Volume 7, Issue 1, January 2017 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Automated Testing Approach in Data Mining System using Genetic Algorithm Framework

K. Thulasiram
Research Scholar, Department of Computer Science,
Sri Venkateswara University, Tirupati,
Andhra Pradesh, India

Dr. S. Ramakrishna
Professor, Department of Computer Science,
Sri Venkateswara University, Tirupati,
Andhra Pradesh, India

DOI: 10.23956/ijarcsse/V7I1/0159

Abstract— Software testing performances are usually designed by human experts, while test automation tools are appraisal of test outcomes is also associated with a considerable effort by software testers who may have imperfect knowledge of the requirements specification. This paper presents a method for optimizing software testing efficiency by identifying the most critical path clusters in a program. The factors discovered are used in evaluating the fitness function of Genetic algorithm for selecting the best possible Test method. This integration will help in improving the common performance of genetic algorithm in search space exploration and exploitation fields with improved convergence rate. To improve testing productivity and reduce costs, it is highly desirable to automate test generation and execution. The extensive software testing is infrequently feasible because it becomes difficult for even medium sized software. Typically only parts of a program can be tested, but these parts are not essentially the most error prone. Consequently, we are developing a additional selective approach to testing by focusing on those parts that are most significant so that these paths can be tested first. By identifying the most significant paths, the testing efficiency can be increased.

Keywords— Automated Testing, Data Mining, Genetic Algorithm, Selenium IDE.

I. INTRODUCTION

Software testing is a process, which is used for evaluating the functionality of a software program. Software testing is also a procedure through which software item is evaluated to identify the defects and then approved them. It is also used to assess the characteristic of a software system and as well as the quality of the product. The software testing should be done during the development process. In other words software testing is approved as confirmation and validation process that a computer program or application or product should meet the prerequisite that is used in software designing and development. The design of the data mining based background for automated black-box testing. The Random Tests Generator (RTG) Module obtains the list of system inputs, their types (discrete, continuous, etc.), and ranges from the Specification of System Inputs (SSI) Module[9]. No information about system functionality is needed, since Data Mining algorithms can automatically disclose the practical requirements from a training set of randomly generated test cases.

II. LITERATURE REVIEW

A. Data Mining

Data mining is used to describe the process of analyzing data to find patterns & relations, classify data and predict results in large data sets of structured data. It can be seen as a region wherever machine learning, computer science and statistics meet on common grounds. Different techniques that also fit in this category include association learning, data classification, clustering and regression. Data mining is the process of extracting knowledge hidden from large databases. The knowledge must be new and one must be proficient to utilize it. Information discovery differs from conventional data retrieval from databases. In conventional DBMS, database reports are returned in response to a query; while in knowledge detection, the data retrieved is not clear in the database. Somewhat, it is hidden patterns. The procedure of discovering such patterns is termed data mining. Data mining finds these patterns and relationships using data analysis tools and techniques to construct models. Here are two most important kinds of models in data mining. First one is predictive model, which utilizes information with identified results to develop a model that can be used explicitly to expect values. Another is descriptive model, which describes patterns in accessible information. It is a dominant new knowledge with grand potential to help companies focus on the most important information in their data warehouses. Data mining tools forecast potential trends and behaviors, allow businesses to make practical, knowledge-driven decisions[10,6].

© 2017, IJARCSSE All Rights Reserved

Data mining has been applied effectively in business environment and also in other fields such as weather conditions forecast, medicine, healthcare, insurance, transportation, government and etc. Data mining brings a set of advantages when used in a specific industry. The main purpose of data mining is to extract patterns from the information at hand, increase its fundamental value and transfer the data into knowledge.

