Received July 22, 2020, accepted July 26, 2020, date of publication August 10, 2020, date of current version August 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3014671

Towards Pattern-Based Change Verification Framework for Cloud-Enabled Healthcare Component-Based

SADIA ALI[1], YASER HAFEEZ[1], N. Z. JHANJHI[2], MAMOONA HUMAYUN[3], MUHAMMAD IMRAN[4], ANAND NAYYAR[5,6], (Senior Member, IEEE), SAURABH SINGH[7], AND IN-HO RA[8], (Member, IEEE)

[1]University Institute of Information Technology, Pir Mehr Ali Shah Arid Agriculture University, Rawalpindi 46000, Pakistan
[2]School of Computer Science and Engineering (S.C.E.), Taylor's University, Subang Jaya 47500, Malaysia
[3]College of Computer and Information Science, Jouf University, Sakaka 72388, Saudi Arabia
[4]National Centre for Physics, Islamabad 44000, Pakistan
[5]Graduate School, Duy Tan University, Da Nang 550000, Vietnam
[6]Faculty of Information Technology, Duy Tan University, Da Nang 550000, Vietnam
[7]Department of Industrial and System Engineering, Dongguk University, Seoul 04620, South Korea
[8]School of Computer, Information and Communication Engineering, Kunsan National University, Gunsan 54150, South Korea

Corresponding authors: Saurabh Singh (saurabh89@dongguk.edu) and In-Ho Ra (ihra@kunsan.ac.kr)

This work was supported in part by the Institute for Information and Communications Technology Promotion (IITP) funded by the Korea Government (MSIT) under Grant 2018-0-00508, in part by the Development of Blockchain-Based Embedded Devices and Platform for M.G. Security and Operational Efficiency, and in part by KETEP, Korean Government, Ministry of Trade, Industry, and Energy (MOTIE), under Grant 20194010201800.

ABSTRACT To survive in the competitive environment, most organizations have adopted component-based software development strategies in the rapid technology advancement era and the proper utilization of cloud-based services. To facilitate the continuous configuration, reduce complexity, and faster system delivery for higher user satisfaction in dynamic scenarios. In cloud services, customers select services from web applications dynamically. Healthcare body sensors are commonly used for diagnosis and monitoring patients continuously for their emergency treatment. The healthcare devices are connected with mobile or laptop etc. on cloud environment with network and frequently change applications. Thus, organizations rely on regression testing during changes and implementation to validate the quality and reliability of the system after the alteration. However, for a large application with limited resources and frequently change component management activities in the cloud computing environment, component-based system verification is difficult and challenging due to irrelevant and redundant test cases and faults. In this study, proposed a test case selection and prioritization framework using a design pattern to increase the faults detection rate. First, we select test cases on frequently accessed components using observer patterns and, secondly, prioritize test cases on adopting some strategies. The proposed framework was validated by an experiment and compared with other techniques (previous faults based and random priority). Hence, experimental results show that the proposed framework successfully verified changes. Subsequently, the proposed framework increases the fault detection rate (i.e., more than 90%) than previous faults based and random priority (i.e., more than 80% respectively).

INDEX TERMS Body sensor, cloud computing, component-based system, design pattern, healthcare systems, regression testing, TCP.

I. INTRODUCTION

Advancement in the technology organization, emphasis on application quality and customers, demand increases their day to day requirements. Thus, due to agility in product delivery, organizations use the concept of reuse and divides into different parts. Therefore, organizations adopt component-based system development (CBSD) based on reusability [1]–[4]. Additionally, CBSD also satisfies extensibility, variability, validity, functionality, portability, and

The associate editor coordinating the review of this manuscript and approving it for publication was Chunsheng Zhu.

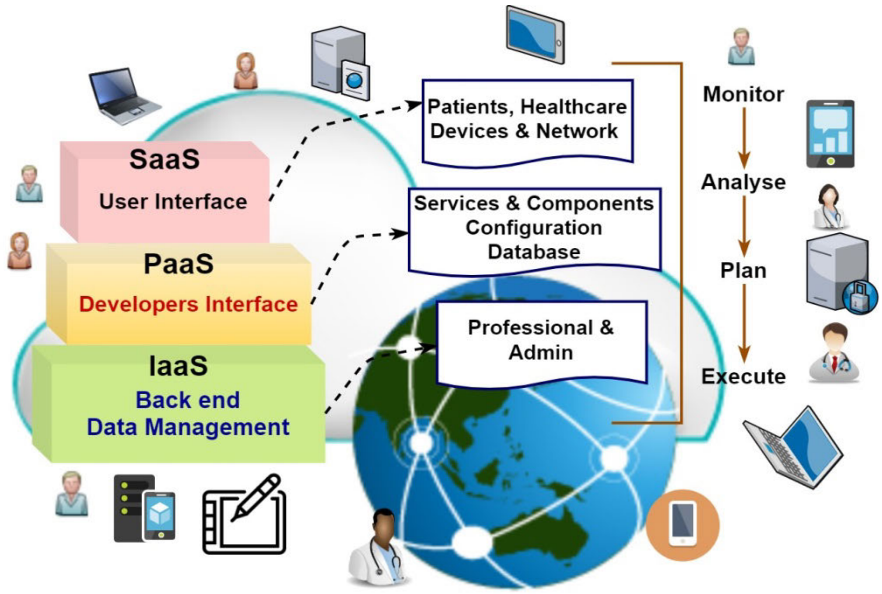


FIGURE 1. Cloud platform for healthcare devices.

consistency of component-based application within limited resources [1], [2], [5]. As in product, several variations and component combinations rise, which consequently results in millions of diverse configurations. In the CBSD context, where products are derived from existing elements of the organization, maintain core assets. Software organization also provides customization facilities to both developers and users; due to the fact error quickly introduced and difficulty to maintain reliability after customization [1], [5], [6]. Notably, in the cloud computing environment (C.E.) becomes more challenging, as the cloud environment increases the availability of software as a service all over the world uses dynamically [7]–[12].

To reduce the cost and complexity of building and maintaining their infrastructure for services providing to customers. There is no need to buy or update different versions of devices. Whereas, frequently accessible and make available 24/7 with continuous modifications increase the competition [13]–[16]. Therefore, smart mobile or laptop devices etc. have the ability to transmit and monitor information processing intelligently using wireless sensors devices like bio meters, healthcare etc. systems [6], [9], [17]–[20]. These devices have different specific functions to monitor and process information, i.e. temperature information, earthquake information, patient health information etc [9], [17], [19], [21].

This increases complexity, cost and effort for reliability after configuration in cloud-based component software development applications [13], [22], [23]. The healthcare body sensor continuously conveys the patient's condition to doctors and medical staff. The health care devices are used for emergency care of the patient to avoid the unnecessary process of registration, laboratory tests, etc. using C.E. different services, i.e., infrastructure as a services (IaaS), platform as a Services (PaaS) and services as a service (SaaS) [7], [16], [17], [21], [24]–[26]. These services provide a different level of services, from Development to usage as IaaS provides users services of resource management and system monitoring interfaces, e.g., Amazon EC2, OpenStack (Open Source). Whereas PaaS provides users capability for application deployment on IaaS using different programming tools and supports, e.g., Microsoft Azure, Google App Engine. Subsequently, SaaS provided running applications which accessible from various client devices, e.g., web browser, web-based Email, Google Apps, etc. instead of handling or control the underlying cloud infrastructure [7], [16], [21], [22], [24], [25], [27]. This information is collected through cloud-based architecture, as shown in Figure 1. Where all the bio front end healthcare systems connected with Bluetooth and Wi-Fi router with a different laptop, mobiles, tablets, etc. devices to monitor multiple patient information for their medical conditions like blood pressure, temperature, heart rate, etc. [9], [18], [28] at a central cloud location. Health care devices are connected to smart, intelligent devices.

For verification of component-based system (C.B.S.), a regression testing (R.T.) approach is used [29]–[32]. Its objective is to decrease testing effort, for reuse, execute, and prioritize test cases [33]–[37]. Existing approaches improve by-product sampling. Prioritization methods to rearrange test cases to detect faults as earlier as possible with feedback translation faster and error rectification earlier [38]. Research on R.T. prioritization suggested numerous systematic strategies to verify changes using code coverage. However, once faults revealed from the test suite, then debugging

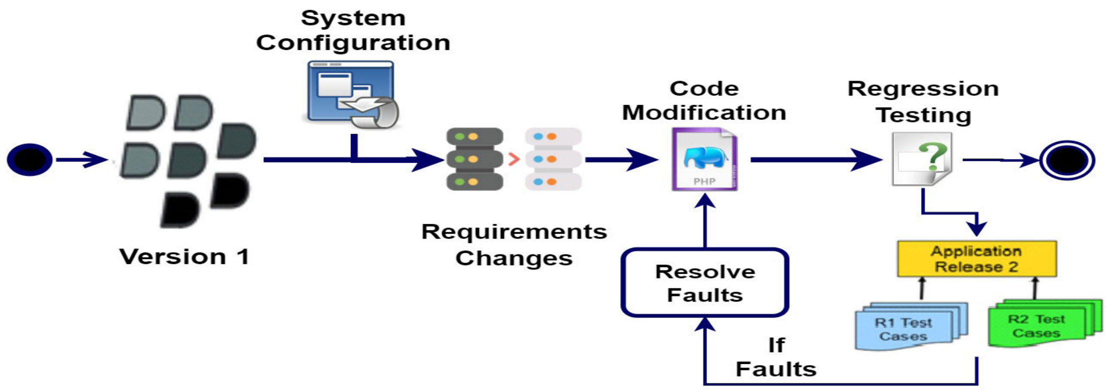


FIGURE 2. R.T. process.

is time-consuming due to difficulty in localizing errors in C.B.S. features [39]–[41].

