International Journal of Information Technology and Applications
Volume 1, Issue 4, December 2024 ISSN 3007-7397 © P-Learn Press

Explorations of Semantic Error Localization and Automatic Repairing through Deep Neural Network and Quantum Intelligent Algorithm

Peng Mei*

Party School of the Central Committee of C.P.C, Beijing, China

meipeng@ccps.gov.cn

Abstract. The aim is to accelerate the compilation of programmers, reduce the error codes in the program, and accurately locate the semantic errors. The deep learning neural networks and Quantum Intelligence Algorithm (QIA) encode the information hidden in the code. Then, a localization-repairing model for semantic errors is constructed based on a deep quantum neural network. This model locates the code semantics through the deep neural attention mechanism and repairs the erroneous codes through QIA. Furthermore, the model's performance is tested and verified in Online Judge (OJ). Other algorithms are also tested for performance comparison. The effectiveness of the proposed localization-repairing model is proven. The results suggest that introducing the attention mechanism to neural networks can improve the model accuracy to 70.91%; meanwhile, introducing QIA can accelerate the convergence and increase the recognition rate of the model. Compared to traditional semantic localization models, the positioning accuracy of the proposed model is increased to 85.24%. Besides, its capability of semantic repairing is significantly improved compared to single algorithm models, and the proportion of program repairing is 89.27%. Tests on the system also prove the advantages of the proposed model in precise localization and excellent repairing. The above results can provide more ideas for the localization and repair of semantic errors in computer programs.

Keywords: Deep learning, Neural network, Quantum intelligent algorithm, Semantic error localization, Program repairing.

1. Introduction

As technology continues to advance, the lives of humans are electronicized, including electronic news, electronic books, and electronic offices; consequently, traditional text-based works are gradually decreasing [1]. The Internet is a crucial platform for human beings. It has no geographical restrictions; all users can publish and answer questions anytime and anywhere. The fast speed of information transmission not only enriches the learning methods of humans but various data sharing also improves the working efficiency of humans [2]. However, due to the limitations of input methods, or the carelessness during typing, semantic errors will appear, which affects not only the quality of the texts but also the subsequent understanding by others [3]. Therefore, a correct semantic understanding is vital for human communication, and the same is valid for computers. Programmers often encounter various while writing programs. This is inevitable but causes programmers to spend a lot of time and energy repairing the codes [4]. Hence, scholars began to work on semantic error detection and repair, specifically for computer programs [5]. According to the repairing methods generated, the programmers can either directly modify the program codes or be provided with a modification hint to improve the code quality [6]. Due to the complexity of programming languages, except for the keywords defined, the programmer can arbitrarily determine the variable's name, leading to many problems in the repairing method of the automatic programs [7]. There are more than 20 automatic detection and repair methods for different programming languages; however, many problems are found in various semantic repair methods for computers [8]. Studying language errors and repairing systems will not only improve the accuracy of communication but also reduce the useless operations of programmers, which is of considerable significance to computer programming.

The most common methods in text detection include the rule-based, corpus-based, and feature learning-based methods [9]. The rule-based method learns the linguistic knowledge in the text, defines the intrinsic rules, and compares the text to be tested with the rules, thereby judging the semantic error information [10]. After learning all the texts, the corpus-based method constructs a corpus and then performs statistics and diagnosis based on the word strings [11]. The feature learning-based method constructs a knowledge base of common errors with excellent effect; then, it extracts features from the text to be tested, scores the text through machine learning or deep learning, and outputs the result [12]. Research on text detections is various. Devlin et al. (2017) utilized the rule-based processor for generating a large number of repairing candidates; then, they adopted a statistical neural network model to score these candidates with a novel neural network architecture; this model could predict 41% of wrong sentences only through one test [13]. Mohan and Jannesari (2019) employed neither any defect datasets nor error-free programming source codes for training; the structure and semantic details of Abstract Syntax Tree (AST) were employed for training; consequently, on the prior dataset, the model accuracy of correct localization and recognition could reach 81% [14]. Jacob Devlin et al. (2017) proposed the Share-Specialize-Compete (SSC) model, in which the AST was input into Recurrent Neural Networks (RNN) for error recognition of text information for the first time [15]. Sumit Gulwani et al. (2018) combined the clustering algorithm with the automatic repairing algorithm for the automatic repairing of program semantics; this method was advantageous on simple codes; however, as the codes became complex, the clustering classification tree would grow linearly, causing the system to run slowly [16]. The above research reveals that deep learning and neural networks can learn the various features of the codes, thereby significantly improving the accuracy and effect of program repair. Hence, they are highly feasible and practical. The focus here is to analyze and localize semantic errors in programs. Deep learning neural networks and Quantum Intelligence Algorithms (QIA) are adopted to encode information hidden in the codes. On this basis, a localization model for semantic program errors is proposed based on deep neural QIA, whose accuracy is tested and verified in the OJ system. Furthermore, after AST is encoded with Gated Recurrent Units (GRU)--based Graph Neural Network (GNN) (GGNN), the codes are input into a deep neural network model, and the program is repaired. The accuracy of such a method is also verified. The results can provide a theoretical basis for the research on semantic recognition of computer programs.