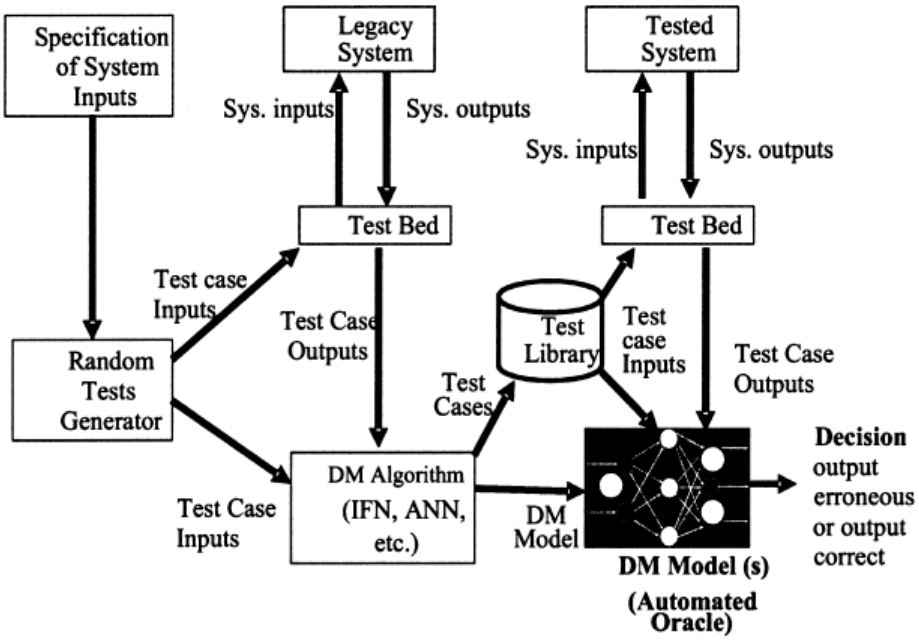

Fig.1: Data Mining based Functional Testing

B. Machine learning

Machine learning is a learning technique in the area of artificial intelligence which deals with making intelligent decisions and drawing conclusions based on the features of the input data rather than preset regulations. The thought is to be able to resolve problems where the algorithm is unknown or the problem becomes unsolvable due to constantly changing requirements. By analysing existing data it is possible to detect patterns and relations in the data. From the conclusions made with existing data, the algorithm can then "learn" to make perfect predictions when encountering fresh data for the similar problem by producing common regulations that can be used. Machine learning can be divided into two different categories: supervised learning and unsupervised learning. Supervised learning deals with methods such as classification, prediction and regression for which the user is required to supervise the algorithms during the learning phase and tweak the settings to optimize the solution. With unsupervised learning which instead deals with methods for clustering and compression, the algorithm themselves handle the modification. A common need in analysing large quantities of data is to divide the dataset into logically well divergent groups such that the objects in every group distribute some property which does not hold for other objects. As such, clustering searches a universal type of data, generally with the main focus on associating every object with a cluster even though in some cases we are interested in understanding where clusters are located in the data space.

C. The Framework

The most common structure for input data when it comes to data mining algorithms consist of a table like structure where data is stored in rows and columns. Each row consists of data associated to each other, separated in dissimilar columns where each column can have its own data type[12,14]. The framework able to master the complexity of the knowledge discovery process over this kind of data needs to maintain at least the following functionalities.

1. *Trajectory Reconstruction*: Trajectory data need to be created from raw observations, approximating the original trajectory from the sample points.
2. *Querying spatio-temporal data:* Querying capabilities are required to filter and combine movement of data therefore spatio-temporal primitives must be embedded in the query language.
3. *Model extraction*: Trajectory models representing combined behavior have to be extracted using mining algorithms specialized on trajectory data.
4. *Model storing and Progressive Mining*: Discovered models have to be explicitly represented and stored therefore, allowing both the re-using of the derive models with different data and the progressive mining.
5. *Data and Model Manipulation*: Data and models may be manipulated and combined.
6. *Privacy control*: A set of privacy preserving methods should be used on data to guarantee the anonymity of the data.
7. *Pattern Understanding*: Reasoning methods on data and models allows to automatically inferring new knowledge according to an application domain representation.
8. *Extensibility* : The framework must provide an easy way to integrate new kind of data, models and algorithms.

© 2017, IJARCSSE All Rights Reserved

D. Tools for Validation and Testing of Mining Models

Analysis Services supports several approaches to validation of data mining solutions, supporting all phases of the data mining test method.

