A workload prediction model in the multi-cloud to reduce SLA violation

Junquan Wei

BSc Industrial Engineering & Innovation Sciences – TU/e 2014
Student identity number 0927403

in partial fulfillment of the requirements for the degree of

Master of Science
in Innovation Management

Supervisors:
dr. C.M. Chituc, TU/e, IS
dr. R.J. de Almeida, TU/e, IS
TUE. School of Industrial Engineering.

Series Master Theses Innovation Management

Subject headings: cloud computing, multi-cloud, service level agreement (SLA), SLA violation, SLA-related costs, and workload prediction
Preface

This paper is a result of the master thesis project, carried out in Eindhoven University of Technology in Eindhoven, the Netherlands. After doing bachelor in China, I decide to continue my studies in innovation management in the Netherlands. And this master thesis finalizes my 2 years study in TU/e, at the faculty of Industrial Engineering & Innovation Sciences.

This dissertation is ultimately based on the experimental data of the Grid Workloads Archive. None of the text of the dissertation is taken directly from previously published or collaborative articles.

There are problems and challenges met during the process of completing my master thesis, but they are solved with the help of my mentors. And I also acquired a lot of new knowledge and experiences. So here, I really would like to thank my supervisors, C. M. Chituc and R.J. de Almeida, who have excellently supported me form the beginning to the end of the whole master thesis. I admire their professional attitude, knowledge, and friendliness along the complete process of conducting master thesis.

It is acknowledged that this master thesis is completed in collaboration with supervisor, C. M. Chituc and R.J. de Almeida, who have made many contributions to the work.

And thanks those people who have provided help with me during the process of finish the master thesis.

Junquan Wei

Eindhoven, August 2016
Abstract

Reducing SLA violations is of significant importance for cloud service providers to leverage customer satisfaction and lower SLA-related costs in the fiercely competitive multi-cloud environment. An effective workload prediction model can assist to optimize resource allocation, ensure QoS, and schedule etc, thereby decreasing the possibility of SLA violations. In this master thesis, an AR(MA)X-LLM model (i.e., AR(MA)X model using log-likelihood maximization) is proposed, which is able to estimate the truncated values accurately so that the predictive results can conform to the reality, and delivering more realistic value to cloud service providers. Then whether the prediction result of AR(MA)X-LLM model can reduce the SLA violations and costs is verified using SLA violation measurements and one of the retrieved cost models. The result shows that the proposed AR(MA)X-LLM model can significantly reduce SLA violations. Unfortunately, it is not necessary to lower the SLA-related costs.
Management summary

It is important for cloud service providers in the multi-cloud environment to accurately predict the upcoming workload to reduce SLA violation, and thereby increasing customer satisfaction and decreasing penalties. There are a variety of prediction models have been collected in section 3.4, which can be used to predict the future workload. Besides, criteria of comparing different models and grades of each criterion are described so that cloud providers can decide which model should be adopted under certain circumstance. Those models are helpful, unfortunately, they are either unable to predict truncated values or have poor prediction accuracy. As a result, we develop an AR(MA)X-LLM model (AR(MA)X-LLM using log-likelihood maximization) that not only has good prediction accuracy, but also enables to predict truncated values. The specific description of the AR(MA)X-LLM model is presented in section 5.4. And the pseudo code of how to implement the proposed method is described in detail in section 5.5.2, so that the cloud providers can easily implement this model. On the purpose to reduce SLA violation and SLA-related costs, measurement of SLA violations and cost models are clearly illustrated in section 6. And totally there are three cost models developed, so that the cloud service providers can select an appropriate cost model in a specific condition. The result shows that AR(MA)X-LLM model can significantly reduce SLA violations. Unfortunately, it is not necessary to reduce SLA-related costs. But cloud service provides can count on price of renting processors and SLA penalty to compute if the cost will be decreased.
Table of Content

1. Introduction ..................................................................................................................... 1

2. Research project ................................................................................................................ 2
   2.1 Background and significance .................................................................................. 2
   2.2 Research objective and structure ........................................................................ 4
   2.3 Project approach ..................................................................................................... 5
   2.4 Project scope ........................................................................................................... 7
   2.5 Deliverables ............................................................................................................. 7
   2.6 Contribution to the literature ................................................................................... 8

3. Literature review ............................................................................................................... 9
   3.1 Cloud computing ..................................................................................................... 10
   3.2 Multi-cloud ............................................................................................................ 11
   3.3 Service level agreement and SLA violation ........................................................... 12
   3.4 Workload prediction methods in the cloud environment ........................................... 14
      3.4.1 Quadratic response surface model (QRSM) ..................................................... 15
      3.4.2 Exponential smoothing (ES) ............................................................................... 16
      3.4.3 Autoregression model (AR) ................................................................................ 17
      3.4.4 Moving Average (MA) ....................................................................................... 17
      3.4.5 Second order autoregressive moving average method ....................................... 18
      3.4.6 Autoregressive moving average (ARMA) .......................................................... 18
      3.4.7 Autoregression integrated moving average (ARIMA) ........................................ 18
      3.4.8 Autoregressive model with exogenous variables (ARX) .................................... 19
      3.4.9 Autoregressive moving average model with exogenous variables (ARMAX) ... 20
      3.4.10 ARIX and ARIMAX model .............................................................................. 20
      3.4.11 A Multiple Time Series Approach based on Hidden Markov Modeling (HMM) 21
      3.4.12 Predicating workload using linear regression model (LRM) ......................... 22
      3.4.13 SMM (Statistical Metric Modeling) .................................................................... 23
      3.4.14 Neural network model to predicate the workload (NN) .................................. 24
      3.4.15 kNN (k nearest neighbours) ............................................................................ 25
      3.4.16 Comparison criteria of different methods........................................................ 26
4. Problem deduction and definition ......................................................... 28
   4.1 Problem deduction ........................................................................ 28
   4.2 Problem definition ........................................................................ 29

5. Method ................................................................................................. 30
   5.1 Data collection and preprocessing .................................................. 30
   5.2 Historical data analysis ................................................................... 31
   5.3 Primary design ............................................................................... 33
   5.4 AR(MA)X-LLM ............................................................................. 36
      5.4.1 ARX-LLM with error term independent on previous error terms ...... 36
      5.4.2 ARMAX-LLM with error term dependent on previous error terms ... 38
   5.5 Implementation and result .............................................................. 40
      5.5.1 Training and testing dataset ..................................................... 41
      5.5.2 Implementation ...................................................................... 41
      5.5.3 Evaluation of the prediction result .......................................... 43
      5.5.4 Result .................................................................................... 43

6. SLA violations and costs .................................................................... 46
   6.1 SLA violations measurement ......................................................... 46
   6.2 Costs measurement ........................................................................ 46
      6.2.1 SLA violation and resource cost model .................................... 47
      6.2.2 VRC cost model .................................................................... 48
      6.2.3 DSI (delivery, storage, and immigration) cost model .................. 49
   6.3 Result of SLA violations and costs .................................................. 51

7. Discussion ............................................................................................ 52

8. Conclusion, limitation and recommendation ....................................... 55
   8.1 Conclusion ..................................................................................... 55
   8.2 Limitation and recommendation .................................................... 56

Bibliography ........................................................................................... 58

Appendix A ............................................................................................ 65

Appendix B ............................................................................................ 65
1. Introduction

Cloud computing is recently becoming increasingly popular and has attracted significant attention by both cloud providers and users because of its huge advantages to provide economical, scalable, elastic and seemingly infinite computing resources, as well as the increasing demands for low-cost, high performance, and energy-efficient computing [1], [24], [28], [65]. While cloud computing provides these advantages to clients, it also creates a number of challenges encountered by the cloud service providers who aim at creating successful businesses [42]. More specifically, quality of service and cost management are two major determinants that influence the success of cloud-based services. As a consequence, cloud service providers should ensure the quality of service and reduce service costs to reach their business goals. Quality of service is typically specified in the service level agreement (SLA), which is a contract between cloud provider and consumer, and it also consists of guarantees, actions and remedies that should be taken for all cases of violations [3], [50]. Regarding cost management, many works have been carried out, which mainly focus on cost optimization, resource allocation, task scheduling optimization, and prediction of the future workload characteristics and volume, etc. [30], [37], [42]. Catering to the SLAs while maintaining costs low is, however, challenging for cloud providers due primarily to the varying number of incoming customers to the service system[62]. Consequently prediction on the future workload plays an important role in cloud computing. Not only does it help quickly plan and provision computing resources, but also decreases costs incurred in overprovisioning of resources and SLA violations, which typically result in penalties of cloud service providers [30], [26]. Besides, initializing additional virtual resources in the cloud is not instantaneous, which means that cloud-hosting platforms introduce several minutes delay in the hardware resource allocation. A precise workload prediction model can aid to allocate resource in advance to avoid inconveniences of end users [8]. It is consequently important for cloud service providers to accurately estimate the upcoming workload in the future to ensure quality of service and reduce costs.

In this research paper, an ARX-based prediction model is presented to predict the future workload. The experiment result shows that the proposed model are superior to
other methods such as ARMA, MEAN and the forth, and it actually helps the cloud provides lower the percentage of SLA violation and the SLA violation costs.

The rest of the paper is organized as follows: section 2 discusses research project including background and significance of the research project, project approach and scope, deliverables, and contribution to the literature. Section 3 indicates the result of literature while section 4 describes problem deduction and definition. The proposed method is discussed in section 5, which is followed by SLA violation and costs in section 6. Then discussion of the whole work is presented in section 7 and conclusions, limitations and recommendations of the work are illustrated in section 8.

2. Research project

This chapter describes the research method to conduct the research performed in this master thesis. Section 2.1 describes the background and significance of this research project while section 2.2 illustrates the research objective and structure. In section 2.3, project approach is presented, and project scope and deliverables are indicated in section 2.4 and 2.5 respectively. Section 2.6 ultimately depicts the contribution to the literature of this work.

2.1 Background and significance

In recent years, cloud computing has become very popular and been accepted by enterprise users and individual users since it can provide economical, scalable, and elastic access to computing resources over Internet[1]. According to Xu [61], the global cloud computing market is anticipated to grow at a 30% CAGR reaching $270 billion in 2020. The compound annual growth rate (CAGR) is the mean annual growth rate of an investment over a specified period of time longer than one year. A large number of firms have entered the cloud market as cloud service providers to compete for large profits (e.g., Amazon, Google, Microsoft, Rackspace, and GoGrid etc.), and the number of providers is still increasing at a rapid speed[20][37][38]. This results in a competitive environment of cloud computing which is referred to as multi-user and multi-provider cloud market. On one hand, different cloud providers compete against each other for both existing and new cloud consumers. On the other hand, cloud providers probably cooperate with each other to
improve their final business profits[58]. These cloud providers offer a variety of pricing schemes, different kinds of instances, or even different value-added features to differ from their competitors, and thereby attracting customers and making profits[21][37]. This emerging multi-cloud environment enables cloud users to optimize performance, availability, costs, lower risks and improve quality of services, and motivates SaaS providers to reduce costs through leasing resources from IaaS providers instead of operating their own data centers[20][47]. However, multi-cloud also makes problems occurred in the cloud computing more challenging and complex. Under the multi-cloud environment and because of the prosperity of cloud computing, countless SLAs are carried out between cloud provider and users. And inevitably, SLAs violations will occur due to limitations on the availability of infrastructure resources[42]. Once cloud providers violate the agreed SLAs, they are responsible for bearing the penalties, which lead to the economical loss of cloud providers. As a consequence, reducing the possibility of SLAs violations is quite important for cloud providers, not only because SLAs violation prevention reduces penalty, but also because it enhances customer satisfaction.

Prediction on the future workload plays an important role in assisting cloud service providers achieve their commercial goals. Cloud providers usually need to provide high assurance in terms of Quality of Service (QoS) metrics such as high throughput, response times, and service availability to their consumers. Service providers will lose their user base and revenues without such assurances, which are typically specified in the service level agreements (SLAs) between the cloud provider and users[51]. Catering to the SLAs while maintaining costs low is, however, challenging for cloud providers due primarily to the varying number of incoming customers to the service system. This is mainly because upcoming workload prediction can help cloud providers optimize resource allocation, scheduling, ensuring QoS, and provisioning, as well as decrease the chances of violating SLAs[24][26][46]. As a consequence, predicating the upcoming workload in the future period is of crucial importance for cloud providers to control QoS and manage costs, which are two major factors that determine the business success of a cloud provider[42].

Given the importance of both workload prediction and SLA violations in the current multi-cloud environment, it is meaningful to take further research on predicating upcoming workload and reducing SLAs violations.
2.2 Research objective and structure

This master thesis aims at reducing SLA violations and SLA-related cost under the cloud environment through proposing an accurate workload prediction model.

In the multi-cloud environment, cloud consumers have many choices in terms of selecting an appropriate cloud service provider to acquire the intended services. And customer satisfaction is an influential factor that determines the customers’ choices. It is subsequently vital for cloud providers to enhance the customer satisfaction to keep and acquire customers. However, violating SLAs by cloud providers not only decreases customer satisfaction, but also produces penalties[35][36]. Maintaining the possibility of SLA violations low is therefore essential for cloud providers. Unfortunately, fulfilling all the service level objectives (SLOs (i.e., preventing SLAs violations) is typically impossible or very costly due to limitations and costs of resource provisioning. For instance, cloud providers may choose to violate a SLA to reduce resource cost or release more resources from other cloud provider to avoid violating SLAs, etc. And for the cloud providers, profit is always an important goal. Reducing SLA violations and SLA-related costs is therefore becoming quite challenging but important in the current multi-cloud environment. Given the compelling power of the prediction of the upcoming workload and its influence on SLA violation and costs as explained above, this research design aims at solving the below problem:

**How to reduce SLA violations and SLA-related costs of cloud providers via accurate workload prediction in the multi-cloud environment?**

In order to elaborate on the research objective, a structured set of search sub questions is required as key elements to gradually achieve research purpose. As a result, a couple of research sub questions are formulated as following:

1. What is the concept of multi-cloud environment?
2. What is the definition of SLA and its correlation with multi-cloud?
3. How are SLA violations generated?
4. What is the influence of SLA violations on cloud providers?
5. How to decrease SLA violations?
6. Why accurate workload prediction is important for reducing SLA violations and SLA-related costs?
7. What are the existing workload prediction methods in the multi-cloud?