R.T. is the most common method for verifying the quality of software application when there is a modification in the system during Development, as shown in Figure 2. This depicts the process of faults identification after modification for reliability before the release of a new device [14], [37], [42]. There are large sets of test suite which needed more time and cost to re-execute in the regression testing process. Therefore, to reduce the costs, time, and size of test suite various practices are used to verify changes such as test case prioritization (TCP), regression test selection (R.T.S.), test suite minimization (T.S.M.) and test suite augmentation (T.S.A.) [29]–[32], [38]–[41]. R.T. S procedures can be widely reused for testing product line applications, but capturing components variabilities is the main challenge [38], [39], [41], [43]–[47].

In a component-based engineering process, a design pattern (D.P.) for recurrent problems solution for improving the development process for the common situation in all phases [37], [42], [48]–[52]. A DP is optimal to general issue solution that the development team faced during CBSE development process improvement [49], [50]. As C.B.S. development D.P. based on reuse in the same situations for designing systems and have different types of D.P., i.e., adapter, template, observer, strategy, etc. suitable in different scenarios [48], [49], [51], [52]. Therefore, D.P. applicable is different in different situations, i.e., for component integration, analysis, testing etc. Consequently, its significant impact on the regression testing process is component verification. And D.P. helps to reduce testing efforts in components confirmation after changes.

In this study, we use observer and strategy patterns for configuration management and reliability analysis using regression testing approaches, i.e., selection and prioritization in cloud component-based systems. By using observer pattern logic, identify changes in component-based healthcare applications and strategy for fault detection using multiple criteria.

Therefore, the main contribution of research study categorized as;

- We proposed a pattern-based change verification (PBCV) framework for the healthcare system in a cloud computing environment. Firstly, the PBCV framework frequently extracts access component by users using observer pattern either by user or developers for repeatedly and recently access components verifications after changes. Secondly, for the prioritization process, extract relevant test cases using a strategy pattern. Thirdly, the identified rate of fault detection using evaluation metric for prioritization technique.
- Through the experimental evaluation is adopted for verification of proposed PBCV Framework. Thus, the synthetic project and real industrial projects are being used to validate the effectiveness of the PBCV framework and compared with two existing techniques, i.e., random prioritization and change based requirement priority with both patterns and without a pattern.
- Therefore, PBCV significant impact on maximum fault detection and outperform than existing technique.

The rest of the study is prepared as; Unit 2 explains related work with a detailed explanation of existing literature reviews. Subsequently, in Unit 3, we described the PBCV framework, which provides a solution to problems analyzed in the literature review. In Unit 4, we illustrate results and discussion of the empirical evaluation method. While in Unit 5 describes threats to validity and how to remove these threats. Finally, in Unit 6, study contributions concluded and highlighted our future work.

II. RELATED WORK

In unit 2, we elaborate current relevant literature in CBSD for modification using R.T. for healthcare application verification in cloud computing, and few studies proposed R.T. techniques for C.B.S. The authors in [1] presented CBSD testing approach after integration dependency validation in an embedded application. But require different metrics for R.T. process enhancement. In [13], the study author proposed a technique for continuous changes in cloud-based application services automatically. Hence, in [16] paper author described an optimization of a cloud-based test case approach for the dynamic random testing strategy to increase the effective fault detection process.

While in [31] presented an approach for TCP in a modified cloud-based system to enhance performance by increasing faults detection rate. Same as in [32] proposed TCP location-based approach using gravitation law for embedded intelligent devices. As the same study [30] researcher have explained the regression testing approach with reinforcement learning according to previous execution, time, and historical information of failed test cases. And used an industrial case study for evaluation for continuous integration. Consequently, in [29] study authors presented selective regression testing remover inter learned test case redundancy of highly configurable system for continuous integration. They used averaged based redundancy metric and historical information of integration tests for removal of redundancy. Thus, in [40] proposed a recommender system for the prioritization of test cases to increase fault detection frequency using user previous access history. But not get considerable results due to current and new users access frequency and frequently configure components. Same in [39] authors described that proposed configuration similarity to prioritize products in the software product line (S.P.L.) to improve reliability.

In [46] demonstrated using a genetic algorithm to generate test cases and fault localization for an S.P.L. and have a significant impact on the quality of product line and C.B.S. Thus, in [41] described that fault localization method not properly work in S.P.L.s context. The faults removal accuracy increases to improve effectiveness by isolating single faults with detection is easier instead of removing multiple faults at the end. In [50] paper author use modifies the form of adapter or wrapper pattern in CBSE for improving component functionalities and their subclasses inheritances. Same the case in [49] author use an architectural pattern for product line variabilities management process. It also helps in the validation of variation, which increases the benefits of design patterns. As in [51] author manages and verifies components in product line development using observer, strategy, template, and composition patterns for software creations, which results in significant improvement in the software industry.

Therefore, from existing studies, we concluded that there is a need for cloud-based healthcare component-based systems regression testing technique using design patterns to mitigate the identified challenges. These challenges explained in Table 1 with their description, i.e., Lack of core assets activities managements (CAMA), Cloud-Based system reliability (CBSR), Lack of change of historical information (H.I.), Observant of frequently access and change components (FACC), Irrelevant and redundant test cases (IRTC), Information detection using health care sensor devices (IDSD), Comprehensive methods for CBS RT and Multi-objective criteria (M.O.C.). Hence, in Table 2 we listed the parametric analysis of existing approaches for R.T.

TABLE 1. Overview of challenges in related work.

Challenges	Description
CAMA	The current techniques have no mechanism to save or updates C.A.M.
CBSR	No mechanism for CBSR in existing techniques.
HI	Less focus on recording and using H.I. for R.T. process.
FACC	To improve the R.T. process in C.B.S. development FACC technique required.
IRTC	IRTC increase cost, time, and efforts, which results in system failures.
IDSD	The mechanism id required for the configuration of IDSN devices.
CBS RT	Few techniques work on CBSRT.
M.O.C.	MOC has a significant impact on R.T.

So, there is a need to propose a framework to cloud-based C.B. healthcare applications approach for regression testing, which works in two steps. Firstly, used to rank components that frequently configured and secondly, prioritize test cases for setting verification approach is proposed to increase faults detection rates and minimize test case features, to facilitate the researcher and practitioners.

III. MATERIALS AND METHODS

In the unit, explain the detail of the PBCV proposed framework (P.F.) for the frequently change C.B.S. regression testing for removal of challenges in the group of related work using observant and strategy patterns for system verification and reliability. In Figure 3, we describe the architecture of regression testing of health care sensors in cloud computing.

The use of an interface layer is the front-end layer where all the users like patients, doctors, admin, etc. connected to C.E. using network gateway consists of a Wi-Fi router and Bluetooth. The gateway connects the health care devices to share information of patients among doctors and medical staff for reducing emergency death due to complicated procedures, lab tests, and first checks by doctors. The gateway to use the transport layer is to link with the cloud platform. Cloud platform links both front end and backend to avoid cost, time and effort with abundant resources/services to customers. In the back end, the data management layer that adopted all the services providing processes like Development, designing, analysis etc. Therefore, for modification verification of cloud-based healthcare body sensors devices in our proposed approach, different steps being adopted. Firstly, extract frequently accessed components by the customers. Secondly, select test cases for prioritization of frequently access components using some strategies, i.e., frequently changed test cases or high code coverage test cases. Thirdly, after the removal of identified faults released or update the modification of the service for customer's uses. These steps performed to detect maximum defects as soon as possible by a minimum number of test case execution.

TABLE 2. Parametric analysis Of component-based regression testing techniques.

Ref.	[2]	[9]	[15]	[16]	[13]	[23]	[21]	[28]	[31]	[32]	[35]	[38]	[39]	[40]	[41]
Component-Based RT	✗	✓	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✗	✓	✗
Component Testing	✓	✓	✓	✗	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗
TCP	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	✗	✗	✓
Cloud Services Reliability and Maintenance	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Health Care Sensors	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
R.T.S.	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✗
Frequently Change Component	✗	✓	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗
Coverage Metric	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	✗	✗	✓
Historical Metric	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓
Tool Supported	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
Multi-Criteria	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
Frequently Change Test Cases	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
APFD	✗	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✗	✓	✓
Empirical Evaluation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Fault Rate	✗	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗	✓	✓
Redundant Faults	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗
Tie In Case Of Similar Priority	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗

