Towards techniques, challenges and efforts of software as a service layer based on business applications in cloud environments

Received 31 July 2019
Revised 26 September 2019
10 December 2019
Accepted 11 December 2019

Mojgan Fardinpour
Department of Computer Engineering, Behbahan Branch, Islamic Azad University, Behbahan, Iran

Alireza Sadeghi Milani
Young Researchers and Elite Club, Tabriz Branch, Islamic Azad University, Tabriz, Iran, and

Monire Norouzi
Young Researchers and Elite Club, Islamshahr Branch, Islamic Azad University, Islamshahr, Iran

Abstract

Purpose – Cloud computing is qualified to present proper limitless storage and computation resources to users as services throughout the internet. Software as a service (SaaS) layer is the key paradigm perspective in the software layer of the cloud computing. SaaS is connected by business applications to access consumers on existing public, private and hybrid cloud models. This purpose of this paper is to present a discussion and analysis on the SaaS layer based on business applications in the cloud environment in form of a classical taxonomy to recognize the existing techniques, challenges and efforts.

Design/methodology/approach – Existing techniques, challenges and efforts are classified into four categories: platform-dependent, application-dependent, data-dependent and security-dependent mechanisms. The SaaS layer mechanisms are compared with each other according to the important factors such as the structural properties, quality of service metrics, applied algorithms and measurement tools.

Findings – The benefits and weaknesses of each research study are analyzed. In the comparison results, the authors observed that the application-based method, the non-heuristic algorithms, the business process method have the highest percentage of the usage in this literature.

Originality/value – The SaaS layer mechanisms based on business applications have some main features such as high accessibility, compatibility, reusability and collaboration to provide activated application and operation services for user with help of Web browsers. A comprehensive analysis was presented as originality on the SaaS layer mechanisms based on business applications for high level of the cloud environment that 46 peer-reviewed studies were considered.

Keywords Cloud computing, QoS, Business application, Software as a service (SaaS)

Paper type Research paper

1. Introduction

Cloud computing is established over the current progresses such as software components, virtualization equipment, distributed systems and service delivery in the internet

Kybernetes
Vol. 49 No. 12, 2020
pp. 2993-3018

0368-492X
DOI 10.1108/K-07-2019-0520

(Mulia *et al.*, 2013; Oliveira *et al.*, 2014). Classically, the cloud service tiers are presented in some main forms such as software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS) (Singh and Chana, 2016; Aliyu *et al.*, 2017; El Kafhali and Salah, 2018a). The existing services can be edited over a cloud client, which could be a Web browser application, mobile applications and so on (Navimipour *et al.*, 2017; Chiregi and Jafari Navimipour, 2017; Keshanchi *et al.*, 2017; Riahi and Krichen, 2018). The SaaS is an important standard method in the cloud computing (Amiri, 2017; Rodríguez *et al.*, 2016) that is connected to business applications to access customers on the existing cloud models such as public, private and hybrid (Kwok and Mohindra, 2008).

1.1 Brief motivation

The SaaS layer architecture has progressed from the resource provisioning model in software delivery process that contains four main procedures including modeling, discovery, monitoring and selection (Toosi *et al.*, 2018; Souri *et al.*, 2017). The SaaS layer architecture has some main features such as high accessibility, compatibility, reusability and collaboration to provide activated application and operation services for user with help of Web browsers (Arabnia and Tinetti, 2018). Also, the SaaS provides various business application services including business resource management, enterprise service planning and cluster resource management (Seethamraju, 2015; Venkata Krishna *et al.*, 2018). Figure 1 illustrates a main description of the SaaS layer architecture in the cloud computing that navigates application and operation services. There are five essential software elements' abilities that permits cloud services to reuse several cloud-based service categories as the main motivation of cloud computing (Balmukund *et al.*, 2017). These abilities are monitoring, scalability, security, configurability and multi tenancy (Imani *et al.*, 2017; Baker *et al.*, 2015).

There are some surveys and review studies that describe the SaaS and cloud challenges. For example, Jatoth *et al.* (2017) have presented a systematic literature-based review for quality of service (QoS) and aware service composition approaches according to the

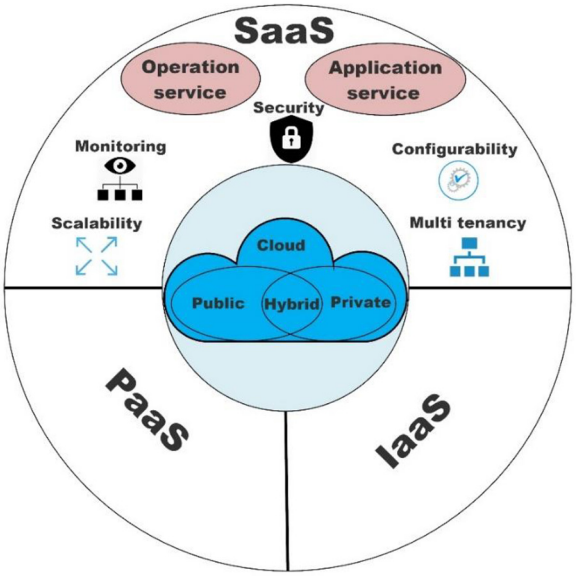

Figure 1.
The software as a service architecture

computational intelligence mechanisms. They have classified the existing methods into three categories that include heuristic, non-heuristic and meta-heuristic approaches. However, this paper suffers some weaknesses; for example, the papers selection procedure is not clear and the studies are not compared completely. Navimipour and Vakili (2017) presented a systematic review of the service composition approaches in cloud computing based on the algorithmic classification including framework-based, agent-based and heuristic-based approaches. The papers selected for the study were journal papers from 2012 to 2016 according to the QoS parameters, without any technical discussion and comparison. Banerjee and Jain (2014) proposed a survey on the SaaS features based on quality models such as pay per use, reusability, customizability, security and availability. The presented quality models are mapped into the SaaS architecture according to the SaaS developers such as three layers including customer, platform and application developer. Some defects of this survey include that the authors have not illustrated a side-by-side comparison between the represented quality models and the SaaS layers. Also, this survey has not mentioned the open issues and new challenges in the SaaS topic. Church and Goscinski (2014) have presented a technical survey on the service computing solutions in the SaaS for the mammalian genomic analysis. This article divides the SaaS approaches into grid, cluster and cloud systems to clarify the genomic analysis when high-performance computing (HPC) applications are executed on the distributed systems. In addition, some structure and behavior features of the SaaS approaches are discussed according to the mammalian genomic study requirements. The main weakness of this survey is that the authors have not mentioned the open challenges and opportunities for the SaaS approaches in the mammalian genomic analysis. Tsai *et al.* (2014) proposed a brief survey on the SaaS architecture including data-oriented, middleware-based PaaS-based and service-oriented architectures. Also, a mapping study is performed for analyzing the technical methods of the SaaS such as multi-tenancy, customization, redundancy and the scalability in this study. There are some defects for this survey as follows:

- the technical comparison of the SaaS architectures is not explained;
- the new challenges and opportunities are not discussed in the SaaS providers; and
- there is no a systematic and comprehensive review in this study for the SaaS approaches.

1.2 Main contributions

Because of accessibility of the cloud applications in business and social commitments, the SaaS layer has important role in the business applications. To the best of our knowledge, there are some technical surveys and review articles on the various parts of the cloud computing (Singh and Chana, 2016; Chiregi and Jafari Navimipour, 2017; Tsai *et al.*, 2014; Souri *et al.*, 2018b); however, they do not concentrate on the SaaS layer mechanisms for business applications in the cloud computing. This paper represents a discussion and analysis on the SaaS layer mechanisms based on business applications in the cloud environments and compares the transformations between declared instruments. Also, the SaaS layer mechanisms are divided into four main categories including platform-based, application-based, data-oriented and secure-based mechanisms. A taxonomy is provided to distinguish between considered SaaS layer mechanisms. The main contributions of this paper are as follows:

- classifying recent challenges of the SaaS layer mechanisms based on business applications in the cloud environments;

- poviding a systematic analysis on the presented SaaS layer mechanisms based on business applications in cloud environments;
- discovering technical aspects of the SaaS layer mechanisms based on business applications; and
- comparing the key aspects of the SaaS layer mechanisms based on business applications in the cloud environments for improving their weaknesses in the futures.

The rest of this research is structured as follows. The basic concepts and some related works are provided in Section 2. Also, we present the systematic article selection for comparing research papers in Section 3. Section 4 discusses the SaaS mechanisms in the cloud computing and classifies them. Section 5 presents the discussion and some open issues on this topic. Finally, Section 6 concludes this paper and presents some limitations.

2. Paper selection approach

This section demonstrates the research classification of the SaaS mechanisms (Charband and Jafari Navimipour, 2016). In this classification, some electronic databases such as ACM, Science Direct, Springer and IEEE have been used according to Table I.