These seven sub questions elaborate on the research problem sequentially and they will scientifically approach the research objective.

2.3 Project approach

The intended project is a science-based design which primarily aims at designing an artifact, thereby applying scientific/design knowledge. As a consequence, it is determined to use regular cycle plus reflective cycle (Figure 1), which is based on the regular cycle from van Strien[59], as the design method to perform the project. The regulative cycle is normative in the sense that the development of a design or plan is guided by an objective derived from the problem under consideration, which is aligned with the case in this master thesis.

![Regulative cycle model](image.png)

*Figure 1: Regulative cycle model*

The first phase of the regular cycle is the identification of a problem which is further diagnosed and analyzed in the second stage. In the next phase a design is developed in order to solve the identified problem. We also reflect on the design to see if there are some potential improvements. In the intervention phase the design is implemented in practice, which is evaluated in the last phase in order to check whether the problem has been solved. This phase may lead to new problems which means that the cycle starts again. Afterwards, the whole process will be analyzed again, which will then be reflected with taking other inputs into account. Ultimately, the whole process is
documented. Reflection on the regulative cycle will be continually conducted until an ultimate solution is agreed.

During the process of acquiring a satisfying design that can solve the intended problem, we primarily investigated in the areas of interest (i.e., research domains of mentors). This means that we mainly researched in the domains of cloud computing, multi-cloud, service level agreement, and SLA violation etc. And it is found that violating SLA in the multi-cloud environment can result in lots of negative consequences. And the initial problem is defined as how to reduce SLA violations in the multi-cloud. Then diagnosis and analysis of initial problem are performed. It is derived from the theory of literatures that an effective workload prediction model can help to reduce SLA violations because of resource optimization, resource automatic scaling, QoS assurance etc. As a result, the primary design aims at finding a workload prediction model that can precisely estimate the future workload of the experimental data. And then the design is implemented in MATLAB software to get the result (i.e., intervention phase). After this phase, the experiment result is evaluated by both student and supervisors. It is discovered that the predicted output can be negative, which is not aligned with the reality after reflecting on the above five stages. Hence we redefine the problem as how to accurately predict the truncated future values in the multi-cloud to reduce SLA violations. The reason why previous prediction model can generate negative output is because the coefficients and error term of the prediction model can be negative (diagnosis and analysis). Subsequently, we set the constraints of model parameters so that the estimated values will be positive. In order to get the constrained parameters that will keep the model goodness, log-likelihood maximization is utilized. And AR(MA)X-LLM (AR(MA)X using log-likelihood maximization) is developed (redesign). The redesign is then implemented and the results are acquired. It is found that the AR(MA)X-LLM have a good prediction accuracy and the produced outputs are non-negative, which means that this model can combat the problem (evaluation phase). Then the complete process from problem definition to evaluation is analyzed to find potential mistakes. And it is eventually documented.
2.4 Project scope

The intended research design is an experimental design which aims at answering the research question. The experiment requires to collect the data of workload of service cloud providers and then test the research hypothesis using the collected data and proposed method. And thereby the scope of selection of sample concentrates on the cloud service providers like Amazon, Microsoft, and Google and the forth. A variety of data sources are acquired through searching online, retrieving correlated literatures. Specifically, Parallel Workload Archives contains a repository of information in terms of the workload on parallel machines. This data source has two parts: raw workload logs from various machines around the world, and workload models. Another data source is CSIRO ATNF data archive, which mainly includes radio astronomy data taken with the Australia Telescope Compact Array (ATCA), Parkes Radio Telescope, Mopra Radio Telescope and Long Baseline Array (LBA). Worldwide LHC Computing Gird (WLCG) is also found to be a data source, which distributes and analyzes the data generated by the Large Hadron Collider (LHC). The last available data source is the Grid Workload Archive provided by TU/d, which offers workload traces from grid and cloud environment to researchers and to practitioners alike. The final experimental data is selected from above four data sources. Since this research focuses on predicting the workload, only the data source that can provider clear and specific measurement of workload will be adopted. The intended study is a meta-analysis, which will thoroughly examine a number of valid studies on a topic and mathematically combine the result using accepted statistical methodology to report the results. In terms of the output, this project will focus on developing a new workload prediction model in the multi-cloud environment so that is can be utilized by cloud providers to estimate the upcoming workload.

2.5 Deliverables

The deliverables of this master thesis are listed as following:

(1) A summary of the existing workload prediction models in the multi-cloud.
(2) The detailed principles of workload prediction’s effects on reducing SLA violation and SLA-related costs.
(3) A novelly feasible and accurate workload prediction model which can be used by
cloud providers to perform the workload prediction.

(4) The functions that can be used to calculate SLA violation and SLA-related costs.

(5) A master thesis.

2.6 Contribution to the literature

SLA attributes, as the core part of SLA, are properties of a service object; each parameter has a name, type and unit. Every parameter is typically associated with high/low watermarks so that customer, provider, or a designated third party is able to appraise the retrieved metrics whether they meet/surpass/fall below defined service level objectives[31]. As a result, each parameter and its permitted range are defined in SLA. However, different levels of SLA attributes are not clearly mapped in the current literature.

The first contribution of this master thesis is to provide a detailed framework, which comprises different levels and categories of SLA attributes that can be defined in the service level agreement, and the mappings of these SLA attributes. The framework is illustrated in Figure 2.

The second contribution to literature of this master thesis is developing the criterion of comparing the goodness of a variety of prediction models that can be applied in the multi-cloud environment. Algorithms, pros and cons, and limitations of these forecasting models are primarily described in detail so that they can save time for researchers when referring to the prediction models. And these criteria are derived from the advantages, disadvantages, and limitation of the retrieved prediction models, and they can be used to select an appropriate estimation model under certain circumstances.

The third and main contribution of this work to literature is to develop the method that enables to reduce SLA violation using a time-series workload prediction method (i.e., AR(MA)X-LLM model), which is able to generate the predicted results that conform to the reality. Current literatures either focus on developing an accurate prediction model or dedicate to developing modes to reduce costs. We propose a method that can obtain SLA violation and SLA-related costs after using a time-series prediction model, so that can verify whether the SLA violation and cost will be decreased.

The last but not least literature contribution developing different SLA-related cost
models that can be utilized given different cost parameters. Even though under the multi-cloud environment, not all information can be collected due to cost, interests of cloud service providers, etc. This means that some cost parameters can be missed and thereby, different cost models comprising different cost parameters are necessary to compute the cost under a certain circumstance.

![Service Measurement Index]

**Figure 2: Categories and Attributes of the Service Measurement Index**

3. Literature review

This chapter discusses headlines from literature that are significant for this research and its research objectives. This research purpose is to explore a workload prediction model to accurately estimate the upcoming workload in the cloud environment so that SLA violations and costs can be reduced. As a result, important concepts such as cloud computing and multi-cloud, SLA and SLA violations, and a variety of existing
workload prediction methods are introduced. Meanwhile, a list of creation to compare the goodness of different prediction methods are recommended based on pros and cons of those methods. In the following sections, we present key findings that played a central role as the foundation of knowledge and theory applied in this master thesis.

### 3.1 Cloud computing

The underlying concepts of cloud computing could be derived from 1961 in the MIT Centennial talk by John McCarthy, who said that “…The computer utility cloud become the basis of new and important industry”. Nonetheless, the complete concepts of cloud computing were potentially first introduced by Erick Schmidt in 2006, on Search Engine Strategies Conferences[48]. And on October in 2007, the term ‘cloud computing’ became popular when IBM and Google declared to collaborate in this domain[6]. Till now, a number of experts and scholars have defined cloud computing in various ways.

According to Armbrust, et al. [1], cloud computing refers to “both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services”. And the data center hardware and software play the roles as so-called ‘cloud’, which is the underpinning infrastructure for capabilities of cloud computing. As a result, constructing and operating large-scale, commodity-computer data centers in low budgets is necessary and important for enabling cloud computing. Mell & Grance[40] describe cloud computing as a model for enabling cloud consumers to acquire ubiquitous, convenient, on-demand computing resources (e.g., networks, storage, and services etc.) that can be quickly provided and released with minimum management effort or service provider interaction. And Above-mentioned two definitions indicate the components and service characteristics of cloud computing. Subsequently, an integrated definition of cloud computing is proposed as below:

“Cloud computing, which consists of applications, systems software and hardware to deliver services via the Internet, aims at quickly and flexibly provisioning ubiquitous, convenient, on-demand computing resources to cloud customers with minimum management efforts”.

From Cochran & Witman [14], cloud computing architectures have:
Highly abstracted resources
• Near instant scalability and flexibility
• Near instantaneous provisioning
• Shared resources (hardware, database, memory, etc.)
• ‘Service on demand’, usually with a ‘pay-as-you-go’ billing system
• Programmatic management (e.g., through WS API)

There are five main essential characteristics of cloud computing, which are on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service[40]. These five characters highlight the greater advantages and capabilities of cloud computing when compared to traditional IT services and, therefore, cloud computing is described by Buyya et al. [7] as the fifth utility (after water, electricity, gas, and telephone) to satisfy people’s daily requirements.

Cloud computing can be categorized into three types in terms of service model[40], and these classifications are shown as below:

• **Software as a service (SaaS).** The capability delivered to users is to employ provider’s application operating on cloud infrastructures.

• **Platform as a Service (PaaS).** The capability delivered to users is to employ provider’s platform to develop, run, and manage web applications through programming languages, libraries, services, and tools offered by provider.

• **Infrastructure as a Service (IaaS).** The capacity delivered to users is to provision computing resources where customers can deploy and run arbitrary software.

Cloud model can also be classified into four kinds regarding employment model[40], which are private cloud, public cloud, community cloud, and hybrid cloud (combination of private, public and community cloud).

3.2 Multi-cloud

Because of the evolution and prosperity of cloud computing, cloud providers are growing rapidly and the competition among cloud providers has resulted in an ever-growing number of solutions provided to cloud consumers[20]. This leads to the emergence of multi-cloud, which means that cloud consumers employ services and resources from multiple cloud providers to optimize performance, availability, costs, lower risks and improve quality of services[20][47]. And the diversity and usage
benefits of the IaaS offers motivate SaaS providers to lease resources from the Cloud rather than operating their own data centers, which helps SaaS providers get rid of the maintenance overheads and better meet with customers who are more demanding regarding service requirements nowadays. This trendency results in the emergence of new ways of service provisioning in which depending on a single cloud provider is insufficient. Namely, the need of harnessing multiple cloud providers with various quality attributes and pricing models has been raised recently[21].

In multi-cloud model, typically there is not an agreement between the different cloud providers on how to share resources, and the cloud user is aware of different Clouds and is responsible, or an intermediary party such as broker is obligated, for coping with the provisioning of services or resources[47]. This denotes that the term multi-cloud can indicate the employment of multiple and independent cloud providers by cloud clients or services. Petcu [47] also points out that an important functionality of multi-cloud is to manage employments on a variety of Clouds. According to[47], multi-cloud can be classified into two categories that are service-based and library-based. In service-based multi-cloud, a particular service provides brokerage between multiple Clouds depending on users’ service level agreements or provisioning regulations; whereas in the library-based multi-cloud, a library facilitates a congruent way to acquire multiple services and resources, as well as supplements of services and resources from multiple Clouds.

However, there are also some problems caused by multi-cloud. Because cloud solutions provided by multiple Clouds are usually heterogeneous and offered characteristics are incompatible, and different Clouds cannot interoperate mutually, users are not able to utilize the full potential of cloud computing[20]. Furthermore, multi-cloud increases the complexity of operations, and generates issues of security, governance, and resiliency etc. And multi-cloud produces more values only when consumers select the right cloud provides.

3.3 Service level agreement and SLA violation

The emergence of service level agreement can be traced back to the early 1990s as a method for Information Technology (IT) departments and service provider in computing networking environment to manage and measure quality of service (QoS)
they were providing to their users[33].

Service level agreement, a contract between consumer and provider in the context of a special service provision, is a document that consists of the descriptions of the agreed services, service level parameters, guarantees, and actions and remedies should be taken for all cases of violations[3][50]. And SLA aims at bridging the gap between service provider and customer[57]. Specifically, SLA generally incorporates six perspectives regarding roles and objectives, which are defining roles and responsibilities, managing expectations, controlling implementation and execution, providing verification, enabling communications, and evaluating return on investments[33].

Even though the nature of requirements of SLA can be quite diversified, the general structure of SLA remains the same. From Keller & Ludwig [31], every analyzed SLA consists of:

• The involved parties.
• The SLA parameters.
• The metrics used as input to compute the SLA parameters.
• The algorithms for computing the SLA parameters.
• The service level objectives (SLOs) and the appropriate actions to be taken if a violation of these SLOs has been detected.

Service level agreements are of vital importance for both provider and consumer as provider can base on these agreements to optimize the use of infrastructure to meet agreed services, and consumer can ensure the level of quality of service with contracts[3]. Under the background of current development of IT technology, a rapidly increasing number of customers tend to delegate their assignments and requirements to cloud providers since cloud computing is able to provide cheap and pay-as-you-go computing and services compared to traditional IT Infrastructure, which means countless SLAs are carried out between provider and consumer daily, even hourly. This makes service level agreement (SLA) between cloud providers and cloud consumers become more and more important and attain more attention[45].

