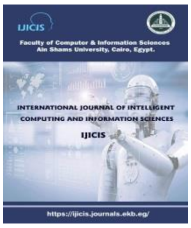
IJICIS, Vol.21, No.1, 84-94
DOI: 10.21608/ijicis.2021.69462.1076

International Journal of Intelligent Computing and Information Sciences

https://ijicis.journals.ekb.eg/



Enhancing Test Cases Prioritization for Internet of Things based systems using Search-based Techniques

N. M.Mohamed *

Department of Information Systems,
Computer and Information Science,
Ain Shams University,
Cairo, 11566, Egypt
noha_medhat@cis.asu.edu.eg

S. M. Moussa

Department of Information Systems,
Computer and Information Science,
Ain Shams University,
Cairo, 11566, Egypt
sherinmoussa@cis.asu.edu.eg

N. L. Badr

Department of Information Systems,
Computer and Information Science,
Ain Shams University,
Cairo, 11566, Egypt
nagwabadr@cis.asu.edu.eg

M. F. Tolba

Department of Scientific Computing,
Computer and Information Science,
Ain Shams University,
Cairo, 11566, Egypt
fahmytolba@cis.asu.edu.eg

Received 2021- 03-25; Revised 2021-4-30; Accepted 2021-4-30

Abstract: *Test cases prioritization has been excessively considered for continious regression and integration testing in Internet of Things based systems to apply multilevel testing activities. Various number of devices, sensors and acctuators are connected together through the internet using different technologies, which requires extensively testing the effeciency of these components and the transferred data between them. Due to the number of the connected components has dramatically increased, causing a direct proportional increase in the number of test cases.Studies that handle the augmentation of the number of test cases for traditional systems lack effeciency when applied for Internet of Things based systems. In this paper, we introduce an enhancement for test cases prioritization using Hill Climbing algorithm as a local search based technique, adapted to achieve tangible effeciency. It is integrated with the LSTM deep learning algorithm for test cases classification purposes. The results of the test cases prioritization using Hill Climbing for regression and integration testing are evaluated using precision, where it achieved 80% and 97% for regression testing, and 93% and 88% for integration testing using two Internet of Things-based system datasets.*

* Corresponding author: N. M.Mohamed

Department of Information Systems, Computer and Information Science, Ain Shams University, Cairo, 11566, Egypt
E-mail address: noha_medhat@cis.asu.edu.eg

Keywords: *Deep learning, Integration testing, Internet of Things based systems, Regression testing, Test case prioritization, Test case selection, Local search-based techniques, Hill Climbing*

1. Introduction

Internet of Things (IoT) based systems differ from traditional systems, in which an enoromous number of heterogenous devices are connected to build IoT-based systems, raising various difficulities and challenges in different aspects. The issues faced are due to the incremental addition of devices, users, sensors, using different technologies and protocols [1][2]. Focusing on testing issues demands high investigation to ensure the effectiveness and validity of IoT-based systems, pushing to find new solutions that are capable of handling the unique nature of IoT-based systems. In this paper, we propose an enhanced prioritization of test cases (TCs) for IoT-based systems during regression and integration testing. Integration testing is conducted everytime a new component joins the IoT-based system's architecture, whether it is a sensor, device, or a new user role[3]. Regression testing is conducted upon receiving any new change requests or emerging bugs after the system is deployed [4]. Both the selection and prioritization of TCs for integration or regression testing face diverse challenges because of the increasing volume of the needed TCs to examine [5]. TCs selection for IoT-based systems was investigated in[6] using the Long Short Term Memory (LSTM) deep learning classifier [7] for reduction purposes. Moreover, a representation of TCs prioritization was discussed in [7] using Genetic algorithm (GA) and Simulated Annealing (SA) algorithm, resulted in a low accuracy level. Thus, we present in this study an enhanced framework for continuous regression and integration testing of IoT-based systems to improve TCs prioritization, considering the specific nature of IoT-based systems. The enhanced prioritization of the selected TCs is achieved using the Hill Climbing (HC) algorithm, as one of the Search Based Techniques (SBT) [8], which indicated to achieve better efficiency for the proposed framework.

