

# Smart Non Redundant Data Extraction for Efficient Testing

Anurag Sahu

**Abstract:** This paper presents a framework for improving effectiveness of automated trying out in the absence of specs. The framework supports a fixed of associated techniques. First, it consists of a redundant-check detector for detecting redundant checks among mechanically generated take a look at inputs. These redundant tests boom testing time with out growing the potential to detect faults or growing our self belief inside the software. Second, the framework consists of a non-redundant-check generator that employs country-exploration techniques to generate non-redundant tests within the first location and makes use of symbolic execution techniques to in addition improve the effectiveness of take a look at generation. Third, because it is infeasible for builders to inspect the execution of a massive range of generated check inputs, the framework consists of a test selector that selects a small subset of take a look at inputs for inspection; those selected take a look at inputs exercise new program behavior that has no longer been exercised with the aid of manually created exams. Fourth, the framework consists of a test abstractor that produces succinct state transition diagrams for inspection; these diagrams summary and summarize the behaviour exercised via the generated check inputs. Finally, the framework includes a software-spectra comparator that compares the internal software behaviour exercised by means of regression assessments executed on software versions, exposing behavioural differences past specific software outputs. The framework has been carried out and empirical consequences have shown that the evolved techniques inside the framework improve the effectiveness of computerized testing by using detecting high percentage of redundant exams among test inputs generated with the aid of current gear, generating non-redundant check inputs to obtain high structural coverage, lowering inspection efforts for detecting problems inside the software, and exposing extra behavioural differences for the duration of regression trying out.

**Keywords:** Software Testing, ANFIS

## 1. Introduction

Software checking out is an crucial part of the software development lifecycle. Software checking out is presently maximum extensively used approach for detecting software screw ups. Due to time fee and other situations exhaustive testing is not viable. Software testing includes dynamic verification of the behavior of a application on a finite set of check cases definitely decided on for the typically endless execution area towards the specified predicted conduct. When checking out software developers create check inputs, run take a look at inputs & check take a look at execution for correctness. This work affords a framework for improving effectiveness of automated trying out within the absence of specs. Software trying out includes 4 important steps in testing software: generating take a look at inputs, generating anticipated output for test inputs, run check inputs and confirm actual output. To reduce the hard human effort in test sports developers automate those steps by way of using checking out tools.

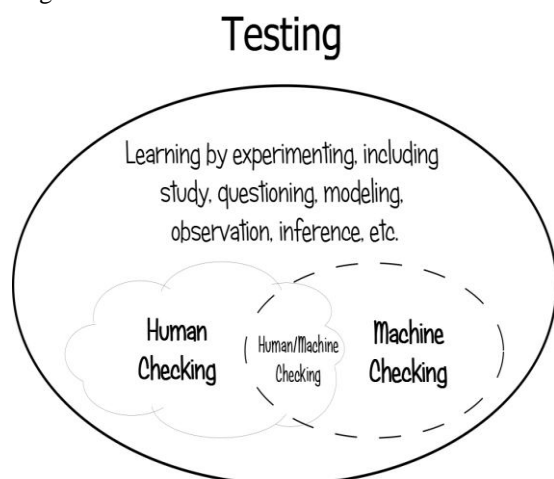


Figure 1: Conventional Testing Framework

Software allows in many components of life & therefore enhancing software reliability is become vital to society. A latest document via National Institute of Standards & Technology located that software errors value US economic system approximately \$60 Billion each yr. Software trying out continues to be the most broadly used method for enhancing software program reliability. However software program trying out is exertions extensive, generally costing for half of the software program improvement effort. To lessen the human attempt in trying out testers can automate software checking out via using computerized gear. This work recognition on growing a framework for improving effectiveness of computerized exams

Software merchandise must be reliable, correct, and scalable. To make sure those traits, it's miles necessary to test the software program at numerous conditions and as a result software testing is an critical issue of the software development manner. Software trying out is presently the most widely used approach for detecting software failure Software checking out consists of the dynamic verification of the conduct of a application on a finite set of take a look at instances definitely selected from the normally countless execution domain towards the required anticipated behavior. When trying out a software developers create test inputs, run check inputs & test execution for corrections.