Under the multi-cloud environment, countless SLAs are carried out between cloud provider and users. And inevitably, SLAs violations (i.e., SLOs specified in SLA are not met) will occur due to limitations on the availability of infrastructure resources[42]. Once cloud providers violate the agreed SLAs, they are responsible for
bearing the penalties, which lead to the economical loss of cloud providers. As a consequence, reducing the possibility of SLAs violations is quite important for cloud providers, not only because SLAs violation prevention reduces penalty, but also because it enhances customer satisfaction.

3.4 Workload prediction methods in the cloud environment

This section is carried out to answer research sub question 7, which seeks out different workload prediction models in the multi-cloud.

Currently, there are a large number of methods have been proposed to predict the workload in the cloud. These methods can be basically categorized into three types, namely, statistical methods, learning methods, and hybrid methods (i.e., the combination of statistical and learning methods). In terms of statistical methods, for example, Quiroz et al. [49] have proposed a pattern-matching model (quadratic response surface model) to identify the similar past occurrences with the current workload history. Roy, Dubey, & Gokhale[51] adopt a second order autoregressive moving average method to perform the workload prediction. Liu[34] use MEAN, exponential smoothing, autoregressive, and moving average models etc. to compare the experiment outcomes with a proposed time-series pattern based interval forecasting strategy (TPBIFS) when predicting the cloud upcoming workload. And Khan, Yan, Tao, & Anerousis [30] adopt a time-series prediction approach based on the Hidden Markov Model (HMM) to estimate the future workload. Sarikaya, Isci, & Buyuktosunoglu[56] present the statistical metric modeling (SMM), a probability distribution over workload patterns, that is system and metric-independent for predicting workload behavior. A linear regression model (LRM) is employed by [62] while a second order autoregressive moving average method is applied by [51] to predict the workload. Besides, autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA) are usually utilized by many researchers to perform the workload prediction [9], [51], [63]. Models such as AR, ARMA, ARIAM with considering integrated noise and exogenous variables (i.e., ARX, ARIX, ARMAX, ARIMAX respectively) are applied in the cloud workload prediction as well.

Regarding learning methods, which typically learn from historical data and build
model to predict the future values, neural network (NN) models and k nearest neighbors (kNN) have been widely adopted to estimate time-series data [2], [16], [27].

When it comes to hybrid methods, Cetinski & Juric [8] proposed a hybrid method which combines statistical and learning method, called AWE-WPC advanced model, to efficiently predict the workload in the cloud. Islam, Keung, Lee, & Liu [28] propose a prediction-based resource provisioning method with combining linear regression and neural network to meet with future resource demands. Montes et al. [43] propose a multi-stage predictor method which uses learning methods (such as KNN, MLP etc.) with a single entity vision of the grid to improve management system performance. We concluded the algorithms, advantages, disadvantages, and limitations of those prediction methods in the following section.

### 3.4.1 Quadratic response surface model (QRSM)

This method relates the achieved wall clock response time \( w \) (representing workload) with the number of virtual machines \( (v) \), the amount of memory allotted per VM in \( m \) (with units KB/VM) and the index of parallelism \( p \) (measured in terms of avg CPU time/VM)[49]. With this set of predictors, \( \{v, m, p\} \), the author aims to model the actual wall clock response time. In the non-linear model, the author captures the achieved parallelization by accounting for the relationship between \( w \) and \( p \) in concert with other predictors. Formally the predicted response surface can be expressed as following with knowing the parameters of the response surface \( b \).

\[
\hat{ω} = b_0 + b_1v + b_2m + b_3p + b_{12}vm + b_{13}vp + b_{23}mp + b_{11}v^2
+ b_{22}m^2 + b_{33}p^2
\]  

(3-1)

Advantage of this methodology is that it well suited to nonlinear dynamics simulation and experiment and it is easy to implement in damage identification settings, which means that relatively few data sets are required to build a model relating inputs and outputs. Disadvantage of this model is that a response surface models only a single response with respect to the uncertain parameters, and the response is normally assumed to be a smoothly-varying value with respect to the change in parameters. Besides, response surfaces are designed to work with continuous parameters, and its methods assume that for a given fixed set of input values, the response is uniquely
determined. The final discovered disadvantage is that RSM can cause "aliasing" effects.

3.4.2 Exponential smoothing (ES)

3.4.2.1 Single exponential smoothing

Single exponential smoothing is used for short-range forecasting, usually just one month into the future[29]. The specific formula for single exponential smoothing is:

$$S_t = \alpha X_t + (1 - \alpha)S_{t-1}$$

where $\alpha$ is the smoothing factor, and $0 < \alpha < 1$, $X_t$ is the current observation, and $S_{t-1}$ is the previous estimated value.

3.4.2.2 Double exponential smoothing

This method is used when the data shows a trend. Exponential smoothing with a trend works much like simple smoothing except that two components must be updated each period - level and trend. The level is a smoothed estimate of the value of the data at the end of each period. The trend is a smoothed estimate of average growth at the end of each period[29]. The specific formula double simple exponential smoothing is:

$$S_t = \alpha y_t + (1 - \alpha)(S_{t-1} + b_{t-1}) \quad 0 < \alpha < 1$$

$$b_t = \gamma (S_t - S_{t-1}) + (1 - \gamma) b_{t-1} \quad 0 < \gamma < 1$$

Note that the current value of the series is used to calculate its smoothed value replacement in double exponential smoothing.

3.4.2.3 Holt-Winter exponential smoothing

This method is used when the data shows trend and seasonality. The model is defined by the equations[9]:

$$S_t = \alpha \frac{y_t}{D_t-K_1} + (1 - \alpha)(S_{t-1} + T_{t-1})$$

$$T_t = \beta (S_t - S_{t-1}) + (1 - \beta) T_{t-1}$$

$$D_t = \gamma \frac{y_t}{S_t} + (1 - \gamma) D_{t-K_1}$$

$$\hat{y}_{t+h,t} = (S_t + hT_t) \times D_{t-K_1 + h}$$

where $S_t$, $T_t$ and $D_t$ stand for the level, trend and seasonal estimates, $K_1$ for the seasonal period, and $\alpha$, $\beta$ and $\gamma$ for the model parameters. When there is no seasonal
component, the $\gamma$ is discarded and the $D_{t-K_l+h}$ factor in the last equation is replaced by the unity.

The advantage of exponential smoothing is that it is easier to implement and more efficient to compute, as it does not require maintaining a history of previous input data values. And disadvantage is that this method can smooth away important trends or cyclical changes if improperly used, and predicating workload using exponential smoothing can lead to lagged effects. Single exponential smoothing is limited to the short-term prediction.

### 3.4.3 Autoregression model (AR)

AR multiplies previous data points with some parameters between 0 and 1 to predict the next value. These parameters can be deemed as weights assigned for each point where normally the closer the time, the larger the weight is. The formula for the generic AR ($K$) is defined as follows:[34]:

$$P_{i+k} = c + \alpha_i Y_i + \alpha_{i+1} Y_{i+1} + \ldots + \alpha_{i+k-1} Y_{i+k-1} + \epsilon_t (\sum_{j=i}^{j=i+k-1} \alpha_j = 1) \quad (3-8)$$

where $P_{i+k}$ is the estimated future value, $c$ is a constant, $\alpha_i, \ldots, \alpha_{i+k-1}$ are the parameters of the model that can be estimated through many ways such as the ordinary least squares or methods of moments, and $Y_i, \ldots, Y_{i+k-1}$ are the previous values in the time series $K$, $\epsilon_t$ is white noise.

The advantage of AR is that it is easy to implement and it does not require many data to forecast the future value, whereas the disadvantage is that AR makes sense only if the data is auto-correlated, and the determination of the value of $\alpha_i, \ldots, \alpha_{i+k-1}$ heavily influences the model’s success. Besides, AR is only applicable to forecast the future phenomenon that is related to the previous values.

### 3.4.4 Moving Average (MA)

The generic MA predicts the next value based on the average value of the latest $K$ points, denoted as MA ($K$). The formula for the generic MA ($K$) is defined as follows:[34]:

$$P_{i+k} = \frac{1}{k} \sum_{j=i}^{i+k-1} Y_i \quad (i \geq 0, k \geq 1) \quad (3-9)$$
where $P_{i+k}$ is the predicated value, $k$ is the average length of the duration sequences, and $Y_i$ is the previous value of the nearest $K$ points.

The advantage of Moving average is that it is simple and it provides a more stable level indicating support or resistance. And the disadvantage of this method is that it does not model seasonality or trend, the historical data must be stored and processed, and it is difficult to determine the optimal number of periods to include in the average.

### 3.4.5 Second order autoregressive moving average method

The equation of this method is given by[51]:

$$
\lambda(t + 1) = \beta \times \lambda(t) + \gamma \times \lambda(t - 1) + (1 - (\beta + \gamma)) \times (\lambda(t - 2))
$$

(3-10)

where $\lambda(t + 1)$ is the estimated future value, $\lambda(t)$, $\lambda(t - 1)$, and $\lambda(t - 2)$ are the previous observation values, $\beta$ and $\gamma$ are given by the values 0.8 and 0.15, respectively.

Advantage of this method is that it is easy to fit an approximation to almost any time series while the disadvantage is that it is easy to take random noise into any time series.

### 3.4.6 Autoregressive moving average (ARMA)

ARMA model is a combination of Autoregressive and moving average models. The number of past observations that $X_t$ depends on, $p$, is the AR degree. The number of past innovations that $X_t$ depends on, $q$, is the MA degree. In general, these models are denoted by ARMA $(p, q)$. The general formula of ARMA model can be described as below[9]:

$$
X_t = c + \varepsilon_t + \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}
$$

(3-11)

where $X_t$ is the estimated value, $c$ is a constant, $\varepsilon_t$ is white noise, $\phi_i$ and $\theta_i$ are the parameters, and $X_{t-i}$ is the past observed value while $\varepsilon_{t-i}$ is the white noise error term.

### 3.4.7 Autoregression integrated moving average (ARIMA)

An ARIMA model is a generalization of the ARMA model that is used for
longitudinal data with long-term trends. The model is generally denoted as ARIMA \((p, d, q)\) where \(p\), \(d\), and \(q\) are the order values of the autoregressive, integrated, and moving average parts of the model.

The generic equation of ARIMA is illustrated as below:

\[
\left(1 - \sum_{i=1}^{p} \phi_i L^i\right) (1 - L)^d X_t = \delta + \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \epsilon_t
\]

where \(X_t\) is a time series of data, \(L\) is the lag operator, the \(\phi_i\) are the parameters of the autoregressive part of the model, the \(\theta_i\) are the parameters of the moving average part and the \(\epsilon_t\) are error terms, \(\delta\) is a constant. No trend in the data is modeled by setting \(d = 0\), a linear trend is modeled by setting \(d = 1\), and a quadratic trend is modeled by \(d = 2\).

The main advantage of this method relies on the accuracy over a wider domain of series and it requires data on the time series in question only. First, this feature is advantageous if one is forecasting a large number of time series. Second, this avoids a problem that occurs sometimes with multivariate models.

The disadvantage of this method is that some of the traditional model identification techniques are subjective and the reliability of the chosen model can depend on the skill and experience of the forecaster (although this criticism often applies to other modelling approaches as well). It is not embedded within any underlying theoretical model or structural relationships. The economic significance of the chosen model is therefore not clear. Furthermore, it is not possible to run policy simulations with ARIMA models, unlike with structural models. ARIMA models are essentially ‘backward looking’. As such, they are generally poor at predicting turning points, unless the turning point represents a return to a long-run equilibrium.

### 3.4.8 Autoregressive model with exogenous variables (ARX)

ARX model is the general autoregressive model with considering the exogenous variables that can influence the predicted output. ARX model structure can be illustrated as below:

\[
y(t) + a_1 y(t - 1) + \cdots + a_{n_a} y(t - n_a) \\
= b_1 u(t - n_k) + \cdots + b_{n_b} u(t - n_k - n_b + 1) + e(t)
\]

The parameters \(n_a\) and \(n_b\) are the orders of the ARX model, and \(n_k\) is the delay.
\[ y(t) \text{— Output at time } t. \]
\[ n_a \text{— Number of poles.} \]
\[ n_b \text{— Number of zeros plus 1.} \]
\[ n_k \text{— Number of input samples that occur before the input affects the output, also called the dead time in the system.} \]
\[ y(t - 1) \ldots y(t - n_a) \text{— Previous outputs on which the current output depends.} \]
\[ u(t - n_k) \ldots u(t - n_k - n_b + 1) \text{— Previous and delayed inputs on which the current output depends.} \]
\[ e(t - 1) \ldots e(t - n_c) \text{— White noise disturbance value.} \]

3.4.9 **Autoregressive moving average model with exogenous variables (ARMAX)**

ARMAX model is the general ARMA model with the inclusion of exogenous variables that possibly affect the estimated output. ARMAX model structure can be described as below:

\[
\begin{align*}
y(t) + a_1 y(t - 1) + \cdots + a_{n_a} y(t - n_a) &= b_1 u(t - n_k) + \cdots + b_{n_b} u(t - n_k - n_b + 1) \\
&+ c_1 e(t - 1) + \cdots + c_{n_c} e(t - n_c) + e(t)
\end{align*}
\]

3.4.10 **ARIX and ARIMAX model**

ARIX model is similar to ARX model, except that it contains an integrator in the noise source \( e(t) \). And ARIMAX model is similar to ARMAX model, except that it
contains an integrator in the noise source $e(t)$. A general ARIAMX model can be structured as below:

$$ A(q) y(t) = B(q) u(t - nk) + \frac{1}{(1 - q^{-1})} e(t) \quad (3-15) $$

### 3.4.11 A Multiple Time Series Approach based on Hidden Markov Modeling (HMM)

Predictable Co-clusters are used to determine, among all co-clusters discovered, the ones that are more predictable. Then autocorrelation function is expressed[30]:

$$ R_C = \sum_{t=1}^{n-1} \left[ (c_t - \bar{c})(c_{t+1} - \bar{c}) \right] / (n\sigma_c^2) \quad (3-16) $$

Here $\bar{c}$ and $\sigma_c$ represent the mean and standard deviation of time series $C = \{c_1, c_2, \ldots, c_n\}$, where $c_i (i = 1, 2, \ldots, n)$ is an indicator variable representing whether this co-cluster exists at time interval $i$.