- Categorize data into testing and training sets.
- Filtering models to train and test different combinations of the similar source data.
- Measuring *lift* and *gain.* A *lift chart* is a scheme of visualizing the enhancement that you get from using a data mining model, when you evaluate it to random guessing.
- Performing *cross-validation* of data sets
- Generating *classification matrices*. These charts sort good and bad guesses into a table so that you can rapidly and simply measure how exactly the model predicts the objective value.
- Creating *scatter plots* to evaluate the fit of a regression formula.
- Creating *profit charts* that associate economic gain or costs with the use of a mining model, so that you can evaluate the value of the recommendations.

III. GENETIC ALGORITHM

A genetic algorithm (GA) is a search heuristic that mimics the process of expected development. This heuristic is regularly used to produce positive solutions to optimization and search problems. Genetic algorithms belong to the bigger class of evolutionary algorithms (EA), which produce solutions to optimization problems using techniques motivated by expected evolution, such as inheritance, mutation, selection, and crossover. Genetic algorithms discover application in bioinformatics, phylogenetics, computational science, engineering, economics, chemistry, manufacturing, mathematics, physics and other fields.

A. Use of Genetic Algorithm in Data Mining

In this paper, we discuss the applicability of a genetic based algorithm to the search method in data mining. Data mining algorithms need a technique that partitions the domain values of an attribute in a restricted set of ranges, only because considering every possible ranges of domain values is infeasible. Genetic Algorithm (GA) is a self-adaptive optimization searching algorithm. Genetic Algorithm obtains the most excellent solution, or the most reasonable explanation through generations of chromosomes constant evolution includes reproduction, crossover and mutation etc[12]. Here is the general description of this problem:

$$\mathrm{F}(x) = a \times S(x) + b \times C(x) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Here a and b are constants, a *≥0, b≥0, S(x)* is the support and *C(x)* is the confidence.

B. Analysis-Based Test Data Generation Approaches

An approach to test data creation via symbolic execution date back to 1976 typically these approaches generate a thorough set of test cases by deducing which combinations of inputs will cause the software to follow given paths. TESTGEN, for example, transforms every situation in the program to one of the form e < 0 or e ≥0, and then searches for values that minimize (resp. maximize) e, thus causing the condition to become true (resp. false). Several approaches bound the range of dissimilar conditions they consider; for instance, TESTGEN's minimization approach cannot be applied to conditions relating pointers. In addition, most analysis based approaches gain heavy memory and processing time costs[3,7]. These restrictions are the main reason why researchers have explored the use of heuristic and metaheuristic approaches to test case creation.

C. Genetic Algorithms for Testing

Genetic algorithms (GAs) were first described by Holland. Candidate solutions are represented as "chromosomes", with solution represented as "genes" in the chromosomes. The probable chromosomes form a search space and are connected with a fitness task representing the value of solutions determined in the chromosome. Search profits by evaluating the fitness of each of a population of chromosomes and then performing point mutations and recombination on the booming chromosomes. GAs can beat simply random search in ruling solutions to difficult problems. Genetic Algorithms is an valuable tool to use in data mining and pattern recognition. There are two diverse methods to applying Genetic Algorithm in pattern recognition:

a) Use Genetic Algorithm as a classifier directly in computation.
b) Use a GA to optimize the results i.e. as an optimizer to display the parameters in new classifiers.

Most applications of GAs in pattern recognition optimize several parameters in the classification development. GAs has been useful to find an best possible set of feature weights that develop classification accurateness. First, a conventional feature extraction technique such as Principal Component Analysis (PCA) is functional, and then a classifier such as k-NN (Nearest Neighbour Algorithm) is used to estimate the fitness function for GA. Mixture of classifiers is a new area that GAs have been used to optimize. GA is also used in selecting the prototypes in the case-based classification[4,8]. According to the second method of genetic algorithm to optimize the outcome from the dataset is more efficient to compute the exact values of observations of data by applying data mining techniques.

