DEVELOPMENT AND RESEARCH ON BIG DATA USING SOFTWARE TESTING TECHNOLOGY

[1]Kolanu Shravani, Research Scholar, Department of CSE, Dr. A.P.J. Abdul Kalam University, Indore, Madhya Pradesh, India
[2]Dhanraj verma, Research Guide, Department of CSE, Dr. A.P.J. Abdul Kalam University, Indore, Madhya Pradesh, India

ABSTRACT: Here large data is introduced to improve the technology in computer science. This data is described as big data which is accessed by using internet. In this paper the development of big data is presented using software testing technology. This system will determine the quality, stability and reliability of the system. High performance is obtained in the system by determining the big data analysis. There is a need to develop a testing framework which is application independent and scalable with the increased requirements of each application. Hence the proposed system gives effective result in terms of reliability and quality.

KEY WORDS: Software Testing, Automated Testing Framework, Software non linear module Framework, Selenium Web Driver.

I.INTRODUCTION

Big data technology plays important role in this present generation. To get high performance software testing is introduced in the big data technology. Basically, for given input, desired output is obtained in the software testing. This software testing uses number of variation while processing the operation in the system. Testing process is done so as to sort any mistakes or discovering programming bugs concerning the real desire for the outcome. Quality of the product is determined by the software testing procedure [1]. Basically, software testing is nothing but determining the quality of software by following the IEEE standards. In IT industries, software testing for each cycle of time is recorded and this is one of the important parts in entire software testing system. The requirements should be given to estimate the procedure of software testing. While performing the developing phase of software, unit testing procedure is introduced.
Hence, while performing software testing time is recorded at every clock cycle in the software development phase [2].

The main intent of software testing is to provide the high quality. This software testing is low cost and reliable, in the same way it is user satisfactory. Basically, testing of software is more complex by using the array programming language. Hence to overcome this complexity, an automated software testing tool is introduced. by using this tool, low cost and reliable output is obtained. In automatic testing process a test case is developed to get effective output. This software testing tool is used to test the ap [placation form depend on the performance of load [3].

Big data is an all-encompassing term for any collection of data sets so large and complex that it becomes difficult to process them using traditional data processing applications. Big data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, curate, manage, and process data within a tolerable elapsed time.

Big data "size" is a constantly moving target, as of 2012 ranging from a few dozen terabytes to many peta bytes of data. Big data is a set of techniques and technologies that require new forms of integration to uncover large hidden values from large datasets that are diverse, complex, and of a massive scale. Big data uses inductive statistics and concepts from nonlinear system identification to infer laws from large sets of data with low information density to reveal relationships, dependencies and perform predictions of outcomes and behaviors.

Due to such large size of data it becomes very difficult to perform effective analysis using the existing traditional techniques. Big data due to its various properties like volume, velocity, variety, variability, value, complexity and performance put forward many challenges. Testing Big data is one of biggest challenge faced by every organization because of lack of knowledge on what to test and how to test. Biggest challenges faced in defining test strategies for structured and unstructured data validation, setting up an optimal test environment, working with non relational database and performing non –functional testing.

These challenges cause poor quality of data in production and delayed implementation and increase in cost. The big data application will handle a large number of structured and unstructured data. The data processing will involve more than one data node and completed in a shorter period of time. Due to the low quality and poor system design code, application performance as data volume growth will decline, even when the amount of data reaches a certain size, the application crashes and cannot provide mission services. If the performance of the application does not meet the service level agreements (Service-Level Agreement, SLA), will lose the goal of building big data systems. Therefore, due to data capacity size and complexity of systems in big application, performance testing has played a very important role to achieve the actual performance ability.

In this paper, the design of big data technology using software testing is presented. This testing process will improves the accuracy of the system. In the same way it ill saves the time of the system. An automated software testing tool will help the testers to authenticate the test cases and in the same way it will test the entire system automatically. The main advantage of this automated testing tool is, it can be reused and less effort is required to implement the system.

