ISSN: 0974-5823 Vol. 6 No. 3 December, 2021

International Journal of Mechanical Engineering

Security Testing Approach of Cloud Computing Environments

Majeda Sultana
M. Tech. Scholar
School of Engineering and I.T.
MATS University, Raipur

Dr. Abhishek Badholia
Associate Professor
School of Engineering and I.T.
MATS University, Raipur

1. INTRODUCTION:

Conventional way of testing process are becoming too expensive in areas of time, money, and other things. As a result, testing procedures have evolved to strategies that help enterprises in terms of business and revenues. Leading corporations like IBM, Microsoft, Google, and Amazon have a strong stake in the business term "CLOUD." As calculations, storage, and consumer contacts have already begun to transfer to the cloud, software testing is following suit. Testing new software necessitates the use of expensive server, storage, and network resources for a limited period. These computational assets are not utilised after testing, resulting in budget overruns. To deliver a dependable service, companies must test their offerings across all formats.

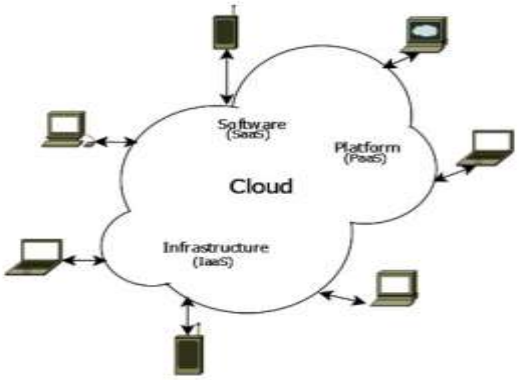


Figure 1.1: Types of Cloud Services

Cloud testing is a method of evaluating apps that uses the clouds as a computational environments and its infrastructures to simulate real-world throughput using current cloud computing services. Cloud testing is fundamentally related to the notions of cloud computing and software as a service (SaaS). Cloud testing allows you to test the cloud by employing cloud services such as gear, communication bandwidth, and workloads to more nearly imitate real-world situations and characteristics. Cloud testing has various challenges, including a restricted budget, fulfilling deadlines, a high cost per test, a huge quantity of sample instances, low test recycles, and geographical dispersion of clients.

Every physical thing becomes locatable, accessible, and reachable in the virtual realm of the Internet of Things (IoT). As more physical things are projected to connect to the Internet, the IoT is predicted to comprise millions or billions of devices that will interact with one another and with other entities. These items include not just computers and laptops, which are already present in traditional networks, but also physical things (such as household appliances), cars, and so on.

The variety of tools and techniques utilized to provide services has a significant influence on IoT device interoperability and administration. Furthermore, numerous equipment possess minimal computing performance and are installed in an outdoor setting, making it vulnerable to becoming manipulated or damaged by bad persons. Because of its intrinsic complexities and diverse architecture, the Internet of Things is vulnerable to a slew of risks and assaults that will disrupt its regular operation. As a result, ensuring the security of IoT devices is a difficult but critical responsibility.

Concept of Cloud Computing Deployment

Cloud services organizations could be private, public, or hybrid. The private cloud organisation is sent to the user by the organization servers. The installation methods provide cloud adaptability and services while protecting the organisation, security, and frequent influence on neighbouring computer servers. 3rd party cloud distributors offer clouds solutions through the internet. Public cloud providers are offered on demand, often on a continuous or hourly basis; nevertheless, long-haul responsibility are available for specific agencies. The CPU use and data transmission capabilities are used by the customer by paying for services depending on their requirements.