© 2017, IJARCSSE All Rights Reserved

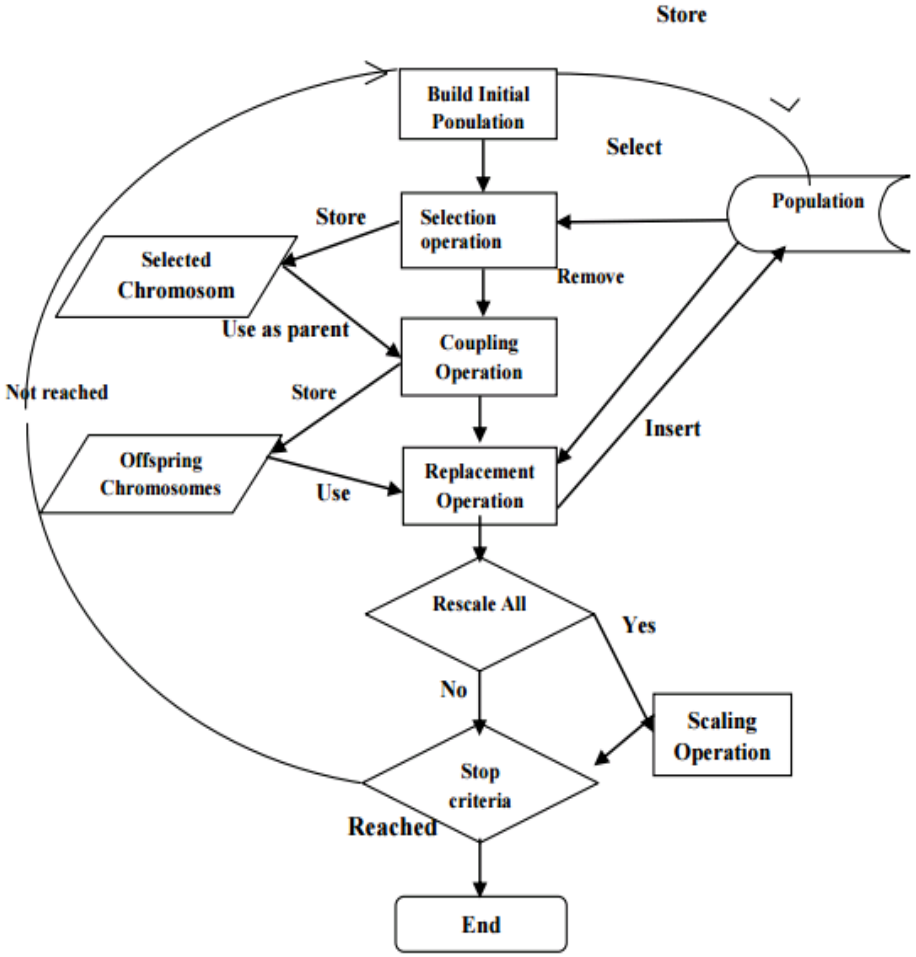

Fig.2:Genetic Algorithm Flowchart

D. Association rules mining based on a novel Genetic Algorithm

Encoding

This paper employs natural numbers to encode the variable A_{ij}. That is, the number of the lines of each range in the matrix A in which the element 1 exists is regarded as a gene. The genes are free of each other. They are marked by A_1, $A_2 \ldots A_j, \ldots, A_n$, in which $A_j \in [1,m]$, $j \in [1, n]$ n may be a repetitively identical natural number. The best way is the generated M individuals randomly that the length is N. Then the chromosome bunch encoded by the natural number is calculated as the initial population.

The Fitness

Formula (1) is properly transformed into:

$$F(xy) = W_S \times \frac{S(xy)}{S_{min}} + W_c \times \frac{C(xy)}{C_{min}} \quad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Here, $W_C + W_s = 1$, $W_c \geq 0$, $W_s \geq 0$, S_{min}, is minimum support, and C_{min} is minimum confidence.

Web Testing

Web testing is completely focused on web based applications. This testing is to help reduce the efforts required to test the web applications, minimize the cost, increase software quality and used to reuse the test cases. There are different web testing are available like functional testing, compatibility testing, load testing, regression testing and performance testing.

