Al-Rafidain Journal of Computer Sciences and Mathematics (RJCM)

www.csmj.mosuljournals.com

Using intelligence techniques to automate Oracle testing

Nour Sulaiman [1],* Safwan O. Hasoon[2]

Department of Software, College of computer science and mathematics, Mosul University, Mosul, Iraq,[1,2]
**Corresponding author. Email: nour.20csp28@student.uomosul.edu.iq[1]*

Article information

Article history:
Received : 15/2/2023
Accepted : 19/3/2023
Available online :

Abstract

Abstract— The software development life cycle includes software testing as a critical step. Oracle testing is necessary to determine if a particular test case detects an error because the purpose of testing is to ensure that the application adheres to its specifications. This test leads to the consumption of time and cost and may lead to errors in which the tester falls. This research aims to present a proposed system based on the automation of the Oracle test using intelligent techniques. The model was trained using the random forest algorithm and using the convolutional neural network. This proposed system is considered one of the box techniques. Test blacks that depend on the output of the system being tested.

Keywords:
Automated software testing, Mutation testing, Black-box testing, Artificial neural networks, CNN Network, Random forest.

Correspondence:
Author: Nour Sulaiman
Email: nour.20csp28@student.uomosul.edu.iq

I. INTRODUCTION

One of the main objectives of testing is to hasten the release of the program while ensuring that there are no bugs that could cause it to be discovered again and undermine the programmer's or developer's confidence in the program [15].

A software system's capabilities can be assessed to see if it can produce the needed results. Software testing is a crucial step in ensuring a certain level of software system performance and quality. More than half of the development time is spent on testing, a crucial component of software development [16].

Automated Software Testing (AST) It is a type of testing where the test case is carried out automatically using different automation tools and test scripts. Automated test cases are carried out using test scripts and automation techniques. Its advantage is that it expedites test execution after automated scripts have been generated [17].

In this paper, a system will be proposed to automate Oracle testing using intelligent techniques. This test aims to predict the output of the system being tested and compare it with the results of the software under test. The random forest algorithm and the convolutional neural network were used to build this system.

A. Test Software

Testing is the process of evaluating a system or its component(s) to determine whether or not it meets the specified requirements. This activity yields the actual, expected, and difference in their outcomes. Simply put, testing is the process of running a system to identify any gaps, errors, or missing requirements that are contrary to the actual desire or requirements [7].

When to Automate: Test Automation should be used for the following software projects:

1- Large and critical projects.

2- Projects that necessitate testing the same areas on a regular basis.

3- Requirements that do not change frequently.

4- Using many virtual users to access the application for load and performance.

5- Stable Software in terms of manual testing.

6- Time availability is required [7].

Black box testing and white box testing are the two primary methods for testing a program.

The project source code is not used to create the tests when using black box testing. only use software specifications. White box testing is a technique where tests are made using source code. The evaluation of the source code reveals that it behaves well under the hidden logic test. black box testing is more efficient with testing large code blocks since only the specification must be evaluated, whereas white box testing is more focused on the inner workings of the program. Black box testing is more focused on the specification [8].

B. Test Level

Software testing consists of several levels starting with the acceptance test that evaluates the system in relation to the requirements. Then comes the system test whose tasks are to evaluate the program in relation to the architectural design. Integration test that performs system testing and evaluation of the program in relation to the design of the subsystem, then comes the module test that evaluates order regarding detailed design. Finally, unit testing is tested by evaluating programs in relation to implementation [9].

C. Test Oracle

A test oracle is a mechanism that can be used to determine whether a method output is correct or incorrect. Testing is performed by executing the method under test with random data and evaluating the output with the test oracle [10].

An oracle ought to treat these two conditions separately:

1- If the condition on the initial state does not hold, then the program is off the hook: since its assumption does not hold, whatever it does must be considered correct.

2- The output condition of the specification is checked only if the input condition [11].