Generally, the software linear data driven testing model will use the specific logic to drive the data and the explanation is divided into three categories mainly which are given below:

(1) At first they will provide the communication between the internal tested object name and interface element name. By using the recording, the application and test scripts will provide level of abstraction. By using mapping procedure, particular object and the elements are interfaced with other. Hence in this system, testing procedure will not effect, only mapping table is affected.

(2) Here the test description is explained in detail manner. By using the description process, one can see an overview about the expected outcome for given specific inputs. Here the details of descriptor is interrelated with each other in the development platform.

(3) At last the dividing of information and test contents are given here. The data is extracted finally from the test scripts. The both data and script maintain their relation and

communication independently [5]. Impact of the system is reduced by performing the test script operation.

II. BIG DATA CHARACTERISTICS AND DATA FORMATS

Generally, with the term Big Data, we get the idea of a huge volume of information. Big data computation frames another pattern for inescapable processing with the degree of information developing and the quickness of data expanding. With the origin of new advancements, an enormous amount of organized and unstructured data is delivered, gathered from different sources like social media, audios, websites, video and so forth which is hard to oversee and process.

Big data possess the characteristic property of volume, velocity, variety, and veracity, which makes its behavior dynamic in nature. And this nature of big data streams makes the testing process crucial, which if not performed efficiently will pose adverse effects on organizations. Big data testing benefits the organizations substantially by providing data accuracy, better decision making; which plays a vital role in any business, better market strategy, reduces the deficit and boost profits and a lot more. For testing big data, it requires that the tester must have an optimal level of understanding what big data framework is followed. Today, with the huge volume of data generated can have a hard time managing them. And this makes it quite crucial to perform big data testing and harness its advantages.

Though there are many V’s introduced, following 4Vs define the big data characteristics. These are Variety, Velocity, Volume and Veracity as shown in figure (1).

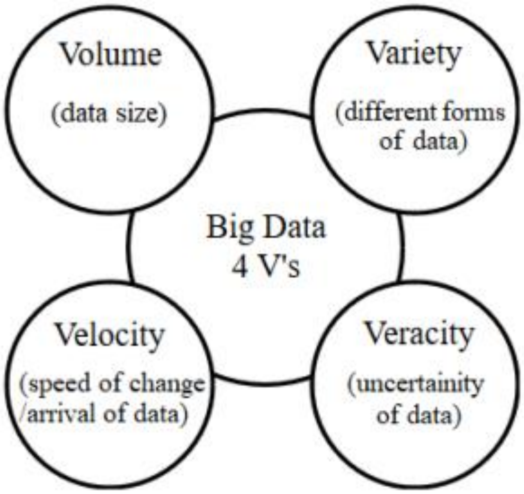

Fig. 1: 4 V’s of big data

Variety: (different forms of data) Data or information that comes in for the processing can be of a variety of forms and formats. For instance data can come in different file formats like .txt, .csv, .xlsx, etc. Sometimes the information may not be formatted in the desired manner e.g.; data can come in the form of SMS, audio, video, pdf or other doc format or something we may not have contemplated it. It makes it quite crucial for the organizations to handle such a wide variety of data efficiently as at present wide range of formats of data are available to seek information from it.

Velocity: (speed of change/arrival of data) This characteristic of big data provides the glimpse of the pace of data i.e., at what rate data is arriving in from various sources like networks, social media, and other business processes. This high-speed real-time data is massive and comes in a continuous fashion which may need immediate processing. There is even possibility of mutation in the data over time

Volume: (data size) At present data comes in/generated from different sources by machines, networks, and media, from which valid information is extracted and aggregated at the organization hub. For instance consider the case of a variety of data sent by smart mobile phones to the network infrastructure, the information collected from various surveys, feedback forms, etc., this aggregated information forms the enormous size of data which needs to be properly analyzed

Veracity: (uncertainty of data) There are a wide variety of sources of data stream available which produce a huge amount of data. With these many available sources, this data becomes vulnerable to outliers or noise. Due to which the nature or behavior of the data may change. The term Veracity describes this as the uncertainty of data which poses a huge impact on the decision-making process of the organization. Based on the above characteristics, data comes in with different sizes, formats, rate, etc., thereby resulting in the following categories of data format.