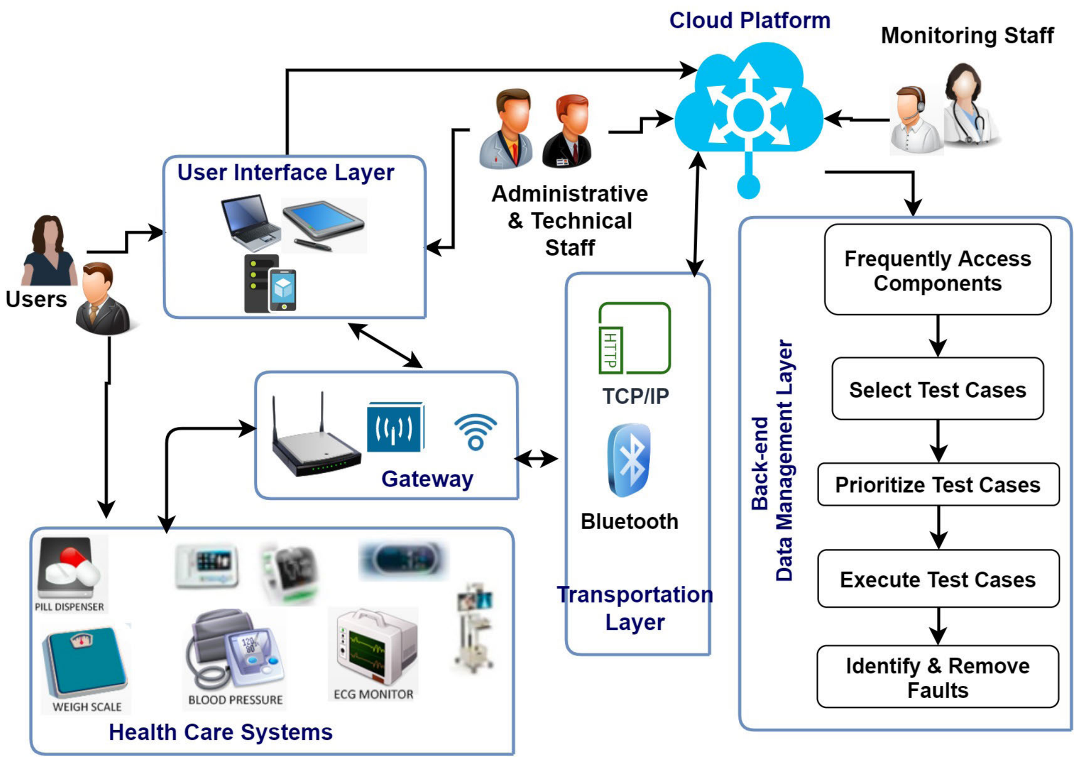


FIGURE 3. The architecture of cloud-based regression testing of health care sensors.

A. PBCV PROPOSED FRAMEWORK

The PBCV provides a comprehensive framework for test case selection and prioritization of C.B.S. change implication verification, as shown in Figure 4.

1) CHANGE INITIATIONS

Change in the C.B.S. project is started, and implementation frequently configures by either by users or developers due to their perspective and use for better quality. The Component Repository (C.R.) is used to manage all versions and C.B.S. relevant information. The information includes all reusable component information, changes historical data, faults relevant details, test case specification, and other relevant information. Frequently Access Components (F.A.C.), firstly, identify access components for that we use the observer pattern method adopted for the extraction of F.A.C. Secondly, we extract F.A.C. highly and sort them according to the highest frequency using the strategy pattern. In change test cases and suite phase, we obtain the test cases based on F.A.C. and identify these components relevant test suites to verify the change functionality of components.

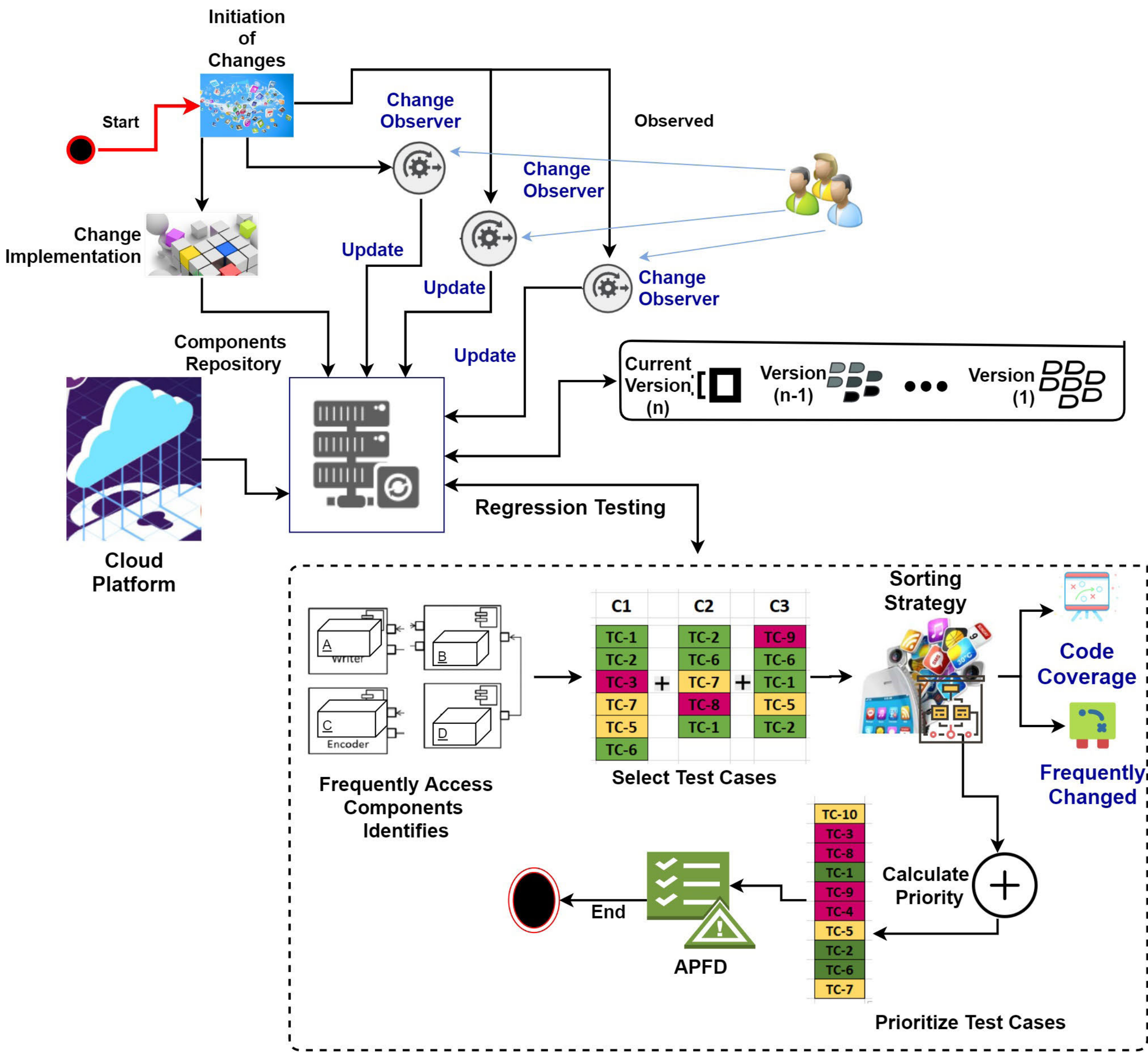


FIGURE 4. Proposed pattern-based change verification (PBCV) framework.

Whereas, in Extract and Select Test Case (ESTC) phase extract F.A.C. test cases for execution to verify components interfaces. The test selection base on the highest frequency of changes in components to avoid test cases irrelevancy and reduction in size.

2) SORTING STRATEGY (S.S.)

Then to reduce redundant faults and higher faults detection rate, use strategy pattern for test case prioritization. Therefore, we define three strategies for the prioritization process, i.e., Frequent changes in test cases, execution rate, and faults history.

The reason for selecting these strategies is to avoid redundant test cases and the use of multi-criteria for maximum faults detection during execution. Consequently, a problem arises when more than one test cases have the same frequency, which results in ambiguity and reliability issues. Therefore, to remove these issues, we defined multiple strategies that mostly ignore existing techniques. Thus, the first criteria we used were frequent changes for sorting, but in case of similarities faults history as second criteria and if still similar situations, then execution rate as the third strategy.

3) CALCULATE PRIORITY (C.P.)

Then CP test cases calculated and sort test cases phase sort test cases and highest frequency test cases executed first to reduce execution time and effort with maximum faults identification. Then Average Percentage of Faults Detected (APFD), which is mostly used evaluation metrics. APFD to detect the fault detection rate over complete faulty test suite in TCP approach. Higher the value of fault detection rate then earlier and maximum faults detected during regression testing [40], [42], [44], [46], [47], using equation (1);

$$APFD = 1 - [(TF_1 + TF_2 + \ldots\ldots + TF_{\mathrm{m}})/nm] + 1/2n \tag{1}$$

$\mathrm{TF_i}$ = Number of first Test Cases in execution order, m = faults' number in application tests, n = test cases' total number in the suite.

In the following section, we investigate the performance and faults rate identification of PBCV through empirical study.

IV. RESULTS AND DISCUSSION

The experimental study was performed to evaluate our PBCV framework. The experimentation was detained for executing PBCV to examine whether it's essentially able to deliver facilities which it promises. The experimental process depicted in Figure 5.

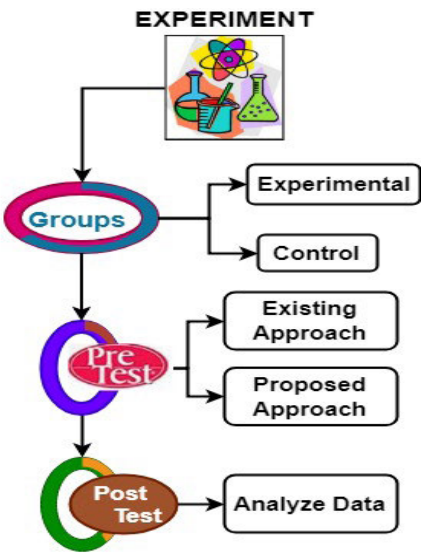


FIGURE 5. Experiment processed.