The rest of the paper is organized as follows: the second section shows the related work, the third section is for demonstrate the proposed Enhanced Framework for Continuous Integration and Regression Testing in IoT- based systems (IoT-ECIRTF), the forth section is for the experimental evaluation, the fifth and the last section is the conclusion to conclude the paper work.

2. Related work

Testing research in IoT-based systems has addressed the need for improved testing techniques to match the nature of such systems. The challenges of testing in IoT-based systems were addressed in [1] on the different testing levels, where a lackage was indicated when testing IoT-based systems during the selection and prioritization of the IoT system TCs . In [6], a testing technique was proposed in order to effectively apply integration and regression testing over IoT based systems, which was by merging deep learning LSTM algorithm for the IoT TCs selection. LSTM classifier was applied in order to classify the IoT system requirements into the main IoT system components which are the user devices, sensors and actuators, data processing, and protocols and gateways. The proposed technique prioritizes IoT system TCs by the application of Search Based Techniques (SBT) using Simulated Annealing (SA) and Genetic Algorithm (GA). However, an improvement for the prioritization accuracy was needed. In [9], several prioritization techniques were discussed for the handling of continuous integration environment, in which an IoT system is typically a continuous integration system. The accuracy of the TCs prioritization was investigated, depending on different metrics such as the statements coverage rate, failure history, execution time, faults detected per cost, or using Model based techniques. This study assured that none SBTs were considered with IoT based systems for TCs prioritization. Authors in [10] have confirmed the need for prioritization techniques when testing IoT based systems, requiring auto testing and TCs generation. Authors have improved TCs generation using Model Based techniques (MBT) and adapted them to work with the IoT-based systems. Thus, more investigations for TCs prioritization techniques in IoT-based systems are still needed to minimize time consumption and associated costs.

In [11], the focus was the requirements prioritization for the highly configuarable systems as in IoT systems. The criteria were set to prioritize the requirements by applying elicitation, analysis, documentation, verification and validation, triggering the lack of covering all kinds of configurable systems, such as the embedded systems, in which testing the connected hardware and software parts should be tested regarding the configuration requirements for scalablility and reliablity. The used requirements prioritization technique was the least squares estimation, but it required direct human involvement. Authors in [12] disscussed the frequent changes in the systems connected through the cloud, in which the periodic changes enforce periodic testing and faults detection after each update affecting the system. The used technique was the Average Percentage of Faults Detected (APFD), in which TCs prioritization was according to the average rate of TCs to test certain components affected with specific updates. This approach required accessing the lines of code for components, which was not always possible. The verification and validation facilities for smart house IoT systems testing was proposed in [13], where data analysis was a core step. Human direct and continuous interaction with the system was mandatory, in order to ensure the data quality of the system prioritization at different IoT data dimensions. The main limitation was the affirmation for human intrusion, which wastes efforts, time and cost.

Prioritization was also disscussed for security testing challenges in [14]. As sensing in the sensor networks for IoT systems should be conducted sequential, the sensing and reading of data were prioritized to assure the IoT system security. This methodology required the insertion of the preffered nodes to the hardware components during the system development, which was hard to sustain maintainability after the system was deployed to the end users. The quality of autonomous vehicle behavior as a part of IoT systems was considered in [15], where all automatic movements based on the end-users requests should be tested to confirm the overall behavior of these vehicles. The prioritization of the metrics and key performance indicators required for system testing was investigated to save cost and time consumption for the system acceptance.

IoT systems characteristics are the reasons behind the challenges faced for testing such system functionalities and confedentiality, networks reliability, and security and privacy of IoT systems as discussed in [16], stressing for the need to develop new models to fit for the quality of IoT systems. A clear call for machine learning models was defined to deal with the requirements to fulfill for systems quality. However, no model was proposed to be taken as a solution for the discovered problems. Recent researches discussing the testing and quality assurance of IoT systems prove having common vulnerabilities when dealing with the characteristics of IoT systems, where there is data diversity, vast connected hardware and software parts, and diversity of network technologies that connect the system parts together. New ideas and testing solutions are needed for testing different levels of IoT systems, specially those related to the regression and integration testing for IoT systems.