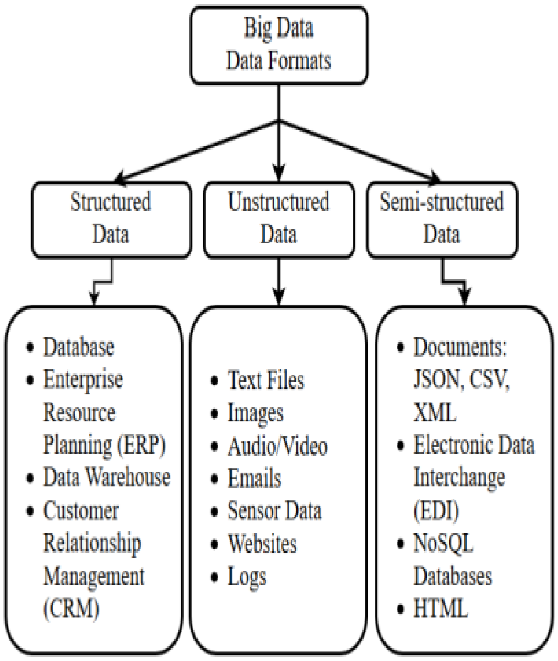

Fig. 2: Big Data: data formats

Structured Data: (high degree of organization) Structured data comprises of definite design in a well-organized manner such that data is easily utilized for processing and analysis. As an example, we can consider the relational database structure in which information is stored with some standards such that the same standards can be utilized in retrieving the information. This data format has a relational key and can be easily mapped into pre-designed fields.

Unstructured Data: (low degree of organization) This type of data does not follow any kind of pattern. This makes it quite difficult to analyze and may require some tools to extract desired information for processing, for example, streaming data, web pages, electronic mails, videos, etc. This lack of organization of data makes it a time-consuming task to process data. Most of the real-time data stream is unstructured. However, by identifying some hidden pattern, we can utilize it for the desired purpose but finding that pattern is difficult and time-consuming.

Semi-Structured Data: (partially organized data) This type of data can be organized by applying a bit of desired operations as in conversions, shifting, etc. There is various software that handles this kind of data like Apache Hadoop. Sometimes we call this type of data as structured data, which is lacking some sequence or pattern and is available in an unorganized manner. Such kind of information can come in the form of tab-delimited text files, CSV files, BibTex files, XML, web data such as JSON (JavaScript Object Notation) files, and other markup languages. This type of data is easier to handle as compared to completely unstructured data where we need to spend little effort to identify the pattern and process the information.

III. BIG DATA TESTING APPROACH AND STRATEGY

Big data testing is often associated with varied sorts of testing, for example, functional testing, performance testing, database testing, and infrastructure testing. Along with these, it is critical to have an unmistakable test plan that permits a simple rendering of big data testing. When performing big data testing, comprehend that the idea is mainly about checking the application capability to handle thousands of gigabytes of data. Big data testing for CPS are often generally partitioned into three vital stages that incorporate:

Data staging validation: Also referred to as a preHadoop stage, the method of big data testing starts with process validation, which aids in guaranteeing if right data is pushed into the “Hadoop Distributed File System (HDFS)”. Validation testing is done for the data which is taken from different references, for example, RDBMS, and online networking. Then the data is coordinated with the data utilized in the Hadoop process so as to check if the two coordinate with one another. A number of normal tools that might be utilized for this stage are “Talented and Datameer”.