By searching keywords of the important attributes, the following search keywords were defined (Navimipour and Vakili, 2017; Aznoli and Navimipour, 2017; Souri and Rahmani, 2014; Souri *et al.*, 2014; Jamshidi *et al.*, 2013; Kitchenham *et al.*, 2010):

"Software as a service" OR "SaaS" OR "Software application""OR "Business application") AND ("Cloud computing""OR "Cloud")

This systematic survey is designed at responding the following questions (Q) based on the objectives and possibilities of the research studies that were published in 2012-2018 (Souri *et al.*, 2018b):

Q1. Which mechanisms of the SaaS layer based on business applications are applied to the cloud computing?

Q2. What specified algorithms support the SaaS layer mechanisms of the cloud environment?

Q3. Which technical methods are selected for the SaaS layer mechanisms as the case study?

Q4. What are the QoS factors usually applied in the SaaS layer mechanisms of the cloud?

Q5. What popular modeling tools are used for evaluating the SaaS layer mechanisms?

Table I. Electronic databases used in article selection

Online database	URL Address	Journal search	Conference search
IEEE	http://ieeexplore.ieee.org/	✓	✘
ACM	http://dl.acm.org/	✓	✘
Elsevier	www.sciencedirect.com/	✓	✓
Springer	http://link.springer.com/	✓	✓
Emerald	www.emeraldinsight.com/	✓	✘
John Wiley	http://onlinelibrary.wiley.com/	✓	✘
Taylor and Francis	http://tandfonline.com	✓	✘

According to above technical questions, we will discuss existing research studies with mapping on the analytical technical questions. Also, we analyze and answer to these questions in Section 4.

Figure 2 displays the distribution of the research studies over time according to various publishers such as IEEE, Elsevier, Springer and Wiley. Because of the number of the published papers, we just analyze the journal articles and high-quality conference papers as the important and peer-reviewed studies for the SaaS mechanisms. According to variety of existing international conference, we just analyze some conferences of ScienceDirect and springer. Also, low-quality conference papers are omitted in this review. According to some important publication factors such as publication years, journal rank and article relevance, the 46 peer-reviewed studies were considered for further analysis of which the detail content is provided in Section 3. Figure 3 shows paper selection diagram according to inclusion and exclusion refinements.

3. Software as a service mechanisms in the cloud computing

This section presents a comprehensive review of the selected SaaS mechanisms in the cloud computing for specified research studies. We have divided the SaaS mechanisms into four main categories: application dependent, data dependent, security dependent and platform dependent approaches. Figure 4 displays a taxonomy of the SaaS mechanisms in the cloud. Each SaaS mechanisms include a set of the main methods.

In the application dependent mechanism, we categorize set of the methods into provisioning method, business process method and scheduling method. In this mechanism, application services are collaborated and analyzed in the cloud environments. Resource provisioning method provides the data assortment, arrangement and run-time controlling of the SaaS applications and PaaS resources to guarantee the performance of the SaaS providers. This method has the interactive communication between the cloud providers and cloud users which confirms the QoS factors (Zhang *et al.*, 2016).

The data dependent mechanism has two methods: business process method and partitioning method. This mechanism provides interactive services to communicate information of cloud data centers. The security dependent mechanism presents two main methods including signature-based method and trust-based method. The platform dependent mechanism has two key modeling methods including scheduling method and provisioning method. Partitioning method in the database cloud services is a main challenge that supports effectively to alleviate risks of the illegal access in the existing information

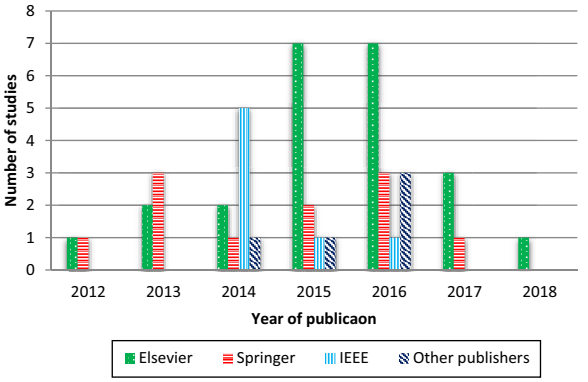

Figure 2.
Distribution of selected research studies by publishers

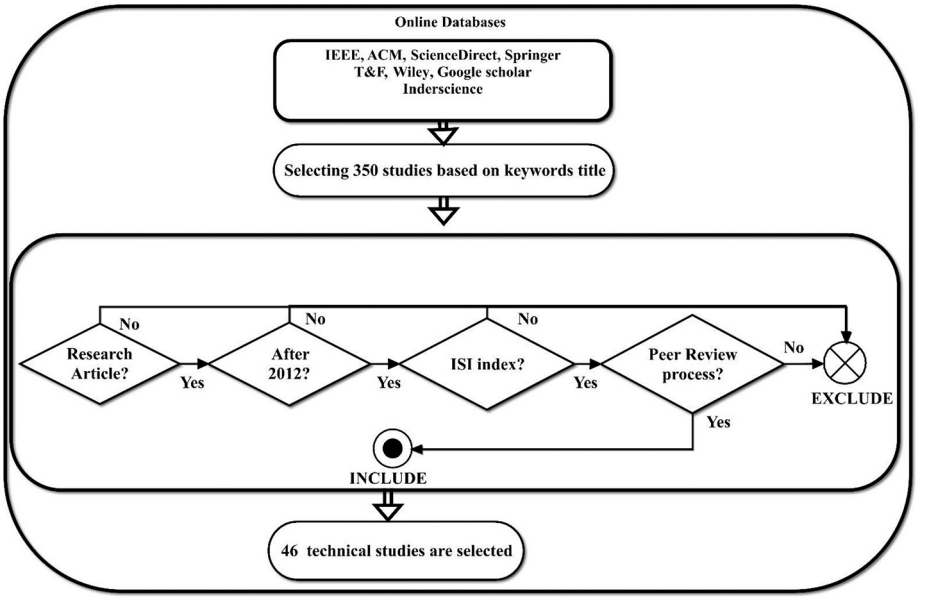

Figure 3.
Paper selection approach to find relevant studies

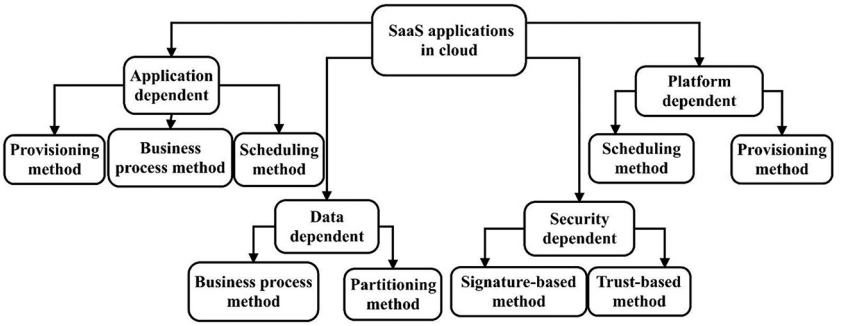

Figure 4.
The taxonomy of the SaaS mechanisms on the cloud environments

systems and cloud systems (Levitin *et al.*, 2017). Now, we describe the above taxonomy based on the illustrated SaaS mechanisms in the cloud.

3.1 Application-dependent software as a service mechanism

This subsection illustrates the application-based methods in the cloud. Also, the application dependent SaaS mechanisms will be compared according to structural properties and evaluation factors. SaaS applications are well-known as the Web-based software or on-demand software that facilitate an easy to use remote application hosting and delivery instead of installing and maintaining software.

In the first study, Amiri (2017) presented two algorithms to explain the complexity of the trusty application assignment problem. Determining the suitable application to install on

virtual provider and to allocate the users while minimizing total cost is examined in this paper. The second algorithm is used to defining every cluster as an application structure among the all probable structures. The wide computational study is directed using large data sets. The experimental results showed that the proposed algorithms have better performance rather than a standard branch-and-bound procedure. Evaluating the quality of the solution which is generated by the heuristic algorithms still needs to develop a new method.

Vázquez-Poletti *et al.* (2017) presented a two-tier explanation to reduce the resource order and to use reception control based on Markov chain methods. Validation process showed that available resources awareness is needed to manage request quality. The proposed method increased the probabilities of completing an order. The proposed reception control algorithms optimized the number of processed orders for a deal with limited resources. In contrast, the other applications pertaining to the simulation and modeling tools should be used to evaluate the proposed algorithms.

Vidhyalakshmi and Kumar (2017) presented a new structure to help the clienteles to effective and reliable SaaS evaluation performance for the assumption. Some technical abbreviations of analytic hierarchy methods are presented for the multi-criteria decision-making (MCDM) approach. This study used MCDM as a suitable approach for the composite decision-making procedure in the proposed structure to receiving the comparative feature priorities and allocating their weights. The allocated weights were determined based on estimated dependability amounts. The advantages of the proposed structure are enabling clienteles to execute SaaS process effectively based on business necessity and their feature priorities. The proposed structure helps the providers to realize their product status and the customers' expectations. But the network feature has been omitted from the feature list because of the responsibility issues.

Huang and Shen (2015) proposed a novel integrated approach to solve the service deployment problem. The service distribution problem receipts cost and the possible similarity between services into deliberation. The communication charges among machines modeled using two graphs as possible parallelism among services. These two graphs are combined into a single graph and the service placement problem is converted into an NP-Complete problem for the result. The experimental results determined that the proposed approaches decreased the service response time for composite SaaS applications.