The main lackage on the previous researchers work is missing to provide an adaptable or convinient methodology for testing IoT based systems. Different reseaches are concentrating on different testing levels such as the application of performance and security testing over IoT based systems, showing the importance of applying prioritization of the test cases while testing IoT based systems. Previous researches have applied prioritization using different traditional techniques as it is mentioned before, resulting to have prioritized test cases with low accuracies. In order of the faced challenges during testing IoT based systems using the traditional testing techniques, the focus in this paper will be on providing a prepared framework for enhancing the priortization of the test cases during the application of testing over the IoT based systems.

3. The Enhanced Framework for Continuous Integration and Regression Testing in IoT-based systems (IoT-ECIRTF)

This paper presents an enhanced framework for integration and regression testing in IoT-based systems by integrating the Hill Climping (HC) algorithm as one of the SBTs for TCs prioritization on the top of the deep learning LSTM algorithm at the IoT-based Continuous Integration and Regression Testing Framework (IoT-ECIRTF) [6], as shown in Fig.1. IoT-ECIRTF consists of four main layers as detailed herein.

3.1 The IoT components training layer in IoT-ECIRTF

This layer aims to train a classifier to cope with the specifications of IoT-based systems. Testing using the high number of TCs in IoT-based systems is a very tedious effort, in which the selection of related TCs according to the changes or the newly added components is required. An IoT specifications pre-processing module is used, where Natural Language Processing (NLP) techniques [17] are applied to clean the IoT system specifications. An IoT components features extraction module follows, in which the LSTM classifier [18] is used to analyze the cleaned specifications of the IoT system [19]. The subsections of is layer is divided as the following:

- **IoT specifications pre-processing module:** in which Natural Language Processing (NLP) techniques are used for the aim to clean and remove the stop conditions of the IoT system specifications, in which it is the used data as the training dataset.

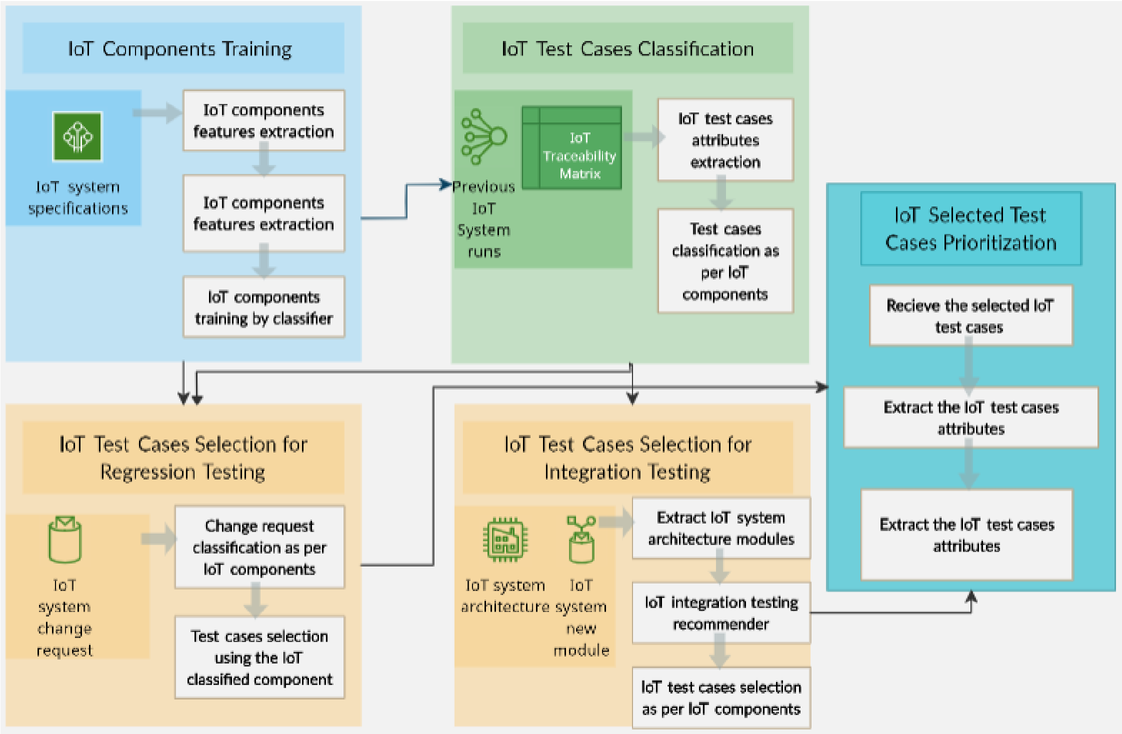


Figure.1: The IoT-based Enhanced Continuous Integration and Regression Testing Framework (IoT-ECIRTF)