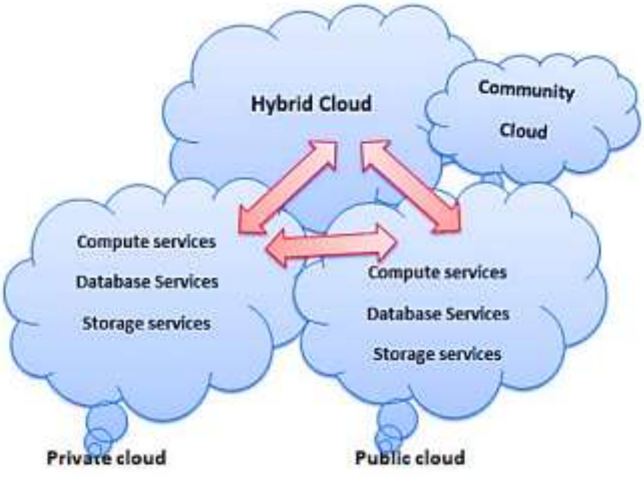


Figure 1.2 Concept of Cloud Computing Deployment

The public cloud includes AWS, IBM, Microsoft Azure, and Google Cloud etc.. A hybrid or crossbreed-cloud is a combination of private and public clouds, involving coordination and automation among the two (Helmi et al. 2018). The staffing levels and programmes upon a private clouds are operated by the enterprise and can benefits from the open clouds to respond any job outbursts.

2. Literature Review

This report addresses a need in cloud testing research: while there have been earlier surveys, none have provided a complete and up-to-date systematic assessment of the area. Table 1 summarizes comparable articles by publication year to ease comparison. The table demonstrates: the focus of the study (a few studies cover a specific aspect, others are broad); in the column 3rd, the year of the most recent referred study; in the 4th column, the research approach and finally, in the 5th column, either the collection of relevant initial analyses, or the amount of selected secondary analyses.

As a general observation, the last column shows that our survey encompasses a far broader group of papers than any previous effort. Several studies provide an informal review of available cloud testing methodologies and tools focused on an ad hoc sampling of the studies, without taking a comprehensive manner. Such publications were obviously valuable in the early years of the topic, providing a short overview to the discipline. A few of studies helped to develop a solid classification of important researching patterns, although others introduced cloud inspection procedures and technologies.

The surveys by Inçki et al. and Priyanka et al., among the early overview publications, conduct a systematic search of the literature and give a detailed classification of research investigations. However, both publications examine the literature through 2012, and significant research has been undertaken since then, necessitating the creation of a new, up-to-date SLR. There are more recent SLRs on cloud testing that cover particular areas within the larger field of cloud testing. Sakellari and Loukas, for example, provide a service to researchers by assessing current mathematical models, simulation methodologies, and testbeds that may be utilised for undertaking cloud testing research.

Zein et al. survey methodologies and resources designed expressly for evaluating smartphone apps, including, between various things, cloud-based techniques. Such studies largely intersect within this study, but neither of those give a comprehensive review of the cloud analysis research area. The research of Jia et. al., is an outlier: this publication recommends using the well-known 5W + 1H trend to assist the structure of research topics for comprehensive mappings investigations. The report subsequently uses a comprehensive mapping analysis of clouds experimental work to categorise more than 50 main papers as a case study to show the technique. Whereas the article was released in 2016, the list of contained publications was chosen in 2012, thus that research too is prior to the timeframe under consideration.

Ultimately, the research by Ahmad et.al., that concentrates on experimental investigations in cloud assessment articles is the nearest study toward this research. The article does a literature survey from 2010 to 2015 and presents a comprehensive mapping examination of 69 main research (from 75 referred papers). In contrast, this article covers a various time span (2012-2017) and includes around twice as many studies. Furthermore, we may see distinct primary study picks for such seasons covered in respective studies (i.e., 2012-2015). This could be due to Ahmad et aluse .'s of a distinctive lookup procedure, as well as their more progressive analysis of the phrase "testing" to include those certain review and evaluation strategies, although this research study just concentrates on assessment metrics.

Table 2.1: Cloud characteristics

Cloud characteristics		Testing problems
A huge quantity of concurrent processes is possible.		Extensive testing packages
Probability to utilize assets whenever required		Carry out inspections throughout the day.
Capability of creating many virtualized digital devices, all having a unique setup		Handle different configurations

3. CLOUD TESTING

A. Functional Testing

i) System Testing: System testing of software or hardware is performed on an integrated and full system in attempt to verify the compliance of the systems including its set criteria. It is a type of black box testing that does not need any understanding of the underlying program or logical architecture.