## 2. Related Work

The verification and validation of engineering designs are of primary significance as [9] P.G. Maropoulos 2010 directly affect manufacturing overall performance and ultimately define product functionality and client belief. Research in factors of verification and validation is widely unfold ranging from gear hired in the course of the virtual design segment, to methods deployed for prototype verification and validation. This paper evaluations the same old definitions of verification and validation in the context of engineering

design and progresses to provide a coherent evaluation and class of these activities from initial layout, to design within the virtual domain and the physical verification and validation of merchandise and techniques. The scope of the paper includes factors of system design and demonstrates how complex products are proven within the context in their lifecycle. Industrial necessities are highlighted and studies trends and priorities recognized.

Nuclear energy industries have growing hobby in using fault detection and diagnosis (FDD) strategies to improve safety, reliability, and availability of nuclear strength vegetation (NPP). A brief overview of FDD techniques by [10] **Jianping Ma, 2011**. FDD strategies are classified into model-primarily based strategies, data-pushed techniques, and sign-based strategies. While sensible applications of model-based totally techniques are very limited, numerous information-pushed strategies and signal-primarily based strategies were implemented for tracking key subsystems in NPPs. In this paper, six regions of such programs are considered. They are: instrument calibration tracking, instrumentation channel dynamic overall performance monitoring, system monitoring, reactor core tracking, free component tracking, and transient identification. The concepts of using FDD strategies in these packages are explained and recent studies of advanced FDD strategies are tested. Popularity of FDD programs in NPPs will constantly growth as FDD theories increase and the protection and reliability requirement for NPP tightens .

Cloud computing affords on-demand get entry to computational resources which together with pay-per-use business models, allow utility carriers seamlessly scaling their offerings. Cloud computing infrastructures allow developing a variable quantity of virtual system instances depending on the software demands. An appealing functionality for Software-as-a-Service (SaaS) companies is having the potential to scale up or down application assets to best devour and pay for the resources which might be simply required in some unspecified time in the future in time; if finished correctly, it will be much less luxurious than jogging on normal hardware by using traditional web hosting. However, even if huge-scale packages are deployed over pay per-use cloud high-overall performance infrastructures, cost-effective scalability isn't always completed because idle techniques and resources (CPU, reminiscence) are unused but charged to software vendors. Over and under provisioning of cloud assets are still unsolved troubles. Even if height hundreds can be successfully predicted, without an effective elasticity model, pricey sources are wasted in the course of nonpeak times (underutilization) or revenues from capacity clients are misplaced after experiencing poor service (saturation). [11] **Javier Espadas, 2013** attempted to establish formal measurements for under and over provisioning of virtualized resources in cloud infrastructures, mainly for SaaS platform deployments and proposes a aid allocation model to set up SaaS packages over cloud computing systems by taking into account their multitenancy, hence creating a fee-effective scalable surroundings.

As using electronics and software inside the contemporary off road vehicles tend to boom, there may be a massive

challenge of getting the electronics and software program tested earlier than making it available for patron use. There are a variety of interactions among the hardware and software for the vehicle to characteristic. And with the quantity of interactions in the hardware and software come plenty of troubles in verifying the capability of the vehicle. If right verification is unnoticed, the customer may grow to be getting an risky automobile that can lead to extreme results. To make these motors safe and greater green, foremost producers are using vehicle simulators where the check engineers can carry out several computerized and manual exams. Once these assessments are achieved, the reports are amassed for in addition evaluation of the faults took place at numerous stages of checking out. The analysis is similarly used to discover the foundation motive of the failure to save you it from reoccurring before the brand new product is surpassed for manufacturing. [12] **Darshak Dodiya, 2014** investigated the quantity of time ate up on results reporting processes involved within the average evaluation of test outcomes in a heavy off road automobile manufacturing organisation in Midwest. The test and recommended enhancements on this study will help make the employer's manner of consequences reporting more efficient and powerful. In the quit of this take a look at end and some pointers for in addition research are mentioned to effectively utilize assets in software program verification and validation system of this Midwestern Company. This observe has delivered a simple however very effective prototype device called ACRS (Automated Central Reporting System) which if implemented properly can reduce the amount of time and money spent on generally omitted assignment of searching effects from a big pool of automatic test results. The research highlighted the left out region of consequences reporting manner in the company which if appeared carefully continues to be not efficient, thereby making the entire technique of PV&V within the company less efficient. The study proved that there is lots of room for improvement inside the current outcomes reporting system if the organization appears to implement automated equipment for reporting consequences from automatic checks