- **IoT components features extraction:** in this module LSTM classifier is used as the analyzer of the cleaned specifications of the IoT systems, in which the extracted features are the words that have highly weighted rates. These features are the used ones on the next module, in which to train the IoT system components. IoT systems components to have are the user devices, gateways and protocols, sensors and actuators, and data processing components. The reason of choosing the LSTM deep learning classifier is that it is working effectively with long sequences of data.
- **IoT components training by classifier:** in here the classifier works on the extracted features that are detected from the previous module in which is called the features extraction module.

The result of this layer is a trained model that is used to classify the TCs of the next layer. The classification is according to the extracted component, where the IoT system specifications are categorized based on the previously mentioned four components. To increase the learning accuracy, the LSTM algorithm runs four times. Sigmoid function is used to measure the Input, Output, Forget gates, new hidden state and new value state of LSTM algorithm.

3.2 The IoT Test Cases classification layer in IoT-ECIRTF

In this layer, the TCs are classified as per the IoT standard components, which are the (1) sensors and actuators, (2) protocols and gateways, (3) user devices, and (4) the data processing [20]. TCs are classified for the selection and prioritization purposes during the continuous integration and regression testing in the IoT based systems. The inputs for this layer are the traceability matrix of the system specifications, where the TCs are mapped to the requirements and the previous runs of TCs. The steps followed through this layer are:

- **IoT test cases attributes extraction:** Some attributes extracted for the TCs are needed for the prioritization for the IoT based systems. The extracted attributes are the TC description, name, Coverage Rate (CR), Fault Detection Rate (FDR), and Execution Time (ET). Other attributes are the TC Id, name, and the description. Evaluated the priority of the TC by these metrics, where if the FDR and the CR are high it triggers to have high priority and if the ET is low it triggers to have high priority of the TC too. The CR of TC, in which the summation of the covered number of the requirements is divided by the total number of the requirements. The FDR is calculated by the total number of the detected faults from the previous IoT system run. The output of this layer is the TCs classified using the trained generated module and the extracted attributes of the previous test runs are used for the selection purpose of the integration and regression testing.

- **IoT Test cases classification as per IoT components:** The IoT system TCs are classified as per the IoT components after receiving the output of the previous layer, in which the LSTM classifier is trained according to the IoT specifications. Classification of the TCs is considered as testing the testing dataset in which the accuracy of the trained classifier is obtained, after evaluating the results of the IoT TCs selection in the next layer. The IoT TCs selection is mentioned in the next two layers.

3.3 The IoT Test Cases selection for regression testing in IoT systems

The selection of the IoT TCs during the regression testing is triggered when receiving a change request as it is described in the following modules:

- **Change request as per IoT components:** When a change request is received, it is classified to its related IoT standard component. This classification is intended to select the relevant TCs, to be further prioritized for the regression testing purposes. The change request is classified to be whether sensors and actuators component, protocols and gateways, user devices, or data processing component. IoT components are used for the selection, in which the TCs to be chosen is detected.

- **TCs selection using the classified IoT components:** This module is responsible of applying the selection of the related TCs according to the testing type. If the received input is a change request, then we apply TCs selection according to the classified category of the change request. The selection of the TCs is according to the IoT components that is detected to need testing depending on the previous module, where the IoT components is whether sensors and actuators component, protocols and gateways, user devices, or data processing component.

3.4 The IoT TCs selection layer for integration testing in IoT systems

The selection of TCs for integration testing is considered if a system architecture is received and a new IoT system module as an input. It consists of three modules as follows:

- **Extract IoT system architecture modules:** Extracting the modules of IoT system architecture is conducted after the conversion of the architecture format into the XML format to convey with the program implementation, where some attributes are checked to be existent. The needed attributes for each module are the name of the module, ID, previous connected modules IDs, and suffix of connected modules IDs.