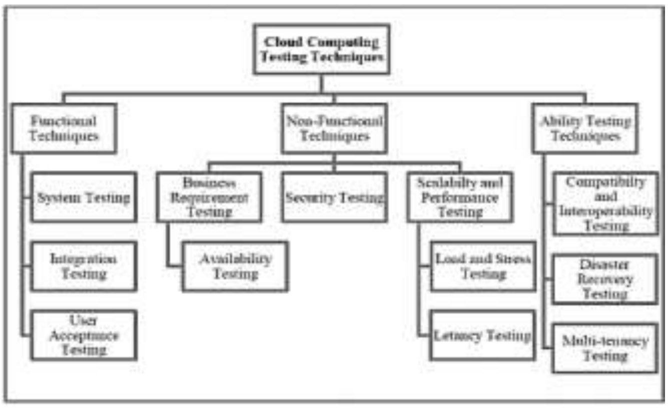

Cloud Computing Testing Techniques
Functional Techniques
Non-Functional Techniques
Ability Testing Techniques
System Testing
Integration Testing
User Acceptance Testing
Business Requirement Testing
Availability Testing
Security Testing
Scalability and Performance Testing
Load and Stress Testing
Latency Testing
Compatibility and Interoperability Testing
Disaster Recovery Testing
Multi-tenancy Testing

Figure 3.1: Techniques for Cloud Testing

All "embedded" application programs that have survived integration assessment, as well as the software program as its own incorporated with any suitable hardware system, are the inputs to system testing. In the coxxntext of a System Required Specifications and a Functionality Requirements Specifications, it is conducted on the overall systems. It examines not just structure but also the customer's behaviour and desires. Systems assessment is indeed obligated to evaluate out and exceeding the application and equipment required specification's declared bounds (s). Its primary goal is to assess operational, economic, and end-user needs.

ii) Integration Testing: Integration testing is a process that involves inspecting each software module as a whole. This testing approach is appropriate for the cloud computing system when it comes to overall company strategy. It assists the organization in determining whether the cloud solution will function with present infrastructures and settings, demonstrating that the cloud service installation might not produce any negative effects on existing platforms. Ultimately, the corporate objectives must be checked and confirmed to ensure that cloud solution's actual outcome fits the company's objectives.

iii) User Acceptance Testing: This should be performed to ensure that the supplied cloud solution fulfils corporate demands and that the client acknowledges the cloud solutions that has been designed. hypothetical and substantial client acceptability testing are also performed. On-site assessment, on the other hand, enables for instant observation and management of testing improvements.

B. Non-Functional Testing

i). Business Requirement Testing: Data, parameters, and tests scenarios are all obtained from the needs in requirements-based testing. This encompasses both functional and non-functional characteristics such as accessibility, efficiency, and dependability.

The testing procedure involves the following steps:

- It must be completed in a reasonable timeframe.
- It may bring meaning to the software development life cycle and hence be sustainable.
- Because comprehensive system testing is unachievable, the t esting procedure must be fast.
- Testing would offer an overview of the project's state; as a re sult, it must be controllable.

ii). Security Testing: Due to rise in vulnerabilities in the commercial world, security testing has become an essential component of application testing. This could ensure that business-critical information is securely kept and transmitted. [10] Network protection is critical in a cloud context. Many security appliances are in widespread use to safeguard data centres and businesses. Anti-virus, data breach protection, firewalls, intrusion avoidance mechanisms, and anti-spam are all encouraged by these gadgets. [11] Permission, Reliability, Secrecy, Verification, Integrity, and Non-repudiation are among the six essential guidelines it aims to prove.

iii). Performance and Scalability Testing: Load testing is one of the most basic types of efficiency analysis. A load test is frequently performed to determine how the system will behave under a given load. This load can be defined as the estimated amount of simultaneous app users doing a particular exchange volume within a specified time frame. All of the major crucial commercial transactions' reaction times are included in this test. Whenever a cloud system is subjected to its intended stress, load testing ensures that it can run at the requisite response times.

iv). Testing for compatibility and interoperability: Compatibility test verifies how well a machine under test performs in a certain setting. As instance, much study has been conducted to determine if equipment used in heart transplantation are suitable with the body of the recipient. Compatibility test in application development examines how apps perform across various browsers and operating systems. Interoperability testing verifies how a system functions whenever it interacts with something else.

v). Disaster Recovery Testing: Tragedies are an unavoidable fact of life for any company, and they're also unexpected. Users should be able to access cloud services at all times, according to the cloud service provider. After a failure, the time it takes to recover from a disaster must be minimal. [5] This is a method of determining if the restoration processes carried out after a catastrophic failure or interruption were successful.

vi). Multi-tenancy Testing: Multi-tenancy is a major feature of both public and private clouds, and it applies to all three cloud layers: IaaS, SaaS, and PaaS. A software design in which a single example of a programme operates on a server and serves numerous tenants is referred to as multitenancy testing.