For this purpose, we developed two types of datasets, i.e. synthetic and real. In synthetic datasets, we selected online software development organization for the experiment. The project-based on the management system for online shopping, and it consists of different components according to different stakeholders' requirements. The information in the dataset about requirements, changes, test cases and faults used were not real, and for experiment purpose, we used some synthetic information.

Subsequently, real dataset local software house selected which had distributed cloud environment for Development. Thus, a health care device project selected, which consists of different components and all components, has a specific function like heart rate, blood pressure, etc. monitoring and transmitting online information to doctors and other monitoring staff. In this dataset, all information about requirements, changes, test cases, and faults used were real.

These selected experimental projects are; management system (M.S.) and healthcare application (H.A.). These projects consist of approximately more than 14000 lines of codes; in the cloud platform, different software versions available and have a large set of test suites with more than thousands of test cases in each suite. The MS and H.A. datasets have different sets of requirements, access features, and numbers of stakeholders. The change in dataset means that there may be updates, remove, and addition of information and features in both projects.

For the evaluation, we record the sessions of user access component interfaces and configuration made by both users and developers. For that, we created two groups of participants i.e. consists of companies' employees, users, and some volunteers to collect relevant data without any ambiguity in both M.S. and H.A. Development in different environments. Therefore, two experiment setups developed and performed using Steps of PBCV. Experiment 1 shown in the online organization on the M.S. dataset, while experiment 2 conducted in the local organization on the H.A. dataset. For these experiments, we gather all relevant data information from all daily transactions from designing and updating web applications and regular access by users and organizations.

Consequently, for investigation, we selected 16 participants for M.S. dataset and 40 members of participants for H.A. datasets. The participants of both teams divided into two Teams, i.e., Team-I (T-I) and Team-II (T-II). The included participants are project manager (PM), transaction manager (TM), system administrator (SA), the resource manager (RM), end-users (E.U.), volunteers (Vs), the app manages (AM) and quality engineer (Q.E.). According to the experiment process on T-I, we apply PBCV treatment means P.F. while in T-II, we used control treatment (C.T.) means no PF/PBCV (NPBCV).

Thus, participants of T-I observed and recorded information of different components which during the experimental period frequently access and change. These components in M.S. were replaced, i.e., login, discounts, location, mini car, payment, services options, rating, etc. and we number them as; $\mathrm{CMS_1}$, $\mathrm{CMS_2}$, ….., $\mathrm{CMS_n}$. Hence, the components in H.A. were changed, i.e., login, appointments, medical records, patients' and doctors' details, etc. and I.D. used for them as; $\mathrm{CHA_1}$, $\mathrm{CHA_2}$, ….., $\mathrm{CHA_n}$. Thus, in Table 3, we listed the selected highly change frequently and recently accessed components (FRAC) and their relevant test cases.

It help to avoid human efforts to execute all test suites and which results in ambiguity and challenges in the faults detection process. Then after the removal of duplication, we select test cases (S.T.C.) and listed in Table 4. Also mentioned their new frequency (N.F.) of some of the S.T.C. to describe our selection and prioritization process due to privacy we are not allowed to display complete information about our datasets; which calculated on the base of strategies described in

TABLE 3. FRAC detail.

FRAC and their relevant test cases of components											
MS						HA					
CMS_1	CMS_2	CMS_3	CMS_4	CMS_5	CMS6	CHA_1	CHA_2	CHA_3	CHA_4	CHA_5	CHA_6
TC_{MS1}	TC_{MS2}	TC_{MS9}	TC_{MS10}	TC_{MS10}	TC_{MS1}	TC_{HA9}	TC_{HA15}	TC_{HA5}	TC_{HA3}	TC_{HA5}	TC_{HA1}
TC_{MS3}	TC_{MS3}	TC_{MS6}	TC_{MS13}	TC_{MS13}	TC_{MS3}	TC_{HA16}	TC_{HA16}	TC_{HA6}	TC_{HA6}	TC_{HA11}	TC_{HA2}
TC_{MS5}	TC_{MS8}	TC_{MS7}	TC_{MS26}	TC_{MS15}		TC_{HA19}		TC_{HA12}	TC_{HA9}	TC_{HA18}	TC_{HA4}
TC_{MS7}	TC_{MS12}	TC_{MS21}						TC_{HA14}	TC_{HA10}	TC_{HA20}	TC_{HA6}
TC_{MS10}	TC_{MS17}								TC_{HA13}	TC_{HA21}	
TC_{MS15}									TC_{HA17}		

TABLE 4. Selected T.C. and priority detail.

STC_{MS}	NF_{MS}	TCP_{MS}	STC_{HA}	NF_{HA}	TCP_{HA}
TC_{MS1}	0.45	TC_{MS1}	TC_{HA9}	0.01	TC_{HA19}
TC_{MS10}	0.13	TC_{MS15}	TC_{HA16}	0.33	TC_{HA3}
TC_{MS17}	0	TC_{MS8}	TC_{HA19}	0.51	TC_{HA16}
TC_{MS21}	0.16	TC_{MS13}	TC_{HA15}	0.12	TC_{HA12}
TC_{MS2}	0	TC_{MS9}	TC_{HA5}	0.22	TC_{HA5}
TC_{MS8}	0.3	TC_{MS21}	TC_{HA6}	0	TC_{HA10}
TC_{MS9}	0.23	TC_{MS10}	TC_{HA12}	0.23	TC_{HA13}
TC_{MS3}	0.03	TC_{MS12}	TC_{HA14}	0.06	TC_{HA15}
TC_{MS12}	0.05	TC_{MS3}	TC_{HA3}	0.44	TC_{HA14}
TC_{MS13}	0.26		TC_{HA10}	0.19	
TC_{MS15}	0.4		TC_{HA13}	0.15	

PBCV section. The in TCP column sorted T.C. listed according to highest frequency to lowest priority.

Therefore, from a large set of T.C.s' selected from M.S. datasets are; {TC_{MS1}, TC_{MS3}, TC_{MS5}, TC_{MS7}, TC_{MS10}, TC_{MS15}, TC_{MS2}, TC_{MS8}, TC_{MS12}, TC_{MS17}, TC_{MS6}, TC_{MS9}, TC_{MS21}, TC_{MS10}, …..} and from H.A. datasets are; {TC_{HA9}, TC_{HA16}, TC_{HA19}, TC_{HA15}, TC_{HA5}, TC_{HA6}, TC_{HA12}, TC_{HA14}, TC_{HA3}, TC_{HA10}, TC_{HA13}, TC_{HA17}, TC_{HA11}, TCM_{S18}, TC_{HA20}, TC_{HA19}, TC_{HA15}, TC_{HA5}, TC_{HA6}, TC_{HA12}, TCM_{S14}, TC_{HA3}, …..} for the first execution. We conjure that by executing this T.C., we identify maximum faults. The S.T.C. order in ascending way on their respective frequencies for detection of errors which we performed and further use for verification in our experiment using APFD evaluation metric, which uses as a benchmark in the existing literature.

After the execution of T.C.; we get different fault detection rate (F.D.R.) which we listed in Table 5. For NPBCV we used previous faults based (P.F.B.) historical information and random priority (R.P.) for sorting T.C. According to Table 5, after executing TCMS1, TCMS16, TCMS10; we get 80, 43 and 40 percent of faults using PBCV, P.F.B. and R.P., respectively. Similarly, we also compared with the results of H.A. experiment and mentioned their results in Table 5. And S.T.C. of C.T. of both M.S. and H.A. were unable to detect maximum faults in first execution and our hypothesis that PBCV detects maximum faults in first execution then the NPBCV accepted.

The F.D.R. as depicted in Figure 5-6 is to describe and compare the results of both approaches and teams for further analysis. And from the analysis, we analyze that our proposed approach outperformed than the other method and significant impact in error-free C.B.S. The x-axis of Figure 6 shows the F.D.R. and y-axis depict the executed test cases. Therefore, in PBCV we identified almost all the faults as compared to P.F.B. and R.P., where we extract less percent of the errors from the total faults. Consequently, we have spent more effort, time and cost to identify all faults in P.F.B. and R.P.

After the verification of PBCV results, we calculate the APFD of both PBCV and NPBCV using equation 1, as mentioned in section III. The results of APFD are; 95 percent for PBCV and 70 Percent for NPBCV, as shown in Figure 7. The x-axis describes the percentage of APFD values, while

TABLE 5. PBCV and NPBCV results.

