Available online at: https://ijact.in

Date of Submission	16/08/2020
Date of Acceptance	20/09/2020
Date of Publication	31/10/2020
Page numbers	3856-3859 (4 Pages)

ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

SOFTWARE TESTING BY USING THE BLACK-BOX METHOD AND THE EQUIVALENCE PARTITION TECHNIQUE TO PREDICT THE ACCURACY OF THE NEURAL NETWORK BASE

Zulkifli*[1], Ford Lumban Gaol[2], Harco Leslie Hendric Spits Warnars[3], Benfano Soewito[4]

[1]Computer Science Department, BINUS Graduate Program – Doctor of Computer Science, Bina Nusantara University Jakarta, Indonesia 11480

[2,3,4]Computer Science Department, BINUS Graduate Program – Doctor of Computer Science, Bina Nusantara University Jakarta, Indonesia 11480

*Correspondent Author Email: Zulkiflist31@gmail.com

Abstract: A neural network algorithm is an artificial nervous system or artificial neural network, it is a physical cellular system that can acquire, store and use the knowledge gained from experience for activation using bipolar sigmoid where the output value ranges from -1 to 1. Because there is a yet of a neural network algorithm model to predict the level of accuracy in terms of software testing, the equivalent partitions black-box technique is used. The black-box software testing method is a testing approach where the data comes from defined functional requirements regardless of the final program structure, and the technique used is equivalent partitions. The design of the accurate prediction of this research is by determining the college application as the software to be tested, then testing it using the black-box method with the equivalence partitions technique. This test was chosen because it will find software errors in several categories. From black-box testing, a dataset is obtained to measure the accuracy of real output and predictive output. The last step is to calculate the error, RSME from the real output, and the predicted output. Furthermore, the final results of the research on the neural network algorithm that is applied to determine the prediction of the accuracy level of black-box software testing with the equivalent partitioning technique is the average accuracy above 80%.
.

Keywords—Neural Network, Black-Box, Equivalence Partitions.

I. INTRODUCTION

Software testing is a program execution process that has an objective of finding errors [1]. Software testing should find unintended errors, and the test is declared successful if it is successful in correcting those errors. Black-box software testing defined as a testing process that tries to find errors in several categories, including malfunction, interface error, data structure error, performance error, initialization error, and termination [2]. From the background, the researcher will make "the design of the neural network algorithm based accuracy prediction for software testing black-box method with the equivalence partitions technique."

II. METHOD

The following is a picture of the method in this research, and the higher education application is the software to be tested.

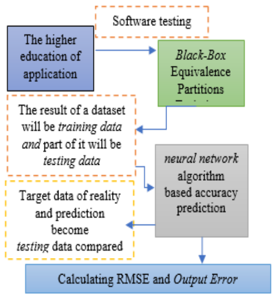

Fig. 1. The method of software testing *black-box* method of *equivalence partitions* technique

The above figure illustrates the method used in this research, with the following stages: the higher education application is software that is being software to be tested, and then the software will be tested using the black-box method of equivalence partitions technique. The results of software testing, this dataset become training data, and part of it will be examining data, then this dataset will be predicted for accuracy with a neural network algorithm by comparing reality target data with predictive data from testing data, then finally counting the output error value and RSME.

A.The Design of Software Testing Prediction Model

The design of the software testing prediction model can be seen in Figure 2, namely:

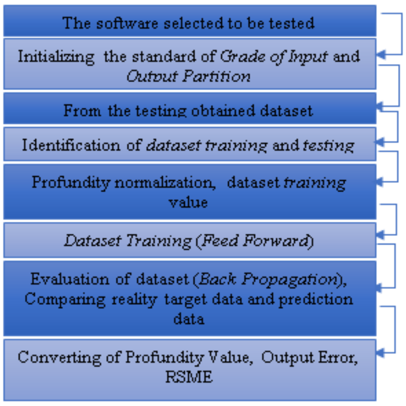

Fig. 2. The design of the software testing prediction model

Fig. 2 describes the design stages of the prediction model for software testing. The first stage begins with selection of software to be tested, namely the higher education application. The black box method is applied by initializing the standard grade of the input and output partitions. The black box method is applied by initializing the standard grade of the input and output partitions. The test results obtained a dataset; this dataset will be identified and considered as training data and test data. The data set will become a neural network algorithm. To change this, what is done is to normalize the training data set and depth values first, and then enter the feed-forward stage to the training data. In next stage, the evaluation of dataset takes place by comparing reality target data and prediction data. In the last step, there is a conversion of profundity value, output error and RSME value.