Capture, coding and verbal exchange of newborn screening (NBS) facts constitute a task for public health laboratories, fitness departments, hospitals, and ambulatory care practices. An increasing number of situations focused for screening and the complexity of interpretation make contributions to a developing need for included statistics-control techniques. [13] **Stephen M Downs, 2018** maked NBS an vital test of equipment and architecture for digital health records change (HIE) in this convergence of character affected person care and population fitness activities. For this cause, the American Health Information Community undertook 3 responsibilities described in this paper. First, a newborn screening use case became mounted to facilitate standards harmonization for commonplace terminology and interoperability specs guiding HIE. Second, newborn screening coding and terminology were developed for integration into electronic HIE sports. Finally, explanation of privateness, security, and clinical laboratory regulatory necessities governing records exchange become supplied, serving as a framework to establish pathways for enhancing

screening software timeliness, effectiveness, and performance of satisfactory patient care offerings.

### 3. Methodology

ANFIS has been proven to be powerful in modeling several processes which includes time collection, actual-time reservoir operations and river flow forecasting. ANFIS possesses residences consisting of functionality of studying, building, expensing and classifying. It has the advantage of permitting the extraction of fuzzy guidelines from numerical records or professional knowledge and adaptively constructs a rule base. Moreover, it could adapt the complicated conversion of human intelligence to fuzzy structures. The major trouble of the ANFIS predicting version is the time required for training structure and figuring out parameters. ANFIS uses the mastering capability of the ANN to outline the input-output dating and assemble the fuzzy regulations by way of figuring out the input structure. The device effects have been acquired via thinking and reasoning capability of the fuzzy common sense. The ANFIS architecture includes 5 layers (Figure three.1). Here the circles denote a hard and fast node whereas squares denote an adaptive node. For simplicity it is assumed that the examined FIS has two inputs and one output. For a primary order Sugeno fuzzy version, a regular rule set with two fuzzy if-then regulations can be expressed as

- Rule 1: IF  $x$  is  $A_1$  and  $y$  is  $B_1$  THEN  $f_1 = p_1x + q_1y + r_1$
- Rule 2: IF  $x$  is  $A_2$  and  $y$  is  $B_2$  THEN  $f_2 = p_2x + q_2y + r_2$

where,  $x$  and  $y$  are the crisp inputs to the node  $i$ ,  $A_i$  and  $B_i$  are the linguistic labels (low, medium, high, etc.) characterized by convenient membership functions and  $p_i$ ,  $q_i$  and  $r_i$  are the consequence parameters ( $i = 1$  or  $2$ )

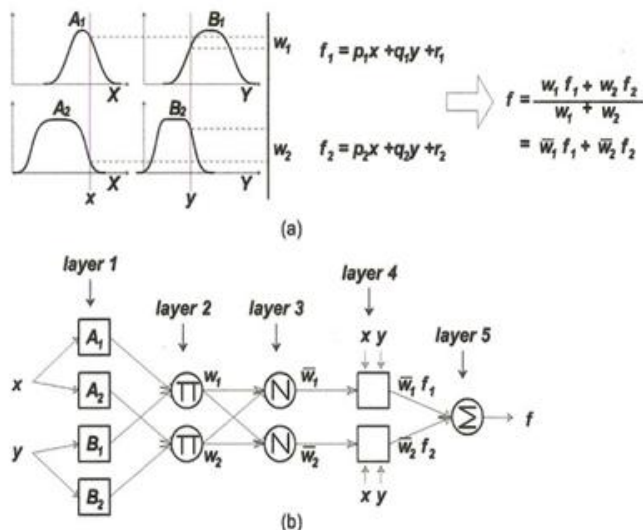


Figure 2: (a) A two input first-order Sugeno fuzzy model with two rules; (b) Equivalent ANFIS Architecture

The model is briefly presented step by step in the following way;

**Input nodes (Layer 1):** Each node in this layer generates membership grades of the crisp inputs which belong to each

of convenient fuzzy sets by using the membership functions. Each node's output  $O_i^1$  is calculated by:

$$O_i^1 = \mu_{A_i}(x) \text{ for } i=1,2; O_i^1 = \mu_{B_i}(y) \text{ for } i=3,4 \quad (1)$$

Where  $\mu_{A_i}$  and  $\mu_{B_i}$  are the appropriate membership functions for  $A_i$  and  $B_i$  fuzzy sets, respectively. Many various membership functions such as trapezoidal, triangular, Gaussian function, etc. can be applied to determine the membership grades. The gauss membership function is used, as;

$$O_i^1 = \mu_{A_i}(x) = e^{-\frac{(x-c)^2}{2\sigma^2}} \quad (2)$$

Where,  $\{a_i, b_i, c_i\}$  is the membership functions' parameter set that changes the shape of membership function from 1 to 0. These parameters are referred to as the premise parameters.