- **IoT integration testing recommender:** When the system receives a new module and the system architecture, it is required to apply integration testing. The dependency between the modules is the indicator of the relevant modules' selection. The integration testing handler helps to decrease the number of required stubs or drivers, that is, having less number of following connected modules or previously connected modules. If the connected following modules are equal to the number of previously connected modules, then it is recommended to use stubs over drivers. If all modules are implemented, then we neither use stubs nor drivers; we select all related TCs of connected needed modules.

- **TCs selection using the classified IoT components:** This module is responsible of applying the selection of the related TCs according to the testing type. If the received input is a new module besides the IoT system architecture, then TCs selection is related to the connected modules according to the newly added module. The classification of the newly added module defines the relevant IoT system components to be ready for the selection of the related IoT system TCs. The selected IoT TCs are used during the integration testing to effectively test the IoT system when receiving new IoT modules.

3.5 IoT selected test cases prioritization

Researches have issued challenges while testing IoT systems, mostly caused due to the complexity of IoT system, where the time and cost are heavily consumed that required techniques for testing IoT systems. In this layer, the focus is to reduce the cost and time by applying IoT TCs prioritization using Search Based Techniques (SBT) that have proved great performance with traditional systems. SBTs care about both the huge search spaces (Global search techniques) and the small search spaces (Local search techniques), in which it varies in the same IoT system we test according to the selected IoT TCs. This triggers having better accuracies when applying local search techniques for the regression testing and the global search techniques, causing better results for integration testing. The number of selected TCs guides to which type of SBT is suitable. When the number of IoT selected TCs is high, global techniques results are better than Local techniques and vice versa.

IoT-CIRTF in [6] prioritizes IoT-based TCs using SA (Local SBT) and GA (Global SBT) algorithms. In this module, the Hill Climbing (HC) algorithm is proposed to provide higher accuracy than using SA for IoT-based TCs prioritization to enhance the framework efficiency. The proposed Enhanced Continuous Integration and Regression Testing Framework (IoT-ECIRTF) calculates the Fitness Function (FF) for each TC using the extracted TCs attributes to decide the priority of TCs. The attributes include the FDR, ET, and CR. The FF equation is shown in (1) as follows:

$$FF_{(TC_{ID})} = \sum_{i=1}^{n} \frac{TC_{index} \cdot (CR_i + FDR_i)}{ET_i} \qquad (1)$$

Where ID is for the test case identity that starts from $i = 1$ until it reaches the total number n of TCs in the sequence of TCs that the fitness function is being calculated for, the test case $index$ is for the order of test case in the sequence, CR is the coverage rate that defines the rate of the requirements covered using the specified test case, FDR_i is the fault detection rate that is being detected by the specified test case, ET_i is the execution time each test case requires to run. The sum of the coverage rate and the fault detection rate is divided over the execution time which is defined as ET_i in (1).

Algorithm 1 describes the Hill Climbing (HC) algorithm utilized in the IoT-ECIRTF. It starts with an initial solution selected randomly, in which the solution in our problem is defined as the sequence of the selected related TCs [21]. The algorithm then tries to reach the optimum sequence of prioritized TCs by evaluating the effectiveness of this solution. It checks the current sequence of TCs with the nearest sequences of TCs to determine if the current sequence is better than the next or previous sequence, deciding whether to keep the current sequence as the best prioritized sequence or to move to the next or previous sequence. The comparison between the sequences is based on the FF calculation as shown in (1). Reaching a better solution than the current one is defined as the Local Maxima [22], in which the solution is either moving forward or backtrack in order to reach the local maxima. The exit from this repetitive check is when there is no more better solutions found, or in case the found solution is the best solution according to the target solution that is set from the beginning if the better sequence of prioritized TCs is known in advance.

Algorithm 1: Hill Climbing Algorithm for IoT TCs Prioritization in IoT-CIRTF.