TC_{MS}	PBCV (%)	TC_{MS}	PFB (%)	TC_{MS}	RP (%)	TC_{HA}	PBCV (%)	TC_{HA}	PFB (%)	TC_{HA}	RP (%)
TC_{MS1}	95	TC_{MS16}	43	TC_{MS10}	40	TC_{HA19}	98	TC_{HA18}	35	TC_{HA1}	45
TC_{MS15}	80	TC_{MS19}	22	TC_{MS14}	10	TC_{HA3}	85	TC_{HA2}	24	TC_{HA15}	25
TC_{MS8}	70	TC_{MS20}	13	TC_{MS16}	43	TC_{HA16}	73	TC_{HA20}	28	TC_{HA8}	20
TC_{MS13}	40	TC_{MS17}	21	TC_{MS18}	22	TC_{HA12}	60	TC_{HA6}	30	TC_{HA13}	52
TC_{MS9}	60	TC_{MS18}	22	TC_{MS20}	13	TC_{HA5}	42	TC_{HA4}	44	TC_{HA9}	37
TC_{MS21}	50	TC_{MS2}	15	TC_{MS22}	20	TC_{HA10}	45	TC_{HA11}	33	TC_{HA21}	30
TC_{MS10}	40	TC_{MS14}	10	TC_{MS24}	18	TC_{HA13}	52	TC_{HA14}	31	TC_{HA7}	40
TC_{MS12}	30	TC_{MS4}	39	TC_{MS25}	23	TC_{HA15}	25	TC_{HA21}	19	TC_{HA17}	14
TC_{MS3}	20	TC_{MS22}	20	TC_{MS26}	25	TC_{HA14}	35	TC_{HA22}	21	TC_{HA20}	28

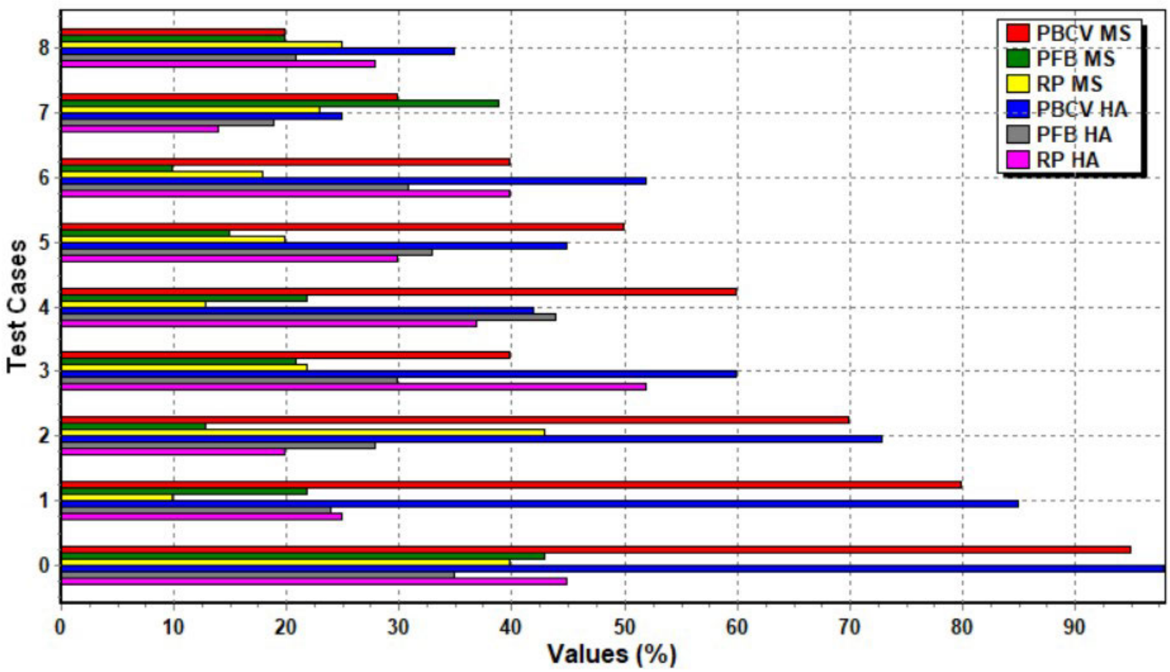


FIGURE 6. S.T.C. for execution.

the y-axis explains the approaches that we evaluated. Hence, evaluation metrics results also proved that PBCV has a significant impact on system reliability.

After experimenting with analysis results, we gathered the data using the questionnaire-based method—the questionnaire constructed to extract the viewpoints of participants included in both teams. The questions formulated on identified parameters from current literature about R.T. process improvement. The parameters identified after the comparative analysis of existing studies and these parameters are; easy implementation (E.I.), Testing effort reduce (T.E.R.), faults identification (F.I.), reduce human effort (R.H.E.), component verification (CV), test case size reduction (TCSR), reduce redundancy (R.R.), improve R.T. (I.R.T.), change reliability (C.R.), component change reliability (C.C.R.), multi criteria important (MCI), improve selection of T.C. (ISTC), improve TCP (ITCP), cloud computing testing (C.C.T.), healthcare system reliability (H.S.R.) and help in similar cases (H.S.C.).

The overall results of H.A. and M.S. datasets participant's viewpoints/analysis about both approaches are depicted in Figure 8. Therefore, the x-axis show rating scales percentage of both approaches' satisfaction, whereas the y-axis describes parameter details.

The results in the figure explain that the satisfaction level of a maximum of PBCV participants more than 50 percent. Whereas, for NPBCV participant figure depicted the level of satisfaction less than 50 percent. The five different rating

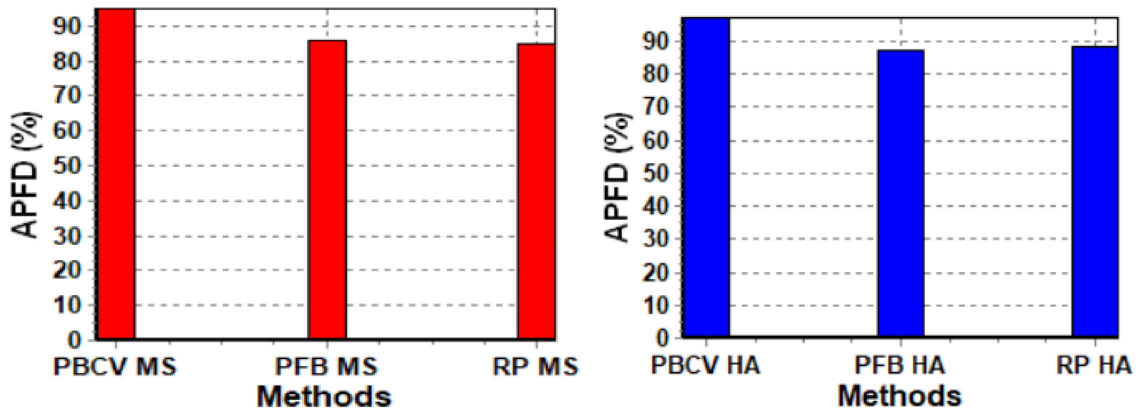


FIGURE 7. APFD Values.

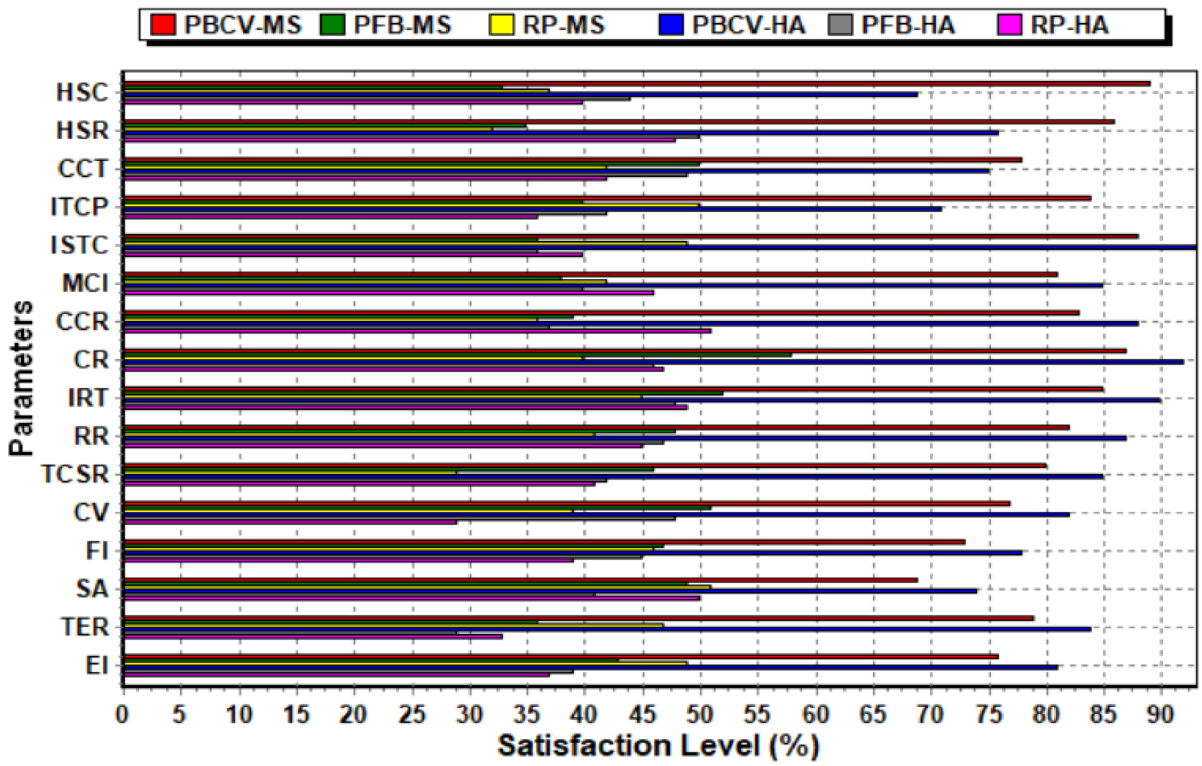


FIGURE 8. Viewpoints analysis results.

scales used for parametric analysis, i.e. Excellent (E), Above Average (A.A.), Average (A), Below Average (B.A.) and Very Poor (V.P.). In Figure 8, we describe the overall satisfaction level of all participants for both datasets using different methods for comparison of PBCV framework with other NPBCV methods.