**Rule nodes (Layer 2):** In this layer, the AND/OR operator is applied to get one output that represents the results of the antecedent for a fuzzy rule, that is, firing strength. It means the degrees by which the antecedent part of the rule is satisfied and it indicates the shape of the output function for that rule. The outputs of the second layer, called as firing strengths  $O_i^2$  are the products of the corresponding degrees obtaining from layer 1, named as  $w$  given below.

$$O_i^2(x) = w_i = \mu_{A_i}(x) \mu_{B_i}(y), i=1, 2 \quad (3)$$

**Average nodes (Layer 3):** Main target is to compute the ratio of firing strength of each  $i$ th rule to the sum of all rules' firing strength. Thus the firing strength in this layer is normalized as;

$$O_i^3 = \bar{w}_i = \frac{w_i}{\sum_{i=1,2} w_i} \quad (4)$$

**Consequent nodes (Layer 4):** The contribution of  $i$ th rules towards the total output or the model output and/or the function defined is calculated

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i), i=1, 2 \quad (5)$$

Where,  $w_i$  is the  $i$ th node's output from the previous layer (i.e., demonstrated in the third layer).  $\{p_i, q_i, r_i\}$  is the parameter set in the consequence function and also the coefficients of linear combination in Sugeno inference system.

**Output nodes (Layer 5):** This layer is called as the output nodes in which the single node computes the overall output by summing all the incoming signals and is the last step of the ANFIS. Hence, each rule's fuzzy results are transformed into a crisp output in this layer by de-fuzzification process, as;

$$f(x,y) = \frac{w_1(x,y)f_1(x,y) + w_2(x,y)f_2(x,y)}{w_1(x,y) + w_2(x,y)} = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2} \quad (6)$$

$$O_i^5 = f(x,y) = \sum_i w_i f_i = \bar{w}_i f_1 + \bar{w}_i f_2 = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (7)$$

There are two major phases for implementing the ANFIS for specific applications: the structure identification phase and the parameter identification phase. The structure identification phase involves finding a suitable number of fuzzy rules and fuzzy sets and a proper partition feature



space. The parameter identification phase involves the adjustment of the premise and consequence parameters of the system. More detailed descriptions of the two phases are provided in the following two sections (b) and (c).

**(b) Parameter Identification using hybrid learning algorithm**

During the learning process, the premise parameters in the layer 1,  $\{c, \sigma\}$ , and the consequent parameters in the layer 4,  $\{p, q, r\}$ , are tuned until the desired response of the FIS is achieved. The two frequently used training methods are the back-propagation (BP) algorithm and the hybrid learning algorithm. The hybrid learning algorithm, which combines the least squares method (LSM) and the BP algorithm, is used to rapidly train and adapt the FIS. This algorithm converges much faster since it reduces the dimension of the search space of the original BP algorithm. When the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters. The output  $f$  can then be written as:

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 = \bar{w}_1 f_1 + \bar{w}_2 f_2 \quad (8)$$

$$= (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + (\bar{w}_2) r_2$$

Which is linear in the consequent parameters  $p_1, q_1, r_1, p_2, q_2$  and  $r_2$ . The hybrid getting to know algorithms of ANFIS consist of the following two components: (a) the learning of the basis parameters by means of back-propagation and (b) the studying of the outcome parameters by means of least-squares estimation. In the forward pass of the hybrid studying set of rules, functional alerts go forward until layer four to calculate every node output. The nonlinear or premise parameters inside the layer 2 stay constant on this pass. The consequent parameters are diagnosed by way of the least squares estimate. In the backward pass, the mistake rates propagate backward from the output cease towards the enter cease, and the basis parameters are updated by means of the gradient descent. Jang provided the exact description and the mathematical heritage of the hybrid getting to know set of rules.