Fan *et al.* (2015a) presented a personalization method for cloud services based on SaaS. The proposed framework is based on customer-side personalization, semantic method, modeling central client, data mining knowledge and recommendation device and dynamic cloud service composition. The advantages of this framework are supporting personalized services in multiple systems and offering a flexible solution for service personalization. But it is not mentioned how to build one solution for all systems.

Anselmi *et al.* (2014) proposed a new approach for a platform provider based on game-theoretic. The proposed cost sample contains a session of convenience functions includes incomes and disadvantages incurred based on the completed performance plane and the energy costs related to platform resources. The evaluation results demonstrated that the proposed solution methods are based on random initialization perform equally which measure the inefficiency because of PaaS and SaaS self-centered behavior. Extending the presented methods did not deliberate in this study.

Rezaei *et al.* (2014) studied the use of a new semantic collaborative framework for cloud SaaS systems. In the proposed framework, the main performers and collaborative mechanisms are clear for each one of the performers who contribute to semantic collaborative. Advantages of this study are defining semantic interoperability states for

SaaS systems, specifying the appropriate design for the proposed framework and constructing an implementation structure for SaaS systems. But the evaluation methods for the proposed framework assessment in SaaS level have not been mentioned in this study.

Yang *et al.* (2015) presented a novel model of SaaS Readiness recommending that the organizational clients need to prepare for technical, organizational and environmental features for the assumption of SaaS using this model. The simulation results supported all of the important existing features for SaaS adoption. The main advantage of this study is extending the TOE (technology–organization–environment) mechanism using formative theories at various levels of technical, organizational and environmental aspects of SaaS assumption to achieve the psychological influences.

Li *et al.* (2017) presented resource allocation method based on optimization for SaaS method in the cloud. The optimization decomposition method used to solve the cloud service allocation for filling the cloud user's requirements and the cloud providers' incomes. The authors also proposed a SaaS cloud resource allocation algorithm for SaaS cloud application. Some tests deliberated for comparing the performance of the presented algorithm with two associated algorithms. In addition, payment, resource consumption and execution ratio in competition with different number of request have been studied in the experiments. The test results showed that the proposed algorithm exhibits better performance obtaining the tradeoff between the system performance and the cost with SaaS clients' particular necessity. But there is not an improvement of the cloud market consideration on which the presented method can be validated.

Seethamraju (2015) studied the challenges of the SaaS systems in supporting middle-sized enterprises. This study outcomes are as follows: defining features in determining to accept SaaS software seller's popularity in the business, the possible readiness of the seller to cover the client throughout the product life sequence, the seller's contribution in the co-creation of value for clients and the general assistance of applying an integrated enterprise resource planning. According to this study, competitive forces faced by the creativity, external aspects, apprehensions in fields of security and system privacy have no effect on adoption conclusion which is the main advantage of this paper.

Candeia *et al.* (2015) developed a function model considering business features related to presenting a SaaS application. This model was designed according two proposed heuristics algorithms. Both heuristics algorithms used to input the estimation of a future workload for an interval time. This estimation achieved from executed historical data of the SaaS application. Analyzing the results showed that all heuristic algorithms develop SaaS provider function in comparison to on-demand instances. Two proposed algorithms improved SaaS provider. But there is not an evaluation method for large workload prediction errors situation.

Ma and Kauffman (2014) proposed a new model for SaaS in a competitive market. The authors focused on numerous specifications of SaaS competition including differences in seller presentations and short information about application performance. Using a game-theoretic model in this study has several advantages. But a dynamic model for evaluating the switching time is needed to represent a customer's decision-making procedure.

Sharif *et al.* (2017) presented a SaaS scheduling broker to protect privacy. The workflows' tasks scheduling performs under customers' deadlines in this framework. This approach evaluated using a real workflow on a VMware tool in the hybrid cloud. The evaluation results revealed that under the proposed scheduling strategies the total cost decreased in comparison with the alternatives. But the workflow structure affection on choosing an efficient method to workflows scheduling in heterogeneous locations did not evaluate.

Wu *et al.* (2014) proposed user-driven resource algorithms based on service level agreement (SLA) to decrease the price. The proposed provider algorithms considered user profiles and providers' quality factors such as response time to manage user requirements dynamically. The authors considered the user-side factors such as the number of upgrade requirements and structure level factors such as the service launching time to compare algorithms. The simulation results displayed that the proposed algorithms reduced the total price and the number of SLA contravention. The SLA negotiation process in the cloud for user approval levels development and other pricing approaches to minimize the cost factor for SaaS providers did not discuss in this study.

Huang (2017) analyzed the hitherto implicit intellectual structure of the cloud computing using the methods of patent clustering and co-word clustering. To improve the quality of patent clustering, the Taguchi method used by the E–I index as the quality specifications. Three effective factors on the quality including the cut-off value for dichotomization, the number of terms, and the scarcity are examined in this study. An adapted co-word algorithm proposed to professionally detect all important R&D themes as a decision-making problem in SaaS, removing the need to analyze the relationship strength, is the obvious advantage of this study. Discovering the relation between R&D themes and competition between corporations using the proposed method, factor analysis, multidimensional scaling analysis and network investigation is another advantage of this research. In contrast, overlapped clustering of the cloud patents has not been examined yet.

Fan *et al.* (2015b) proposed innovative approach improvements in launching personalization for the user-side on cloud services based on SaaS. This approach has semantics, a private client model, server-side edge module, synchronization instrument, hybrid viewpoint section with some algorithms and user-side MapReduce with Web workers. Moreover, this approach delivered a complete sample for the semantic user-side personalization method and evaluated with some metrics. The outcomes proved the correctness of the proposed approach with effective high performance and recommendation utility satisfaction.

Iranpour and Sharifian (2016) introduced a smart controller on an field gate array to control the admission rate of demands for a SaaS server in a cloud environment. The brain emotional intelligent controller considered as the foundation of the controller, which controls the admission rate of the server. A conceptual design of a server offered and simulated showing the behavioral features of an actual server. The system executed on proposed hardware. The controllers' effectiveness compared in terms of server consumption, admission rate and the drop rate of appeals and the ability of the controllers. Achieving an improvement of about 14 per cent in the number of self-confessed requests in a worse case by applying intelligent controller is the main advantage of this paper.

El Kafhali and Salah (2018b) presented a diagnostic model to manage the SaaS cloud services dynamically. This model provides a queuing mathematical approach to estimate the lowest number of multi-core VMs for conforming existed QoS parameters. The authors used discrete event simulator (DES) environment to evaluate the proposed model.

3.1.1 Analysis of the reviewed application-dependent mechanisms. Table II illustrates the categorization of the above studies and the effective factors to analyze the application-dependent SaaS mechanisms in the cloud. The applied method, measurement environment, case study and applied algorithm for each study are compared and illustrated.

Table III explains a side-by-side valuation for the above studies using evaluation factors in the SaaS of the cloud computing. The following factors include availability, time, scalability, cost and reliability. Also, in the application-dependent mechanisms, most research studies evaluated their proposed approach in the time and cost conditions.

Table II.
Categorization of recent studies and other information in application-dependent mechanisms

Research	Method	Implementation environment	Case study	Algorithm
Amiri (2017)	Customer Business Process	Visual C#	Reliable application placement	Heuristic, probabilistic greedy algorithm
Vázquez-Poletti *et al.* (2017)	Provisioning	Matlab	large–scale Bayesian inference problems	Non-heuristic
Vidhyalakshmi and Kumar (2017)	Customer business process	Analytic Hierarchy Process Tool	simple client-server model	Non-heuristic
Huang and Shen (2015)	Provisioning	ASP.Net via Amazon EC2 environment	Composite SaaS application	Non-heuristic
Fan *et al.* (2015a)	Provisioning	ASP.Net via Amazon EC2 environment	User's profile and activities	Non-heuristic
Anselmi *et al.* (2014)	Provisioning	Visual C#	Transactional Web-services	Heuristic
Rezaei *et al.* (2014)	Provisioning	ClouSim	Cloud SaaS systems	Non-heuristic
Yang *et al.* (2015)	Customer business process	MATLAB	173 organizations business catalogs and observations	Heuristic machine learning
Li *et al.* (2017)	Provisioning	CloudSim	SaaS Cloud Application	Non-heuristic
Seethamraju (2015)	Customer business process	Vanilla software	Multiple organizations such as SAP, Oracle and Microsoft	Non-heuristic
Candeia *et al.* (2015)	Customer business process	GEIST	IaaS provider workloads	Heuristic
Ma and Kauffman (2014)	Customer business process	–	SaaS markets	Non-heuristic
Sharif *et al.* (2017)	Scheduling	CloudSim	Workflow scheduling broker	Heuristic
Wu *et al.* (2014)	Provisioning	CloudSim	Data center with 500 physical machines from Amazon EC2	Heuristic
Huang (2017)	Customer business process	Matlab	Relation between R&D themes	Heuristic machine learning
Fan *et al.* (2015b)	Customer business process	ASP.NET MVC 3 on the Amazon EC2	Music recommendation data set	Heuristic machine learning
Iranpour and Sharifian (2016)	Customer business process	Matlab	The SLA requests	Heuristic machine lLearning
El Kafhali and Salah (2018b)	Provisioning	DES simulator	Estimating the lowest number of multi-core VMs	Non-heuristic/ queuing networks

(*continued*)

Table II.