MapReduce validation: It is the idea of programming that takes into account colossal adaptability crosswise over a huge number of servers in a Hadoop cluster. Amid big data testing, Map Reduce second stage is the validation in which a tester analyzes the legitimacy of business logic on every joint which is taken post the validation of the previous data after running opposite various joints.

Output Validation: On effectively performing the initial two stages, the last stage of the method is “output validation”. It incorporates processed files which are prepared to be passed to an “Enterprise Data Warehouse (EDW)” or some other system in the view of particular necessities. Output validation stage incorporates the belowmentioned steps:

- Need to validate change rules are effectively applied.

- Need to validate the data respectability and in addition effective data lading into the resulting organization.

- Guaranteeing that any data defilement by differentiating the objective data and the HDFS file organization data.

Data ingestion and Throughout: During this activity, the tester checks for the speediness of the system in which it can consume data from numerous data references. Testing incorporates the recognizing of different messages that can be processed by the queue in a given time. It additionally incorporates how rapidly data can be implanted into the underlying data store for instance insertion rate into a Mongo and Cassandra database.

Data Processing: It incorporates the conformance of the speed with which the queries or map-reduce jobs are performed. It likewise incorporates testing of the data handling in segregation when the repressed data store dwells inside the data sets.

Software testing is among the most critical parts of the software development process. The creation of tests plays a substantial role in the evaluation of software quality yet being one of the most expensive tasks in software development. This process typically involves intensive manual efforts and it is one of the most labor-intensive steps during software testing.

To reduce manual efforts, automated test generation has been proposed as a method of creating tests more efficiently. In recent decades, several approaches and tools have been proposed in the scientific literature to automate the test generation. Yet, how these automated approaches and tools compare to or complement manually written is still an open research question that has been tackled by some software researchers in different experiments. In the light of the potential benefits of automated test generation in practice, its long history, and the apparent lack of summative evidence supporting its use, the present study aimed to systematically review the current body of peer-reviewed publications on the comparison between automated test generation and manual test design. We conducted a systematic literature review and meta-analysis for collecting data from studies comparing manually written tests with automatically generated ones in terms of test efficiency and effectiveness metrics as they are reported.

This used a set of primary studies to collect the necessary evidence for analyzing the gathered experimental data. The overall results of the literature review suggest that automated test generation outperforms manual testing in terms of testing time, test coverage, and the number of tests generated and executed. Nevertheless, manually written tests achieve a higher mutation score and they prove to be highly effective in terms of fault detection. Moreover, manual tests are more readable compared to the automatically generated tests and can detect more special test scenarios that the ones created by human subjects. Our results suggest that just a few studies report specific statistics (e.g., effect sizes) that can be used in a proper meta-analysis.

The results of this subset of studies suggest rather different results than the ones obtained from our literature review, with manual tests being better in terms of mutation score, branch coverage, and the number of tests executed. The results of this meta-analysis are inconclusive due to the lack of sufficient statistical data and power that can be used for a meta-analysis in this comparison. More primary studies are needed to bring more evidence on the advantages and disadvantages of using automated test generation over manual testing.

IV. MODULES OF DATA TECHNOLOGY USING SOFTWARE TOOLS

By defining data module of software testing, the software testing procedure is controlled. This entire software data module is divided into three modules mainly which are given. They modules are script language module, support library module and the core module. Among these three, the core data module plays an important role to drive the non linear data in software testing module. This core module will control the implementation of scripts procedure. The control module is mainly divided into four parts they are the script actuator, data parser, the middle layer and the script parser. To call the test middle layer is used, to implement the script, script actuator is used, to analyze the script, script parser is used and at last to analyze the logic, data parser is used.

A. Core module:

This module will control and implement the no linear data while testing the software. The control module for non linear data is mainly divided into four parts they are the script actuator, data parser, the middle layer and the script parser. To call the test middle layer is used, to implement the script, script actuator is used, to analyze the script, script parser is used and at last to analyze the logic, data parser is used.