4. Testing Tools and Methodology

Cloud-based devices may be tested using a variety of methods at several layers, which include the device interfaces, platforms interface, database unit, and application system.

4.1. SOASTA

It was inspired by the need to assessment in a real-world setting rather than a lab setting. Agile approaches, such as frequently builds and rapid modification rates, are common in today's web apps. In terms of scalability, configuration, user profiles, and network conditions, stress testing with traditional tools in the lab might differ dramatically from analysis in the development environment. When opposed to lab procedures, running testing on live websites can reach a better level of accuracy and confidence. SOASTA CloudTest is a Web app efficiency analysis tool in development. It is capable of simulating millions of virtual objects. A service that provides public cloud infrastructures. The worker nodes might be spread over public and private clouds to collaborate in massive load testing.

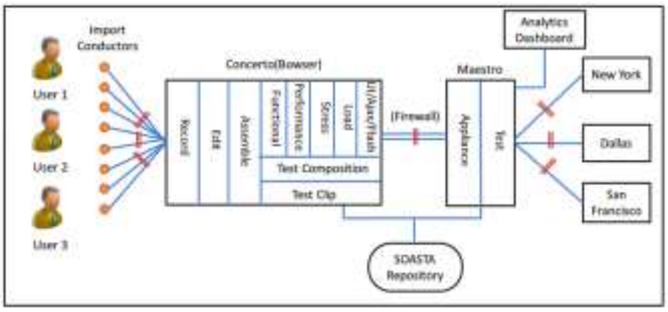


Figure 4.1: SOASTA Architecture

For analysis, test data from scattered test agents are combined. Analytical methods based on memory approaches are used to handle the massive amounts of data generated by large-scale testing in real time. On a synchronized time-line, provisioning data is shown through analytical dashboards. Testers may control and monitor the whole process using an Ajax-based online UI, which includes initiating 100s of load generating servers, building and operating globally dispersed test agents, and reviewing test results.

4.2. ITKO LISA

The LISA product suite from ITKO is aimed to help app development groups, particularly those working on bespoke apps, SOA, and cloud computing, become more productive. iTKO LISA seeks to deliver a cloud-based atmosphere and virtual services for the creation, verification, and validation of composite applications. It claims that its novel approach to supporting continuous addition for growth and testing has reduced software delivery timelines by 30 percent or more. Virtualisation technologies lies at the heart of the LISA design. LISA provides virtualized services for unavailable or inaccessible resources by replicating the target system's dynamic behavior so that it may reply as if it were a live system. It breaks the dependency limitations of system integration in this way, allowing for continuous testing.

4.3. Load Runner

Hewlett-HP Packard's Load Runner is an autonomous performing and test automation software for load testing, which involves studying system behavior and performance while producing real-world demand. In November 2006, HP purchased Load Runner as part of its acquisition of Mercury Interactive. HP LoadRunner is a software testing tool that works by simulating actual users by establishing remote individuals that utilize clients programs such as Microsoft Edge and send HTTP responses to IIS or Apache web servers. The data may then be dissected further to learn more about why certain behaviors occur.