The time-lagged cross-correlation between the ordered pair $(C_1, C_2)$ is

$$ R_{C_1,C_2} = \sum_{t=1}^{n-1} \left[ (c_{1,t} - \bar{c}_1)(c_{2,t+1} - \bar{c}_2) \right] / (n\sigma_1\sigma_2) \quad (3-17) $$

where $\bar{c}_1, \bar{c}_2$ and $\sigma_1, \sigma_2$ are the means and standard deviations of $C_1, C_2$, respectively.

Consider a co-cluster $C$ as predictable, if its autocorrelation $R_C > \gamma = 0.4$, or if there exists at least one other co-cluster $C'$, so that the time-lagged cross correlation $R_{C',C} > \gamma$.

Prediction Model is then used to divide predictable co-clusters into prediction groups to predicate workload changes.

The workload variations across all co-clusters in a prediction group are modeled as a continuous-time Markov process. Each Markov state represents a specific type of application behavior, hence corresponds to the appearance of some co-clusters with certain probabilities. Using a Hidden Markov Model (HMM), the following parameters are defined:

- $R$ states, denoted as $H = \{H_1, H_2, \ldots, H_R\}$, that represents $R$ different application behaviors in a prediction group; the state at time $t$ is $q_t$.

$2^k$ distinct observations per state. Assume there are $k$ co-clusters in the prediction group, each can either appear or not in an observation interval, our observation space
has $2^k$ possible outcomes: $O = \{O_0, O_1, \ldots, O_2^k\}$

The state transition probabilities $A = \{a_{ij}\}$, where

$$a_{ij} = P\{q_{t+1} = H_j | q_t = H_i\}, H_i, H_j \in H.$$

The observation probabilities in state $j$, $B = \{b_k\}$, where

$$b_j(k) = P\{O_k \text{ at } t | q_t = H_j\}, H_j \in H, O_k \in O.$$

All parameters $H$, $O$, $A$ and $B$ can be estimated using the expectation-maximization method.

Once we estimated the state transition and observation probabilities, given the current observation $o_t$ at time $t$, we calculate the probability of observing $o_{t+1}$ at time $t+1$ as:

$$P(O_{t+1}|O_t) = \sum_{H_t} \{P(H_t|o_t) \sum_{H_{t+1}} P(H_{t+1}|H_t)P(o_{t+1}|H_{t+1})\} \tag{3-18}$$

The first term $P(H_t|o_t)$, can be determined using the Viterbi’s algorithm, while the second and third terms are the state transition and observation probabilities that have been estimated. With this probability estimation, we then predict the most likely observation at time $t+1$ as:

$$o'_{t+1} = \arg\max_{o_t} P(o_t|o_t), o_t \in O. \tag{3-19}$$

The advantage of this method is that it is based on multiple time series and explores the cross-correlation among them through the discovery of co-clusters and prediction groups; also the prediction accuracy is improved. And the disadvantage is that this method is complex, and is typically used to forecast workload for the “hot spots” in a cloud.

### 3.4.12 Predicating workload using linear regression model (LRM)

Linear regression model is quite simple and it can be expressed as[62]:

$$Y_t = \beta_1 + \beta_2 X_t \tag{3-20}$$

$Y$ is the workload, and $X$ is the time. The coefficients $\beta_1$, $\beta_2$ are determined by solving a linear regression equation based on previous workloads $Y_{t-1}$, $Y_{t-2}$, $Y_{t-3}$ and so on.

The Ordinary Least Squares are used to solve this equation, then we can get the results as below:

$$\sum Y_t = n \beta_1 + \beta_2 \sum X_t$$
\[ \sum X_i Y_i = \beta_1 \sum X_i + \beta_2 \sum X_i^2 \]

According to the Cramer’s Rule, we can obtain the solution of linear simultaneous equations of \( \beta_1, \beta_2 \), as shown in below:

\[ \begin{align*}
\beta_1 &= \frac{\sum X_i^2 \sum Y_i - \sum X_i \sum X_i Y_i}{n \sum X_i^2 - (\sum X_i)^2} \\
\beta_2 &= \frac{n \sum X_i Y_i - \sum X_i \sum Y_i}{n \sum X_i^2 - (\sum X_i)^2}
\end{align*} \]

### 3.4.13 SMM (Statistical Metric Modeling)

The SMM model is a probability distribution, \( P(s) \), over \( L \) samples \( S = s_1, s_2, ..., s_L \), that attempts to reflect the frequency with which each finite sequence \( s = s_1, s_2, ..., s_l \) \((l < L)\) occurs in a metric[56].

\[
P(s) = P(s_1)P(s_2|s_1) ... P(s_l|s_{l-1} ... s_1) = \prod_{i=1}^{l} P(s_i|s_{i-1}, s_{i-2}, ..., s_1) \quad (3-21)
\]

In SMM, the probability of a string \( P(s) \) is expressed as the product of the probabilities of the samples that compose the sequence, with each sample probability is conditional on the identity of the last \( n-1 \) samples. Without loss of generality, we can express the probability of a \( s \), \( P(s) \) as:

\[
P(s) = \prod_{i=1}^{l} P(s_i|s^{i-1}_{i}) \approx \prod_{i=1}^{l} P(s_i|s^{i-1}_{i-n+1}) \quad (3-22)
\]

where \( s^{i}_{i} \) denotes samples \( s_i, ..., s_j \). In order to simplify the description and formulation of the SMM we consider the case \( n = 2 \). The extension of formulation and results to higher order models are trivial. By setting \( n = 2 \), we make the approximation that the probability of a sample only depends on the identity of the immediately preceding sample, hence we can approximate \( P(s) \) as:

\[
P(s) = \prod_{i=1}^{l} P(s_i|s_{i-1})
\]

This probability distribution can be estimated with maximum likelihood estimation (MLE) technique:

\[
P(s_i|s_{i-1}) = \frac{C(s_{i-1}, s_i)}{C(s_{i-1})}
\]
where $C(x)$ denotes the number of times the sequence $x$ occurs in the metric. This is called the maximum likelihood (ML) estimate for $P(s_i|s_{i-1})$.

In order to improve the accuracy of ML estimates, “model smoothing” is applied to the relative frequencies to make sure that each probability estimate is larger than zero. A widely used set of smoothing methods is based on absolute discounting, which interpolates higher order n-gram models with lower order n-gram models. When there is insufficient data to estimate a probability in the higher order model, the lower order model can often provide useful information. The higher order distribution is created by subtracting a fixed discount $D < 1$ from each nonzero count. For finite metric sequences with nonzero counts, this distribution has the following general form:

$$P_{\text{int}}(s_i|s_{i-n+1}^{i-1}) = \frac{C(s_{i-n+1}^{i-1}) - D \sum s_i C(s_{i-n+1}^{i-1})}{\sum s_i C(s_{i-n+1}^{i-1})} + \alpha(s_{i-n+1}^{i-1})P_{\text{int}}(s_i|s_{i-n+2}^{i-1})$$ (3-23)

where $P_{\text{int}}(s_i|s_{i-n+2}^{i-1})$ is the lower order smoothing distribution. Normalization constraints fix the value of $\alpha(s_{i-n+1}^{i-1})$:

$$\alpha(s_{i-n+1}^{i-1}) = D \frac{n_{1+}(\ast, s_{i-n+1}^{i-1})}{\sum s_i C(s_{i-n+1}^{i-1})}$$ (3-24)

where $n_{1+}(\ast, s_{i-n+1}^{i-1})$ represents the number of bins for which $C(s_{i-n+1}^{i-1}) > 0$.

There are four main strengths of SMM compared to existing predictors. First, it models long term global patterns in application behavior. Second, the predictor can respond to variable-length patterns. Third, it is resilient to small fluctuations in the observed patterns. Last, the SMM predictor has the ability to adapt it as it learns more it predicts better.

### 3.4.14 Neural network model to predicate the workload (NN)

The neural network consists of three layers with an input layer, one output layer and one hidden layer[27]. In the input layer, the neural network receives the information of workloads from $X_1$ to $X_a$ at various time steps of one complete sequence that constitutes the first epoch. At each time step the output is feedback to be employed as the input $k$ for the next time step. At any given time $t$, the input of the neural network composed of the vector is shown as:

$$x(t) = [x(t), x(t-1), ..., x(t-p)]^T$$ (3-25)
where \( p \) is the number of selected delay line memory. The vector means the neural network can remember the numbers of workloads. For this vector, it works to select the number of \( p \) and decides how long the neural network can remember the workloads in the past.

For a neural network with one hidden layer, the output of a single neuron \( j \) in the hidden layer is given by the equation:

\[
y_j = \varphi \left( \sum_{l=0}^{p} w_j(l) x(n - 1) + b_j \right)
\]

(3-26)

Here \( \varphi(\cdot) \) is the activation associated to the neurons \( j \), \( w_j \) is synaptic weight and \( b_j \) is the bias.

Therefore the output of a neural network with \( m \) neurons in the hidden layer is:

\[
y(n) = \sum_{j=1}^{m} w_j y_j(n)
\]

(3-27)

And the output vector of the neural network maps to the input vector is shown as:

\[
y(n) = \sum_{j=1}^{m} w_j \varphi \left( \sum_{l=0}^{p} w_j(l) x(n - 1) + b_j \right) + b_0
\]

(3-28)

Advantages are that this method requires less formal statistical training, ability to implicitly detect complex nonlinear relationships between dependent and independent variables, ability to detect all possible interactions between predictor variables, and the availability of multiple training algorithms. And disadvantages are that this method is unable to detect unusual patterns and results in greater computational burden, proneness to over-fitting, and the empirical nature of model development.

### 3.4.15 kNN (k nearest neighbours)

The k-Nearest Neighbours algorithm is a simple, yet powerful and versatile, pattern recognition method. The algorithms are briefly introduced as below[2]:

1. The time series considered, \( \{X_t\} \) with \( t = 1, \ldots , n \), is transformed into a series of \( d \)-dimensional vectors:

\[
X_t^{d,\tau} = (X_t, X_{t-\tau}, \ldots, X_{t-(d-1)\tau})
\]

(3-29)

where \( d, \tau \in \mathbb{N} \), with \( d \) being the number of lags and \( \tau \) the delay parameter. If \( \tau = 1 \), the resulting time series of vectors is denoted by \( \{X_t^d\} \), with \( t = d, \ldots , n \), where

\[
X_t^d = (X_t, X_{t-1}, \ldots, X_{t-(d-1)})
\]
is a vector of \(d\) consecutive observations that can be represented as a point in a \(d\)-dimensional space.

(2) The distance between the last vector \(X_n^d = (X_n, X_{n-1}, \ldots, X_{n-d-1})\) and each vector in the time series \(\{X_t^d\}\) with \(t = d, \ldots, n - 1\) is computed, and the \(k\) vectors closest to \(X_n^d\) are selected. They are denoted by \(X_{T_1}^d, X_{T_2}^d, \ldots, X_{T_k}^d\). The Euclidean distance is typically applied in this step.

(3) Given the \(k\) neighbouring vectors, \(X_{T_1}^d, X_{T_2}^d, \ldots, X_{T_k}^d\), their subsequent values \(X_{T_1+1}, X_{T_2+1}, \ldots, X_{T_k+1}\) are averaged to obtain the forecast \(X_{n+1}\).

This is the most simple kNN, and more sophisticated kNNs are also carried out by other scholars.

The advantage of this method is that the cost of learning process of kNN is low, and there are no assumptions about the concept characteristics. This method is robust to noisy training data and effective if the training data is large. And the disadvantage is that the model is difficult to be interpreted, it is computationally expensive to find the \(k\) nearest neighbors when the dataset is very large, performance depends on the number of dimensions.

### 3.4.16 Comparison criteria of different methods

In order to compare the goodness of each prediction approach a couple of criteria are selected based on advantages and disadvantages of each prediction method. And we grade each method on every selected criteria from 1 to 5 (the higher the grade, the better the criteria is), the specific result is shown in the below table.

Characteristics of prediction approaches influence the cloud service provider’s selection of approach. The first criteria is defined as “computational cost” since the cloud service providers anticipate the chosen approach to generate less cost so that the profit can be increased. And to ensure the selected approach to produce results in time, the prediction model is expected to be easily implemented. Consequently the criteria “easiness of implementation” is defined. In the multi-cloud, sometimes the future workloads need to be estimated based on limited samples since there are not enough samples generated or time is too short. We subsequently define the criteria “sample size requirement”. The workload pattern is uncertain and it can evolve overtime so that the workload prediction model should be conveniently adapted to change the
model. Thereby the criteria “adaptability” is created. The criteria “complexity” is coined because some models are so complicated that mistakes are easily generated and it will cause extra cost. Accurate workload prediction can significantly help to reduce SLA violation and cost, and therefore “prediction accuracy” is involved as an important criterion.

A five-scale grade criteria is sued since a criteria is typically evaluated as “very good”, “good”, “neutral”, “poor”, and “very poor” when acquiring the advantages, disadvantages, and limitations of the retrieved workload prediction methods. Consequently, a criterion is graded as 5 if it is very good, 4 if it is good, 3 if it is neutral, 2 if it is poor, and 1 if it is very poor.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Computational cost</th>
<th>Easiness of implementation</th>
<th>Sample size requirement</th>
<th>Adaptability</th>
<th>Complexity</th>
<th>Prediction accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>QRSM</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>ES</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>AR</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>MA</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>ARMA</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>ARIMA</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>ARX</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>ARMAX</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>HMM</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>TPBIFS</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>SMM</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>LRM</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>kNN</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>NN</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>AME-WPC</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 1: Comparison criteria and result of different prediction methods
4. Problem deduction and definition

4.1 Problem deduction

In the multi-cloud environment, cloud consumers have many choices in terms of selecting an appropriate cloud service provider to acquire the intended services. And customer satisfaction is an influential factor that determines the customers’ choices. It is subsequently vital for cloud providers to enhance the customer satisfaction to keep and acquire customers. This also means that lower customer satisfaction will lead to profit loss that is contradictory to the commercial goal of cloud providers. SLA violation is a determinative factor that causes lower customer satisfaction. Besides, SLA violation will also generate SLA penalties that increase the profit loss. Furthermore, SLA violation can be detrimental for the creditability of cloud providers. This results in lower customer acquisition, and thereby decreasing profit.