D. Regression Test

Regression testing is described as "the process of retesting the modified parts of the software and ascertaining that no new errors have been introduced into previously tested code.[12]

Regression testing is used to revalidate software modifications. Regression testing is an expensive process that involves running test suites to ensure that no new errors are introduced into previously tested code[13]. There are numerous methods for regression testing:

1- Retest all: A traditional method of regression testing is that all tests in the current test suite are redone. This is very expensive compared to other types of regression tests

2- Selection of Regression Tests: is used instead of the "retest all" technique because it is less expensive.

3- Prioritization of Test Cases: This regression testing approach prioritizes test cases more highly in order to improve the rate of fault discovery, or how quickly a test suite can identify mistakes in the altered program to increase reliability.

4- Hybrid Approach : The fourth regression technique is Test Case Prioritization and Selection. On this strategy, numerous researchers are working and have proposed a wide range of algorithms [14].

II. Related Works:

1- In 2017, Mark Harman, Shin Yoo, and William B. Langdon used search-based learning from open-source tests to automatically generate precise test models. In order to predict whether the program's output is accurate enough to be useful, it was determined that testing for mutations and n-version computing (with the aid of deep learning and other computer science methods) is necessary[1].

2- In 2018, Nermin, Mobyen, and Adnan proposed a solution for the test oracle problem using machine learning techniques. The system prototype uses classification using support vector machines. they laid out the necessary procedures for classifying multivariate time series. Time-series data can be applied to a variety of scientific fields, including industry, where signals are derived from sensors, biological data, speech, sounds, economics, etc [2].

3- In 2018, Alberto Joffe outlined a plan to develop a low-cost Oracle testing system that can identify bugs that result in non-trivial and semantic issues. He presented two methods for developing oracle testing: one uses software recursion and the other relies on plain-language comments that describe the source code of software systems. It introduces a method known as cross-validation oracles (CCOracles), which employs redundant sequences of method calls to encapsulate program recursion and produce test oracles automatically [3].

4- In 2020, K. Kamaraj, C. Arvind, and K. Srihari proposed a weight-optimized ANN that employs stochastic diffusion search to pinpoint the ideal weights with a particular fitness function, lowering computational time and misclassification rate. Automation of the development of test cases and test oracles has been the subject of extensive research. Among the automated test oracles, the artificial neural network (ANN) was heavily utilized [4].

5- Ke Chen, Yufei, and other researchers presented "automating the test oracle" in 2021 to find bugs in complex graphics-enhanced applications that don't crash. They suggested GLIB, a method for improving data based on codes for spotting "GUI glitches" in video games. The results show that GLIB can detect non-crashing bugs like GUI bugs in video games with high precision and recall when tested on 20 applications for real-world games [5].

6- In 2020, M. Valueian1, N. Attar, H. Haghighi, and M. Vahidi-As proposed an innovative black box method for developing automated oracles that can be used with low-observability software systems. The "Multi ANN Network" artificial neural network, which is used in the proposed method, trains on the input values and associated pass/fail results of the program being tested. application of the proposed method to software systems that have a lower Observational ability and a higher degree of accuracy than the current machine learning approach. After running an SUT with each input vector, a value has been assigned to each one, indicating whether the program was successful or unsuccessful[6].

7- Ke Chen, Yufei, and other researchers presented "automating the test oracle" in 2021 to find bugs in complex graphics-enhanced applications that don't crash.They suggested GLIB, a method for improving data based on codes for spotting "GUI glitches" in video games.GLIB was tested on 20 applications for real-world games, and the results show that GLIB has a high level of recall and precision when it comes to detecting non-crashing bugs like GUI bugs in video games. (Ke Chen, Yufei, & etc., 2021) [18].

III. Proposed System

Intelligent techniques can be used to automate Oracle testing for software testing and speed up regression testing. In this research, a system design will be proposed to implement Oracle testing, which is based on software testing by predicting the output of the program being tested and comparing it with the results of the application under test by calculating the distance. between the two results.