Additionally, the overall results show that regression testing is an important part of C.B.S. development to verify the reliability of C.B.S. configurations. F.A.C. is the crucial factor for the detection of an error in R.T. and maintenance of component-based healthcare applications, specifically in dynamic cloud computing scenarios. Thus, our proposed PBCV framework significantly removes extracted limitations and improves the process of fault detection.

V. THREATS TO VALIDITY (T.V.)

In this unit, we discuss different types of T.V. [53], [54] for our experiment i.e. internal validity threats (IVT), external validity threats (EVT), conclusion validity threats (CVT) and construct validity threats (CsVT) and how to resolve these limitations.

IVT relevant to find impact treatment and outcome relationship for existing R.T. approaches F.D.R. ability with less cost and time. To decrease IVT we compared PBCV with P.F.B. and R.P. methods for comparison. EVT relevant to experimental results generalization, we experimented with the real-world cloud-based company which develops component-based health care devices and compare with another approach. And it required more projects need for further results generalization. Therefore, we trained participants of the experiment using PBCV to increase the proficiency of PBCV in the real scenario. Hence, in the future, we aimed to investigate different cloud-based companies' results comparison using PBCV all over the world.

CVT is relevant to derive the impact of conclusion from both M.S. and H.A. experiments finding and its comparison with NPBCV. So, we select the significant size of the sample and for reliable results APFD for PBCV framework analysis. In future, we investigate the statistical significance of results we will be adopted different statistical tests for reliability analysis. CsVT used to find the impact of selected factors on experiment outcomes, i.e. the F.D.R. ability of PBCV framework in the existing literature. As faults numbers not known therefore in experiment APFD used and compared with P.F.B. and R.P. methods. And also performed participant's survey and given enough time for answering all survey questions to detect the impact of dependent variables of both PBCV and NPBCV on independent variables. So, in comparison to other approach results proved that PBCV framework outperformed over P.F.B. and R.P. methods.

Moreover, cloud efficiency security, optimization including load balancing [55]–[58] and cloud location services have also have impact on cloud based health cares. In addition, cloud design, authentication mechanism such as using blockchain and wireless sensor network [14], [59]–[65], security and privacy in remote health care also consider a high impact on this.

VI. CONCLUSION

Due to continuous integration and advancement in features and services of health care, component-based devices using cloud platform increases reliability analysis. To improve the rate of fault detection in the cloud environment for reliability analysis become more complicated and time-consuming. Therefore, to help reliability analysis of modified components for high fault detection ability with no redundant faults and test cases using the proposed approach. So, the proposed approach was designed to resolve challenges in regression testing in C.B. healthcare cloud-enable systems while supporting continuous dynamic change decision and implementation activities in modern software development organizations. As was enlightening to see that the existing approach unsuccessful in increasing fault detection rate and component change verification's due to redundant faults or irrelevant test cases and frequent changes. The most existing approach relies on code coverage and ignores change verification with other criteria, especially in the component-based system. Thus, the proposed approach improves the C.B. healthcare systems quality using regression testing and provides significant implications in cloud bases services. An experimental method was conducted to evaluate the validity of the proposed approach, and the results demonstrated that the fault detection rate increased and more rapidly identified the maximum number of faults.

VII. FUTURE WORK

In future work, this work will be extended to resolve commonalities and in addition, the variability change decision implementation analysis in component and software product line regression testing with mapping change from requirements to trial. And proposed data management and frequently change elements using fog computing strategies to increase reliability verification.

REFERENCES

[1] B. Hendradjaya, "A proposal for new software testing technique for component based software system," *Int. J. Electr. Eng. Informat.*, vol. 10, no. 1, pp. 60–78, Mar. 2018, doi: 10.15676/ijeei.201.10.1.5.

[2] R. M. Parizi, "Microservices as an evolutionary architecture of component-based development: A think-aloud study," May 2018, *arXiv:1805.11757*. Accessed: Mar. 31, 2020. [Online]. Available: http://arxiv.org/abs/1805.11757

[3] C. Ayala, A. Nguyen-Duc, X. Franch, M. Höst, R. Conradi, D. Cruzes, and M. A. Babar, "System requirements-OSS components: Matching and mismatch resolution practices—An empirical study," *Empirical Softw. Eng.*, vol. 23, no. 6, pp. 3073–3128, Dec. 2018, doi: 10.1007/s10664-017-9594-1.

[4] P. Chatzipetrou, E. Papatheocharous, K. Wnuk, M. Borg, E. Alégroth, and T. Gorschek, "Component attributes and their importance in decisions and component selection," *Softw. Qual. J.*, vol. 28, no. 2, pp. 567–593, Jun. 2020.

[5] F. Meyerer and O. Hummel, "Towards plug-and-play for component-based software systems," in *Proc. 19th Int. Doctoral Symp. Compon. Archit. (WCOP)*, Marcq-en-Bareul, France, 2014, pp. 25–30, doi: 10.1145/2601328.2601334.

[6] S. Singh, I. H. Ra, W. Meng, M. Kaur, and G. H. Cho, "SH-BlockCC: A secure and efficient Internet of Things smart home architecture based on cloud computing and blockchain technology," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 4, 2019, Art. no. 1550147719844159.

[7] F. Peng, Q. Long, Z. X. Lin, and M. Long, "A reversible watermarking for authenticating 2D CAD engineering graphics based on iterative embedding and virtual coordinates," *Multimedia Tools Appl.*, vol. 78, Jan. 2019, pp. 26885–26905.

[8] M. Al-Qurishi, M. Al-Rakhami, F. Al-Qershi, M. M. Hassan, A. Alamri, H. U. Khan, and Y. Xiang, "A framework for cloud-based healthcare services to monitor noncommunicable diseases patient," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 3, Mar. 2015, Art. no. 985629, doi: 10.1155/2015/985629.

[9] M. M. Gaber, A. Aneiba, S. Basurra, O. Batty, A. M. Elmisery, Y. Kovalchuk, and M. H. U. Rehman, "Internet of Things and data mining: From applications to techniques and systems," *Wiley Interdiscipl. Rev. Data Mining Knowl. Discovery*, vol. 9, no. 3, p. e1292, May 2019, doi: 10.1002/widm.1292.

[10] Q. Tang, K. Yang, P. Li, J. Zhang, Y. Luo, and B. Xiong, "An energy efficient MCDS construction algorithm for wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, no. 1, pp. 83–97, Dec. 2012.

[11] H. Li, W. Li, S. Zhang, H. Wang, Y. Pan, and J. Wang, "Page-sharing-based virtual machine packing with multi-resource constraints to reduce network traffic in migration for clouds," *Future Gener. Comput. Syst.*, vol. 96, pp. 462–471, Jul. 2019.

[12] H. Li, W. Li, H. Wang, and J. Wang, "An optimization of virtual machine selection and placement by using memory content similarity for server consolidation in cloud," *Future Gener. Comput. Syst.*, vol. 84, pp. 98–107, Jul. 2018.

[13] Y. Luo, K. Yang, Q. Tang, J. Zhang, P. Li, and S. Qiu, "An optimal data service providing framework in cloud radio access network," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, pp. 1–11, Dec. 2016.

[14] W. Zeng, P. Chen, H. Chen, and S. He, "PAPG: Private aggregation scheme based on privacy-preserving gene in wireless sensor networks," *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 9, pp. 4442–4466, 2016.

[15] M. N. Abadeh and S.-H. Mirian-Hosseinabadi, "Delta-based regression testing: A formal framework towards model-driven regression testing," *J. Softw. Evol. Process*, vol. 27, no. 12, pp. 913–952, Dec. 2015, doi: 10.1002/smr.1752.

[16] H. Pei, B. Yin, and M. Xie, "Dynamic random testing strategy for test case optimization in cloud environment," in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops (ISSREW)*, Memphis, TN, USA, Oct. 2018, pp. 148–149, doi: 10.1109/ISSREW.2018.000-9.

[17] B. Yin, S. Zhou, S. Zhang, K. Gu, and F. Yu, "On efficient processing of continuous reverse skyline queries in wireless sensor networks," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 4, pp. 1931–1953, 2017.

[18] X. Yin, K. Zhang, B. Li, A. K. Sangaiah, and J. Wang, "A task allocation strategy for complex applications in heterogeneous cluster-based wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 8, 2018, Art. no. 1550147718795355.

[19] C.-C. Lee, C.-W. Hsu, Y.-M. Lai, and A. Vasilakos, "An enhanced mobile-healthcare emergency system based on extended chaotic maps," *J. Med. Syst.*, vol. 37, no. 5, p. 9973, Oct. 2013, doi: 10.1007/s10916-013-9973-0.

[20] H. Min-Shiang, L. Cheng-Chi, and T. Yuan-Liang, "Two simple batch verifying multiple digital signatures," in *Information and Communications Security*, vol. 2229, S. Qing, T. Okamoto, and J. Zhou, Eds. Berlin, Germany: Springer, 2001, pp. 233–237.

[21] R. Yu, T. W. C. Mak, R. Zhang, S. H. Wong, Y. Zheng, J. Y. W. Lau, and C. C. Y. Poon, "Smart healthcare: Cloud-enabled body sensor networks," in *Proc. IEEE 14th Int. Conf. Wearable Implant. Body Sensor Netw. (BSN)*, Eindhoven, The Netherlands, May 2017, pp. 99–102, doi: 10.1109/BSN.2017.7936017.

[22] C. M. S. Magurawalage, K. Yang, L. Hu, and J. Zhang, "Energy-efficient and network-aware offloading algorithm for mobile cloud computing," *Comput. Netw.*, vol. 74, pp. 22–33, Dec. 2014.