1. *Functional Testing:* It is s software testing process, which is used to test the functionality of the application. It will check the validations on all fields; verify page redirection, business logic & calculation.
2. *Compatibility Testing:* Web based applications are tested on different browsers. It makes sure that the application will be reliable on all browsers. Applications are well-suited with diverse devices like mobile, notebook etc.
3. *Performance Testing:* The performances of web based applications are tested. It is the process of determining the speed of computer, software program and scalability& reliability. Load and stress tests are one of the performance test types.
4. *Load Testing:* Load testing is the testing with the target of determining how well the product handles competition for system resources. It will be in the form of network traffic, CPU utilization or memory allocation. For example; multiple applications are running on a compute concurrently.
5. *Stress Testing:* This test is conducting to calculate the behaviour when the system is pushed away from the breaking point. It is to determine, if the system manages recover gracefully.

© 2017, IJARCSSE All Rights Reserved

E. Web Automation Testing

Manual testing is complicated to test the high competitive websites and web applications. It will be avoid for using web automation testing. It provides the ability to reuse, tests multiple browsers, platforms & programming languages.

Features:

- It saves time
- Minimize the cost
- Improves accuracy
- Less effort and get more results

Selenium Suite

Selenium was created by Jason Huggins working in Thought Works in 2004. He was working on a web application that required regular testing. He realized that manual testing replication was becoming more and more inefficient; he created a JavaScript program that would routinely manage the browser's action. He named this program JavaScriptTestRunner. Afterward he concluded this JavaScriptRunner open source which was later re-named as Selenium Core. Selenium is an open source browser automation tool, commonly used for testing the web applications. It automates the control of a web browser so that repeated tasks can be automated[11,13]. Selenium is a set of testing tools, working with multiple browsers, operating systems and writing tests in different languages like C#, java, Ruby and Python.

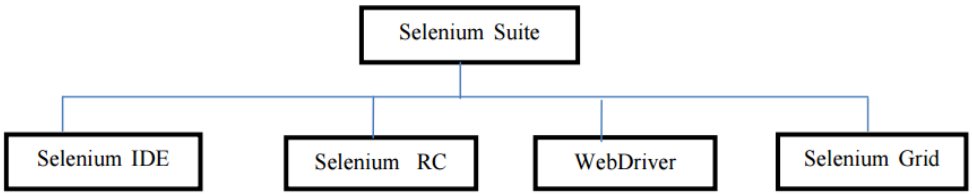

Fig.3 : Selenium Suite

Selenium is a suite of four components. First is Selenium IDE, which is an extension for Firefox that allows users to record and playback tests. Second element is Selenium RC which is a server written in java. It accepts commands for the browser via HTTP. Third element is Selenium Webdriver which provides APIs in variety of languages to permit more control and the application of typical software development practices. Lastly Selenium Grid, it is potential to use the Selenium APIs to control browser instances scattered over a grid of machines. It allowing more tests to run in similar.

F. Selenium IDE

Selenium IDE (Integrated Development Environment) is a tool to develop Selenium test cases. Selenium IDE was initially created by Shinya Kasatani and donated to Selenium project in 2006. It is implemented as a Firefox Plug-in that allows recording, editing and debugging the selenium test cases. Selenium name comes from Selenium Recorder. On start-up of the Firefox, the recording option is automatically turned on. This choice allows user to record any action done within the web page. In Selenium IDE scripts are recorded in Selenese, an extraordinary test scripting language which is a set of Selenium commands[. It is used to test web application. Actions, Accessors, Assertions are the classification of selenium.