2. Methods

2.1. Traditional Semantic Localization and Repairing

Traditional methods for the localization and repair of program semantics include rule-based, corpus-based, and feature-learning-based methods. A typical flowchart is shown in Figure 1. First, according to the original program, the program will follow the error localization method and give the algorithm rules. Each bugged statement in the program code is calculated to obtain the probability that each statement has a bug. Then, the statements are ordered according to the likelihood. The ordered statements are successively input into the given patch generation algorithm. The patch generation algorithm will determine whether the patch can be output here based on the bugs of the statements. If the patch can be output, the wrong program code will be repaired according to the patch, and the test set will be used for patch verification. If the test cases in the test set are verified, the repairing plan is output for bug repairing. If the statement here cannot output a suitable repairing plan, the following statement with a bug will be chosen from the previously generated bug sequence, and another error repairing plan for the program will be generated again [17].

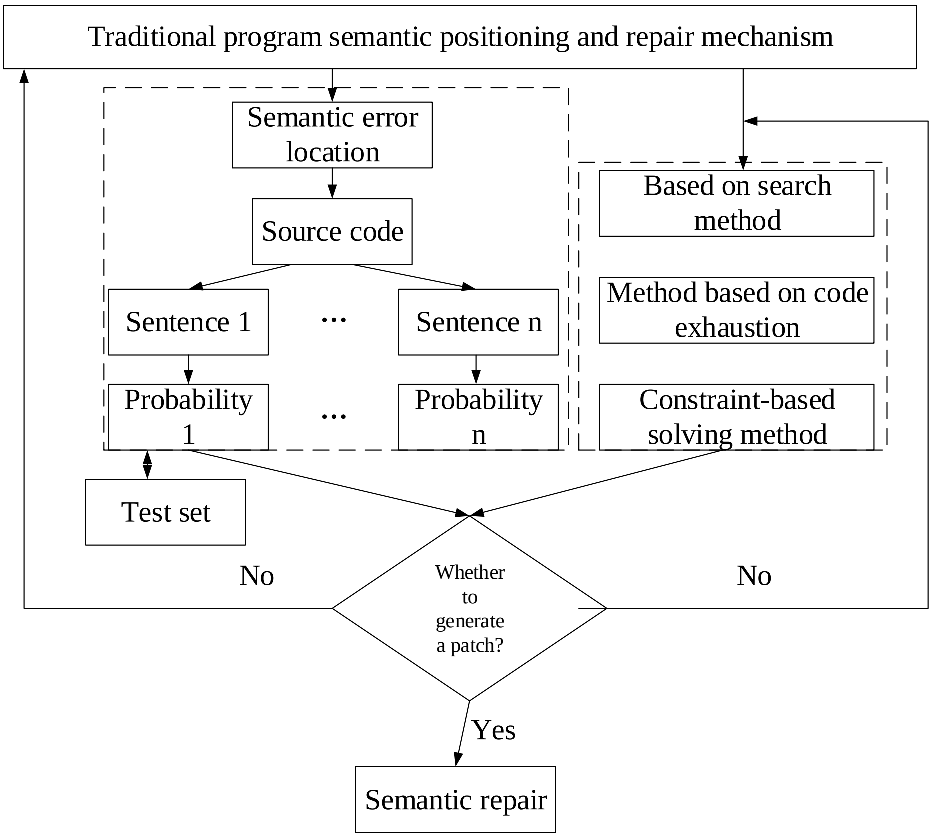

Figure 1. Traditional semantic localization and repairing.

