, three malicious m4.10xlarge instances were found and deleted. We can see both metrics returned to close to 100. Increased usage in 3 instances on Day 20 leads to a 100 Cloud Cost Efficiency, implying we may need to upgrade the instances. Both metrics decreased to around 80, after we upgraded 3 instances to t2.small, t2.large, and m4.xlarge, respectively.

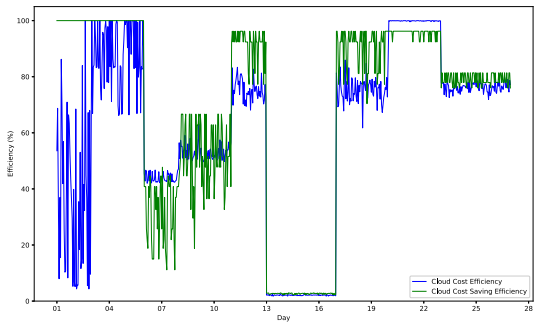

Fig. 3: Experiment results of two metrics of Cost Efficiency and Cost Saving Efficiency with a sequence of Cloud activities.

VII. Conclusion

In this paper, we presented a cloud cost management system which is able to automatically monitor customers' public cloud environment and track the changes of cloud resource efficiency and cost saving. Four efficiency metrics

are introduced to measure the efficiency of a cloud environment, including resource utilization, instance utilization, cost utilization, and cost saving efficiency. With these collected metric data, we apply temporal learning algorithms to predict their future values and to detect anomalies. The proposed cloud cost management system can automatically monitor the cloud environment in real time and identify potential opportunities to optimize the cloud cost and performance. Using the Amazon Web Service Cloud as the public cloud provider, we also show that the proposed system can effectively manage the cloud environment such that the demands of cloud computing applications are always automatically fitted with a minimum cost.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computingthe business perspective," *Decision support systems*, vol. 51, no. 1, pp. 176–189, 2011.

[3] "Gartner says worldwide public cloud services market to grow 18 percent in 2017." `http://www.gartner.com/newsroom/id/3616417`. Accessed: 2017-07-06.

[4] P. Mell, T. Grance, *et al.*, "The nist definition of cloud computing," 2011.

[5] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of ?big data? on cloud computing: Review and open research issues," *Information Systems*, vol. 47, pp. 98–115, 2015.

[6] B. R. Kandukuri, A. Rakshit, *et al.*, "Cloud security issues," in *Services Computing, 2009. SCC'09. IEEE International Conference on*, pp. 517–520, IEEE, 2009.

[7] S. A. Almulla and C. Y. Yeun, "Cloud computing security management," in *Engineering Systems Management and Its Applications (ICESMA), 2010 Second International Conference on*, pp. 1–7, IEEE, 2010.

[8] T. Mather, S. Kumaraswamy, and S. Latif, *Cloud security and privacy: an enterprise perspective on risks and compliance*. " O'Reilly Media, Inc.", 2009.

[9] P. M. Robinson, "Statistical inference for a random coefficient autoregressive model," *Scandinavian Journal of Statistics*, pp. 163–168, 1978.

[10] B. Tang, H. He, and S. Kay, "Adaptive signal detection and parameter estimation in unknown colored gaussian noise," *arXiv preprint arXiv:1607.08259*, 2016.

[11] R. Gnanadesikan and J. R. Kettenring, "Robust estimates, residuals, and outlier detection with multiresponse data," *Biometrics*, pp. 81–124, 1972.

[12] B. Tang and H. He, "A local density-based approach for outlier detection," *Neurocomputing*, vol. 241, pp. 171–180, 2017.

[13] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*, vol. 589. John wiley & sons, 2005.