B. Data access module:

This module will store the data and modify the scripts in this the script will divide the procedure into three layers; they are high level layer, low level layer and medium level layer. The relationship between the modules is maintained by the both high level layer script and low level layer script.

C. Interface module:

To enhance the framework of the system interface module is used. This will allow the user to drag and resize the script. This done only when there will be communication between graphical interface and elliptical curve interface. This is user friendly to use and understand the process. Here the users can modify the existed scripts.

D. Support module

There are two parts in the support module. They are library and log library. The library will provide the sharing of test that is performed. In the same way, log library is used to test and support the libraries.

E. Interpret scripting language module

The interpret module is mainly divided into three types. They are parser, interpreter and analyzer. If the words are passed in a flow depend on the analysis then that process is known as analyzer. In the same way, if the words are passed in sequence manner, then that process is known as parser. At last interpreter is nothing but which is responsible to translate the information in translate way.

V. BIG DATA USING SOFTWARE TESTING TECHNOLOGY

From figure (3) the architecture of big data using software testing technology is presented. In this architecture mainly data specification, data requirement, mapping, test cases generation and execution process is done. The entire framework is depending on the testing procedure. Let us discuss each device in detail manner.

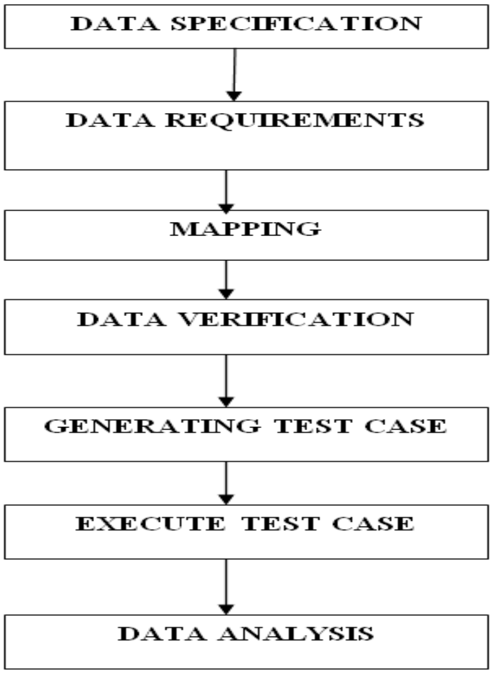

Fig. 3: PROPOSED SYSTEM

The First step: create the data specification. Create a data specification based on the user's requirements. S

econd step: verify the data specification. Verify the data specification from the consistency and effectiveness of the data model.

Third step: determine the rules and Strategies of the test. Determine test standards, test coverage, test selection strategy, and test usage algorithms, etc.

Fourth step: generate test cases. This is the core process of the whole test, generating test cases based on the rules and policies of the test. In this process, it is also necessary to test the test drive module.

The fifth step: executing test cases and tracking. In this process, we can perform manual or automatic execution according to the test cases, compare the actual output results with the expected results, and record the data results.

The sixth step: the analysis of the test results.

Finally, the results of the test are analyzed, and the results can be used as the basis of the test case generation and optimizationn.

Mapping operation is performed by using the test scripts. The obtained reports in the system are modified, and kept in a particular format. While running the test script, the test suit will give an authentication in the case of failing position. This software non linear testing module is many applications to get the exact test sript or output. There will be effective communication between the user and interface module. Coming to the open browse module application it gives effective output. Here the non linear data is used to open the chrome and browser. After opening, this open browse module will navigate the communication and information to the system. The opened file is saves as an URL file and open web browser will save that information in driver script.

Next the data is given as input to the text case. This test case is linked to the email address and password. Now the obtained text is saved in the workbook. The non linear data for software testing is verified using the verifying element. To load the web page it will take some time. If the driver script is missed then the web page will not be loaded. Now to store this data, the web page will be attached to the console window. This system will print the text after data is stored in effective way. At last after performing all this test cases, the web page will be closed. At last the proposed system gives test results in effective way.