Research	Method	Implementation environment	Case study	Algorithm
Banerjee *et al.* (2013)	Scheduling	C++	A cloud compiler for scheduling SaaS applications	Non-heuristic
Liao *et al.* (2017)	Provisioning	–	Resource provisioning for load changes in SaaS cloud service	Meta-heuristic/Genetic algorithm
Mohammad *et al.* (2013)	Customer business process	C++, Java	Biological evolution of SaaS applications	Non-heuristic
Fowley and Pahl (2018)	Customer business process	C#	Cost-aware migration approach	Non-heuristic
Bellavista *et al.* (2013)	Provisioning	WordPress	Provisioning management approach	Non-heuristic
Zhang *et al.* (2013)	Customer business process	Java	Business rule engine for SaaS applciation	Formal/Decision rules

Research	Availability	Time	Scalability	Cost	Reliability
Amiri (2017)	✓	✓	✘	✓	✘
Vázquez-Poletti *et al.* (2017)	✓	✓	✘	✓	✘
Vidhyalakshmi and Kumar (2017)	✓	✓	✓	✓	✓
Huang and Shen (2015)	✘	✓	✘	✓	✘
Fan *et al.* (2015a)	✘	✓	✘	✘	✘
Anselmi *et al.* (2014)	✘	✓	✓	✓	✘
Rezaei *et al.* (2014)	✘	✓	✘	✓	✘
Yang *et al.* (2015)	✘	✓	✘	✓	✓
Li *et al.* (2017)	✘	✓	✘	✓	✘
Seethamraju (2015)	✘	✓	✓	✓	✓
Candeia *et al.* (2015)	✓	✓	✘	✓	✘
Ma and Kauffman (2014)	✘	✓	✘	✓	✘
Sharif *et al.* (2017)	✘	✓	✘	✓	✘
Wu *et al.* (2014)	✓	✓	✘	✓	✘
Huang (2017)	✘	✘	✘	✘	✘
Fan *et al.* (2015b)	✘	✓	✘	✘	✘
Iranpour and Sharifian (2016)	✘	✓	✓	✘	✘
El Kafhali and Salah (2018b)	✘	✓	✓	✓	✘
Banerjee *et al.* (2013)	✘	✓	✘	✘	✘
Liao *et al.* (2017)	✘	✓	✘	✓	✘
Mohammad *et al.* (2013)	✘	✓	✘	✘	✘
Fowley and Pahl (2018)	✘	✘	✘	✓	✘
Bellavista *et al.* (2013)	✓	✓	✘	✘	✘
Zhang *et al.* (2013)	✘	✓	✘	✘	✘

Table III.
A side-by-side comparison of the existing evaluation factors in the application-dependent mechanism

3.2 Data-dependent software as a service mechanism

This subsection illustrates the data-oriented mechanisms. Finally, the existing studies on this mechanism will be compared according to structural properties and evaluation factors.

Lin *et al.* (2015) presented a new scheduling framework including a set of scheduling algorithms obtained from the conservative backfilling algorithm. Prevention in the job waiting for lines never occur using Two-Tier Strict Backfilling algorithm proposed in this study. Simulation results indicated that proposed framework decreased the job turn-around the mean turn-around time of high-priority approaches. However, the off-line description of the scheduling problem needs an ideal algorithm.

Martins *et al.* (2016) introduced a novel framework including technology, organization and environmental views, the new specification and institutional stress. In addition, some features such as cost reserves, relative benefit and complexity affect the SaaS adaption. This study did not examine various type of SaaS-based application.

Church *et al.* (2015) presented a new structure for the organization of SaaS clouds to support scientific research. This structure expresses the cloud usability to admission the cloud and executes distributed applications. A novel resource selection method planned to automate complex deployment methods. The structure and resource selection method was implemented in the Uncinus tool. The implementation results revealed that Uncinus decreased the workflow running time while occurring delays. But the Uncinus tool should reduce the deployment and publication processes to make the middleware library.

Ali *et al.* (2016) presented a Cloud Interoperability Broker framework in the cloud SaaS level. The proposed framework implementation results showed that using the proposed framework mapping information from one cloud provider to other provider is possible. Moreover, it is proved that the migration was completed according to a specified mapping

model. But different features including accuracy, performance and time should be evaluated too.

Kim *et al.* (2012) presented a new solution for the associated problems using master table and code. The master table is dependent on the development master table, and some other client information tables are associated to form the client information. The master table incorporated numerous software databases. Also, the proposed ASP-based software can transfer information from the master table to SaaS environment. Using the master table and code to the integrated system implementation is the advantage of this paper.

Chen *et al.* (2017) studied the ideas of specialists and of three case businesses in internet application areas. In cloud customer relationship management (CRM) projects, two multi-criteria decision-making examination tools are DEMATEL[1]-based analytical network processes (ANP) and the VIKOR[2] methods which do not need previous expectations to discover the weights and presentations among project risk, project management and organizational performance. The experimental results displayed that the extreme criterion of qualified weight is mainly related to the risk measurement, representing experts' assessments of project risk. Moreover, the cloud CRM specialists and businesses discovered that financial performance should be better during the course of a project. The findings of this study offer a valuable reference for the cloud CRM internet service explanations. The advantage of this study is discussing the significance of the cloud CRM project risk management and performance.

Cheng *et al.* (2015) proposed a SaaS private cloud-based construction quality collaboration. The whole structure of the proposed framework introduced completely in this research. In addition, the authors designed and developed the system functions and structure, namely, the quality management section, quality examination section and quality receipt section. The new framework was compared to the old framework. The experimental results showed that the proposed method has good perspectives and social efficiency.

Cho and Chan (2015) proposed a new structure to assess SaaS adoption through four perspectives. This framework formulated theories to forecast the difference in SaaS adoption for commercial processes. This framework evaluated using data from 269 corporations through various activities in Hong Kong. The assessment results support the proposed integrative approach. Supposed benefit of cost has a confident impact on SaaS approach for commercial jobs. Also, supposed quality of service has a main effect, and management approach toward possession and control has a negative impact on SaaS adoption for both categories of processes.

Li *et al.* (2016) proposed a partition model configuration and partition method for the SaaS multi-tenant stored in the assigned scheme. The Relevance Matrix defined using the transactions admission information in the dynamic development of the system scale and then an abstract weighted graph is determined. Data partition model tested by experiments in environments with numerous nodes and the results showed that the number of distributed operations span partitions increased significantly. Using the partition algorithm is the main advantage of this study in system performance and scalability compared to other partition strategies. But data caching and data placement approaches on numerous nodes in the cloud did not examine.

Ge and Huang (2014) analyzed the use of stochastic frontier analysis (SFA) to examine the financial prudence of scale in SaaS firms while contrasting the productivity differences between SaaS applications and traditional software. Moreover, the productivity analysis showed that SaaS firms have smaller prudence of scale than traditional software. Many other features of the SaaS model such as customization should be studied, too.

Afify *et al.* (2017) introduced a new approach to the service reputation calculation from the customer reactions. SaaS Recommender (SaaSRec) addresses many challenges met by the generic recommender systems. Furthermore, the SaaSRec delivered a combined explanation for the recommended services to maximize the client's receipt. The evaluation results indicated that the proposed approach surpassed the other collaboration filtering-based recommendation methods, which is the obvious advantage of this study. This performance enhancement verified using different matrix density points. But the cloud providers' social network rating should be considered.

3.2.1 Analysis of the reviewed data-dependent mechanisms. Table IV illustrates the categorization of the above studies and the effective factors to analyze the data-dependent SaaS mechanisms in the cloud.

Table V explains a side-by-side valuation for the above studies using evaluation factors in the cloud. The following factors include availability, time, scalability and cost. Also, in the data-dependent mechanism, most research studies evaluated their proposed approach in the time and cost conditions.

3.3 Platform-dependent software as a service mechanism

This subsection illustrates the platform dependent mechanism. Moreover, the platform-based mechanisms with three papers will be compared according to structural properties and evaluation factors.

Fortino *et al.* (2014) proposed a Cloud-based SaaS architecture called BodyCloud for the management of body sensor and the complete life cycle of the data analysis. BodyCloud provided a framework to build and organize requests based on society body sensor networks. The BodyCloud methodology provides a flexible programming model adjusted on a few Web-based programming concepts. The ECGaaS atop BodyCloud implementation results demonstrated that proofing the efficiency and usability of the system is the obvious advantage of this study. But there is a need for the description of an extensible module distributed between the cloud-side and structure-side. Moreover, the global definition security framework for the BodyCloud methodology is needed.

Motavaselalhagh *et al.* (2015) presented a scheduling algorithm for effective resource allocation and maximize profit and customer satisfaction level (CSL) for SaaS suppliers. The scheduling algorithms consist of knowledge-based adaptable admission control. The simulation outcome specified that the suggested algorithm provided a significant improvement by cost saving.