Output: TCs prioritized with the highest priority value.

```
Begin
  Initialize  iteration i: i=1
  Generate sequence of n number of TCs
  Loop
     Calculate FF for the initial sequence of TCs     (ISTC)
     Select next neighbor sequence of TCs (NNTC)
     Calculate FF for the next selected solution
     If (FF (ISTC)> FF (NNTC)) then
        Keep the Initial sequence of TCs (ISTC)
     Else If (FF (NNTC)> FF (ISTC)) then
        Choose the Next Neighbor of TCs (NNTC)
        Set ISTC=NNTC
     End if
     If (TCs sequence (ISTC) is not changing after a number of iterations) then
        Break
     End if
  End Loop
  Return prioritized TCs
End
```

The application of the Hill Climbing algorithm for prioritizing test cases using the proposed Fitness Function (FF) as shown in equation (1) is proving better accuracy for prioritization, compared to the results of prioritization using other Search Based Techniques (SBT). Results and detailed comparison with percentages is shown in the next section.

4. The experimental evaluation

This section discusses the experimentations applied to evaluate our proposed enhanced IoT TCs prioritization layer using the Hill Climbing algorithm (IoT-ECIRTF). We considered the Global System for Mobile communication (GSM) [23] as our IoT-based system case study, where the specifications and TCs of both the IoT device connection efficiency [24] and Mobile IoT (MIoT) [25] datasets are used for evaluation. The GSM system architecture and component diagram are used for the integration testing, calculating the dependency between the modules[26]. LSTM classifier was trained for both datasets, where a number of iterations achieved more accuracy with four layers of LSTM and hundreds of epochs to cover the IoT system datasets requirements. The accuracy of the prioritizing IoT system TCs is measured by the precision equation, as shown in (2):

$$Pr\,e\,cision = \frac{TP}{TP+P} \qquad (2)$$

FF is applied over the created IoT TCs sequences to evaluate the TCs order generated by the HC algorithm. The better ordered sequences are triggered at higher values of FF. FF is muted to work with IoT TCs sequences, as the main factors of IoT TCs are the CR, FDR, and ET. The HC precision percentages achieve 80% and 97% for the IoT device connection efficiency dataset and the MIoT dataset respectively. The precision values of HC are compared to those of SA and GA algorithms implemented in IoT-CIRTF. Fig.2 presents a comparison between the FF results of HC, SA and GA algorithms with respect to regression testing for both datasets. The HC algorithm achieves better precision percentages of 80% and 97% respectively for the IoT device connection efficiency and MIoT datasets respectively, compared to the results obtained when applying SA that achieved 72% and 81% respectively. The use of GA as a Global SBT achieved accuracies of 88% and 81% for both datasets respectively.

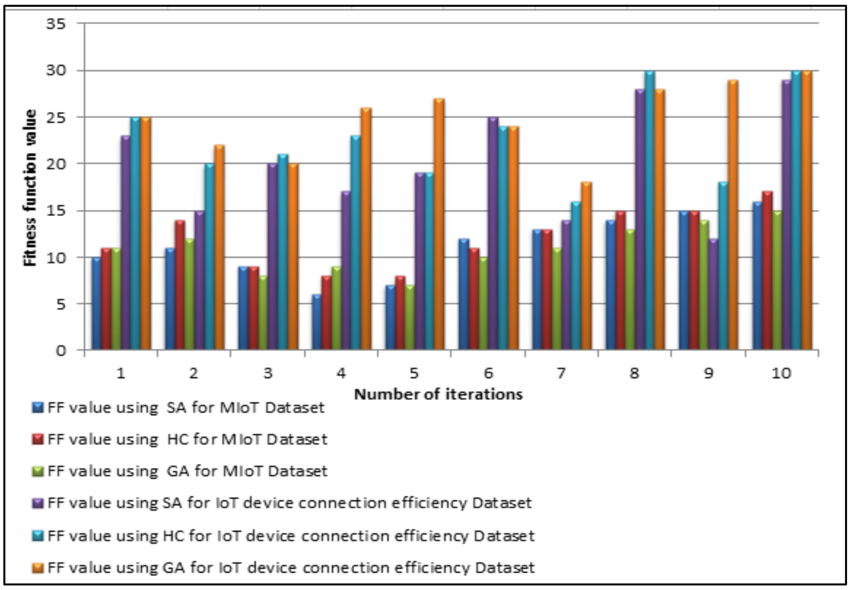


Figure.2: Comparison of TCs prioritization results for regression testing using SBTs.