VI. RESULTS

From below figure (4) shows the output graph of proposed system.

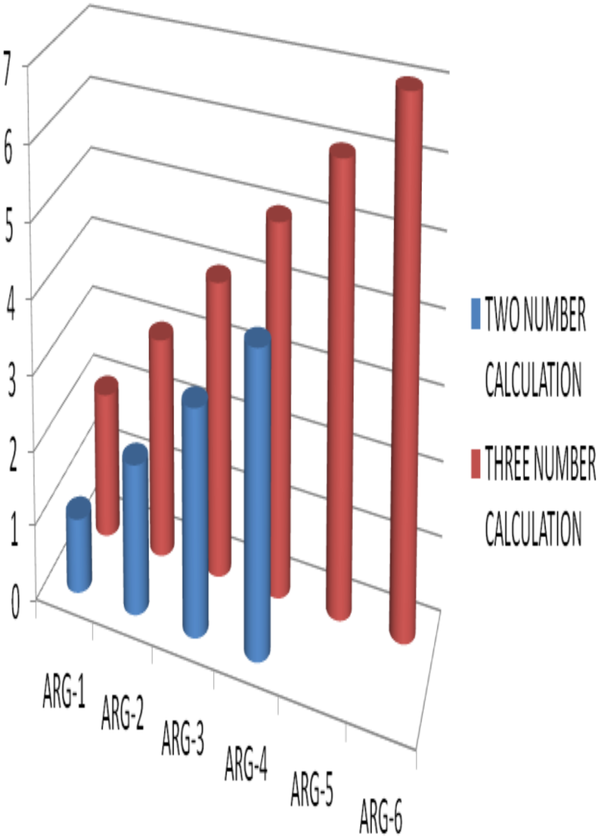

Fig. 4: OUTPUT OF BIG DATA CALCULATION

The below figure (5) shows the accuracy and reliability of proposed system.

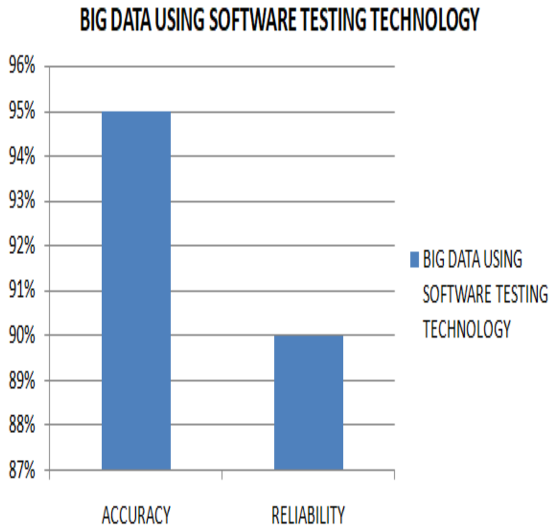

Fig. 5: ACCURACY AND RELIABILITY

Table. 1: Accuracy And Reliability Of Big Data Using Software Testing Technology

Parameters	**Big Data Using Software Testing Technology**
Accuracy	95%
Reliability	90%

VII. CONCLUSION

Big data technology using software testing tool design is presented in this paper. Software testing is the strategy for testing application so as to discover the variety between the yield and the real yield. Computerized testing doesn't required much human exertion as the testing of use is performed with the assistance of other application however it is a repetitive occupation as we have to test the application over and over for each little change in the application. To defeat with these issues a big data technology Framework is proposed in this paper.

Here an exceed expectations exercise manual which contains every one of the watchwords is utilized for which the capacities or strategies are made in the Framework content record. The proposed system doesn't require the information of programming language for testing web application as all the code is as of now implanted in Framework content record. Besides there is no compelling reason to modify the code to test various applications and just the exceed expectations exercise manual is changed which is simple. The trial results demonstrate that the system is cost productive and simple to utilize.