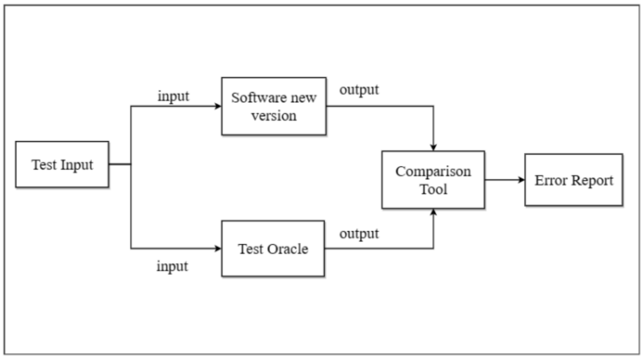


Figure I: Overview of the proposed system

This system consists of two stages:

The first stage: the training stage, which is the stage of training the model on the results of the program, and the inputs are entered into the application, and the data is entered in the model that will be trained using one of the intelligent technologies. At the same time, the software output is entered into the model that will be trained, as its results will be approved for training. Train the model The result of the trained model is Oracle's software outcome prediction model

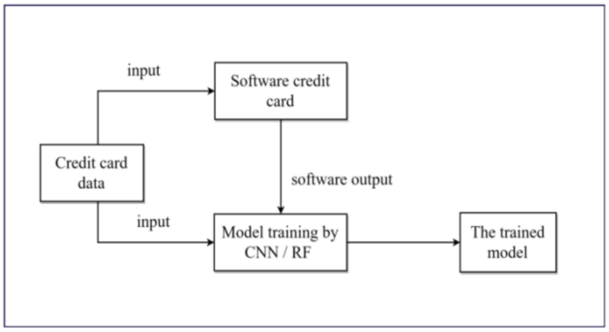


Figure II : Training stage

The second stage of the proposed system is the evaluation stage, in this stage is the software testing stage by predicting the model values of the software results and comparing them with the results of the application being tested. The comparison tool calculates the absolute distance between the values of the Oracle model and the results of the software under test. All values are between zero and one.

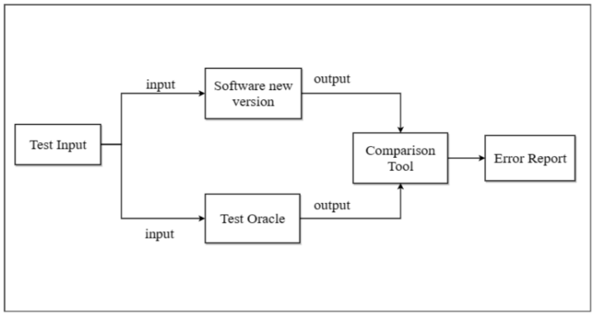


Figure III : Evaluation stage

A. Test Cases

The data that was used in this proposed system is randomly generated data that matches the specifications of the credit card approval application that will be tested using the proposed system. 10,000 samples were generated that will be considered credit card users, Application requirements and attribute descriptions were used to generate the training data used in this study.

Data details consist of nine attributes The number of columns to be created is 9 and their headings will be depending on the attributes of the application: (Region, Age, Nationality, State, marital status, number of dependents, Gender, income class and approved credit).

This data will be entered into the credit card approval algorithm to obtain the approved / not approved results. After obtaining these results, the training data set is ready to train the Oracle model.

This data will be entered into the credit card approval algorithm to obtain the approved / not approved results. After obtaining these results, the data will be pre-processed by deleting redundant data, after which the training data set will be ready to train the Oracle model. Below is a table of the types of data that will be generated and used in the proposed system.

TABLE I : Attribute Of Data

Name of the feature	***Data Type***	***feature Type***
ID number	int	I
Citizenship	int	I
State	int	I
Age	int	I
Gender	int	I
Region	int	I
Income Class	int	I
Numberofdependents	int	I
Marital status	int	I
Credit approved	inte	O
Amount	int	O

B. Implementation of the proposed system

In this proposed system The credit card approval system will be tested , the random forest algorithm and convolutional neural network will be used to train the model and adopt the highest accuracy model to be the approved Oracle system for testing and predicting software accuracy.

The steps of executing the test consist of three main steps: generating the test data, applying them to the system, and finally reporting the errors.

Create test data

The test data will be generated depending on the features and requirements of the credit card approval system on which the terms of the credit card algorithm will be based. The number of columns to be generated is 9 and their titles will depend on the application attributes: (region, age, nationality, state, marital status, number of dependents, gender income category, approved credit).