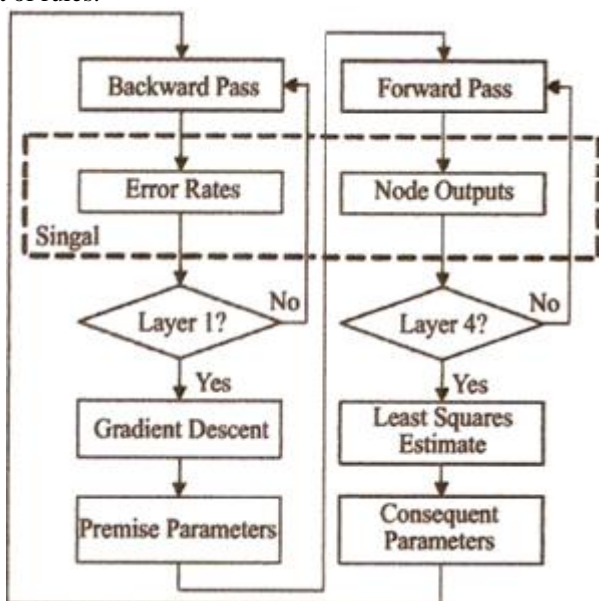


Figure 3: Data Flow diagram for Hybrid learning algorithm

**(c) Structure Identification**

Clustering is a system wherein facts are positioned into groups or clusters, such that facts in a given cluster tend to be much like every other, and records in specific clusters have a tendency to be multiple. When the clustering estimation is carried out to a set of input output statistics, each cluster centre may be considered as a fuzzy rule that describes the characteristic behavior of the machine. Each cluster centre corresponds to fuzzy rule, and the cluster diagnosed represents the antecedent of this rule. This step paperwork the structure identity.

Clustering algorithms are used drastically not most effective to prepare and categorize data, but are also beneficial for data compression and model production. By finding similarities in statistics, one can represent comparable information with fewer symbols. The density function for a information factor is defined because the measure of capability for that statistics factor. It is predicted based on the distance of this statistics point from all different records factors. Therefore, a records factor mendacity in a heap of different records factors will have a excessive chance of being a cluster centre, even as a information point that is located in an area of diffused and no longer focused statistics points can have a low hazard of being a cluster centre.

Subtractive clustering is a method for mechanically generating fuzzy inference systems by detecting clusters in enter-output training information. Subtractive clustering considers every facts factor as a capacity cluster centre. The degree of ability for a statistics point is predicted primarily based on the gap of this data factor from all other facts factors. Therefore, a facts factor lying in a heap of other information factors can have a high danger of being a cluster centre, whilst a information point that is placed in an area of subtle and not focused records factors could have a low threat of being a cluster middle.

After measuring the capacity of each information point, the statistics point with the finest capability price is elected as the first cluster centre. To discover the subsequent cluster centre, potentials of records factors should be revised. For every records point, an quantity proportional to its distance to the primary cluster centre might be subtracted. This reduces the threat of a information point centre. After revising the capacity of all facts points, the data point with the maximum capacity may be decided on as the next cluster centre. The capability of data factors inside the first step is measured as given by Dirk [4]. Given a group of  $n$  information points  $\{x_1, \dots, x_n\}$ , the subtractive clustering algorithm considers every statistics factor as a capacity cluster center.

A density measure at a data point  $x_i$  is defined as

$$D_i = \sum_{j=1}^n e^{-|x_i - x_j|^2 / (r_a / 2)^2} \quad (9)$$

where the cluster radius  $r_a$  is a positive constant. Thus, a data point that has many neighbouring data points will have a high potential of being a cluster center. The radius  $r_a$  defines a neighbourhood. Data points outside this radius have little effect on the density measure. The choice of  $r_a$  plays an important role in determining the number of clusters. Large values of  $r_a$  will generate a limited number of clusters, while

small values of  $r_a$  will generate a large number of clusters. After the density measure of each data point has been calculated, the first cluster center is chosen to be the data

point with the highest density measure. Suppose  $x_{c_1}$  is the point selected and  $D_{c_1}$  is its density measure, then the density measure for each data point  $x_i$  is revised by the formula