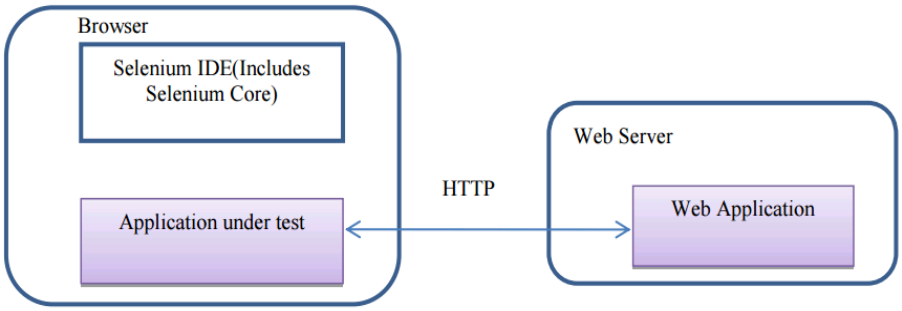

Fig.4: Architecture of IDE

Features

- It is trouble-free and easy record and playback.
- Selenium IDE supports logical field selection options like ID's, XPath and Names.
- It saves test scripts in some formats like Selenese, Ruby etc.
- IDE allow to customization through plug-ins.
- Selenium IDE having an option for adding different asserts options in scripts.
- It allows setting breakpoints and debugging the scripts.
- It also supports auto complete commands.

© 2017, IJARCSSE All Rights Reserved

IV. IMPLEMENTATION AND TEST CASE GENERATION OF GENETIC ALGORITHM

Step 1: Start with randomly generated test cases from the population.
Step 2: Calculate the fitness f(x) of each pair of test cases (chromosome x) in the population.
Step 3: Repeat the following steps until a n child test cases have been generated.
Step 3(a): Select a pair of parent test cases from the current population where the probability of selections an increasing function of fitness. Selection is done "with replacement," meaning that the same pair of test case can be selected more than once to become a parent. i.e. (Selection process is carried out)
Step 3(b): With the crossover probability Pc, cross over the pair at a randomly chosen point to form two child cases or off springs.
Step 3(c): If no crossover takes place, form two test cases that are exact copies of their respective parent cases.
Step (d): Mutate the two child cases with mutation probability Pm, and place the resulting pair of test cases in the new population. If n is odd, one new population member can be discarded at random.
Step 4: Replace the current test cases with the new test cases.

A. Genetic Algorithm Implementation in C++

Pseudo-code for genetic algorithm:

```
choose initial_population:
evaluate individual_fitness function
determine population's_average
        fitness_function
Repeat
        select best_case individuals to
        reproduce;
        mate_pairs at random;
        apply crossover_operator;
        apply mutation_operator;
        evaluate Individual fitness;
        determine population's average
      fitness;
```

The second step consists for generating data consists of the outer loop, which will generate the possible test cases remaining. To account for the possibility of unfeasible test requirements, which includes branches and statement values, the loop will produces iterations until it satisfies the test results for the given population values. The algorithm will produce values which will be applied for the crossover and mutation operator. Then the fitness function for individual values are generated and the population's individual average fitness functions in calculated. In the final step, the algorithm will assign the combined values of the test cases and find at least one individual desired fitness function values until enough test generations have been passed.

Population size = 50
Number of generations = 250
Crossover rate=0.7
Mutation rate = 0.001

Table 1: Implementation of Iterations

Iteration	Best fitness	Average fitness	Standard Deviation
1	17.79	12.51	5.47
2	21.30	14.91	3.69
3	21.30	14.72	3.25
4	22.99	15.49	3.93