Liu *et al.* (2014) presented an analysis method to optimize the power-performance in SaaS cloud. The goals of this research are maximizing the operating income when serving heterogeneous SaaS applications with unpredictable user requests and minimizing the power utilization when processing the client requirements. To reach these purposes, the authors made a unified profit-maximizing objective to equally consider income and cost in an economic opinion. The advantages of this paper are having both the optimization in the cost-effective power-performance tradeoff and constancy in robustness and adaptation to time-varying and bursty client requirements which succeeded by the proposed control structure. More effective cost reduction mechanisms should be explored by altering the number of activated servers in the data center dynamically.

3.3.1 Analysis of the reviewed platform- dependent mechanism. Table VI illustrates the categorization of the above studies and the effective factors to analyze the platform-dependent method in the cloud.

Table VII explains a side-by-side valuation for the above studies using evaluation factors in the cloud. The following factors include availability, time, scalability, cost and reliability.

Table IV. Categorization of recent studies and other information in the data-dependent mechanism

Research	Method	Implementation environment	Case study	Algorithm
Lin *et al.* (2015)	Partitioning	CSIM.20	Newly submitted jobs	Heuristic
Martins *et al.* (2016)	Customer business process	Kolmogorov Smirnov test-bed	Data collected from 265 firms	Non-heuristic
Church *et al.* (2015)	Partitioning	Uncinus tool	Genomics case study	Non-heuristic
Ali *et al.* (2016)	Customer business process	Visual C#	Microsoft Dynamics CRM 2015 and Sugar CRM SaaS providers	Non-heuristic
Kim *et al.* (2012)	Customer business pProcess	ASP.net	Four existing software's based on a Web portal site	Non-heuristic
Chen *et al.* (2017)	Customer business process	C#	cloud CRM experts data set	Non-heuristic
Cheng *et al.* (2015)	Customer business process	C#	Developed in construction quality supervision in Wuhan city, China	Non-heuristic
Cho and Chan (2015)	Customer business process	MATLAB	Data set from 269 companies in Hong Kong	Non-heuristic
Li *et al.* (2016)	Partitioning	Cloudsim	Experimental data set from the research group project	Non-heuristic
Ge and Huang (2014)	Customer business process	C#	The historical financial data of publicly listed firms	Non-heuristic
Afify *et al.* (2017)	Customer business process	MATLAB	WS-DREAM data set as a real-world services data set	Heuristic machine learning
Gutierrez-Milla *et al.* (2015)	Partitioning	C	Modeling bi-dimensional spaces	Non-heuristic/Manhattan and chessboard calculation
Meena *et al.* (2016)	Customer business process	C#	Managing image storages in cloud	Non-heuristic/wavelet algorithm

Also, in the platform-dependent mechanism, most research studies evaluated their proposed approach in the time and scalability conditions.

3.4 Security-dependent software as a service mechanism

This subsection illustrates the security dependent mechanisms. Also, the existing research studies in this mechanism will be compared according to structural properties and evaluation factors.

Cotroneo *et al.* (2016) introduced a structure to analyze large volumes of security alarms and to support the automated recognition of the alarms root reasons. The structure accepts weighting and theoretical clustering methods to fill the hole between the unstructured textual alarms and the decision tree formalization. The main advantage of this paper is that filtering step of the structure reduced the volume of the UNIX alarms.

Tang and Liu (2015) proposed a holistic model to identify the security controls and help the organization in the selection of a reliable service provider. Moreover, a logic method

Table V.
A side-by-side comparison of the existing evaluation factors in the data-dependent mechanism

Research	Availability	Time	Scalability	Cost	Reliability
Lin *et al.* (2015)	✓	✓	✘	✘	✘
Martins *et al.* (2016)	✓	✓	✘	✘	✓
Church *et al.* (2015)	✓	✓	✓	✓	✘
Ali *et al.* (2016)	✘	✓	✘	✘	✘
Kim *et al.* (2012)	✓	✓	✘	✓	✘
Chen *et al.* (2017)	✘	✓	✘	✓	✓
Cheng *et al.* (2015)	✘	✓	✓	✓	✘
Cho and Chan (2015)	✓	✓	✓	✓	✓
Li *et al.* (2016)	✓	✓	✓	✓	✓
Ge and Huang (2014)	✘	✓	✘	✓	✓
Afify *et al.* (2017)	✘	✓	✘	✓	✘
Gutierrez-Milla *et al.* (2015)	✘	✓	✘	✘	✓
Meena *et al.* (2016)	✘	✓	✓	✘	✘

Table VI.
Categorization of recent studies and other information in the platform-dependent mechanism

Research	Method	Implementation environment	Case study	Algorithm
Fortino *et al.* (2014)	WSN platform	–	ECG (electrocardiogram) as a Service	Non-heuristic
Motavaselalhagh *et al.* (2015)	Scheduling platform	Cloudsim	Social data for 300 user	Heuristic machine learning
Liu *et al.* (2014)	Provisioning	MATLAB	real-world datacenter	Non-heuristic

Table VII.
A side-by-side comparison of the existing evaluation factors in the platform-dependent mechanism

Research	Availability	Time	Scalability	Cost	Reliability
Fortino *et al.* (2014)	✓	✓	✓	✘	✘
Motavaselalhagh *et al.* (2015)	✘	✓	✓	✓	✘
Liu *et al.* (2014)	✘	✓	✓	✓	✓

proposed to help an organization and finding which security controls are required in its special SaaS context. Revealing that the organized and systematic approach proposed by the model is an objective and effective process to save organization's time is the advantage of this research.

Goode *et al.* (2015) presented a former understanding of the security role in SaaS customer pleasure. A former model of SaaS satisfaction is modified in this study. Recognizing the role of security in the SaaS model is the advantage of this paper. Examining the security requirements from the cloud provider for understanding how the SaaS providers navigate security policy is still needed.

Almorsy *et al.* (2013) introduced a model-driven security designing methodology for multi-tenant, cloud-hosted SaaS requests called MDSE@R. The proposed approach validated by applying it to some Web applications. The advantage of MDSE@R is bridging the gap between two specifications service description models and security specification models by combining the service and security prototypes into a combined service security model.

Du *et al.* (2014) presented a novel integrated service attestation for multitenant SaaS systems called IntTest. The IntTest occupations randomized replay-based reliability check to confirm the integrity of distributed service. A prototype of the IntTest system implemented and verified on a commercial data stream processing framework running inside a construction virtualized cloud computing structure. The advantage of this paper is that the IntTest reached higher pinpointing correctness than existing alternative structures upon experimental results.

3.4.1 Analysis of the reviewed secure-dependent mechanism. Table VIII illustrates the categorization of the above studies and the effective factors to analyze the security dependent method in the cloud.

Table IX explains a side-by-side valuation for the above studies using evaluation factors in the cloud. The following factors include availability, time, scalability, cost and reliability. Also, in this mechanism, most research studies evaluated their proposed approach in the time and scalability factors.

4. Discussion

Previous sections described the review process of the selected studies in the SaaS mechanisms in the cloud computing. In this section, a statistical analysis of declared SaaS mechanisms in the cloud is deliberated. Also, we present the analytical reports of the technical questions raised in Section 2 as follows:

Research	Method	Implementation environment	Case study	Method
Cotroneo *et al.* (2016)	Secure signature	Matlab	Two data sets named Unix and Windows contain 163,370 alerts	Heuristic machine learning
Tang and Liu (2015)	Secure trust	–	Canadian school board	Non-heuristic
Goode *et al.* (2015)	Secure trust	Unmediated modeling	Korean ASP (application service provision)	Heuristic mchine learning
Almorsy *et al.* (2013)	Secure signature	–	Seven web-based applications	Non-heuristic
Du *et al.* (2014)	Secure trust	CloudSim	IBM System S stream processing applications	Non-heuristic

Table VIII. Categorization of recent studies and other information in security-dependent mechanism

Q1. Which mechanisms of the SaaS layer based on business applications are applied to the cloud computing?

Figure 5 presents a comparison side of the SaaS mechanisms up to now according to the proposed taxonomy in Section 4. In this literature, there are 4 SaaS mechanisms that include application-dependent, platform-dependent, security-dependent and data-dependent mechanisms. The application-dependent mechanism has the highest percentage of the usage with 46 per cent in the literature. Of course, the data-dependent has 30 per cent, the security-dependent has 16 per cent and the platform-dependent has 8 per cent usage in the cloud computing.

Q2. What specified algorithms support the SaaS layer mechanisms of the cloud environment?