$$D_i = D_i - D_{c_1} e^{-|x_i - x_{c_1}| / (r_b / 2)^2} \quad (10)$$

Where  $r_b$  is a positive constant. Note that the data points near the first cluster center  $x_{c_1}$  will have significantly reduced density measures, so that they are unlikely to be selected as the next cluster center. The constant  $r_b$  is usually greater than  $r_a$  to prevent closely spaced cluster centers. Generally  $r_b$  is specified as 1.5 times of  $r_a$ . After the density measure for each data point is revised, the next cluster center  $x_{c_2}$  is selected, and all of the density measures for data points are revised again. Subtractive clustering can be used as a standalone approximate clustering algorithm in order to estimate the number of clusters and their locations.

### 3.1 Algorithm

The code for bifurcation of testing data is performed using the ANFIS based model. The algorithm is described below:

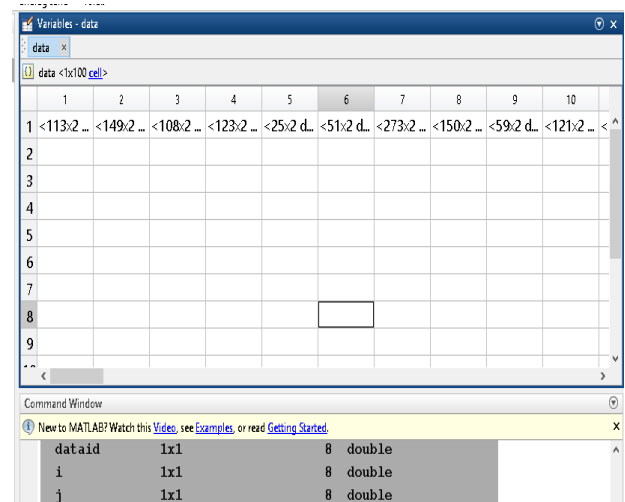
- 1) Select the testing code.
- 2) Justify data based dependency of the codes.
- 3) Generate data set.
- 4) Extract training data randomly (30% of total data)
- 5) Define erroneous and non erroneous data set ids.
- 6) Define number of input variables.
- 7) Extract input and output training data.
- 8) Train ANFIS model using the selected data at 2 and 3 number of membership function.
- 9) Generate ids of non redundant data by the developed fuzzy model.
- 10) Perform test on generated non redundant data ids.
- 11) Analyze the code performance.

## 4. Result and Discussion

We have taken a two codes executing in MATLAB platform. The codes are saved as 'partE.m' and 'partF.m'. Both codes are kept in the same folder. The software testing tool whose performance is to be measured is mlcovr-0.2. Both of the codes are data dependent. They use following data:

Name	Size	Bytes	Class
L	1x100		double
M1	1x1		double
M2	1x1		double
Max	1x1		double
N	1x100		double
data	1x100		cell
dataid	1x1		double
i	1x1		double
j	1x1		double
n	243x2		double

Main data is generated randomly for 100 times and saved as data.mat as a 1x100 cell structure. Each data cell has different size and values as shown in figure 4.



**Figure 4:** Demonstration of data block having several testing datasets

The code partE.m is healthy code and runs for all the data but partF.m is some complex variant of partE.m and does not runs for all the data set.

The minimum error in 2numf is 0.011729 and for 3 numf is 0.043419. Thus the prediction error in 3 numf is less than 2 numf. The data id for which both fuzzy structures predicted to be erroneous are given in table 1:

**Table 1:** The data id for which both fuzzy structures

2 numf		3numf	
10	F	10	F
11	F	13	T
13	T	15	F
15	F	19	T
18	F	20	T
20	T	29	T
21	F	31	F
27	F	32	T
29	T	34	T
35	F	43	T
44	F	44	F
45	T	52	T
46	F	54	T
49	F	60	T
50	T	61	T
51	F	62	T
54	T	65	T
55	F	68	F
60	T	69	T
62	T	76	F
66	F	77	T
67	T	86	F
72	F		
73	F		
76	F		
77	T		
89	F		
94	F		
95	F		
98	F		
<b>Total:30</b>	<b>True (t):10</b>	<b>Total:22</b>	<b>True (T):15</b>

%accuracy:	30%	%accuracy:	68.18%
------------	-----	------------	--------

Thus in table 1 it has been observed that the accuracy of predicting the erroneous data is above than 68%. The result represents that out of 100 data id 22 data ids is erroneous. On checking the code for these data id the final prediction accuracy of 68% is observed. Thus it can be justified that the fuzzy logic rules at 3numf are capable of classifying the data sets. It is only 22 data ids are required to check the performance of given code instead of all the 100 data. Thus total coverage time will be reduced by  $100-22=78\%$  percent.