Number of data generated: 10,000 cases Sequential identification starts from 500,001 All data columns are merged together into one table Data frames will be saved in a csv file. This data will be entered into the credit card approval system and will be used to train the test model.

C. Build the model

After generating the data, the model will be built and trained using the test data with the output values of the credit card application, which will be represented by (0 or 1), which means

1: approved

0: not approved

First, the model will be trained with the random forest algorithm. A pre-processing will be performed on the data set and divided into training data (80% of the total data) and test data (20% of the total data). After completing the training of the model with the random forest algorithm, the test model will be built and trained using the neural network. convolutional with the same preprocessed data used with the output value of the credit card application.

D. comparison tool

After the model is trained, the credit card application will be tested using a comparison tool, which will compare the results of the application with the results of the model's prediction by calculating the absolute distance.

The root mean square error (RMSE) is used to determine the distance between the two values and is represented by the following equation:

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n} (y_j - \hat{y}_j)^2}$$

Where

N : is the number of sample.
y(i) : is the i-th measurement.
y ̃(i) is its corresponding prediction.

The comparison tool performs the following actions, first of all a threshold value is created, if the network prediction and application output match and the absolute distance is 0.0, then this means that the network prediction matches the program output means that both outputs are correct, and if the network prediction and application output fall in the interval less From the threshold value both outputs are likely to be correct, and finally if the network prediction and application output lie in the interval greater than the threshold value, there will be an error in the result.

Table II: Type Of Output

Type of Output	Period	Same
	Threshold < RMSE	● Both correct ● Both wrong
	Threshold > RMSE	● Model correct ● Software correct

E. injection errors

One of the important tests is the mutation test that injects errors and the system test to see if the program will be able to change the output value or will it remain the same. In this proposed system, logical errors will be achieved, and a slight change will be made to the algorithm of the application that will be under test.

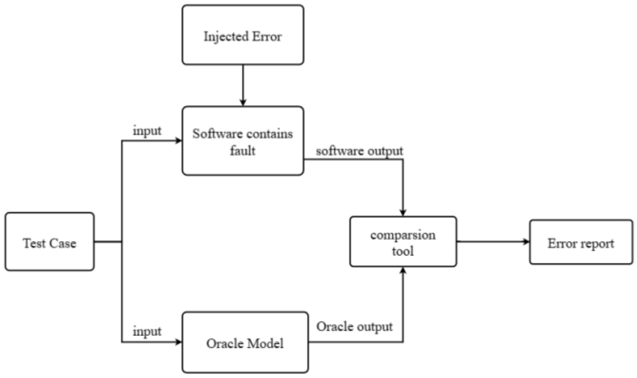
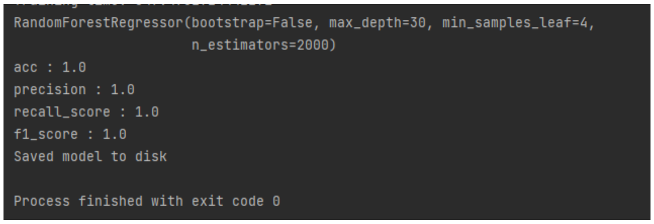


Figure V : injection errors

V. Result

After training the model on the random forest algorithm, the results were calculated as follows:



```
RandomForestRegressor(bootstrap=False, max_depth=30, min_samples_leaf=4,
                      n_estimators=2000)
acc : 1.0
precision : 1.0
recall_score : 1.0
f1_score : 1.0
Saved model to disk

Process finished with exit code 0
```