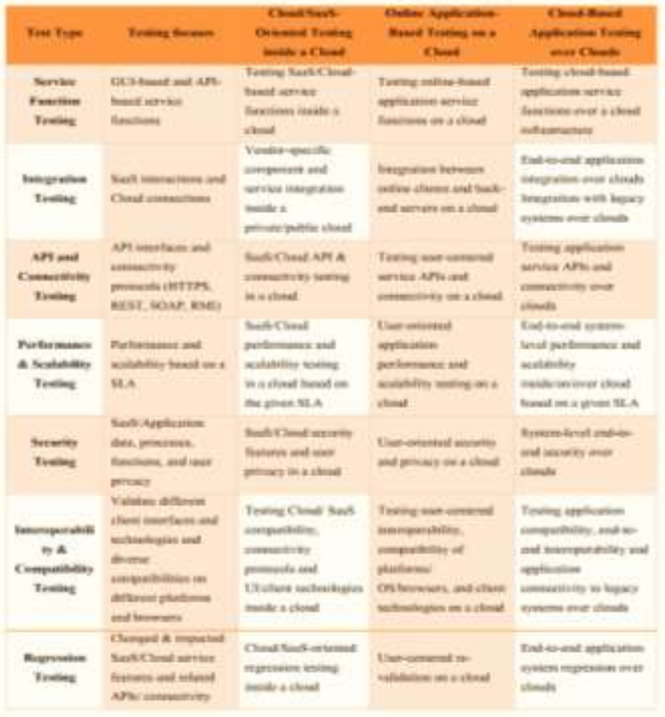
HP LoadRunner can be used as a stand-alone application for one or two people utilising each controllers, or as part of HP Performance Centre (which pools numerous controllers, all stress producers, and adds a web site, a scheduler, and other features to allow several people to share LoadRunner resources).

4.4. Blitz

As from clouds to clouds, Blitz is a load-testing tool. Clients of Blitz are often app and webpage creators who employ the service all across the the iterative development of mobile apps, webpages, and APIs. Throughout the development process, Blitz gives programmers a number of tools:

- Scalability testing for Web apps and APIs using load testing.
- PaaS vendors, simultaneous integration tools, and browsers integration x on a pay-per-test basis, it may test up to 50,000 virtual users at the same time.
- It's cloud-based, so there's no software to download. But, this means it can't test apps behind firewalls or in other places where they're shielded from the Web.

Table 4.1: COMPARISION OF DIFFERENT TESTINGS



Test Type	Testing focuses	Cloud/SaaS-Oriented Testing inside a Cloud	Online Application-Based Testing on a Cloud	Cloud-Based Application Testing over Clouds
Service Function Testing	GUI-based and API-based service functions	Testing SaaS/Cloud-based service functions inside a cloud	Testing online-based application service functions on a cloud	Testing cloud-based application service functions over a cloud infrastructure
Integration Testing	SaaS interactions and Cloud connections	Vendor-specific component and service integration inside a private/public cloud	Integration between online clients and back-end servers on a cloud	End-to-end application integration over clouds Integration with legacy systems over clouds
API and Connectivity Testing	API interfaces and connectivity protocols (HTTPS, REST, SOAP, RMI)	SaaS/Cloud API & connectivity testing in a cloud	Testing user-centered service APIs and connectivity on a cloud	Testing application service APIs and connectivity over clouds
Performance & Scalability Testing	Performance and scalability based on a SLA	SaaS/Cloud performance and scalability testing in a cloud based on the given SLA	User-oriented application performance and scalability testing on a cloud	End-to-end system-level performance and scalability inside/on/over cloud based on a given SLA
Security Testing	SaaS/Application data, processes, functions, and user privacy	SaaS/Cloud security features and user privacy in a cloud	User-oriented security and privacy on a cloud	System-level end-to-end security over clouds
Interoperability & Compatibility Testing	Validate different client interfaces and technologies and diverse compatibilities on different platforms and browsers	Testing Cloud/ SaaS compatibility, connectivity protocols and UI/client technologies inside a cloud	Testing user-centered interoperability, compatibility of platforms/ OS/browsers, and client technologies on a cloud	Testing application compatibility, end-to-end interoperability and application connectivity to legacy systems over clouds
Regression Testing	Changed & impacted SaaS/Cloud service features and related APIs/ connectivity	Cloud/SaaS-oriented regression testing inside a cloud	User-centered re-validation on a cloud	End-to-end application system regression over clouds

5. Conclusion

Cloud testing is currently the most popular empirical subject between new studies. More study is needed to solve the open difficulties and concerns in cloud testing as testing as a service and cloud technologies evolve. This study provides an overview of the various testing approaches available as well as the issues faced in the cloud ecosystem. To imitate user action, functional testing necessitates extensive use of application and equipment. Non - functional assessment, on the other hand, enables the linkage and assessment of the testing of non-functional properties of software systems. Some testing issues in the cloud atmosphere have been identified, and with the ongoing improvements in each of these approaches, we can't argue that one is superior than another, since each assessment method seems to have its specific collection of advantages and limits.