Figure 6 presents a comparison side of the specified algorithms including heuristic, non-heuristic and hybrid algorithms. In this literature, the non-heuristic algorithms have the

Table IX.
A side-by-side comparison of the existing evaluation factors in the security-dependent mechanism

Research	Availability	Time	Scalability	Cost	Reliability
Cotroneo *et al.* (2016)	✘	✓	✓	✘	✓
Tang and Liu (2015)	✓	✓	✘	✓	✘
Goode *et al.* (2015)	✘	✘	✓	✘	✓
Almorsy *et al.* (2013)	✘	✓	✓	✓	✘
Du *et al.* (2014)	✘	✓	✓	✘	✘

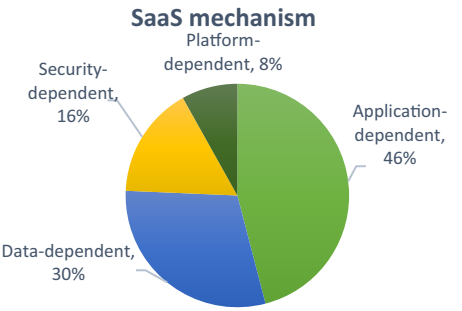

Figure 5.
The SaaS mechanisms in the cloud computing

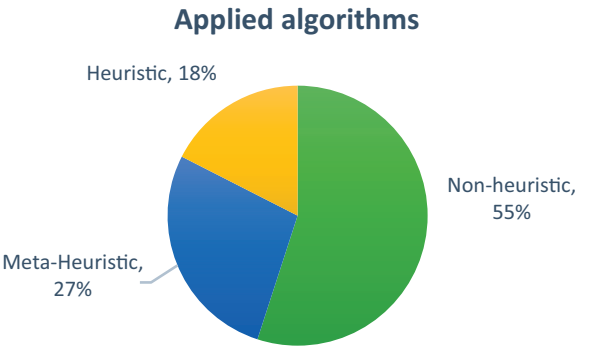

Figure 6.
The specified algorithms in the SaaS of the cloud computing

highest percentage of the usage by 55 per cent in the literature. Of course, meta-heuristic has 27 per cent and heuristic has 18 per cent usage in the cloud computing.

Q3. Which methods are selected for SaaS layer mechanisms as the case study?

Figure 7 presents a comparison side of the technical methods including provisioning, business process, scheduling, partitioning, signature, the trust and the WSN methods. In this study, the business process method has the highest percentage of the usage, that is, 49 per cent. Of course, the provisioning method has 24 per cent, trust method has 11 per cent, partitioning has 8 per cent, scheduling has 5 per cent and the signature method has 5 per cent usage in the cloud computing.

Q4. What are the evaluation factors usually applied in SaaS layer mechanisms of the cloud?

Figure 8 presents a comparison side of the evaluation factors including availability, time, scalability, cost and reliability factors. In this study, the time factor has the highest percentage of the usage of 35 per cent in the literature. Of course, the cost has 25 per cent, scalability has 14 per cent, availability has 14 per cent and reliability has 12 per cent usage in the cloud computing.

Q5. What popular modeling tools are used for evaluating the SaaS layer mechanisms?

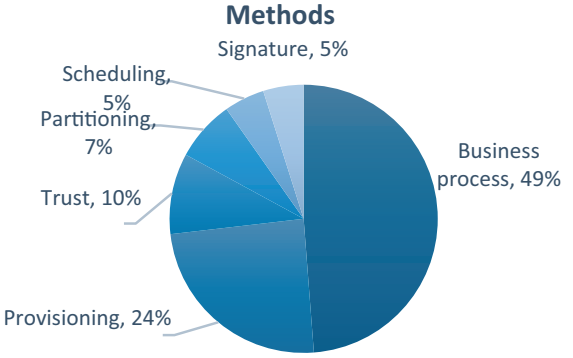

Figure 7.
The methods of SaaS in the cloud

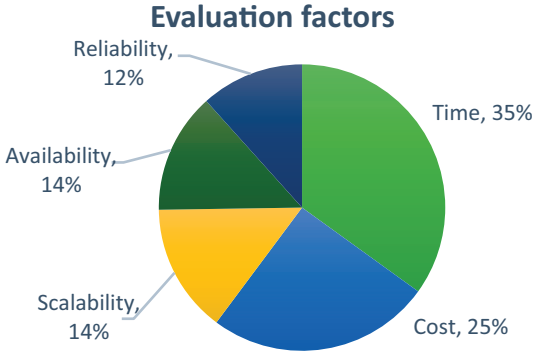

Figure 8.
The evaluation factors of SaaS in the cloud

Figure 9 describes a comparison side of the modeling and simulation tools for evaluating the SaaS mechanisms. In this study, the CloudSim has the highest percentage of the usage of 29 per cent in the literature. Of course, MATLAB has 20 per cent, Visual C# has 20 per cent, ASP.Net has 8 per cent, Analytical Hierarchy Process (AHP) has 6 per cent, GEIST tool has 6 per cent and the other tools has 11 per cent usage in the cloud computing.

Table X depicts the number of published papers in each scientific journal per publisher. The *FGCS* journal has five publications in this topic. The *TPDS* journal has three article

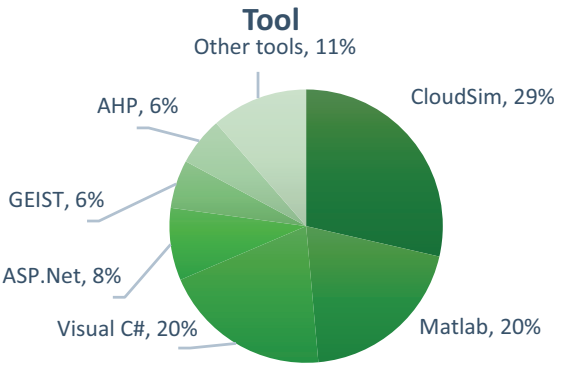

Figure 9.
The measurement and simulation tools of SaaS in the cloud

Publisher	Journal	No.
Elsevier	*Future Generation Computer Systems (FGCS)*	5
Elsevier	*Journal of network and computer application*	2
Elsevier	*Computers in Human Behavior*	2
Elsevier	*Journal of Systems and Software*	1
Elsevier	*European Journal of Operational Research*	1
Elsevier	*Expert Systems with Applications*	1
Elsevier	*Computers and Mathematics with Applications*	1
Elsevier	*Computer Physics Communications*	1
Elsevier	*Computer Security*	1
Elsevier	*Decision Support Systems*	1
Springer	*Information Systems Frontiers*	2
Springer	*Journal of Scheduling*	1
Springer	*Information Technology and Management*	1
Springer	*Automated Software Engineering*	1
Springer	*The Journal of Supercomputing*	1
Springer	*Journal of Intelligent and Robotic Systems*	1
Springer	*Soft Computing*	1
Springer	*Human-Centric Computing and Information Sciences*	1
IEEE	*IEEE Transactions on Parallel and Distributed Systems (TPDS)*	3
IEEE	*IEEE Transactions on Engineering Management*	2
IEEE	*IEEE Transactions on Services Computing*	1
IEEE	*IEEE Transactions on Cloud Computing*	1
Taylor and Francis	*Technology Analysis and Strategic Management*	1
John Wiley	*Concurrency and Computation: Practice and Experience*	2
SAGE	*Transactions of the Institute of Measurement and Control*	1
SAGE	*Information Development*	1

Table X.
The number of articles in each journal

publications in the SaaS approaches. Also, *Elsevier* publications have must ethics in the SaaS approaches of the cloud computing.

5. Open issues

In this section, some challengeable perspectives are presented for the SaaS mechanisms in the cloud computing.

- Interoperability (Rezaei *et al.*, 2014; Arunkumar and Venkataraman, 2015) is a main challenge for communicating SaaS applications in the large-scale cloud environments. SaaS interoperability manages an integrated architecture to interconnect the SaaS providers with other cloud providers. Data replication, data migration, service consolidation are some key features to evaluate the SaaS interoperability issue.
- Multi-tenancy as the important issue allows instance SaaS applications to serve multiple users by sharing existing resources. This issue supports the entity and integrity factors to facilitate user requests. Some key challenges of this open issue include trust management between users and the SaaS applications, cost-effective factor for data architecture of multi-tenant systems and the robustness of the applied SaaS applications for sharing information.
- Scalability is an emerged paradigm for the SaaS layer in the cloud computing as the important open issue. By increasing the customer requests and the varieties of the existing services and having the scalable service provider to represent software applications is more and more challengeable topic. Some key challenges of this issue include cost-efficient shared services with supporting amount number of the users in same time, runtime responsibility in the scalable SaaS provider to manage ease of the user requirements.
- Resource management in the SaaS layer is one of the important issues that generally is estimated according to limited storage and computational power of cloud providers while the workload they are supposed to assume is enormous. Availability condition for cloud applications is essential in the resource management. Scheduling methods as part of resource management can effect on service delivery and service reputation.
- Mobility is a key concept for the SaaS applications than other traditional software applications to provide available resources to users in any location with each device for business efforts. Software developers can manage SaaS applications at everywhere and anytime to connect in the Internet with high mobility condition. Also, mobility can influence on scalability for changing location and coverage extension between IoT devices and users.
- Security and privacy are one of the important challenges in the realization of the cloud computing (Karam *et al.*, 2012; Asim *et al.*, 2018). SaaS applications will be dealing with a lot of personal data and privacy of such data is of prime importance. Privacy preserving algorithms can be run on SaaS approaches (Ghafir *et al.*, 2018). From the security point of view man-in-the-middle (MITM), authentication, distributed denial of service, reliability and access control have been identified as the main security challenges in the SaaS applications (Tariq *et al.*, 2019).