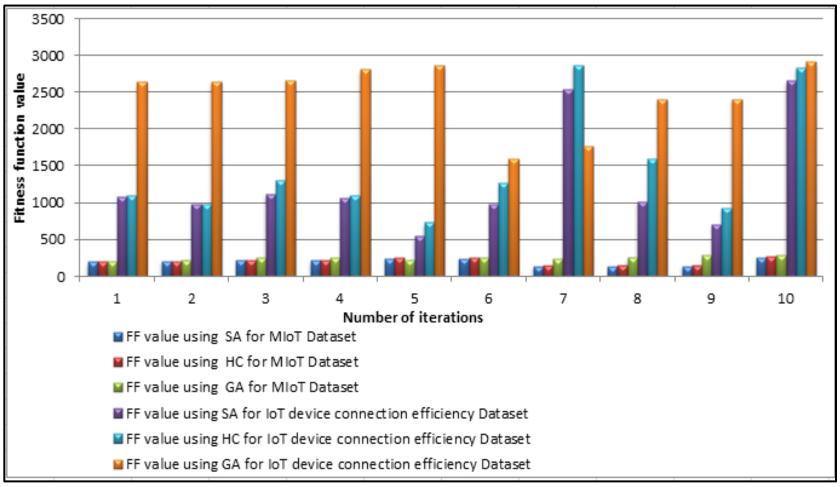


Figure.3: Comparison of TCs prioritization results for integration testing using SBTs.

Fig.3 presents a comparison between the FF results of HC, SA and GA algorithms with respect to integration testing for both datasets. The results prove to have better accuracies for HC Local SBT over the prioritization results of using SA Local SBT for both datasets with percentages of 93% and 88% respectively for HC algorithm, compared to 80% and 77% for the SA algorithm. The use of GA as a Global SBT has resulted accuracies of 92% and 89% for both datasets respectively.

5. Conclusion

In this paper, the IoT-based Enhanced Continuous Integration and Regression Testing Framework (IoT-ECIRTF) for IoT-based test cases (TCs) prioritization using Hill Climbing (HC) local search-based technique is proposed. The framework uses the LSTM deep learning algorithm after being modified to fit the long sequences of the IoT-based systems requirements and TCs. We reached 4 layers to gain better accuracy for TCs selection for both regression and integration testing. The TCs prioritization using HC for regression testing have proved higher accuracy values compared to the SA, receiving percentages of 80% and 97% for the IoT device connection efficiency and MIoT datasets respectively compared to the results achieved when applying SA that achieved 72% and 81%. The precision accuracy for integration testing using IoT device connection efficiency and MIoT datasets were 93% and 88% for the HC algorithm compared to 80% and 77% for the SA algorithm, which triggered better accuracies. When applying GA for regression testing, it gained accuracies of 88% and 81% for both datasets respectively, and 92% and 89% for integration testing for both datasets respectively. As a future work, it is intended to handle other testing techniques rather than the integration and regression testing techniques and to increase the testing accuracy.

References

[1] N. Medhat, S. Moussa, N. Badr, and M. F. Tolba, “Testing Techniques in IoT-based Systems,” *Proc. - 2019 IEEE 9th Int. Conf. Intell. Comput. Inf. Syst. ICICIS 2019*, pp. 394–401, 2019.

[2] O. Olusola, S. Chinye, and K. C. Ukaoha, “Assessing the stability of selected software components for reusability,” *IJICIS*.

[3] P. Chamoso, A. González-Briones, S. Rodríguez, and J. M. Corchado, “Tendencies of Technologies and Platforms in Smart Cities: A State-of-the-Art Review,” *Wirel. Commun. Mob. Comput.*, vol. 2018, 2018.

[4] T. Kuroiwa, “A Hybrid Testing Environment between Execution Test and Model Checking for IoT System,” *2019 IEEE Int. Conf. Consum. Electron.*, pp. 1–2.

[5] M. Leotta *et al.*, “An acceptance testing approach for internet of things systems,” *IET Softw.*, vol. 12, no. 5, pp. 430–436, 2018.

[6] N. Medhat, S. M. Moussa, N. L. Badr, and M. F. Tolba, “A Framework for Continuous Regression and Integration Testing in IoT Systems based on Deep Learning and Searchbased Techniques,” *IEEE Access*, vol. 8, pp. 1–1, 2020.