6. Reference

[1]. F. Hujainah, R. B. A. Bakar, M. A. Abdulgabber, and K. Z. Zamli, ‘‘Software requirements prioritisation: A systematic literature review on significance, stakeholders, techniques and challenges,’’ IEEE Access, vol. 6, pp. 71497–71523, 2018.

[2]. J. N. Goel and B. M. Mehtre, ‘‘Vulnerability assessment penetration testing as a Cyber defence technology,’’ Procedia Comput. Sci., vol. 57, pp. 710–715, Jan. 2015.

[3]. G. Chu and A. Lisitsa, ‘‘Penetration testing for Internet of Things and its automation,’’ in Proc. IEEE 20th Int. Conf. High Perform. Comput. Commun. IEEE 16th Int. Conf. Smart City IEEE 4th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS), Jun. 2018, pp. 1479–1484.

[4]. M. I. Sahu and U. Pandey, ‘‘Mobile cloud computing: Issues and challenges,’’ in Proc. Int. Conf. Adv.

Comput., Commun. Control Netw. (ICACCCN), Oct. 2018, pp. 247–250.

[5]. K. Akherfi, M. Gerndt, and H. Harroud, ‘‘Mobile cloud computing for computation offloading: Issues and challenges,’’ Appl. Comput. Informat., vol. 14, no. 1, pp. 1–16, 2018.

[6]. A. S. Al-Ahmad and H. Kahtan, ‘‘Fuzz test case generation for penetration testing in mobile cloud computing applications,’’ in Proc. Int. Conf. Intell. Comput. Optim., 2018, pp. 267–276.

[7]. W. Xu, B. Groves, and W. Kwok, ‘‘Penetration testing on cloud—Case study with own cloud,’’ Global J. Inf. Technol., vol. 5, no. 2, pp. 87–94, 2016.

[8]. M. Denis, C. Zena, and T. Hayajneh, ‘‘Penetration testing: Concepts, attack methods, and defense strategies,’’ in Proc. IEEE Long Island Syst., Appl. Technol. Conf. (LISAT), Apr. 2016, pp. 1–6.

[9]. J. N. Goel, M. H. Asghar, V. Kumar, and S. K. Pandey, ‘‘Ensemble based approach to increase vulnerability assessment and penetration testing accuracy,’’ in Proc. Int. Conf. Innov. Challenges Cyber Secur. (ICICCSINBUSH), Feb. 2016, pp. 330–335.

[10]. J. Zhao, W. Shang, M. Wan, and P. Zeng, ‘‘Penetration testing automation assessment method based on rule tree,’’ in Proc. IEEE Int. Conf. Cyber Technol. Automat., Control, Intell. Syst. (CYBER), Shenyang, China, Jun. 2015, pp. 1829–1833.

[11]. H. A. Turner, ‘‘Optimizing, testing, and securing mobile cloud computing systems for data aggregation and processing,’’ Ph.D. dissertation, Virginia Tech, Blacksburg, VA, USA, 2015.

[12]. J. N. Goel and B. M. Mehtre, ‘‘Vulnerability assessment penetration testing as a Cyber defence technology,’’ Procedia Comput. Sci., vol. 57, pp. 710–715, Jan. 2015.

[13]. A. S. Al-Ahmad, S. A. Aljunid, and A. S. A. Sani, ‘‘Mobile cloud computing testing review,’’ in Proc. Int. Conf. Adv. Comput. Sci. Appl. Technol. (ACSAT), Kuala Lumpur, Malaysia, Dec. 2013, pp. 176–180.

[14]. S. H. Chandane and M. M. Bartere, ‘‘New computing paradigm: Software testing in cloud, issues, challenges and need of cloud testing in today’s world,’’ Int. J. Emerg. Res. Manage. Technol., vol. 50, pp. 68–75, Feb. 2013.