[23] M. Noorian, E. Bagheri, and W. Du, "Toward automated quality-centric product line configuration using intentional variability," *J. Softw. Evol. Process*, vol. 29, no. 9, p. e1870, Sep. 2017, doi: 10.1002/smr.1870.

[24] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, and H. J. , "An intelligent data gathering schema with data fusion supported for mobile sink in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 15 no. 3, 2019, Art. no. 1550147719839581.

[25] S. Singh, Y.-S. Jeong, and J. H. Park, "A survey on cloud computing security: Issues, threats, and solutions," *J. Netw. Comput. Appl.*, vol. 75, pp. 200–222, Nov. 2016.

[26] C.-C. Lee, P.-S. Chung, and M.-S. Hwang, "A survey on attribute-based encryption schemes of access control in cloud environments," *Int. J. Netw. Secur.*, vol. 15, no. 4, pp. 231–240, Jul. 2013, doi: 10.6633/IJNS.201307.15(4).01.

[27] H. Chen, X. Zhu, G. Liu, and W. Pedrycz, "Uncertainty-aware online scheduling for real-time workflows in cloud service environment," *IEEE Trans. Services Comput.*, early access, Aug. 21, 2019, doi: 10.1109/TSC.2018.2866421.

[28] Y. Santur, S. G. Santur, and M. Karaköse, "Architecture and implementation of a smart-pregnancy monitoring system using Web-based application," *Expert Syst.*, vol. 37, no. 1, p. e12379, Feb. 2020, doi: 10.1111/exsy.12379.

[29] D. Marijan and M. Liaaen, "Practical selective regression testing with effective redundancy in interleaved tests," in *Proc. 40th Int. Conf. Softw. Eng. Softw. Eng. Pract. (ICSE-SEIP)*, Gothenburg, Sweden, 2018, pp. 153–162, doi: 10.1145/3183519.3183532.

[30] H. Spieker, A. Gotlieb, D. Marijan, and M. Mossige, "Reinforcement learning for automatic test case prioritization and selection in continuous integration," in *Proc. 26th ACM SIGSOFT Int. Symp. Softw. Test. Anal. (ISSTA)*, Santa Barbara, CA, USA, 2017, pp. 12–22, doi: 10.1145/3092703.3092709.

[31] H. Srikanth and M. B. Cohen, "Regression testing in software as a service: An industrial case study," in *Proc. 27th IEEE Int. Conf. Softw. Maintenance (ICSM)*, Williamsburg, VA, USA, Sep. 2011, pp. 372–381, doi: 10.1109/ICSM.2011.6080804.

[32] X. Wang, H. Zeng, H. Gao, H. Miao, and W. Lin, "Location-based test case prioritization for software embedded in mobile devices using the law of gravitation," *Mobile Inf. Syst.*, vol. 2019, pp. 1–14, Jan. 2019, doi: 10.1155/2019/9083956.

[33] J. Wang, Y. Gao, W. Liu, W. Wu, and S.-J. Lim, "An asynchronous clustering and mobile data gathering schema based on timer mechanism in wireless sensor networks," *Comput., Mater. Continua*, vol. 58, no. 3, pp. 711–725, 2019.

[34] S. Souto and M. d'Amorim, "Time-space efficient regression testing for configurable systems," *J. Syst. Softw.*, vol. 137, pp. 733–746, Mar. 2018, doi: 10.1016/j.jss.2017.08.010.

[35] Y. Bian, Z. Li, J. Guo, and R. Zhao, "Concrete hyperheuristic framework for test case prioritization," *J. Softw. Evol. Process*, vol. 30, no. 11, p. e1992, Nov. 2018, doi: 10.1002/smr.1992.

[36] M. Khari, P. Kumar, D. Burgos, and R. G. Crespo, "Optimized test suites for automated testing using different optimization techniques," *Soft Comput.*, vol. 22, no. 24, pp. 8341–8352, Dec. 2018, doi: 10.1007/s00500-017-2780-7.

[37] S. K. Harikarthik, V. Palanisamy, and P. Ramanathan, "Optimal test suite selection in regression testing with testcase prioritization using modified Ann and Whale optimization algorithm," *Cluster Comput.*, vol. 22, no. S5, pp. 11425–11434, Sep. 2019, doi: 10.1007/s10586-017-1401-7.

[38] B. Miranda and A. Bertolino, "Scope-aided test prioritization, selection and minimization for software reuse," *J. Syst. Softw.*, vol. 131, pp. 528–549, Sep. 2017, doi: 10.1016/j.jss.2016.06.058.

[39] M. Al-Hajjaji, S. Lity, R. Lachmann, T. Thum, I. Schaefer, and G. Saake, "Delta-oriented product prioritization for similarity-based product-line testing," in *Proc. IEEE/ACM 2nd Int. Workshop Variability Complex. Softw. Design (VACE)*, Buenos Aires, Argentina, May 2017, pp. 34–40, doi: 10.1109/VACE.2017.8.

[40] M. Azizi and H. Do, "A collaborative filtering recommender system for test case prioritization in Web applications," in *Proc. 33rd Annu. ACM. Symp. Appl. Comput. (SAC)*, 2018, pp. 1560–1567, doi: 10.1145/3167132.3167299.

[41] P. Mahali and D. P. Mohapatra, "Model based test case prioritization using UML behavioural diagrams and association rule mining," *Int. J. Syst. Assurance Eng. Manage.*, vol. 9, no. 5, pp. 1063–1079, Oct. 2018, doi: 10.1007/s13198-018-0736-7.

[42] H. Jahan, Z. Feng, and S. M. H. Mahmud, "Risk-based test case prioritization by correlating system methods and their associated risks," *Arabian J. Sci. Eng.*, vol. 45, no. 8, pp. 6125–6138, Aug. 2020, doi: 10.1007/s13369-020-04472-z.

[43] M. A. Hasan, M. A. Rahman, and M. S. Siddik, "Test case prioritization based on dissimilarity clustering using historical data analysis," in *Information, Communication and Computing Technology*, vol. 750, S. Kaushik, D. Gupta, L. Kharb, and D. Chahal, Eds. Singapore: Springer, 2017, pp. 269–281.

[44] S. Fischer, R. E. Lopez-Herrejon, and A. Egyed, "Towards a fault-detection benchmark for evaluating software product line testing approaches," in *Proc. 33rd Annu. ACM Symp. Appl. Comput. (SAC)*, Pau, France, 2018, pp. 2034–2041, doi: 10.1145/3167132.3167350.

[45] D. Garg, A. Datta, and T. French, "New test case prioritization strategies for regression testing of Web applications," *Int. J. Syst. Assurance Eng. Manage.*, vol. 3, no. 4, pp. 300–309, Dec. 2012, doi: 10.1007/s13198-012-0134-5.

[46] X. Li, W. E. Wong, R. Gao, L. Hu, and S. Hosono, "Genetic algorithm-based test generation for software product line with the integration of fault localization techniques," *Empirical Softw. Eng.*, vol. 23, no. 1, pp. 1–51, Feb. 2018, doi: 10.1007/s10664-016-9494-9.

[47] C. Magalhães, J. Andrade, L. Perrusi, and A. Mota, "Evaluating an automatic text-based test case selection using a non-instrumented code coverage analysis," in *Proc. 2nd Brazilian Symp. Systematic Automated Softw. Test. (SAST)*, Fortaleza, Brazil, 2017, pp. 1–9, doi: 10.1145/3128473.3128478.

[48] W. Jin, Y. Gao, X. Yin, F. Li, and H. J. Kim, "An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–21, Dec. 2018.

[49] O. Hummel and C. Atkinson, "The managed adapter pattern: Facilitating glue code generation for component reuse," in *Formal Foundations of Reuse and Domain Engineering*, vol. 5791, S. H. Edwards and G. Kulczycki, Eds. Berlin, Germany: Springer, 2009, pp. 211–224.

[50] J. S. Fant, H. Gomaa, and R. G. Pettit, "A pattern-based modeling approach for software product line engineering," in *Proc. 46th Hawaii Int. Conf. Syst. Sci.*, Wailea, HI, USA, Jan. 2013, pp. 4985–4994, doi: 10.1109/HICSS.2013.52.

[51] C. Seidl, S. Schuster, and I. Schaefer, "Generative software product line development using variability-aware design patterns," in *Proc. ACM SIGPLAN Int. Conf. Generative Program. Concepts Exp. (GPCE)*, Pittsburgh, PA, USA, 2015, pp. 151–160, doi: 10.1145/2814204.2814212.

[52] S. Hussain, J. Keung, M. K. Sohail, A. A. Khan, M. Ilahi, G. Ahmad, M. R. Mufti, and M. A. Noor, "A methodology to rank the design patterns on the base of text relevancy," *Soft Comput.*, vol. 23, no. 24, pp. 13433–13448, Dec. 2019, doi: 10.1007/s00500-019-03882-y.

[53] G. Catolino and F. Ferrucci, "An extensive evaluation of ensemble techniques for software change prediction," *J. Softw. Evol. Process*, vol. 31, no. 9, Sep. 2019, doi: 10.1002/smr.2156.

[54] C. Ma and J. Provost, "A model-based testing framework with reduced set of test cases for programmable controllers," in *Proc. 13th IEEE Conf. Autom. Sci. Eng. (CASE)*, Xi'an, China, Aug. 2017, pp. 944–949, doi: 10.1109/COASE.2017.8256225.