III. EXPERIMENT RESULT

The following are the stages of applying the *black-box* method of *equivalence partitions* techniques and *neural network* algorithms[12], here are the results of this research, namely:

a. Initialization of the Standard Grade Partition of Input and Output

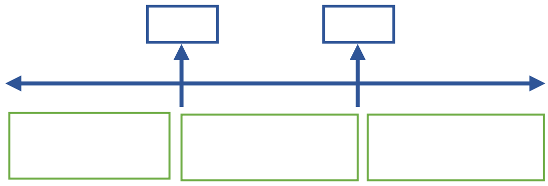

Fig. 3: Equivalence partitions for Exam mark input

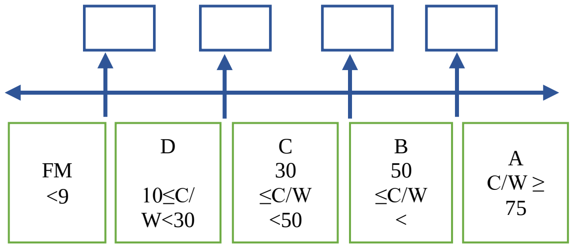

Fig. 4: Equivalence partitions for grade outputs

b. Dataset Testing with the black-box method of equivalence partitions technique

In software testing, the documentation of the black-box method will be carried out. At the grade values that are found to be errors and in each form will be divided into five error models, including errors in function, data structure, interface, initialization, and performance. As for the score grade, the error value found in each form, namely:

Second Form of the Fifth Testing	
Test Case Value Input (Error)	**Score Error**
Input (Exam Mark Performance)	0
Total Error (as Calculated)	10
Partiton Tested (Of Exam Mark)	10 ≤ C/W < 30
Expected Output	D

Based on the second form of the fifth testing, the input value calculation = 10, the output value is obtained (Total Error = 0, and the Partition Grade is D).

Second Form	
Test Case Value Input (Error)	**Score Error**
Function	0
Structure	0
Interface	0
Initialization	0
Performance	10
Total Error (as Calculated)	10
Partiton Tested (Of Exam Mark)	10 ≤ C/W < 30
Expected Output	D

From the second form, the calculation of the input value (0 + 0 + 0 + 0 + 10), the output value is obtained (Total Error = 10, and the Grade Partition is D).

Table 1: A dataset of software testing results by using *black-box* method *equivalence partitions* technique

a	b	c	d	e	F	g	Testing Status
1	1	1	1	1	10	1	Correct
2	1	2	1	2	20	1	Correct
3	1	3	1	3	30	1	Correct
4	1	4	1	4	40	1	Correct
198	40	3	1	0	0	0	InCorrect
199	40	4	1	0	0	0	InCorrect
198	40	3	1	0	0	0	InCorrect
199	40	4	1	0	0	0	InCorrect

a=no, b=form, c= testing, d= tester, e= total error, f= equivalence partitions, g= defect.

Table 1 shows that the *dataset* produced from software testing with the *black-box* method of *equivalence partitions* technique, for the *correct test status* label it means that the tester test results have the same results as the *defect* test results, while the *incorrect* meaning of the *tester* test results is not the same as the *defect* results.

c. Profundity normalization, Dataset Training Values

In normalization of the profundity values on the *dataset training* is carried out, for the process of converting the testing data into range data to 0.1 and 0.9 because the activation function used is the sigmoid function, where the function value never reaches 0 or 1[11], namely:

$$X' = \frac{0.8 * (X_a)}{b * a} + 0.1$$

Table 2: Dataset of Normalization Results

a	b	c	d	e	f	Testing Status
2	1	0,12	0,1	0,1	0,1	InCorrect
2	2	0,12	0,1	0,1	0,1	InCorrect
2	3	0,12	0,1	0,1	0,1	InCorrect
2	4	0,12	0,1	0,1	0,1	InCorrect
2	5	0,12	0,12	0,3	0,12	Correct
25	1	0,1	0,1	0,1	0,1	Correct
25	2	0,1	0,1	0,1	0,1	Correct
25	3	0,1	0,1	0,1	0,1	Correct
25	4	0,1	0,1	0,1	0,1	Correct
25	5	0,1	0,1	0,1	0,1	Correct

a=no, b=form, c= testing, d= tester, e= total error, f= equivalence partitions, g= defect.