6. Conclusion and limitations

This paper presented a survey for the SaaS mechanisms in the cloud computing. During this research, a complete understanding addicted to the SaaS mechanisms and considerations on open issues to synthesize the collected data. In this literature, we applied research studies that were published in 2012-2018. Finally, we examined 46 studies that focused on the SaaS mechanisms in the cloud computing. We observed that the application-based method has the highest percentage of the usage with 46 per cent in the literature. Regarding TQ2, we find that the non-heuristic algorithms have the highest percentage of the usage that is 55 per cent in the literature. According to TQ3, we observed that the business process method has the highest percentage of the usage by 49 per cent in the literature. And for TQ4, we find that the time factor has the highest percentage of the usage of 35 per cent in the literature. For TQ5, the CloudSim has the highest percentage of the usage of 29 per cent in the literature. Of course, MATLAB has 20 per cent, Visual C# has 20 per cent and ASP.Net has 8 per cent. Some limitations of this study are presented as follows:

- omitting non-English studies;
- removing non-ISI indexed studies;
- removing all of the thesis dissertations, book chapters, white papers; and
- omitting not peer-reviewed published studies and low-quality conference papers.

We believe that this review addresses the conceptual features of the SaaS mechanisms for the cloud computing. In future work, researchers can analyze and review other cloud layers such as IaaS, Data as a Service and Process as a Service for applying a systematic literature review for recent studies.

Notes

1. Decision-making trial and evaluation laboratory.
2. VlseKriterijumska Optimizacija I Kompromisno Resenje.

References

Afify, Y.M., *et al.* (2017), "A personalized recommender system for SaaS services", *Concurrency and Computation: Practice and Experience*, Vol. 29 No. 4, p. e3877,

Ali, H., Moawad, R. and Hosni, A.A.F. (2016), "A cloud interoperability broker (CIB) for data migration in SaaS", *Proceedings of 2016 IEEE International Conference on Cloud Computing and Big Data Analysis, ICCCBDA 2016*, pp. 250-256.

Aliyu, A., *et al.* (2017), "Cloud computing in VANETs: architecture, taxonomy, and challenges", *IETE Technical Review*, pp. 1-25.

Almorsy, M., Grundy, J. and Ibrahim, A.S. (2013), *Adaptable, Model-Driven Security Engineering for SaaS Cloud-Based Applications*.

Amiri, A. (2017), "Application placement in computer clustering in software as a service (SaaS) networks", *Information Technology and Management*, Vol. 18 No. 2, pp. 161-173.

Anselmi, J., Ardagna, D. and Passacantando, M. (2014), "Generalized Nash equilibria for SaaS/PaaS clouds", *European Journal of Operational Research*, Vol. 236 No. 1, pp. 326-339.

Arabnia, H.R. and Tinetti, F.G. (2018), "Grid, cloud, and cluster computing and applications".

Arunkumar, G. and Venkataraman, N. (2015), "A novel approach to address interoperability concern in cloud computing", *Procedia Computer Science*, Vol. 50, pp. 554-559.

Asim, M., *et al.* (2018), "Security policy monitoring of BPMN-based service compositions", *Journal of Software: Evolution and Process*, Vol. 30 No. 9, p. e1944.

Aznoli, F. and Navimipour, N.J. (2017), "Cloud services recommendation: reviewing the recent advances and suggesting the future research directions", *Journal of Network and Computer Applications*, Vol. 77, pp. 73-86.

Baker, T., *et al.* (2015), "Security-oriented cloud platform for soa-based SCADA" *in 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*.

Balmukund, R.S., Sandip, B. and Chandra, M.S. (2017), "Identifying the moderating effect of trust on the adoption of cloud-based services", *International Journal of Communication Systems*, Vol. 30 No. 11, p. e3253.

Banerjee, S. and Jain, S. (2014), "A survey on software as a service (SaaS) using quality model in cloud computing", *International Journal of Engineering and Computer Science*, Vol. 3 No. 1, pp. 3598-3602.

Banerjee, C., Kundu, A. and Dattagupta, R. (2013), "SaaS oriented generic cloud compiler", *Procedia Technology*, Vol. 10, pp. 253-261.

Bellavista, P., *et al.* (2013), *Automated Provisioning of SaaS Applications over IaaS-Based Cloud Systems. in Advances in Service-Oriented and Cloud Computing*, Springer Berlin Heidelberg, Berlin, Heidelberg.

Candeia, D., Santos, R.A. and Lopes, R. (2015), "Business-driven long-term capacity planning for SaaS applications", *IEEE Transactions on Cloud Computing*, Vol. 3 No. 3, pp. 290-303.

Charband, Y. and Jafari Navimipour, N. (2016), "Online knowledge sharing mechanisms: a systematic review of the state of the art literature and recommendations for future research", *Information Systems Frontiers*, pp. 1-21.

Chen, Y.S., *et al.* (2017), "Analysis of performance measures in cloud-based ubiquitous SaaS CRM project systems", *Journal of Supercomputing*, pp. 1-25.

Cheng, Y., *et al.* (2015), "Development of a construction quality supervision collaboration system based on a SaaS private cloud", *Journal of Intelligent and Robotic Systems*, Vol. 79 Nos 3/4, pp. 613-627.

Chiregi, M. and Jafari Navimipour, N. (2017), "Cloud computing and trust evaluation: a systematic literature review of the state-of-the-art mechanisms", *Journal of Electrical Systems and Information Technology*.

Cho, V. and Chan, A. (2015), "An integrative framework of comparing SaaS adoption for core and non-core business operations: an empirical study on Hong Kong industries", *Information Systems Frontiers*, Vol. 17 No. 3, pp. 629-644.

Church, P.C. and Goscinski, A.M. (2014), "A survey of Cloud-Based service computing solutions for mammalian genomics", *IEEE Transactions on Services Computing*, Vol. 7 No. 4, pp. 726-740.

Church, P., Goscinski, A. and Lefèvre, C. (2015), "Exposing HPC and sequential applications as services through the development and deployment of a SaaS cloud", *Future Generation Computer Systems*, Vols 43/44, pp. 24-37.

Cotroneo, D., Paudice, A. and Pecchia, A. (2016), "Automated root cause identification of security alerts: evaluation in a SaaS cloud", *Future Generation Computer Systems*, Vol. 56, pp. 375-387.

Du, J., *et al.* (2014), "Scalable distributed service integrity attestation for software-as-a- service clouds", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25 No. 3, pp. 730-739.

El Kafhali, S. and Salah, K. (2018a), "Modeling and analysis of performance and energy consumption in cloud data centers", *Arabian Journal for Science and Engineering*, Vol. 43 No. 12.

El Kafhali, S. and Salah, K. (2018b), "Performance analysis of multi-core VMs hosting cloud SaaS applications", *Computer Standards and Interfaces*, Vol. 55, pp. 126-135.

Fan, H., *et al.* (2015a), "An integrated personalization framework for SaaS-based cloud services", *Future Generation Computer Systems*, Vol. 53, pp. 157-173.

Fan, H., *et al.* (2015b), "Semantic client-side approach for web personalization of SaaS-based cloud services", *Concurrency and Computation: Practice and Experience*, Vol. 27 No. 8, pp. 2144-2169.

Fortino, G., *et al.*, (2014), "BodyCloud: a SaaS approach for community body sensor networks", *Future Generation Computer Systems*, Vol. 35, pp. 62-79.

Fowley, F. and Pahl, C. (2018), "Cloud migration architecture and pricing – mapping a licensing business model for software vendors to a SaaS business model", in *Advances in Service-Oriented and Cloud Computing*, Springer International Publishing, Cham.

Ge, C. and Huang, K.-W. (2014), "Analyzing the economies of scale of software as a service software firms: a stochastic frontier approach", *IEEE Transactions on Engineering Management*, Vol. 61 No. 4, pp. 610-622.

Ghafir, I., *et al.* (2018), "Security threats to critical infrastructure: the human factor", *The Journal of Supercomputing*, Vol. 74 No. 10, pp. 4986-5002.

Goode, S., *et al.* (2015), "Rethinking the role of security in client satisfaction with software-as-a-Service (SaaS) providers", *Decision Support Systems*, Vol. 70, pp. 73-85.

Gutierrez-Milla, A., *et al.* (2015), "Crowd evacuations SaaS: an ABM approach", *Procedia Computer Science*, Vol. 51, pp. 473-482.

Huang, J.Y. (2017), "An analysis of the intellectual structure of the cloud patents of SaaS", *Technology Analysis and Strategic Management*, Vol. 29 No. 8, pp. 917-931.

Huang, K.C. and Shen, B.J. (2015), "Service deployment strategies for efficient execution of composite SaaS applications on cloud platform", *Journal of Systems and Software*, Vol. 107, pp. 127-141.

Imani, M., *et al.*, (2017), "A survey on asynchronous quorum-based power saving protocols in multi-hop networks", *The Journal of Information Processing Systems*, Vol. 13 No. 6, pp. 1436-1458.

Iranpour, E. and Sharifian, S. (2016), "An FPGA implemented brain emotional learning intelligent admission controller for SaaS cloud servers", *Transactions of the Institute of Measurement and Control*.