There are a variety of determinants that lead to SLA violations. Specifically, inaccurate dynamic resource scaling will enable systems in the cloud not to work seamlessly and result in poor resource allocation, therefor increasing the opportunity of SLA violation. Dynamic resource scaling is influenced by a lot of factors (e.g. current state of the system, number of users, projects, upcoming events and projects such as software development projects or delivering a service to a customer). These decisions can be identified on the basis of future workload prediction. It is consequently important to establish an effective workload prediction mechanism to enable more reliable decisions about current and future resource scaling [8].

In addition, upcoming workload prediction can help cloud providers with optimizing resource allocation, scheduling, ensuring QoS, which will help decrease SLA violations[24][26][46].

A cause-effect diagram of the problem deduction is subsequently depicted as following given above information.

As can be seen from above figure, an effective workload prediction model enables cloud service providers to optimize resource allocation, schedule, ensure quality of service, and scale resources dynamically. And SLA violations of cloud service providers can be therefor decreased. This means that more customers’ needs are satisfied, and both the providers’ creditability and customers’ satisfaction are
leveraged. Since less service level agreements are violated by cloud providers, penalties generated by SLA violations decrease. And Eventually the profit of cloud service providers increased.

![Figure 3: Cause and effect diagram of problem deduction](image)

### 4.2 Problem definition

It can be drawn form the problem deduction that effective workload prediction methods will ultimately assist cloud providers increase the profit theoretically, and achieving their commercial goals. This is the reason why we would like to search for new workload prediction methods that can raise the prediction accuracy and be able to predict the required outputs in reality, so that cloud providers are able to achieve their business purposes.

Even though a variety number of prediction methods have been proposed in the existing literature, a few of them can precisely predict the workload under different circumstances [51]. For example, maximum prediction will result in overprovisioning of resources while MEAN will lead to the shortage of resource provisioning. And they will eventually cause high cost or low customer satisfaction. As a consequence, seeking a new precise prediction method is necessary and important to reduce SLA violation and costs.

Furthermore, estimation results of a variety of prediction methods may not conform to
the reality. This probably requires to adapt to the existing methods or develop new methods to perform the prediction so that the forecasted values are aligned with the real situations. In our case, the final estimated results are supposed to be a non-negative value so that estimated output is meaningful for stakeholders. Nonetheless, majority of current prediction methods typically generate negative outputs, which is meaningless in this case. And even though there are also some prediction methodologies that will only produce positive results, the prediction result of these methods are inaccurate. As a result, the ultimate problem is defined as below:

How to predict the truncated outputs with high accuracy in the multi-cloud to reduce SLA violations and SLA-related costs?

5. Method

5.1 Data collection and preprocessing

The experiment data is obtained from the Grid Workloads Archive (http://gwa.ewi.tudelft.nl/), which aims to provide workload traces from grid environment to researchers and to practitioners alike and is widely adopted in the academic research field. GWT-T-2 Grid5000 is eventually selected as the experiment data as it provides the affluent and suitable parameters of the required data (http://gwa.ewi.tudelft.nl/datasets/gwa-t-2-grid5000). GWT-T-2 Grid5000 consists of the data from the beginning of the project to the 10th of November 2006, and the data of a whole year is used in this research paper. The acquired data consists of 29 attributes, of which SubmitTime, RunTime, NProc, ReqNProc, ReqTime and Status are used to measure the intended variables such as the workload, SLA violation etc. ReqNProc stands for the required number of processors (i.e., CPU) of each job. In the cloud environment, SaaS and PaaS typically lease resources, which are VMs, CPUs, from IaaS to provide services to the clients. And the cost of leasing resources is measured by the number of rented VMs or CPUs. As a result, ReqNProc is adopted as the measurement of workload.

In order to estimate the upcoming workload in a certain future period a time-series is
created. Specifically, the workload is acquired hourly (i.e., the sum of the \textbf{ReqNProc}
in an hour). And to acquire the hourly workload, the original data is coped with in the Excel. The specific operational process to acquire the hourly workload is described in Appendix A. Eventually, an hourly time-series workload of a year is obtained, which can be immediately used for the experiment. The trace of the obtained data of a whole year is illustrated in Figure 4.

![Figure 4: Workload trace](image)

### 5.2 Historical data analysis

It is useful to check for different characteristics of the historical data (for example, periodicity, randomness, trends, etc.) before making decisions about which prediction model and which features to include into the learning dataset. The purpose of historical data analysis is discovering useful information, suggesting conclusions, and supporting decision-making. Specifically, testing randomness is used to analyze the distribution of a set of data to see if it is random (patternless). And a random dataset in computer simulation can help to show that the data are valid for use in simulation runs. Trend test is carried out to assess for the presence of an association between a variable with two categories and a variable with $k$ categories. And periodicity test is performed to see if the experiment data emerges a pattern and rule in a certain time.
The randomness of the historical data is examined by the run test in the MATLAB, which returns a test decision for the null hypothesis that the values in the data vector come in random order, against the alternative that they do not. The test is based on the number of runs of consecutive values above or below the mean of the data. The result h is 1 if the test rejects the null hypothesis at the 5% significance level, or 0 otherwise. According to the test outcome, h=1, which means to reject the null hypothesis and the data is non-random. The experiment data is therefore valid for simulation runs as a result of the data randomness.

The trend of the historical data is tested using KPSS test in the MATLAB, which returns a logic value (h) to reject the null hypothesis that the data is trend stationary, against the alternative that the data is not. The KPSS test outcome, h=1, rejects the null hypothesis, which means that the examined data is not trend stationary.

Periodicity is tested through the autocorrelation, which is a mathematical tool for finding repeating patterns such as the presence of a periodic signal obscured by noise. Figure 5 shows the tested result. It can be seen that most points fall inside the 95% confidence interval, which means that the corresponding lags are not statistically significant while statistical significance implies strong periodicity of the signal. Besides, we can see that autocorrelation coefficients do not have high autocorrelation values. These denote that the historical workload does not have strong periodicity, which can be slightly viewed from Figure 4. This also means that using the trend or periodicity prediction methods such as double exponential smoothing or Holt-Winter exponential smoothing would not be suitable.
Figure 5: Autocorrelation result

5.3 Primary design

A prediction model can be built by acquiring the data in the training window (i.e., historical data of workload) and utilize it to estimate the workload throughout a prediction window (i.e., testing data). It can be seen from the workload trace of the whole year that there are some hourly workload values that are way bigger than most values, these extremely huge workload values can easily result in overall model misfit and reduce the model accuracy. It is thereby determined to use 40 days data as training window and 10 days data as prediction window. Initial training and prediction windows are shown in Figure 6.

Figure 6: Training and prediction windows

In order to assess different prediction models' stability and accuracy over time, rolling window analysis is used in the primary experiment design. The rolling window moves
forward by 1 day with keeping 40 days as training window and 10 days as testing window.

Given the result of the historical data analysis, firstly, it is assumed that the future workload is simply influenced by the historical workload data. And a variety of forecasting methods are built via mining the data in the training window (e.g., AutoRegression, Moving average, ARMA, ARIMA, MEAN, neural networking, and exponential smoothing etc.). And then models such as ARX, ARX ARMAX, ARIMAX which include the exogenous inputs are built. In this case, 5 exogenous variables are included, which are “day”, “hour”, “last value”, “average hourly workload of the previous day”, and “workload of the same hour of the previous day”.

The result is evaluated by the model fit, i.e., the percentage of the output variations that is reproduced by the model; a high number means a better model. The precise definition of the model fit is:

$$FIT = \left[ 1 - \frac{NORM(Y - YHAT)}{NORM(Y - MEAN(Y))} \right] \times 100$$

Where $Y$ is the measured output and $YHAT$ is the simulated/predicted output.

The whole year data are examined as described above (40 days as training data and 10 days as testing data) to see if there are some regulations that can be drawn. The best prediction models of each different 40 days are mainly ARX model, ARMAX model, ARIMAX model, ARMA and ARIMA model. In total, 314 types of 40 days data are tested, of which there are 210 ARX models, 7 ARMAX models, 15 ARIMAX models, 45 ARMA models and 37 ARIMA models. This indicates that the selected exogenous variables are related to the estimated output in the most time. It can also be checked that the accuracy of prediction model will be significantly improved especially including the variable “hour” and “workload of the same hour of the previous day” and the accuracy of the prediction model can be still slightly improved by continuing including more exogenous variables such as the “day”, “last value”, and the “ average hourly workload of the previous day”.

As described above, there are totally 314 sets of training and testing data used to retrieve the best prediction model, which means that there are also 314 prediction results generated. Due to space limitations test readability, only the prediction result of the first set of training and testing data is illustrated in Figure 7 and Figure 8.
It can be seen and concluded from the prediction result of the first training data that ARX model has the best prediction result whereas MEAN has the poorest prediction accuracy. Even though ARX has a relatively better prediction accuracy, part of
forecasted results of ARX model are negative, which cannot conform to the reality that predicted values should always be positive. This problem occurs as well in ARIMA, ARMA, ARIMAX, and ARIX etc. SecondOrderARMA and MEAN only generate positive result, of which the prediction accuracy is poor nonetheless. This inspires to carry out the redesign so that prediction outputs are non-negative while prediction accuracy is good.

5.4 AR(MA)X-LLM

It can be seen from the historical workload trace (Figure 4) that hourly workload can change significantly and rapidly. This could easily lead to significant prediction errors if the whole historical data is used. In addition, it is found that exogenous variable “hour” and “workload of the same hour of the previous day” can remarkably improve the model fit. And normally there is a “busy hour” and “free hour” of each day, when workload can be quite different. As a result, it is determined to use the “same hour” workload instead of the whole historical data to perform the prediction. After acquiring the experiment data, we use a variety of models to see if they can fit to the data. In order to evaluate the goodness of each model the model fit (described above) is adopted as a criterion. The model with relatively higher model fit will be considered to perform the prediction. In this case in the section of primary design, ARX model shows higher model fit than other models.

5.4.1 ARX-LLM with error term independent on previous error terms

Even though ARX model has a relatively better model fit than other models, the prediction result of ARX can be negative since a part of ARX model coefficients and white noise are negative. This probably results in meaningless of prediction outcomes, which cannot be negative in nature. As a consequence, we proposed a new model based on ARX model, ARX-LLM (ARX model using log-likelihood maximization), of which the estimation result will always be positive so that the forecasting outcome conforms to the reality. The ARX-LLM model is specifically described in below.

First of all, the function of the general ARX model is illustrated as below:
\[ y(t) = \sum_{n_a=1}^{n_a} \alpha_{n_a} y(t - n_a) + \sum_{n_b=1}^{n_b} \beta_{n_b} u(t - n_k - n_b + 1) + e(t) \] (5-1)

The parameters \( n_a \) and \( n_b \) are the orders of the ARX model, and \( n_k \) is the delay.

- \( y(t) \) — Output at time \( t \).
- \( n_a \) — Number of poles.
- \( n_b \) — Number of zeros plus 1.
- \( n_k \) — Number of input samples that occur before the input affects the output, also called the dead time in the system.
- \( y(t - 1) \ldots y(t - n_a) \) — Previous outputs on which the current output depends.
- \( u(t - n_k) \ldots u(t - n_k - n_b + 1) \) — Previous and delayed inputs on which the current output depends.
- \( e(t) \) — White noise disturbance value.

The predicted output and all exogenous inputs of the experimental data are non-negative. As a consequence, in order to ensure that the prediction result of ARX model is non-negative, the following restrictions have to be complied with.

\[ \alpha_{n_a} \geq 0, \text{for all } n_a = 1, \ldots, n_a \] (5-2)
\[ \beta_{n_b} \geq 0, \text{for all } n_b = 1, \ldots, n_b \] (5-3)
\[ e(t) \geq 0 \] (5-4)

The standard ARX model does not have a truncation (sign restriction) on the outcome variable. Therefore, a common assumption for the error term \( e(t) \) is a normal distribution. Unfortunately, the current condition requires a distribution for \( e(t) \) that only takes non-negative values. A common distribution for this form is the gamma distribution. We thereby define a gamma distribution instead of a normal distribution for the error term:

\[ e(t) \sim \text{gamma}(\kappa, \psi) \] (5-5)

where \( \kappa > 0, \psi > 0 \) are the additional model parameters to estimate. The expectation of the error term is:

\[ e(t) = \kappa \times \psi \] (5-6)

Using ARX model function the error term can be expressed as:
\[ e(t) = y(t) - \sum_{n_a=1}^{n_a} \alpha_{n_a} y(t - n_a) - \sum_{n_b=1}^{n_b} \beta_{n_b} u(t - n_k - n_b + 1) \] (5-7)

Given the assumption that error term is a gamma distribution, the likelihood of \( t = 1, \ldots, T \) observations can be described as follow:

\[ L(y_1\ldots y_T) = \prod_{t=1}^{T} p(e(t)|\alpha, \beta, \kappa, \psi) \]

\[ = \prod_{t=1}^{T} \frac{1}{\Gamma(\kappa) \psi^\kappa} e(t)^{\kappa-1} e\left(\frac{e(t)}{\psi}\right) \] (5-9)

where \( e(t) \) can be calculated from (7).