[7] X. Wang, H. Zeng, H. Gao, H. Miao, and W. Lin, “Location-Based Test Case Prioritization for Software Embedded in Mobile Devices Using the Law of Gravitation,” *Mob. Inf. Syst.*, vol. 2019, 2019.

[8] A. Stuhlsatz, J. Lippel, and T. Zielke, “Feature extraction for simple classification,” *Proc. - Int. Conf. Pattern Recognit.*, no. 2, pp. 1525–1528, 2010.

[9] J. A. Prado Lima and S. R. Vergilio, “Test Case Prioritization in Continuous

Integration environments: A systematic mapping study," *Inf. Softw. Technol.*, vol. 121, no. January, p. 106268, 2020.

[10] V. Rechtberger, M. Bures, and B. S. Ahmed, "Alternative Effort-optimal Model-based Strategy for State Machine Testing of IoT Systems," *ACM Int. Conf. Proceeding Ser.*, no. September, pp. 141–145, 2020.

[11] A. Ali, Y. Hafeez, S. Hussain, and S. Yang, "Role of Requirement Prioritization Technique to Improve the Quality of Highly-Configurable Systems," *IEEE Access*, vol. 8, pp. 27549–27573, 2020.

[12] S. Ali *et al.*, "Towards Pattern-Based Change Verification Framework for Cloud-Enabled Healthcare Component-Based," *IEEE Access*, vol. 8, pp. 148007–148020, 2020.

[13] R. Alrae, Q. Nasir, and M. Abu Talib, "Developing House of Information Quality framework for IoT systems," *Int. J. Syst. Assur. Eng. Manag.*, vol. 11, no. 6, pp. 1294–1313, 2020.

[14] D. G. Costa and F. P. de Oliveira, "A prioritization approach for optimization of multiple concurrent sensing applications in smart cities," *Futur. Gener. Comput. Syst.*, vol. 108, pp. 228–243, 2020.

[15] D. Minovski, C. Ahlund, and K. Mitra, "Modeling Quality of IoT Experience in Autonomous Vehicles," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3833–3849, 2020.

[16] S. Bagchi *et al.*, "New Frontiers in IoT: Networking, Systems, Reliability, and Security Challenges," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11330–11346, 2020.

[17] M. A. Khan and R. Karim, "SS symmetry A Scalable and Hybrid Intrusion Detection System Based on the Convolutional-LSTM Network," 2019.

[18] X. Ye, F. Fang, J. Wu, R. Bunescu, and C. Liu, "Bug Report Classification Using LSTM Architecture for More Accurate Software Defect Locating," *Proc. - 17th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2018*, pp. 1438–1445, 2019.

[19] A. Rashad, F. Maghraby, M. Fouad, Y. Lashin, and A. Badr, "Association Rules Based Classification for Autism Spectrum Disorder Detection," *Int. J. Intell. Comput. Inf. Sci.*, vol. 18, no. 2, pp. 13–28, 2018.

[20] L. Antão, R. Pinto, J. Reis, and G. Gonçalves, "Requirements for testing and validating the industrial internet of things," *Proc. - 2018 IEEE 11th Int. Conf. Softw. Testing, Verif. Valid. Work. ICSTW 2018*, pp. 110–115, 2018.

[21] Z. Li, Y. Xi, and R. Zhao, "A Hybrid Algorithms Construction of Hyper-heuristic for Test Case Prioritization," pp. 1749–1754, 2020.

[22] S. Tarwani and A. Chug, "Investigating optimum refactoring sequence using hill-climbing algorithm," vol. 2667, 2020.

[23] C. Notice and A. Notice, "GSM, 2015. IoT Device Connection Efficiency Common Test Cases 30 January 2015," pp. 1–51, 2015.

[24] G. Requirements, "MIoT Test Requirements," pp. 1–24, 2017.

[25] J. Esquiagola, L. Costa, P. Calcina, G. Fedrecheski, and M. Zuffo, "Performance Testing of an Internet of Things Platform," no. IoTBDS, pp. 309–314, 2017.

[26] T. Xia, Y. Song, Y. Zheng, E. Pan, and L. Xi, "Computers in Industry An ensemble framework based on convolutional bi-directional LSTM with multiple time windows for remaining useful life estimation," *Comput. Ind.*, vol. 115, p. 103182, 2020.