[15]. H. Qi and A. Gani, ‘‘Research on mobile cloud computing: Review, trend and perspectives,’’ in Proc. 2nd Int. Conf. Digit. Inf. Commun. Technol. Appl. (DICTAP), Bangkok, Thailand, May 2012, pp. 195–202.

[16]. H. Muccini, A. Di Francesco, and P. Esposito, ‘‘Software testing of mobile applications: Challenges and future research directions,’’ in Proc. 7th Int. Workshop Autom. Softw. Test, Zurich, Switzerland, Jun. 2012, pp. 29–35.

[17]. B. Kirubakaran and V. Karthikeyani, ‘‘Mobile application testing— Challenges and solution approach through automation,’’ in Proc. Int. Conf. Pattern Recognit., Informat. Mobile Eng. (PRIME), Salem, India, Feb. 2013, pp. 79–84.

[18]. C. Costea, ‘‘Applications and trends in mobile cloud computing,’’ Carpathian J. Electron. Comput. Eng., vol. 5, p. 57, Jan. 2012.

[19]. N. Fernando, S. W. Loke, and W. Rahayu, ‘‘Mobile cloud computing: A survey,’’ Future Generat. Comput. Syst., vol. 29, no. 1, pp. 84–106, 2013.

[20]. B.-G. Chun, S. Ihm, P. Maniatis, and M. Naik, ‘‘Clonecloud: Boosting mobile device applications through cloud clone execution,’’ 2010, arXiv:1009.3088. [Online]. Available: https://arxiv.org/abs/1009.3088

[21]. S.-G. Kang, K.-W. Lee, and Y.-S. Kim, ‘‘Preliminary performance testing of Geo-spatial image parallel processing in the mobile cloud computing service,’’ Korean J. Remote Sens., vol. 28, no. 4, pp. 467–475, 2012.

[22]. B. Stepien, L. Peyton, and P. Xiong, ‘‘Using TTCN-3 as a modeling language for Web penetration testing,’’ in Proc. IEEE Int. Conf. Ind. Technol. (ICIT), Athens, Greece, Mar. 2012, pp. 674–681.

[23]. T. Paananen, ‘‘Smartphone Cross-Platform Frameworks: A case study,’’ M.S. thesis, Degree Programme Media Eng. School Technol., JAMK Univ. Appl. Sci., Jyväskylä, Finland, 2011.

[24]. B. Xing, L. Gao, J. Zhang, and D. Sun, ‘‘Design and implementation of an XML-based penetration testing system,’’ in Proc. Int. Symp. Intell. Inf. Process. Trusted Comput. (IPTC), Huanggang, China, Oct. 2010, pp. 224–229.

[25]. J. Gao, X. Bai, and W.-T. Tsai, ‘‘Cloud testing-issues, challenges, needs and practice,’’ Softw. Eng., Int. J., vol. 1, pp. 9–23, Sep. 2011.

[26]. C. Mainka, J. Somorovsky, and J. Schwenk, ‘‘Penetration testing tool for Web services security,’’ in Proc. 8th IEEE World Congr. Servicess, Honolulu, HI, USA, Sep. 2012, pp. 163–170.

[27]. K. Karnad and S. Nagenthram, ‘‘Cloud security: Can the cloud be secured,’’ in Proc. 7th Int. Conf. Internet Technol. Secured Trans. (ICITST), London, U.K., Dec. 2012, pp. 208–210.

[28]. B. Arkin, S. Stender, and G. McGraw, ‘‘Software penetration testing,’’ IEEE Secur. Privacy, vol. 3, no. 1, pp. 84–87, Jan. 2005.

[29]. D. Geer and J. Harthorne, ‘‘Penetration testing: A duet,’’ in Proc. 18th Annu. Comput. Secur. Appl. Conf., Las Vegas, NV, USA, Dec. 2002, pp. 185–195.

[30]. P. Xiong and L. Peyton, ‘‘A model-driven penetration test framework for Web applications,’’ in Proc. 8th Annu. Int. Conf. Privacy, Secur. Trust (PST), Ottawa, ON, Canada, Aug. 2010, pp. 173–180.

[31]. R. LaBarge and T. McGuire, ‘‘Cloud penetration testing,’’ Int. J. Cloud Comput., Services Archit., vol. 2, pp. 43–62, Jan. 2013.