VIII. REFERENCES

[1] Zhenhua Zhang. The challenge and Prospect of software testing under the background of large data. Electronic Technology and Software Engineering, pp.61,2016.

[2] Bing Fu. Research on the status and development trend of software testing technology. Computer Programming Skills and Maintenance, pp. 31-32,2016

[3] Nengji Chen, Zhiguo Huang. Software testing technology Daquan: test foundation popular tool project real battle (Third Edition) . Posts and Telecom Press, 2015.

[4] Xun Yang. Software testing technology research. Computer Knowledge and Technolgy, vol. 11, issue 28, pp. 207-208,2015 .

[5] Lizhi Cai, ting Yan. The challenge and Prospect of software testing under the background of Big Data. Computer Applications and Software, vol. 31, issue 2, pp. 5-8, 2014 .

[6] Hua Ning,Yongzheng Chen,Zhenglong Zhang. Software testing technology and tools. Modern Enterprise Education, pp. 590, 2014.

[7] L. Singer, F. Figueira Filho, B. Cleary, C. Treude, M.-A. Storey, and K. Schneider, "Mutual assessment in the social programmer ecosystem: an empirical investigation of developer profile aggregators," in Proceedings of the 2013 conference on Computer supported cooperative work, 2013, pp. 103-116.

[8] Jihua Liu, Ce Chen. Progress in the research of software testing technology. Microcomputer Information, vol. 28, issue 10, pp. 494- 496, 2012.

[9] W. Shang, Z. M. Jiang, B. Adams, and A. E. Hassan, "MapReduce as a general framework to support research in Mining Software Repositories (MSR)," in Mining Software Repositories, 2009. MSR'09. 6th IEEE International Working Conference on, 2009, pp. 21-30.

[10] Yunzhan Gong. Research progress in software testing technology. The 10th National Conference on Fault-tolerant Computing Academic, pp. 44-50,2003.

[11] Chen, Min, Shiwen Mao, and Yunhao Liu. "Big data: A survey." Mobile networks and applications 19.2 (2014): 171-209.

[12] Adiba Abidin, Divya Lal, Naveen Garg, Vikas Deep “Comparative Analysis on Techniques for Big Data Testing,” 2016 InCITe

[13] Zikopoulos, Paul, and Chris Eaton. Understanding big data: Analytics for enterprise class hadoop and streaming data. McGraw-Hill Osborne Media, 2011.

[14] Shumeli, Galit, Mohit Dayal, and Bhimasankaram Pochiraju. "Testing Theories with Big Data: A Super-Power Approach." (2012).

[15] Batterywala, Mustafa, and Shirish Bhale. "Performance Testing of Big Data Applications." Impetus Technologies, STC (2013).

[16] Mahesh Gudipati, Shanthi Rao, Naju D. Mohan and Naveen Kumar Gajja, ³Big Data: Testing Approach to Overcome Quality Challenges´, Infosys Labs Briefings, Vol 11, No 1, 2013

[17] Campos, Jaime, Pankaj Sharma, Unai Gorostegui Gabiria, Erkki Jantunen, and David Baglee. "A big data analytical architecture for the Asset Management." Procedia CIRP 64 (2017): 369-374.

[18] H. M. Sneed, K. Erdoes, "Testing big data (Assuring the quality of large databases)", 2015 IEEE Eighth International Conference on Software Testing Verification and Validation Workshops (ICSTW), pp. 1-6, 2015.

[19] Mukherjee, Rajendrani, and Pragma Kar. "A comparative review of data warehousing ETL tools with new trends and industry insight." In 2017 IEEE 7th International Advance Computing Conference (IACC), pp. 943-948. IEEE, 2017.

[20] Alexandrov, Alexander, Christoph Brücke, and Volker Markl. "Issues in big data testing and benchmarking." In Proceedings of the Sixth International Workshop on Testing Database Systems, p. 1. ACM, 2013.