The log-likelihood of \( t = 1, \ldots, T \) observations can be consequently expressed as following:

\[ \ln L(y_1\ldots y_T) = \sum_{t=1}^{T} -\ln \Gamma(\kappa) - \kappa \ln \psi + (\kappa - 1) \ln e(t) - \frac{e(t)}{\psi} \] (5-10)

The purpose of the proposed method is to find out a set of parameters \( \alpha_{n_a}, \beta_{n_b}, \kappa, \text{and} \psi \) which will maximize the log-likelihood of \( t = 1, \ldots, T \) observations. And then we are able to predict the upcoming workload according to function (1) given values of \( \alpha_{n_a}, \beta_{n_b}, \kappa, \text{and} \psi \).

Steps of maximizing log-likelihood estimation are described as following:

1. Set initial values of parameters that conform to parameter constraints.
2. Get initial observations \( y_1\ldots y_T \) for \( T \) periods from the experiment data.
3. Calculate error terms \( e(t) \) according to function (7) using parameters of (5-2), (5-3), and recorded data. For \( t \leq \max (n_a + 1, nb+nk) \), set \( e(t) = \kappa \times \psi \).
4. Calculate the log-likelihood from (5-10).
5. Update parameters and repeat steps from 2 to 4 until the log-likelihood value converges.

5.4.2 ARMAX-LLM with error term dependent on previous error terms

In the ARX model, error term simply depends on the historical observations and different parameter coefficients. However, there is a situation where error term also depends on errors terms of previous prediction result, for example, ARMAX model. As a subsequence, we also provide the detailed process of acquiring eligible
parameter coefficients of model that comprises the error term depending on the previous error terms. And we take the ARMAX model as an example; the specific methodology (ARMAX-LLM) is described as following.

The formula of a standard ARMAX model is initially illustrated.

\[
y(t) = \sum_{n_a=1}^{n_a} \alpha_{n_a} y(t - n_a) + \sum_{n_b=1}^{n_b} \beta_{n_b} u(t - n_k - n_b + 1) \\
+ \sum_{n_c=1}^{n_c} c_{n_c} e(t - n_c) + e(t)
\]

\[(5-11)\]

\(y(t)\) — Output at time \(t\).

\(n_a\) — Number of poles.

\(n_b\) — Number of zeros plus 1.

\(n_c\) — Number of \(C\) coefficients.

\(n_k\) — Number of input samples that occur before the input affects the output, also called the dead time in the system.

\(y(t - 1) \ldots y(t - n_a)\) — Previous outputs on which the current output depends.

\(u(t - n_k) \ldots u(t - n_k - n_b + 1)\) — Previous and delayed inputs on which the current output depends.

\(e(t - 1) \ldots e(t - n_c)\) — White noise disturbance value.

As a consequence, in order to ensure that the prediction result of ARX model is non-negative, the following restrictions have to be complied with.

\[
\alpha_{n_a} \geq 0, \text{ for all } n_a = 1, \ldots, n_a
\]

\[(5-12)\]

\[
\beta_{n_b} \geq 0, \text{ for all } n_b = 1, \ldots, n_b
\]

\[(5-13)\]

\[
c_{n_c} \geq 0, \text{ for all } c_{n_c} = 1, \ldots, c_{n_c}
\]

\[(5-14)\]

\[
e(t) \geq 0
\]

\[(5-15)\]

Similar with ARX model, a gamma distribution is defined for the error term.

\[
e(t) \sim \text{gamma}(\kappa, \psi)
\]

\[(5-16)\]

where \(\kappa > 0, \psi > 0\) are the additional model parameters to estimate. The expectation of the error term is:

\[
e(t) = \kappa \times \psi
\]

\[(5-17)\]

Using the function of ARMAX model the error term can be expressed as:
\[ e(t) = y(t) - \sum_{n_a=1}^{n_a} \alpha_{n_a} y(t - n_a) - \sum_{n_b=1}^{n_b} \beta_{n_b} u(t - n_k - n_b + 1) \quad (5-18) \]

\[- \sum_{n_c=1}^{n_c} c_{n_c} e(t - n_c) \]

Given the assumption that error term is a gamma distribution, the likelihood of \( t = 1, \ldots, T \) observations can be described as follow:

\[
L(y_1 \ldots y_T) = \prod_{t=1}^{T} p(e(t)|\alpha, \beta, c, \kappa, \psi) = \prod_{t=1}^{T} \frac{1}{\Gamma(\kappa)\psi^\kappa} e(t)^{\kappa-1} e\left(\frac{e(t)}{\psi}\right) \quad (5-20)
\]

The log-likelihood of \( t = 1, \ldots, T \) observations can be consequently expressed as following:

\[
\ln L(y_1 \ldots y_T) = \sum_{t=1}^{T} -\ln \Gamma(\kappa) - \kappa \ln \psi + (\kappa - 1) \ln e(t) - \frac{e(t)}{\psi} \quad (5-21)
\]

Steps of maximizing log-likelihood estimation are described as following:

1. Set initial values of parameters that conform to parameter constraints.
2. Get initial observations \( y_1 \ldots y_T \) for \( T \) periods from the experiment data.
3. Set initial error terms to their expectations \( e(1), \ldots, e(t - n_c) = \kappa \times \psi \)
4. Calculate error terms \( e(t) \) according to function (18) using parameters of (5-12), (5-13), (5-14) and recorded data.
5. Calculate the log-likelihood from (5-21).
6. Update parameters and repeat steps from 2 to 5 until the log-likelihood value converges.

This proposed method can be easily expanded to other prediction models consisting of errors terms such as AR, ARMA, and ARIMA and so on so that the non-negative results will be produced.

### 5.5 Implementation and result

In this section, we primarily describe the different training data that will be used to perform the prediction, and then the implementation of the proposed method is
presented. The evaluation of the prediction result is depicted in the following, and the prediction result is illustrated finally.

### 5.5.1 Training and testing dataset

On the purpose to verify the goodness of the proposed prediction model, three datasets of different lengths are used as training data. And the day just behind the training data is adopted as the testing data, which means there are three testing data as well. The reason for adopting one day behind the training data is because required information can be collected less than one day. The specific training and testing data are shown in Table 2.

<table>
<thead>
<tr>
<th>Training data</th>
<th>Period</th>
<th>Testing data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train_1</td>
<td>2\textsuperscript{nd} day of month 1 ~ 31\textsuperscript{st} day of month 3 (89 days)</td>
<td>1\textsuperscript{st} day of month 4 (1 day)</td>
</tr>
<tr>
<td>Train_2</td>
<td>2\textsuperscript{nd} day of month 1 ~ 16\textsuperscript{th} day of month 6</td>
<td>17\textsuperscript{th} day of month 6</td>
</tr>
<tr>
<td>Train_3</td>
<td>2\textsuperscript{nd} day of month 1 ~ 29\textsuperscript{th} day of month 12</td>
<td>30\textsuperscript{th} day of month 12</td>
</tr>
</tbody>
</table>

Table 2: Training and testing data

### 5.5.2 Implementation

The method is implemented in the MATLAB software. First of all, in order to estimate the values of initial parameters $na, nb, and nk$ (or $nc$) the system identification app is utilized. Given the values of $na, nb, and nk$ (or $nc$), algorithms then can be written. Pseudo code of algorithms are described as following.

<table>
<thead>
<tr>
<th>Input: $na =$ Number of poles of AR(MA)X model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$nb =$ Number of zeros plus 1 of AR(MA)X model</td>
</tr>
<tr>
<td>$nc =$ Number of C coefficients of ARMAX model</td>
</tr>
<tr>
<td>$nk =$ Number of input samples that occur before the input affects the output of AR(MA)X model, also called the dead time in the system.</td>
</tr>
<tr>
<td>$\kappa =$ Shape parameter of gamma distribution of error terms</td>
</tr>
<tr>
<td>$\psi =$ Scale parameter of gamma distribution of error terms</td>
</tr>
<tr>
<td>$R =$ The number of exogenous variables</td>
</tr>
</tbody>
</table>

41
\( r \in R \)

T = The number of the workload of training samplest

\( t \in T \)

U = The exogenous variables

\( u \in U \)

e = Error term

\( y = \) Workload

\( z = \) Log-likelihood of observations

\( \alpha_{na} = \) Coefficients of \( y(t-na) \)

\( \beta_{nb} = \) Coefficients of \( u(t-nb-nk+1) \)

\( c_{nc} = \) Coefficients of \( e(t-nc) \)

\( C = [na, nb+nk-1, nc] \)

\[
\text{for } t=\text{max}(C) \\
\quad e(t) = \kappa \times \psi; \\
\text{end}
\]

\text{Algorithms of ARX model}

\[
\text{for } t = \text{max}(C) + 1: T \\
\quad e(t) = y(t) - \sum_{n_a=1}^{na} \alpha_{na} y(t-n_a) - \sum_{n_b=1}^{nb} \beta_{nb} u(t-n_k-n_b+1) \\
\text{end}
\]

\text{Algorithms of ARMAX model}

\[
\text{for } t = \text{max}(C) + 1: T \\
\quad e(t) = y(t) - \sum_{n_a=1}^{na} \alpha_{na} y(t-n_a) - \sum_{n_b=1}^{nb} \beta_{nb} u(t-n_k-n_b+1) - \sum_{n_c=1}^{nc} c_{nc} e(t-n_c) \\
\text{end}
\]

\[
\text{for } t=\text{max}(C)+1:T; \\
\quad \text{if } (e(t) < 0) \\
\quad \quad e(t) = \kappa \times \psi; \\
\quad \text{else } e(t) = e(t); \\
\quad \text{end}
\]

\text{end}

\[
\text{for } t=1:T;
\]
\[
\begin{align*}
\ln(l) &= -\ln(\Gamma(k)) - k\ln(\psi) + (k - 1)\ln(e(t)) - \frac{e(t)}{\psi} \\
\text{end}
\end{align*}
\]

\[M = \text{maximize } (z)\]

**Output of ARX model**

\[
y(t) = \sum_{n_a=1}^{n_a} \alpha_{n_a} y(t - n_a) + \sum_{n_b=1}^{n_b} \beta_{n_b} u(t - n_b - n_a + 1) + \kappa \times \psi
\]

**Output of ARMAX model**

\[
y(t) = \sum_{n_a=1}^{n_a} \alpha_{n_a} y(t - n_a) + \sum_{n_b=1}^{n_b} \beta_{n_b} u(t - n_b - n_a + 1) + \sum_{n_c=1}^{n_c} c_{n_c} e(t - n_c) + \kappa \times \psi
\]

Figure 8: Computing algorithms

### 5.5.3 Evaluation of the prediction result

For the purpose of comparing prediction results of different prediction models, mean squared prediction errors are computed. In statistics, the mean squared error (MSE) of an estimator measures the average of the squares of the difference between the estimator and what is estimated (predicted):

\[
MSE = \frac{1}{n} (\hat{Y}_t - Y_t)^2
\]  
(5-22)

The random variable \(Y\) presents the actual value while \(\hat{Y}_t\) stands for the predicated value, and \(n\) is the total number of the predicated values. The smaller the value of MSE, the better the prediction model is.

### 5.5.4 Result

In this section, the experiment and evaluation results of three different testing data are presented. Furthermore, proposed prediction model is compared with other workload prediction models such as neural networking (NN), SecondOrderARMA, MEAN, and ARX using complete historical data. The number of rented processors (i.e., workload) is an integer in reality; the prediction result is thereby rounded to an integer so that it
makes more sense for cloud service providers.

Prediction results of three different testing data and their comparison with other prediction models are displayed in Figure 9, Figure 10, and Figure 11 respectively. It can be seen from those three figures that most actual values fall into 95% confidence interval of estimated output, which means the prediction accuracy of the proposed model is good.

The evaluation result of the proposed model and other prediction models is indicated in Table 3. It can be concluded that in both Train_1 and Train_3, the proposed model has the best prediction output since its MSE is the smallest of all. Even though the proposed model ranked second (after SecondARMA) in terms of the prediction result in Train_2, it is much better than MEAN, ARX using complete historical data, and neural networking. And in the prediction result of both Train_1 and Train_3, AR(MA)X-LLM model’s prediction accuracy is way better than that of SecondARMA model (that is, 70.96 < 1052.9 and 1795.5 < 7352.2 respectively).

Figure 9: Prediction result of the 1st testing data
Figure 10: Prediction result of the 2\textsuperscript{nd} testing data

Figure 11: Prediction result of the 3\textsuperscript{rd} testing data

<table>
<thead>
<tr>
<th>MSE</th>
<th>Train_1</th>
<th>Train_2</th>
<th>Train_3</th>
</tr>
</thead>
</table>
Table 3: Prediction result of three different testing data

<table>
<thead>
<tr>
<th>Method</th>
<th>Method</th>
<th>70.96</th>
<th>3512.8</th>
<th>1795.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR(MA)X-LLM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARX model using complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>historical data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neural networking</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SecondARMA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MEAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It is therefore concluded that the AR(MA)X-LLM model to acquire the constrained model parameters is able to accurately predict the truncated values and its prediction accuracy is better than other prediction models such as neural networking, MEAN, etc. overall. As a result, AR(MA)X-LLM can assist cloud service providers efficiently predict the future workload.

6. **SLA violations and costs**

In this chapter, the measurement of calculating SLA violations and three cost models of computing costs are primarily described. Then the result of SLA violations and costs is illustrated using the SLA violation measurement and one of three proposed cost models.

6.1 **SLA violations measurement**

SLA violations can be detected once SLA parameters are not met with. And SLA metrics are used to specify SLA parameters and define how SLA parameters can be measured, thus specifying values of measureable parameters[3][45]. As a result, a list of metrics of important SLA parameters is described to detect and measure SLA violations. Both definitions and functions of these SLA metrics are illustrated in Appendix B.

6.2 **Costs measurement**

On the purpose to check whether the proposed method is able to reduce the SLA-related costs, a function of SLA-related costs is required. A couple of
SLA-related cost functions are derived from literatures and they are described in the following.