Jamshidi, P., Ahmad, A. and Pahl, C. (2013), "Cloud migration research: a systematic review", *IEEE Transactions on Cloud Computing*, Vol. 1 No. 2, pp. 142-157.

Jatoth, C., Gangadharan, G.R. and Buyya, R. (2017), "Computational intelligence based QoS-Aware web service composition: a systematic literature review", *IEEE Transactions on Services Computing*, Vol. 10 No. 3, pp. 475-492.

Karam, Y., Baker, T. and Taleb-Bendiab, A. (2012), "Security support for intention driven elastic cloud computing", in *2012 Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation*.

Keshanchi, B., Souri, A. and Navimipour, N.J. (2017), "An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: formal verification, simulation, and statistical testing", *Journal of Systems and Software*, Vol. 124, pp. 1-21.

Kim, W., *et al.*, (2012), "An innovative method for data and software integration in SaaS", *Computers and Mathematics with Applications*, Vol. 64 No. 5, pp. 1252-1258.

Kitchenham, B., *et al.*, (2010), "Systematic literature reviews in software engineering – a tertiary study", *Information and Software Technology*, Vol. 52 No. 8, pp. 792-805.

Kwok, T. and Mohindra, A. (2008), "Resource calculations with constraints, and placement of tenants and instances for multi-tenant SaaS applications", in *Service-Oriented Computing – ICSOC 2008: 6th International Conference, Sydney, Australia, December 1-5, 2008. Proceedings*, in Bouguettaya A., Krueger, I. and Margaria, T., (Eds), *Springer Berlin Heidelberg*, *Berlin, Heidelberg*. pp. 633-648.

Levitin, G., Xing, L. and Dai, Y. (2017), "Optimal data partitioning in cloud computing system with random server assignment", *Future Generation Computer Systems*, Vol. 70, pp. 17-25.

Li, X., *et al.*, (2016), "A partition model and strategy based on the stoer? Wagner algorithm for SaaS multi-tenant data", *Soft Computing*, pp. 1-12.

Liao, W.-H., Chen, P.-W. and Kuai, S.-C. (2017), "A resource provision strategy for software-as-a-service in cloud computing", *Procedia Computer Science*, Vol. 110, pp. 94-101.

Li, C., Liu, Y.C. and Yan, X. (2017), "Optimization-based resource allocation for software as a service application in cloud computing", *Journal of Scheduling*, Vol. 20 No. 1, pp. 103-113.

Lin, Y.D., *et al.* (2015), "Two-tier project and job scheduling for SaaS cloud service providers", *Journal of Network and Computer Applications*, Vol. 52, pp. 26-36.

Liu, F., *et al.* (2014), "On arbitrating the power-performance tradeoff in SaaS clouds", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25 No. 10, pp. 2648-2658.

Ma, D. and Kauffman, R.J. (2014), "Competition between software-as-a-service vendors", *IEEE Transactions on Engineering Management*, Vol. 61 No. 4, pp. 717-729.

Martins, R., Oliveira, T. and Thomas, M.A. (2016), "An empirical analysis to assess the determinants of SaaS diffusion in firms", *Computers in Human Behavior*, Vol. 62, pp. 19-33.

Meena, M., Bharadi, V.A. and Vartak, K. (2016), "Hybrid wavelet based CBIR system using software as a service (SaaS) model on public cloud", *Procedia Computer Science*, Vol. 79, pp. 278-286.

Mohammad, A.F., *et al.* (2013), "Software evolution as SaaS: evolution of intelligent design in cloud", *Procedia Computer Science*, Vol. 19, pp. 486-493.

Motavaselalhagh, F., Safi Esfahani, F. and Arabnia, H.R. (2015), "Knowledge-based adaptable scheduler for SaaS providers in cloud computing", *Human-Centric Computing and Information Sciences*, Vol. 5 No. 1, pp. 16-16.

Mulia, W.D., *et al.* (2013), "Cloud workload characterization", *IETE Technical Review*, Vol. 30 No. 5, pp. 382-397.

Navimipour, N.J. and Vakili, A. (2017), "Comprehensive and systematic review of the service composition mechanisms in the cloud environments", *Journal of Network and Computer Applications*, Vol. 81, pp. 24-36.

Navimipour, N.J., Keshanchi, B. and Milani, F.S. (2017), "Resources discovery in the cloud environments using collaborative filtering and ontology relations", *Electronic Commerce Research and Applications*, Vol. 26, pp. 89-100.

Oliveira, T., Thomas, M. and Espadanal, M. (2014), "Assessing the determinants of cloud computing adoption: an analysis of the manufacturing and services sectors", *Information and Management*, Vol. 51 No. 5, pp. 497-510.

Rezaei, R., *et al.* (2014), "A semantic interoperability framework for software as a service systems in cloud computing environments", *Expert Systems with Applications*, Vol. 41 No. 13, pp. 5751-5770.

Riahi, M. and Krichen, S. (2018), "A multi-objective decision support framework for virtual machine placement in cloud data centers: a real case study", *The Journal of Supercomputing*, Vol. 74 No. 7, pp. 2984-3015.

Rodríguez, G., Soria, Á. and Campo, M. (2016), "AI-based web service composition: a review", *IETE Technical Review*, Vol. 33 No. 4, pp. 378-385.

Seethamraju, R. (2015), "Adoption of software as a service (SaaS) enterprise resource planning (ERP) systems in small and medium sized enterprises (SMEs)", *Information Systems Frontiers*, Vol. 17 No. 3, pp. 475-492.

Sharif, S., *et al.*, (2017), "Privacy-aware scheduling SaaS in high performance computing environments", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 28 No. 4, pp. 1176-1188.

Singh, S. and Chana, I. (2016), "A survey on resource scheduling in cloud computing: issues and challenges", *Journal of Grid Computing*, Vol. 14 No. 2, pp. 217-264.

Souri, A., Asghari, P. and Rezaei, R. (2017), "Software as a service based CRM providers in the cloud computing: challenges and technical issues", *Journal of Service Science Research*, Vol. 9 No. 2, pp. 219-237.

Souri, A., Navimipour, N.J. and Rahmani, A.M. (2018b), "Formal verification approaches and standards in the cloud computing: a comprehensive and systematic review", *Computer Standards and Interfaces*, Vol. 58, pp. 1-22.

Souri, A., Pashazadeh, S. and Navin, A.H. (2014), "Consistency of data replication protocols in database systems: a review", *International Journal on Information Theory*, Vol. 3 No. 4, pp. 19-32.

Souri, A. and Rahmani, A.M. (2014), "A survey for replica placement techniques in data grid environment", *International Journal of Modern Education and Computer Science*, Vol. 6 No. 5, p. 46.

Souri, A., Rahmani, A.M. and Jafari Navimipour, N. (2018b), "Formal verification approaches in the web service composition: a comprehensive analysis of the current challenges for future research", *International Journal of Communication Systems*, Vol. 31 No. 17, p. e3808.

Tang, C. and Liu, J. (2015), "Selecting a trusted cloud service provider for your SaaS program", *Computers and Security*, Vol. 50, pp. 60-73.

Tariq, N., *et al.* (2019), "The security of big data in fog-enabled IoT applications including blockchain: a survey", *Sensors*, Vol. 19 No. 8, p. 1788.

Toosi, A.N., Sinnott, R.O. and Buyya, R. (2018), "Resource provisioning for data-intensive applications with deadline constraints on hybrid clouds using aneka", *Future Generation Computer Systems*, Vol. 79, pp. 765-775.

Tsai, W., Bai, X. and Huang, Y. (2014), "Software-as-a-service (SaaS): perspectives and challenges", *Science China Information Sciences*, Vol. 57 No. 5, pp. 1-15.

Vázquez-Poletti, J.L., *et al.*, (2017), "SaaS enabled admission control for MCMC simulation in cloud computing infrastructures", *Computer Physics Communications*, Vol. 211, pp. 88-97.

Venkata Krishna, J. Apparao Naidu, G. and Upadhayaya, A. (2018), "Lion-whale optimization-based migration of virtual machines for data centers in cloud computing", *International Journal of Communication Systems*, Vol. 31 No. 8, p. e3539.

Vidhyalakshmi, R. and Kumar, V. (2017), "CORE framework for evaluating the reliability of SaaS products", *Future Generation Computer Systems*, Vol. 72, pp. 23-36.

Wu, L., *et al.* (2014), "SLA-based resource provisioning for hosted software as a service applications in cloud computing environments", *IEEE Transactions on Services Computing*, Vol. 7 No. 3, pp. 1-1.

Yang, Z., *et al.* (2015), "Understanding SaaS adoption from the perspective of organizational users: a tripod readiness model", *Computers in Human Behavior*, Vol. 45, pp. 254-264.

Zhang, X., *et al.* (2013), "On-demand business rule management framework for SaaS application", in *Cloud Computing and Services Science*, Springer International Publishing, Cham.

Zhang, J., Huang, H. and Wang, X. (2016), "Resource provision algorithms in cloud computing: a survey", *Journal of Network and Computer Applications*, Vol. 64, pp. 23-42.

Corresponding author
Monire Norouzi can be contacted at: monirenorouzi.research@gmail.com