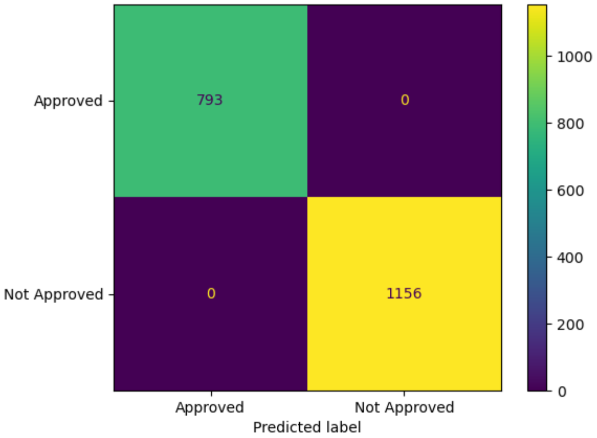


Figure VI: confusion matrix of the Random Forest model

1- CNN Training

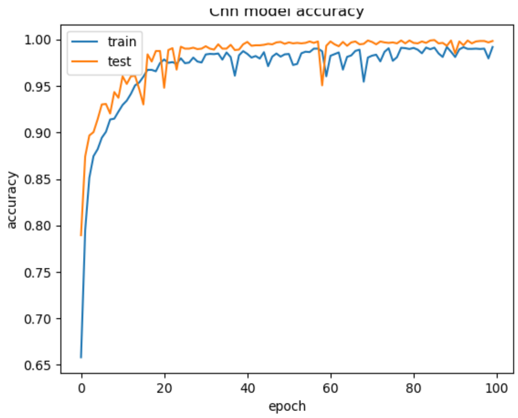


Figure VI : CNN model accuracy

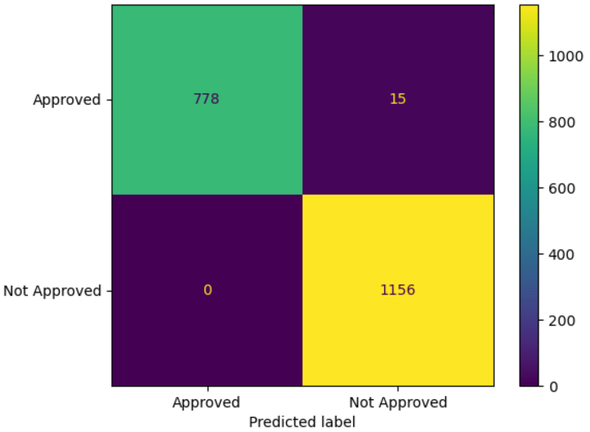


Figure VII : the confusion matrix of the CNN model

From the results of the training, it was found that the accuracy of the random forest algorithm was 100%. However, the CNN network has a very high accuracy of 99%.

Below is a comparison table showing the previous works with the algorithms used and the percentage of accuracy obtained.

Table III Comparison Table

Author (s)	***Address the propose***	***Algorithm for (classification)***	***Accuracy***
Vanmali, M Last, A Kandel	Using A Neural Network In The Software Testing Process	ANN	82%
Nermin Imamovic	Test Oracle Automation With Machine Learning: A Feasibility Study	Support Vector Machine Classification	92%
Seyed Reza Shahamiri, Wan Mohd Nasir Wan, and others	An Automated Frameworkfor Software Test Oracle	Multinetwork	98%
		Proposed System-CNN	99%
		Proposed System-RF	100%

VI. Conclusion

A system is proposed to test the validity of program results through its ability to predict program output. And comparing its results with the results of the program under test and discovering the different values, and the proposed system was able to convert the black box technology test into an automatic test and implement the Oracle test automatically, the system facilitated the regression test, and it was able to implement the mutation test on the credit card credit code. This system can be applied to business applications or even applications that rely on multiple inputs and one output. The proposed system was implemented using the convolutional neural network and the random forest algorithm, and the random forest algorithm showed an accuracy of 100% which is a slight difference from the CNN model which was 99%.

Acknowledgement

To the one who supported me during adversity throughout my life ... my father

To the generous heart and the tender chest ... my beloved mother

To those whom God strengthened my arms, they were the best supporter … my brothers and sisters

To all friends and everyone who had a good impact on the completion of the thesis, I dedicate my scientific research to you.

REFERENCES

[1] Langdon, W. B., Yoo, S., & Harman, M. (2017, May). Inferring automatic test oracles. In 2017 IEEE/ACM 10th International Workshop on Search-Based Software Testing (SBST) (pp. 5-6). IEEE.