## 5. Conclusion

The experiments that we have conducted in this work primarily focus on unit testing of individual structurally complex data structures. The redundant-test detection approach is evaluated against existing test generation tools, which generate a large number of tests but a relatively small number of non-redundant tests. The non-redundant-test generation and test abstraction approaches are evaluated against a relatively low bound of exhaustive testing. The test selection approach and testing approach are evaluated on a set of relatively small programs, being limited in fact by the scalability of the underlying test generation tool or dynamic invariant detection tool.

## References

- [1] David Chays, Yuetang Deng, Phyllis G. Frankl, Saikat Dan, Filippos I. Vokolos and Elaine J. Weyuker An AGENDA for testing relational database applications, Published online 20 January 2004 Copyright c 2004 John Wiley & Sons, Ltd.
- [2] James C. Corbett, Matthew B. Dwyer, John Hatcli, Shawn Laubach, Corina, Robby, Hongjun Zheng, Bandera : Extracting Finite-state Models from Java Source Code, To appear in the proceedings of the 2000 International Conference on Software Engineering (ICSE 2000).
- [3] Tao Xie, David Notkin Checking Inside the Black Box: Regression Testing Based on Value Spectra Differences, 20th IEEE International Conference on Software Maintenance.
- [4] Dirk Beyer Adam J. Chlipala Thomas A. Henzinger Ranjit Jhala, Generating Tests from Counterexamples, Proceedings of the 26th International Conference on Software Engineering (ICSE'04) 0270-5257/04 \$20.00 © 2004 IEEE
- [5] Tony Andrews, Shaz Qadeer A Model Checker for Concurrent Software, <https://www.researchgate.net/publication/221402890>
- [6] Tao Xie, Jianjun Zhao, Darko Marinov, David Notkin, Automated Test Generation for AspectJ Programs, 2005
- [7] Katja Karhu, Tiina Repo, Ossi Taipale, Kari Smolander, Empirical Observations on Software Testing Automation, 2009
- [8] W Stone<sup>1\*</sup>, B Hogan<sup>1</sup>, C Flesher<sup>1</sup>, S Gulati<sup>1</sup>, K Richmond<sup>1</sup>, A Murarka<sup>1</sup>, G Kuhlman<sup>1</sup>, M Sridharan<sup>1</sup>, V Siegel<sup>1</sup>, R M Price<sup>1</sup>, P T Doran<sup>2</sup>, and J Priscu, Design and deployment of a four-degrees-of-freedom hovering autonomous underwater vehicle for sub-ice

- exploration and mapping, Proc. IMechE Vol. 224 Part M: J. Engineering for the Maritime Environment
- [9] P.G. Maropoulos, D. Ceglarek Design verification and validation in product lifecycle, 0007-8506/\$ – see front matter 2010 CIRP. doi:10.1016/j.cirp.2010.05.005
  - [10] Jianping Ma a, Jin Jiang, Applications of fault detection and diagnosis methods in nuclear power plants: A review 2010 Elsevier Ltd. All rights reserved. doi:10.1016/j.pnucene.2010.12.001
  - [11] Javier Espadas, Arturo Molina b, Guillermo Jiménez a, Martín Molina, A tenant-based resource allocation model for scaling Software-as-a-Service applications over cloud computing infrastructures, © 2011 Elsevier B.V. All rights reserved.
  - [12] Darshak Dodiya An analysis of use of automated tools for improving the process of software verification and validation in a Midwestern company, Copyright 2014 - Darshak Dodiya Follow this and additional works at: <https://scholarworks.uni.edu/etd>
  - [13] Stephen M Downs, Peter C van Dyck, Piero Rinaldo, Clement McDonald, R Rodrey Howell, Alan Zuckerman, Gregory Downing, Improving newborn screening laboratory test ordering and result reporting using health information exchange J Am Med Inform Assoc 2010;17:13–18. doi:10.1197/jamia.M3295