### 6.2.1 SLA violation and resource cost model

The SLA-related cost is a combination of various factors such as cost of SLA violations, leasing cost of resources and a cost associated with the changes to the configuration [51]. Cost of leasing resources can be calculated using the number of rented processors $M_k$ in the $k_{th}$ interval (between $k-1_{th}$ hour and $k_{th}$ hour) and the cost of leasing a processor per hour $W_c$. Thereby the function of cost of leasing processors can be described as:

$$Cost_{Leasing\, processors} = W_c \times M_k$$  \hspace{1cm} (6-1)

Considering the given parameters of the used data, occurrence of SLA violations can be evaluated by two aspects (i.e., actual wait time exceeds the agreed wait time and actual run time surpasses the required run time). As a result, the function of SLA violation costs can be illustrated as:

$$Cost_{SLA\, violations} = W_r \times (S_r - S) + W_d \times T_w$$  \hspace{1cm} (6-2)

The configuration cost can be calculated using the price of configuring application and the number of installed processors. The function of configuration cost is thereby expressed as below:

$$Cost_{configuration} = W_f \times |(M_k - M_{k-1})|$$  \hspace{1cm} (6-3)

The meaning of parameters are describe in the below table

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_c$</td>
<td>Cost of leasing a processor per hour</td>
</tr>
<tr>
<td>$M_k$</td>
<td>Number of the processors used in the $k^{th}$ interval</td>
</tr>
<tr>
<td>$M_{k-1}$</td>
<td>Number of the processors used in the $k-1^{th}$ interval</td>
</tr>
<tr>
<td>$W_r$</td>
<td>Penalty for SLA violation regarding run time violations</td>
</tr>
<tr>
<td>$W_d$</td>
<td>Penalty for SLA violation regarding wait time</td>
</tr>
<tr>
<td>$W_f$</td>
<td>Cost of reconfiguring application</td>
</tr>
<tr>
<td>$S$</td>
<td>Required time</td>
</tr>
<tr>
<td>$S_r$</td>
<td>Actual run time</td>
</tr>
<tr>
<td>$T_w$</td>
<td>Wait time</td>
</tr>
</tbody>
</table>

Table 4: Components of the cost model 1
And the formula of SLA-related cost can be expressed as below [51]:

\[
Cost = W_c \times M_k + W_r \times (S_r - S) + W_d \times T_w + W_f \times |(M_k - M_{k-1})|
\]  

(6-4)

The cost of SLA violations regarding run time violation will be set as zero if \( S_r < S \), which means that the cloud providers do not violate the agreement of required time.

### 6.2.2 VRC cost model

According to Yang et al. (2014)[62], SLA-related cost can be comprised of three components. The first part is the virtual resource cost in the resource-level scaling \( (VRC_r) \), which is the product of the number of resource units and the cost of unit resource. A unit resource is the resource consumed by each request of this service.

The second part is the virtual resource cost in the VM-level scaling \( (VRC_v) \), which is the sum of costs of every type of VMs. Costs of different kinds of VMs vary with their capacities, and usually costs increase in proportion to capacities. The third part is the cost of license cost in the VM-level scaling \( (LC_v) \), which is the sum of license cost of every VM. Consequently, the SLA-related cost model can be expressed as below with considering the cost of SLA penalty:

\[
cost = VRC_r + VRC_v + LC_v + SLA_{penalty}
\]  

(6-5)

where

\[
VRC_r = n_r \times \text{cost(resource)}
\]

\[
VRC_v = \sum_{i=1}^{n_v} nVM_i \times \text{cost(VM}_i)\]

\[
LC_v = \sum_{i=1}^{n_v} nVM_i \times \text{cost(license)}
\]

In order to guarantee the resources to be increased can satisfy user requests and the resources will be available in this cluster node, the following constraints have to be complied with.

\[
n_r + \sum_{i=1}^{n_v} nVM_i \times wVM_i > n
\]

\[
n_r < a_r
\]

The following table describes the components of this cost model.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( VRC_r )</td>
<td>VRC of resource-level scaling</td>
</tr>
</tbody>
</table>
6.2.3 DSI (delivery, storage, and immigration) cost model

From Zhao et al. (2014)[64], the total operational cost of cloud providers can be made up of three components. The first portion the delivery cost that is introduced by sending data from data centers to user groups. The second component is the cost triggering from storing the data of cloud users. And the third part is the replica immigration cost due to inter-DC movement replicas as a result of optimizing algorithms’ new decisions on replicas’ locations. With considering the resource cost and SLA penalty, the formula of the cost model can subsequently be expressed as following:

\[
C_{\delta t} = \sum_{k=1}^{W} \sum_{i=1}^{M} \sum_{j=1}^{N} c_{ij} \times X_{ijk} \times \delta t + \sum_{k=1}^{W} \sum_{i=1}^{M} k_{i} \times u_{k} \times \delta t + \sum_{k=1}^{W} \sum_{i=1}^{M} \sum_{i'k=1}^{Y_{ik}=1} \sum_{i'k'=0}^{Y_{ik}'} r_{ik} \times v_{k} + \text{cost}(\text{resources}) + \text{cost}(\text{license})
\]

(6-6)

Constraints of this model are listed in the following:

\[
\text{if } \sum_{j=1}^{N} X_{ijk} > 0 \text{then } Y_{ik} = 1, i = 1, \ldots, M \text{ and } k = 1, \ldots, W,
\]
\[
\sum_{k=1}^{W} \sum_{j=1}^{N} X_{ijk} \leq B_i, i = 1, \ldots, M,
\]
\[
\sum_{k=1}^{W} v_k \cdot Y_{ik} \leq S_i, i = 1, \ldots, M,
\]
\[
X_{ijk} \geq 0, i = 1, \ldots, M, j = 1, \ldots, N, \text{and } k = 1, \ldots, W,
\]
\[
Y_{ik} \in \{0,1\}, i = 1, \ldots, M, \text{ and } k = 1, \ldots, W,
\]
\[
\sum_{i=1}^{M} X_{ijk} \geq Q_{jk}, l = 1, \ldots, M, j = 1, \ldots, N, \text{ and } k = 1, \ldots, W,
\]

The following table illustrates the components of this cost model:

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M )</td>
<td>Total number of data centers in the system</td>
</tr>
<tr>
<td>( N )</td>
<td>Total number of user groups</td>
</tr>
<tr>
<td>( W )</td>
<td>Total number of channels to be replicated</td>
</tr>
<tr>
<td>( c_{ij} )</td>
<td>Communication distance between DC ( D_i ) and user group ( U_j )</td>
</tr>
<tr>
<td>( \delta t )</td>
<td>The interval between two runs of the optimization algorithm</td>
</tr>
<tr>
<td>( k_i )</td>
<td>Storage cost factor of DC ( D_i ) that describes the cost of storing a unit data per unit of time</td>
</tr>
<tr>
<td>( v_k )</td>
<td>Size of the channel ( V_k )</td>
</tr>
<tr>
<td>( r_{ik} )</td>
<td>Communication distance from ( D_i ) to its nearest neighboring DC that has ( V_k )’s replica</td>
</tr>
<tr>
<td>( \text{cost(\text{resources})} )</td>
<td>Cost of resources</td>
</tr>
<tr>
<td>( \text{cost(\text{SLA_penalty})} )</td>
<td>Cost of SLA penalty</td>
</tr>
<tr>
<td>( X_{ijk} )</td>
<td>The bandwidth assigned by DC ( D_i ) to user group ( U_j ) for downloading channel ( V_k )</td>
</tr>
<tr>
<td>( Y_{ik} )</td>
<td>If channel ( V_k ) is replicated at ( D_i ), ( Y_{ik} = 1 ), otherwise ( Y_{ik} = 0 )</td>
</tr>
<tr>
<td>( B_i )</td>
<td>Bandwidth capacity of DC ( D_i )</td>
</tr>
<tr>
<td>( S_i )</td>
<td>Storage capacity of DC ( D_i )</td>
</tr>
<tr>
<td>( Q_{jk} )</td>
<td>The expected bandwidth of ( U_j ) when it downloads channel ( V_k )</td>
</tr>
</tbody>
</table>

Table 6: Components of the cost model 3


6.3 Result of SLA violations and costs

After acquiring the prediction model to estimate the upcoming workload of the cloud providers, we will examine whether this proposed prediction model will help to reduce the SLA violations and costs. The first testing data (1st day of month 4) is used to check if the SLA violations and costs are reduced.

Since the adopted data provides the information about if a job is violated regardless of its SLA parameters, the percentage of SLA violation of the first testing data is therefore calculated by the below formula:

\[
\text{Percentage of SLA violation} = \frac{\text{The number of violated jobs}}{\text{The total number of jobs}}
\]  \hspace{1cm} (6-7)

According to the observed data, the percentage of SLA violation of the first testing data is originally 40.7%. And the percentage of SLA violation will be reduced from 40.7% to approximately 11.1% through using the proposed prediction method. This demonstrates that the proposed method can help to reduce the SLA violation significantly.

Given the attributes of the used data, the cost model 1 is found to be the most appropriate model to perform the cost calculations.

Unfortunately, none of the parameters provided by the chosen data can be used to measure the changes to the configuration. Consequently, cost of SLA violations and leasing resources are used to measure the SLA-related cost eventually. And information of the actual run time and wait time of some jobs is missed. It is therefore determined to not use those missed information instead of giving some new values to the missed information. Given above information, the SLA-related cost before using the proposed prediction model to estimate the future workload can acquire, which is described as below:

\[
\text{Cost} = 258W_c + 231W_r + 20W_d
\]  \hspace{1cm} (6-8)

The percentage of SLA violation of using the proposed prediction method to estimate the future workload is about 11.1%, which is way smaller than 40.7% (i.e., the proportion of SLA violation without using prediction). This means that the proposed prediction method can help to lower the chances of SLA violation, thereby decreasing the corresponding SLA penalties and increasing customers’ satisfaction. SLA-related cost with using proposed prediction method can be acquired through the
above-mentioned SLA cost function. And it is shown as below:

\[
Cost_{prediction} = 339W_c + 5W_d
\]  

(6-9)

It can be seen from above two functions that the proposed method significantly decreases the costs of SLA violations (penalty of violated run time and wait time). Unfortunately, the cost of leasing processors will be leveraged using the proposed method (339 > 258) even though the cost of SLA penalty is apparently decreased. As a result, whether the proposed method will lower the SLA-related cost depends on the specific value of \( W_c, W_r, W_d \).

To sum up, the proposed method significantly decreases the percentage of SLA violation, which will be beneficial for decreasing costs of SLA violations and increasing customer satisfaction. And whether the proposed method will lower the SLA-related cost counts on the specific value of \( W_c, W_r, W_d \), which means that the proposed method cannot necessarily decrease the SLA-related cost.

### 7. Discussion

In this master thesis, we primarily describe the background and significance of the research objective so that the conducted research can significantly contribute to both literature and cloud providers. Then we clearly depict the definitions of important terms such as cloud computing, multi-cloud, service level agreement, and SLA violations etc. in the literature review to answer research sub question 1, 2, and 3. Furthermore, a variety of workload prediction methods are retrieved from literatures, and their algorithms, advantages, and disadvantages are summarized. A couple of criteria to compare the goodness of different prediction models is derived from pros and cons of those retrieved models, and each criteria of the prediction model is graded in order to select an appropriate prediction methodology under certain circumstances. As a result, the research sub question 7 is answered.

Given the importance of reducing SLA violations by cloud providers to increase profit and the significance of an accurate workload prediction model lowering the SLA violation, the initial problem is to explore an accurate workload prediction model in the multi-cloud environment. Through the problem deduction and definition, research sub question 4, 5, and 6. Since the predicted outputs are supposed to be truncated to conform to the reality, we define the final problem as:
How to predict the truncated outputs with high accuracy in the multi-cloud to reduce SLA violations and SLA-related costs?

The experiment data is obtained from the Grid Workloads Archive which provides the workload traces from cloud environment to researchers. After acquiring the experimental data, we check for different characteristics of the data such as trend, periodicity, and randomness, etc. to preliminarily decide which prediction model would be appropriate for the historical data. Then we perform different prediction models on the historical data using 40 days data as training window and 10 days data as testing window. The result shows that ARX model has the best prediction output among all conducted models. Although ARX model has relatively better prediction accuracy, the predicted output can be negative, which is not aligned with the real situation. As a result, we make a redesign which utilizes ARX model and log-likelihood maximization to acquire the constrained model parameters so that the estimated values will be non-negative. It can be found from historical data trace that hourly workload can change significantly and rapidly. This could easily lead to significant prediction errors if the whole historical data is used. In addition, it is found that exogenous variable “hour” and “workload of the same hour of the previous day” can remarkably improve the model fit. And normally there is a “busy hour” and “free hour” of each day, when workload can be quite different. As a result, it is determined to use the “same hour” workload instead of the whole historical data to perform the prediction. The error term is supposed to be non-negative in theory so that the predicted output is non-negative. A gamma distribution is thereby is defined to the error term so that the observations likelihood can be expressed using scale and shape parameter of the defined gamma distribution, and error terms. Then observations log-likelihood can be acquired through implementing logarithm on the observation likelihood.

Since computational functions of independent and dependent error terms are different, algorithms of prediction models either comprising error terms dependent on previous errors or including error terms independent on previous errors are described separately in detail. Then the proposed method is implemented in the MATLAB software and the results can be generated. On the purpose of verifying the goodness of the proposed method, we adopt three sets of training data and use the data of one day right behind the training data as testing data. Meanwhile, we compare the proposed
method with other prediction models such as neural networking, second order ARMA, MEAN and the forth. MSE (mean squared error) is adopted as the criterion to evaluate the goodness of each prediction model.