[2] Imamovic, N. (2018). Test oracle automation with machine learning: A feasibility study.

[3] Goffi, A. (2018). Automating test oracles generation (Doctoral dissertation, Università della Svizzera italiana). Kamaraj, K., Arvind, C., & Srihari, K. (2020).

[4] A weight optimized artificial neural network for automated software test oracle. Soft Computing, 24(17), 13501-13511.

[5] Chen, K., Li, Y., Chen, Y., Fan, C., Hu, Z., & Yang, W. (2021, August). Glib: towards automated test oracle for graphically-rich applications. In Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 1093-1104).

[6] Valueian, M., Attar, N., Haghighi, H., & Vahidi-Asl, M. (2020). Constructing automated test oracle for low observable software. Scientia Iranica, 27(3), 1333-1351.

[7] Point, T. Tutorials Point simply easy learning. línea]. Available: http://www. tutorialspoint. com/operating_system/os_process_scheduling_algorithms. htm.[Último acceso: 28 05 2016].

[8] Saes, L. (2018). Unit test generation using machine learning. Universiteit van Amsterdamg.

[9] Ammann, P., & Offutt, J. (2016). Introduction to software testing. Cambridge University Press.

[10] Saes, L. (2018). Unit test generation using machine learning. Universiteit van Amsterdamg.

[11] Mili, A., & Tchier, F. (2015). Software testing: Concepts and operations. John Wiley & Sons.

[12] Aggarwal, K. K. (2005). Software engineering. New Age International.

[13] Duggal, G., & Suri, B. (2008, March). Understanding regression testing techniques. In Proceedings of 2nd National Conference on Challenges and Opportunities in Information Technology.

[14] Duggal, G., & Suri, B. (2008, March). Understanding regression testing techniques. In Proceedings of 2nd National Conference on Challenges and Opportunities in Information Technology.

[15] Padmini, C. (2006). Beginners Guide To Software Testing. Indianapolis: Lulu Enterprises, Inc.

[16] Fagerström, M., & İsmail, E. E. (2016). Automated Performance Regression Analysis: An Industrial Case Study (Master's thesis).

[17] Rafi, D. M., & Reddy, K. M. K. (2012). Automated software testing: A study of the state of practice.

[18] Chen, K., Li, Y., Chen, Y., Fan, C., Hu, Z., & Yang, W. (2021, August). Glib: towards automated test oracle for graphically-rich applications. In Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 1093-1104).

استخدام التقنيات الذكائية لأتمتة اختبار اوراكل

نور سليمان
قسم البرمجيات, كلية علوم الحاسوب والرياضيات , جامعة الموصل , الموصل العراق
nour.20csp28@student.uomosul.edu.iq

صفوان عمر حسون
قسم البرمجيات, كلية علوم الحاسوب والرياضيات , جامعة الموصل , الموصل العراق
Dr.safwan1971@uomosul.edu.

تاريخ الاستلام:15/2/2023　　**تاريخ القبول: 19/3/2023**

المخص

تتضمن دورة حياة تطوير البرامج اختبار البرامج كخطوة حاسمة. يعد اختبار اوراكل ضروريًا لتحديد ما إذا كانت حالة اختبار معينة تكشف عن خطأ لأن الغرض من الاختبار هو التأكد من أن التطبيق يلتزم بمواصفاته. يؤدي هذا الاختبار إلى استهلاك الوقت والتكلفة وقد يؤدي إلى أخطاء يقع فيها المختبر. يهدف هذا البحث إلى تقديم نظام مقترح يعتمد على أتمتة اختبار أوراكل باستخدام تقنيات ذكية. تم تدريب النموذج باستخدام خوارزمية الغابة العشوائية واستخدام الشبكة العصبية التلافيفية. يعتبر هذا النظام المقترح أحد تقنيات الصندوق. اختبار اللون الأسود الذي يعتمد على إخراج النظام الذي يتم اختباره.

الكلمات المفتاحية: اختبار البرمجيات الآلي ، اختبار الطفرات ، اختبار الصندوق الأسود ، الشبكات العصبية الاصطناعية ، شبكة CNN ، الغابة العشوائية.