Table 3: Dataset of *neural network algorithm* based accuracy prediction of software testing by using *Black-Box* method and Equivalence partitions technique, RMSE value = 0.00142438 in 481 iterations

	Reality		Prediction			
a	***b***	**c**	***d***	***e***	**Error**	
2	0	InCorrect	0,006872	InCorrect	-0,00687	0,993128
2	0	InCorrect	0,014721	InCorrect	-0,01472	0,985279
2	0	InCorrect	0,013219	InCorrect	-0,01322	0,986781
2	0	InCorrect	0,012621	InCorrect	-0,01262	0,987379
2	1	Correct	0,994963	Correct	0,005037	0,994963
2	0	InCorrect	0,006872	InCorrect	-0,00687	0,993128
25	1	Correct	0,99241	Correct	0,00759	0,99241
25	1	Correct	0,99647	Correct	0,00353	0,99647
25	1	Correct	0,996065	Correct	0,003935	0,996065
25	1	Correct	0,995877	Correct	0,004123	0,995877
25	1	Correct	0,995469	Correct	0,004531	0,99547

a=no, b=*defect*, c= target, d= *defect*, e= *output*

Based on the graph of *Dataset* 4 *Hidden Layer* Prediction Results, at *Epoch* = 900, *Learning rate* = 0.1, and the number of testing data = 20 data (20% of the 100 total training data) can be seen in the figure below:

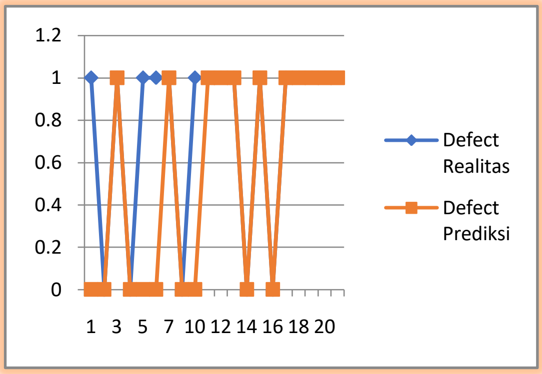

Fig. 5. Graph of Dataset 4 *Hidden* Layer Prediction Results, at *Epoch* = 900, *Learning rate* = 0.1, and the number of testing data = 20 data (20% of the 100 total training data)

d. The result of Accuracy Prediction Level based on the Neural Network Algorithm of Software Testing with the Black-Box Method and Equivalence Partitions Technique

Software testing by using the *black-box* method of the *equivalence partitions* technique based on the *neural network algorithm* has an accuracy of 95% out of 100%, and this is very accurate.

Total Hidden Layer	***Epochs***	***Learning rate***	**Confusion Matrix (Accuracy)**	**AUC**
4	900	0,1	82.00%	0.840 +/- 0.071

4	900	0,1	95.00%	0.954 +/- 0.071
5	900	0,1	81.00%	0.814 +/- 0.071

IV. CONCLUSION

Based on the results of this research, it will be concluded as follows:

Making a prediction model of the level of accuracy by using the neural network algorithm is used for software testing using the black-box method. It is comparable to the partition technique method and can be applied to determine the prediction of the accuracy level of black-box software testing with the equivalence partition technique with very fine accuracy because the prediction shows the average value. The average above 80%, namely: 82% (4 hidden layers, epoch = 900, learning rate = 0.1), 95% (4 hidden layers, epoch = 1000, learning rate = 0.1), and 81% (5 hidden layer, epoch = 1000, learning rate = 0.1), and the most accurate *neural network training* design model is with 4 *hidden layers*, epoch = 1000, learning rate = 0.1) with an accuracy rate of 95%.

Refrences

[1] Albert Endres, Cs. (2003). Hanbook software and System Engineering, Empirical Observations, Laws and Theories.

[2] B. B. Agarwal, C. (2010). Software Engineering & Testing. Boston.

[3] Beizer, B. (1990). Software Testing Techniques, Published by Van Nostrand Reinhold, New York.

[4] Myers, G. (1979). The Art of Software Testing, Wiley; 2nd edition (June 21, 2004).

[5] Berard, C. (1994). Issues in the Testing of Object-Oriented Software.

[6] Fournier, Cs. (2009). Essential Software Testing: A Use-Case Approach.

[7] Mark Last, Cs. (2002). Effective Black-Box Testing with Genetic Algorithms.

[8] Hetzel, W. C. (1988). The Complete Guide to Software Testing, 2nd ed.

[9] Jacek. M. Zuranda. (1992). Introduction to artificial neural systems.

[10] Albert Endres, Cs. (2003). Hanbook software and System Engineering, Empirical Observations, Laws and Theories.

[11] Simon, H. (1999:p20). Neural networks – A comprehensive Foundation.

[12] Patrick J, C. (2000). Black-Box Test Reduction Using Input-Output Analysis. ACM.