According to the prediction result, the proposed method owns the best prediction accuracy in the first and third testing data. Moreover, the MSE of the proposed model is way less than other prediction models in these two testing window. In the second testing data, second order ARMA has the best estimation result, followed by the AR(MA)X-LLM model. Despite the fact that second order ARMA possesses the best output in the second testing data, its prediction accuracy is the poorest among all prediction models in both testing data 1 and 3. Besides this, it is found that most actual predicted values fall into the 95% confidence interval of prediction result, which means that AR(MA)X-LLM model has good prediction accuracy. It is consequently concluded that the proposed model, which gets the constrained model parameter through maximizing the log-likelihood of previous observations, is able to predict the truncated values with a good model precision.

After acquiring the prediction result, we develop functions of SLA violations and SLA-related cost to see if the proposed method is able to reduce the percentage of SLA violation and SLA-related cost. The proportion of SLA violation is measured by the number of violated jobs dividing the total number of served jobs. Regarding SLA-related costs, three different cost models are developed, which compute the cost with considering different parameters in the multi-cloud. Eventually, the first cost model is adopted since its components best map with the parameters of the experiment data. The first cost model is made up of three elements, i.e., cost of leasing processors, cost of penalty of SLA violations and configuration cost. The result shows that the percentage of SLA violations can be significantly decreased, and whether SLA-related cost can be decreased will be dependent on the specific pricing of resource costs and SLA punishments. Saying this, it is summarized that the SLA violation can be significantly lowered and thereby lowing cost and enhancing customer satisfaction. Although the proposed method does not necessarily reduce the SLA-related costs, the cloud provider can base on the pricing of leasing resources and SLA penalty to optimize their SLA-related costs.
8. Conclusion, limitation and recommendation

In this chapter, the main findings and contribution of this master thesis are concluded. And we clarify the limitations of this work that the other researchers should be attentive. Finally we recommend the future work that researchers and scholars can explore.

8.1 Conclusion

In this research paper, we introduce an ARX-LLM (i.e., ARX model using log-likelihood maximization) model to predict the truncated workload in the cloud environment using the historical data. Different from the traditional ARX model that estimates parameters simply based on the data, the proposed method estimates parameters through setting initial parameter constraints and maximizing log-likelihood of observations. Furthermore, parameters and error terms in the standard ARX or ARMAX model can be negative, which potentially results in negative estimated output. This is not aligned with the requirements of estimated output of our case. In our proposed method, we set initial parameter constraints to ensure that the predicted result will always be positive so that the estimated values conform to the reality. And we proposed a log-likelihood maximization method to find out the appropriate parameters. The prediction result shows that the ARX-LLM model is accurate in terms of predicting the future workload and it shows better prediction accuracy than other prediction models, for instance, neural networking, MEAN, Second Order ARMA, and standard ARX model. Besides, the proposed model can be easily expanded to ARMAX, ARMA, ARIMA, and ARIMAX model etc. with changing or adding a few parameters.

After acquiring the estimated outcome using the proposed method, we test if the prediction result of the proposed model can reduce the percentage of SLA violations and SLA-related costs. In terms of SLA violation, a list of metrics of important SLA parameters are described in detail so that they can be adopted by cloud service providers to measure the SLA violation. Regarding SLA-related costs, three different cost models are developed based on different components in the cloud environment.

The result shows that the prediction result can significantly reduce the proportion of SLA violations. Unfortunately, it is not necessary to lower the SLA-related costs
either. Whether the prediction result of AR(MA)X-LLM model will decrease the SLA-related costs depends on the values of the price of renting the processors, values of penalty of SLA violations regarding run time and wait time. Given specific values of these factors, cloud providers are able to calculate if the AR(MA)X-LLM model can reduce SLA-related cost and they can take the corresponding actions to maximize their profit.

Generally, the AR(MA)X-LLM model enables the cloud service provider to precisely estimate the upcoming workload and it shows better prediction accuracy than other methods. And cloud service providers are able to reduce SLA violations as a result of AR(MA)X-LLM model, despite the fact that SLA-related cost cannot be necessarily reduced. Consequently, our approach (AR(MA)X-LLM model) can be considered efficient in terms of resource management and SLA violation reductions.

8.2 Limitation and recommendation

Even though the AR(MA)X-LLM model has good prediction accuracy, there are also some limitations in this research paper. Initially, the adopted data has some missing values of some attributes, which results in the acquirement of a limited number of exogenous inputs. It is potential that some other variables in the obtained dataset can significantly influence the prediction result, for instance, user ID and group ID can affect the prediction result. Unfortunately, those variables cannot be acquired because of some missing values. It is thereby suggested that a dataset without missing values can be investigated in the future work. And due to time limitations, some attributes in the obtained data probably cannot precisely measure the intended variables, which can also cause experimental mistakes. We encourage researchers to seek for new datasets that have enough parameters to measure all experimental variables. Then in terms of the AR(MA)X-LLM model, we mainly use statistical methods to perform the estimation, some but a few learning methods are also used though. It is subsequently advised to employ more learning methods such as regression tree and the forth, as well as some hybrid methods that combine statistical and learning methods. Another limitation is that we assume the error term of the model is a gamma distribution, which is reasonable but cannot be verified. This probably leads to prediction mistakes.
In our proposed method, finally 5 exogenous variables are used, which are potentially not enough. And we suggest future researchers to explore more exogenous variables that potentially have impacts on the prediction in theory. The AR(MA)X-LLM model also involves huge computing volume, this situation will be more difficult with the number of parameters keeping increasing. As a result, this problem is supposed to be attentive for future work and a trade-off analysis can probably be conducted.

Due to the parameters limitations of the experimental data, the function of calculating SLA-related cost does not consist of comprehensive elements that take all cost parameters into account. This can influence the accuracy of the final outcome. It is thereby advised to develop a novel cost model that can well fit to the experiment data, and the result will deliver more values to the stakeholders in the multi-cloud environment.

Even though the AR(MA)X-LLM can significantly reduce SLA violations, it is not necessary to reduce the SLA-related cost. This is another limitation of our work. And hereby, we strongly recommend the future researchers developing new workload prediction model that can decrease both SLA violation and SLA-related costs.
**Bibliography**


Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems. In *Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on* (pp. 887-894). IEEE.


Appendix A

This chapter presents the specific procedures of coping with the original experiment data to acquire the hourly workload to conduct the experimentation.

The unit of parameter SubmitTime originally is second, and it is divided by 3600 so that the unit of measurement of SubmitTime is hour. Then SubmitTime measured by hour is rounded up to an integer using the ROUNDUP function (i.e., roundup (number; num_digits)) in the Excel (in this case, num_digits is 0). And those integers are copied to another column and the repeated integers are deleted to prepare for the next step. After this step, the SUMIF function (SUMIF (range; criterion; [sum_range])) is utilized to calculate the workload of each hour. However, the workload of some hours is probably zero and they are not shown after executing above steps. As a consequence, a time-series measured by hour is created, which ranged from 1 to 8760 (the 365th day). And then IFERROR and VLOOKUP function (IFERROR(VLOOKUP(lookup_value;table_array;col_index_num;[range_lookup]));)) are together used to get the workload of all hours (0 means that if a certain hour can not be found, then workload of it is assigned 0).

Appendix B

This section describes the metrics of important SLA parameter so that the cloud service providers are able to measure SLA violations.

A. Service Response time

Service response time can be used to measure the efficiency of a service, i.e. how fast a service can be made available for usage. Various sub-factors determine the service response time, for instance, average response time, maximum response time promised by service provider, and proportion of time this response time level is missed.

- Average response time is calculated by \( \frac{\sum_{i} T_i}{n} \) where \( T_i \) is time between user \( i \) requested for a service and when it is actually available, and \( n \) is the total number of service requests.

- Maximum response time is the maximum promised response time by the Cloud provider for the service.

- Response time failure is measured by the proportion of occasions when the
response time was higher than the promised maximum response time. Consequently, it is measured by \( n' / n \times 100 \), where \( n' \) is the number of occasions when service provider failed to fulfill their promise.

B. Sustainability

Sustainability can be defined in two ways: regarding the service life cycle or environmental impact of the Cloud service employed. It is therefore decided to subdivide sustainability into two attributes: service sustainability and environmental sustainability.

- Service sustainability is defined as how many components of a service can be reused without changing evolution of user requirements. This also means that more sustainable service will have more features than required. As a result, service is measured by:
  \[
  \frac{\text{number of features provided by service}}{\text{number of features required by the customer}}
  \]

- Environmental sustainability can be calculated as the average carbon footprint of the service, which can be acquired from Carbon calculators such as PUE calculator.

C. Suitability

Suitability is defined as the degree to which a customer’s requirements are met by a Cloud provider. Now, there are two sub cases before defining suitability. First, if after filtering the Cloud providers, there are more than one Cloud provider which satisfy all the essential and non-essential requirements of customer, then all are suitable. Otherwise, if filtering results in an empty Cloud provider list, then those providers who satisfy essential features are chosen. In this case, suitability will be the degree the service features come closer to user requirements. The resultant metric is:

\[
\text{Suitability} = \frac{\text{number of non - essential features provided by service}}{\text{number of non - essential features required by the customer}}
\]

if only essential requirements are satisfied

= 1 if all features are satisfied

= 0 otherwise

D. Accuracy

The accuracy of the service functionality measures the degree of closeness to user expected actual value or result generated by using the service. For computational resources such as Virtual Machines, accuracy’s first indicator is the number of times
the Cloud provider deviated from a promised SLA. It is defined as the frequency of failure in fulfilling promised SLA in terms of Compute unit, network, and storage. If \( f_i \) is the number of times the Cloud provider fails to meet with promised values for user \( i \) over the service time \( T \), then accuracy frequency is defined as \( \sum f_i \) where \( n \) is the number of previous consumers. Another indicator of accuracy is the accuracy value which is defined by \( \sum (\alpha - \alpha_i) / \alpha_i \) where \( \alpha \) can be computational, network or storage unit of the service and \( T_i \) is service time \( T \) for user \( i \).

E. Transparency

Because of the fast evolution of Cloud service, transparency becomes increasingly important feature. It can be defined as a time for which the performance of the user’s application is influenced during a change in the service. Transparency can be calculated regarding frequency of such effect. Therefore, it can be measured by

\[
\sum \text{time for service affect}_i \over \sum \text{number of such occurrence}_n
\]

where \( n \) is the number of clients using the service and \( i \) denotes the customer.

F. Interoperability

Interoperability is the capability of a service to interact with other services provided either by the same provider or other providers. It is still determined to define interoperability even though it is more qualitative and based on user experience. Interoperability is defined as:

\[
\frac{\text{number of platforms offered by the provider}}{\text{number of platforms required by users for interoperability}}.
\]

G. Availability

Availability is the percentage of the time a user can access the service. It is measured by

\[
\frac{\text{total service time} - \text{total time for which service was not available}}{\text{total service time}}.
\]

H. Reliability

Reliability indicates how a service operates without failure during a given time and condition. Subsequently, it is defined depending on the mean time to failure promised by the Cloud provider and previous failures experienced by the customers. If \( \text{numfailure} \) is the number of users who experienced failure in the amount of time less than promised by the provider and \( n \) is the number of users. Let \( p_{\text{mtt}} \) be the promised mean time to failure. Reliability is then measured by:

\[
\frac{\text{promise} \times p_{\text{mtt}}}{p_{\text{mtt}} + \text{numfailure}}.
\]
Reliability = probability of violation × p_mttf = \((1 - \frac{num\text{failure}}{n}) \times p\text{_mttf}\)

Reliability of storage can be defined in terms of durability that is chance of failure of storage device.

1. **Stability**

Stability is defined as the variability in the performance of a service. For storage, it is the variance in the average read and write time. For computational resources, it is the deviation from the performance specified in SLA i.e., \(\sum \frac{\alpha_{\text{avvy},i} - \alpha_{\text{sla},i}}{T} \frac{T}{n}\) where \(\alpha\) can be computational unit, network unit or storage unit of the resource; \(\alpha_{\text{avvy},i}\) is the observed average performance of the user \(i\) who leased the Cloud service, \(\alpha_{\text{sla},i}\) is the promised values in the SLA; \(T\) is the service time while \(n\) is the total number of users.

2. **Cost**

Cost is based on two attributes: acquisition and on-going. It is defined as a volume based metric i.e. cost of one unit of CPU unit, storage, RAM, and network. Accordingly, if a VM is priced at \(p\) for cpu cpu unit, net network, data data, RAM for RAM, then the cost of VM is \(p \frac{\alpha_{\text{cpu}}^a + \alpha_{\text{net}}^b + \alpha_{\text{data}}^c + \alpha_{\text{RAM}}^d}{a + b + c + d = 1}\). The weight of each attribute can be quite diversified of different applications.

3. **Adaptability**

Adaptability is the ability of the service provider to adjust changes in the services based on customer’s request. It is defined as the time taken to adapt to changes or upgrading the service to next level.

4. **Elasticity**

Elasticity is defined in terms of how much a Cloud service can be scaled during peak times. This is defined by two attributes: mean time taken to expand or contract the service capacity, and maximum capacity of service. The capacity is the maximum number of compute unit that can be provided at peak times.

5. **Usability**

Usability refers to the ease of using a Cloud service. The components such as operability, learnability, installability and understandability can be quantified as the average time experienced by the previous users of the Cloud service to operate, learn, install and understand, respectively.
Besides the metrics of important SLA parameters listed above, Comuzzi et al. (2009) [13] also provide the metric of accessibility.

N. Accessibility

Assuming operation $O$ of service $S$; time $T_1$ as the beginning of monitoring time; time $T_2$ as the time of evaluating accessibility; monitoring duration $T = T_2 - T_1$; $R_a$ as the number of all invocations to $O$ during time $T$; $R_d$ as the number of invocations that were not served (i.e. were dropped) during time $T$; accessibility for operation $O$ is then defined as $C_O = (R_a - R_d)/R_a$.

As a matter of fact, above metrics of important SLA parameters are also confirmed by some other scholars and experts. For example, (Dingle, Knottenbelt, & Wang, 2008)[17] indicate the importance of metrics of availability, response time, and usability etc. Availability, time behavior and accuracy are also highlighted by[4].