[55] N. Zaman and M. Ahmad, "Towards the evaluation of authentication protocols for mobile command and control unit in healthcare," *J. Med. Imag. Health Informat.*, vol. 7, no. 3, pp. 739–742, Jun. 2017.

[56] Z. A. Almusaylim, N. Zaman, and L. T. Jung, "Proposing a data privacy aware protocol for roadside accident video reporting service using 5G in vehicular cloud networks environment," in *Proc. 4th Int. Conf. Comput. Inf. Sci. (ICCOINS)*, Kuala Lumpur, Malaysia, Aug. 2018, pp. 1–5, doi: 10.1109/ICCOINS.2018.8510588.

[57] D. A. Shafiq, N. Jhanjhi, and A. Abdullah, "Proposing a load balancing algorithm for the optimization of cloud computing applications," in *Proc. 13th Int. Conf. Math., Actuarial Sci., Comput. Sci. Statist. (MACS)*, Karachi, Pakistan, Dec. 2019, pp. 1–6, doi: 10.1109/MACS48846.2019.9024785.

[58] Z. A. Almusaylim and Z. N. Jhanjhi, "Comprehensive review: Privacy protection of user in location-aware services of mobile cloud computing," *Wireless Pers. Commun.*, vol. 111, pp. 541–564, Oct. 2020, doi: 10.1007/s11277-019-06872-3.

[59] A. Nayyar, *Handbook of Cloud Computing: Basic to Advance Research on the Concepts and Design of Cloud Computing*. New Delhi, India: BPB Publications, 2019.

[60] G. Deep, R. Mohana, A. Nayyar, P. Sanjeevikumar, and E. Hossain, "Authentication protocol for cloud databases using blockchain mechanism," *Sensors*, vol. 19, no. 20, p. 4444, Oct. 2019.

[61] P. K. D. Pramanik, G. Pareek, and A. Nayyar, "Security and privacy in remote healthcare: Issues, solutions, and standards," in *Telemedicine Technologies*. New York, NY, USA: Academic, 2019, pp. 201–225.

[62] J. Vora, A. Nayyar, S. Tanwar, S. Tyagi, N. Kumar, M. S. Obaidat, and J. J. P. C. Rodrigues, "BHEEM: A blockchain-based framework for securing electronic health records," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–6.

[63] J. Zhang, K. Yang, L. Xiang, Y. Luo, B. Xiong, and Q. Tang, "A self-adaptive regression-based multivariate data compression scheme with error bound in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 3, Mar. 2013, Art. no. 913497.

[64] J. Chunwei, Y. Gao, A. K. Sangaiah, and G.-J. Kim, "A PSO based energy-efficient coverage control algorithm for wireless sensor networks," *Comput., Mater. Continua*, vol. 56, no. 3, pp. 433–446, 2018.

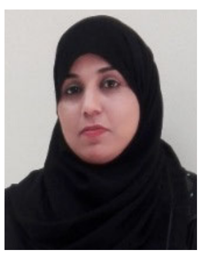
[65] S. Singh, P. K. Sharma, B. Yoon, M. Shojafar, G. H. Cho, and I.-H. Ra, "Convergence of blockchain and artificial intelligence in IoT network for the sustainable smart city," *Sustain. Cities Soc.*, vol. 63, Jul. 2020, Art. no. 102364. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210670720305850

SADIA ALI received the Master of Science degree in computer science from PMAS Arid Agriculture University, Rawalpindi, Pakistan, in 2017. She is currently pursuing the Ph.D. degree with the University Institute of Information Technology, PMAS Arid Agriculture University, under the supervision of Dr. Y. Hafeez (University Institute of Information Technology). Her main current research interests include developing requirements engineering, requirement management, data mining, text mining, and testing solutions for component-based systems.

YASER HAFEEZ received the B.Sc. (Hons.) and master's degrees in computer science and the Ph.D. degree from International Islamic University, Islamabad, Pakistan. He has 19 years of teaching, research, and administrative experience with diversified learners at the undergraduate and graduate levels, taught variety of computing related courses. He is currently an Associate Professor/the Director with the University Institute of Information Technology, PMAS Arid Agriculture University, Rawalpindi, Pakistan. He has supervised various master's and Ph.D. thesis. His research interests include requirements engineering process, global software engineering, agile software development practices, software quality engineering, data mining, and software testing.

N. Z. JHANJHI was with ILMA University and King Faisal University (K.F.U.), Saudi Arabia, for a period of ten years. He has 20 years of teaching and administrative experience. He has great international exposure in academia, research, administration, and academic quality accreditation. He has an intensive background of academic quality accreditation in higher education besides scientific research activities. He was with Academic Accreditation, for a period of ten years. He was with the National Commission for Academic Accreditation and Assessment (NCAAA) and the Education Evaluation Commission Higher Education Sector (EECHES) formerly NCAAA, Saudi Arabia, for Institutional Level Accreditation. He was also with the National Computing Education Accreditation Council (NCEAC). He is currently an Associate Professor with Taylor's University Malaysia. He is also a Moderator with the IEEE TechRxiv, a keynote speaker with the several IEEE international conferences globally, an external examiner/evaluator for the master's and Ph.D. degrees for several universities. He has supervised several postgraduate students, including the master's and Ph.D. He has edited/authored more than 13 research books with international reputed publishers. He has earned several research grants and a great number of indexed research articles on his credit.

He is an active TPC member of reputed conferences around the globe. He received the ABET Accreditation twice for three programs from CCSIT, King Faisal University. He has awarded as a Top Reviewer one % globally from WoS/ISI (Publons), in 2019. He serves as a Guest editor for several reputed journals and a member of the editorial board for several research journals. He serves as an Associate Editor for IEEE ACCESS.

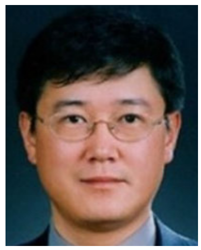


MAMOONA HUMAYUN received the Ph.D. degree in computer architecture from the Harbin Institute of Technology, China. She has 12 years of teaching and administrative experience internationally. She has supervised various master's and Ph.D. thesis. Her research interests include global software development, requirement engineering, knowledge management, cyber security, and wireless sensor networks. She serves as an active reviewer for a series of journals.

MUHAMMAD IMRAN received the M.I.T. degree in information technology from the University of the Punjab, Pakistan, in 2007, the M.S.I.T. degree in information technology from the National University of Science and Technology, in 2011, and the Ph.D. degree in electronic engineering from the School of Electronic Engineering, Dublin City University, Ireland. He has been a Senior Scientific Officer (Permanent) with the National Centre for Physics, Research Institute, Pakistan, since July 2008. He is currently with the C.M.S. Offline Group, CERN, Geneva, Switzerland. His research interests include software engineering, big data, cloud computing, and software defined networks.

ANAND NAYYAR (Senior Member, IEEE) received the Ph.D. degree in computer science (wireless sensor networks) from Desh Bhagat University, in 2017. He is currently with the Graduate School, Duy Tan University, Da Nang, Vietnam. He was a Certified Professional with more than 75 professional certificates from CISCO, Microsoft, Oracle, Google, Beingcert, EXIN, GAQM, Cyberoam, and so on. He has published more than 300 research papers in various national and international conferences and international journals, such as Scopus/SCI/SCIE/SSCI Indexed. He has authored/coauthored cum edited 25 books of computer science. He holds two Patents in the Internet of Things and speech processing. His current research interests include wireless sensor networks, MANETS, swarm intelligence, cloud computing, the Internet of Things, blockchain, machine learning, deep learning, cyber security, network simulation, and wireless communications. He is a member of more than 50 Associations, as a Senior Member and a Life Member. He is also an A.C.M. Distinguished Speaker. He was associated with more than 400 international conferences, as a programme committee/advisory board/review board member. He received more than 20 Awards for Teaching and Research—the Young Scientist, the Best Scientist, the Young Researcher Award, the Outstanding Researcher Award, and the Publons-Top one % Reviewer Award (computer science, engineering, and cross-fields). He serves as an Editor-in-Chief for the *International Journal of Smart Vehicles and Smart Transportation (IJSVST)* (IGI-Global, USA).

SAURABH SINGH received the bachelor's degree from Uttar Pradesh Technical University, the master's degree in information security from Thapar University, and the Ph.D. degree from Chonbuk National University, Jeonju, South Korea. He held a postdoctoral position with Kunsan National University, South Korea. He has an experience of being a Laboratory Leader. He is currently an Assistant Professor with Dongguk University, Seoul, South Korea. He holds a strong academic record. He has published many SCI/SCIE journals and conference papers. His research interests include blockchain technology, ubiquitous security, cloud computing and security, the IoT, deep learning, and cryptography. He received the Best Paper Award from KIPS and CUTE Conference, in 2016. He served as a Guest Editor for *Sustainability Journal* (MDPI). He serves as a Reviewer for the IEEE INTERNET OF THINGS, *IJCS*, IEEE ACCESS, the *IEEE Magazine*, *FGCS*, *Journal of Supercomputing*, *IJDSN*, *Compeleceng*, and so on.



IN-HO RA (Member, IEEE) received the Ph.D. degree in computer engineering from Chung-Ang University, Seoul, South Korea, in 1995. He was a Visiting Scholar with the University of South Florida, Tampa, FL, USA, from February 2007 to August 2008. He was a Faculty Member with the School of Computer, Information and Communication, Kunsan National University, where he is currently a Professor. His major research interests include wireless ad hoc and sensor networks, blockchain, the IoT, PS-LTE, and microgrid.