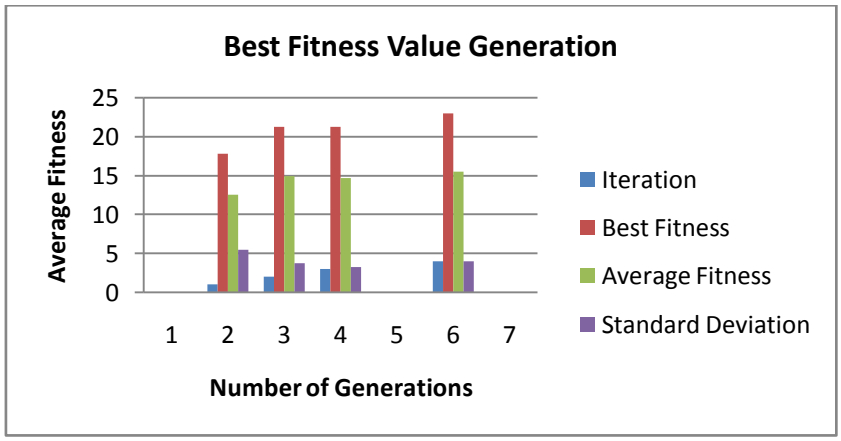

Fig.5 : Best Fitness Generation

© 2017, IJARCSSE All Rights Reserved

V. CONCLUSION

In this paper, we analysed how evolutionary techniques such as Genetic Algorithm helped in software testing. The key advantage of using automated tools is to avoid manual effort. Selenium is a web based automation framework which uses individual platforms and programming languages. These features are record and playback and run in parallel tests. The results show how software testing using Genetic Algorithms becomes proficient even with rising number of test cases. It can reduce the time and provide free software and easier for developers and programmers. Genetic Algorithms are being used and give us a resource of an automatic test case generator. The evolutionary generation of test cases can be applied and proves to be efficient and cost effective than Random Testing. Future enhancements are selenium is to test window based application. So now-a-days selenium is the best available tool for web applications.

REFERENCES

[1] Bo Zhang, Chen Wang, "Automatic generation of test data for path testing by adaptive genetic simulated annealing algorithm", IEEE, pp. 38 – 42, 2011.

[2] Chandraprabha, Ajeet Kumar, Sajal Saxena, " SYSTEMATIC STUDY OF A WEB TESTING TOOL:SELENIUM" International Journal Of Advance Research In Science And Engineering ,IJARSE, Vol. No.2, Issue No.11, November 2013

[3] D.J Berndt, A. Watkins, "High volume software testing using genetic algorithms", Proceedings of the 38t h International Conference on system sciences (9), IEEE, pp. 1- 9, 2005.

[4] Fei Wang and Wencaai Du, "A Test Automaton Framework Based on WEB" proc. IEEE 11th International Conference on Computer and Information (ACIS 12),IEEE Press, pp. 683-687, doi:10.1109/ICIS.2012.21, 2012.

[5] M. Srinivas, L.M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms", IEEE Transactions on Systems,Man and Cybernetics 24 (4), 17–26,1994.

[6] M. Last and O. Maimon, "A compact and accurate model for classification", IEEE Trans. on Knowledge and Data Engineering 16(2), 203-215, 2004.

[7] Nidhika Uppal, Vinay Chopra, "Design and Implementation in Selenium IDE with Web Driver" International Journal of Computer Applications (0975 – 8887) Volume 46– No.12, May 2012.

[8] Nashat Mansour, Miran Salame, "Data Generation for Path Testing", Software Quality Journal, 12, 121–136, Kluwer Academic Publishers, 2004.

[9] Praveen Ranjan Srivastava et al, "Generation of test data using Meta heuristic approach" IEEE TENCON India available in IEEEXPLORE, 19-21 NOV 2008.

[10] Oded Maimon, Lior Rokach, "Data Mining and Knowledge Discovery Handbook", Springer 2006.

[11] R. Kuhn, R. Kacker, Y. Lei, and J. Hunter. Combinatorial software testing. *IEEE Computer*, 42(8):94-96, 2009.

[12] Tapia, J.J., Vallejo, E.E.: "A Clustering Genetic Algorithm for Inferring Protein- Protein Functional Interactions from Phylogenetic Profiles". In: 2008 IEEE World Congress on Computational Intelligence, 2008.

[13] T. Mens and T. Tourwe, "A survey of software refactoring", Software Engineering, IEEE Transactions on, vol.30, no. 2, pp. 126-139, Feb 2004

[14] Wang D W, Yung K L, Ip W H. "A Heuristic Genetic Algorithm for Subcontractor Selection in a Global Manufacturing Environment [J]". IEEE Trans. On SMC Part–C,31(2):189–198, 2001.

[15] Y.C. Kulkarni, Y.C. Kulkarni, "Automating the web applications using the selenium RC", ASM's International Journal of Ongoing Research in Management and IT e-ISSN-2320-0065, 2011.

© 2017, IJARCSSE All Rights Reserved