2.2. Semantic Localization Algorithm

Deep Neural Network (DNN) is an Artificial Neutral Network (ANN) with multiple layers between the input and output layers. Despite the linear and nonlinear relationships, DNN will find the correct mathematical operation to convert the input to the output [18]. Layers in DNN are divided into three categories: the input layer, the hidden layer, and the output layer, as shown in Figure 2. In DNN, the calculation of a neuron is as follows:

$$a_j^l = \sigma(\sum_k w_{jk}^l a_k^{l-1} + b_j^l) \tag{1}$$

In Eq. (1), $w_{jk}^{l}$is the weight value of data, l represents the number of layers, j represents the j-th neuron, k represents the k-th neuron, a_j^l is the weight of a neuron, and b_j^l represents bias. This neural network adopts the gradient descent algorithm to optimize the value of a single parameter. The update calculation for parameters is as follows:

$$\theta = \theta_1 \text{-} \eta \Delta J(\theta) \tag{2}$$

In Eq. (2), θ_1 is the parameter of the last neuron, $J(\theta)$ is the loss function, and η is the learning rate. When the algorithm is propagated back, the chain rule of derivatives is utilized. Then, the derivative of a point (x, y) can be obtained:

$$\frac{\mathrm{dz}}{dx} = \frac{dz \cdot dy}{dy \cdot dx} \tag{3}$$

This point is generalized to vector form:

$$\Delta_x{}^z = (\frac{\partial y}{\partial x})^T \Delta_y{}^z \tag{4}$$

In Eq. (4), $\Delta_x{}^z$ and $\Delta_y{}^z$ are the vector values of point (x, y), z is the propagation distance, and T is the propagation time. Through the chain rule, the imperial capital of each layer can be reversed continuously from the final loss function, and a minor storage cost can be adopted in exchange for an increase in speed. The commonly used loss function is the Mean Square Error (MSE) loss function. The lower the value of this function is, the better the model fits in the training set and the better the effect is. The equation of the MSE loss function is:

$$J = \frac{1}{N} \sum_{i=1}^{N} (Y_i - y_\theta(x_i))^2 \tag{5}$$

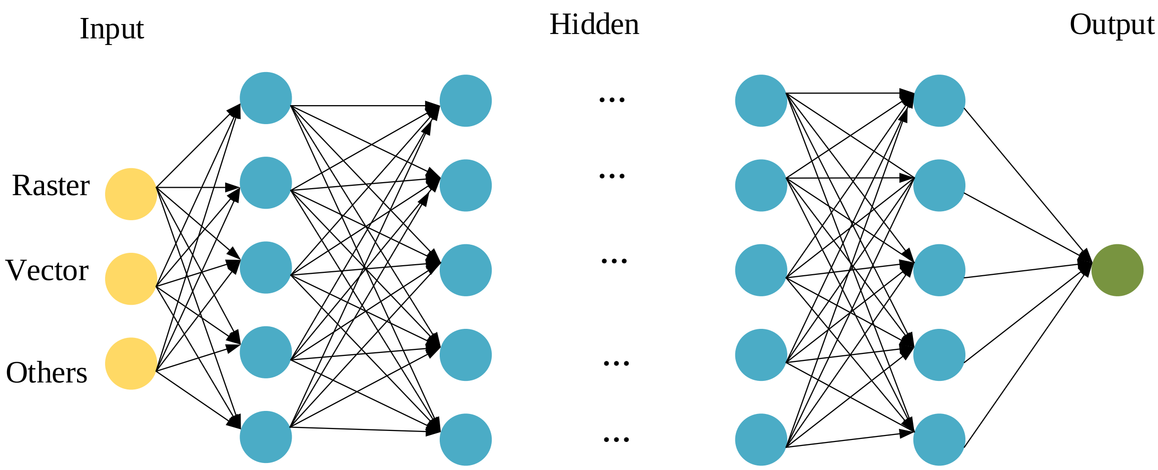

Figure 2. Structure of DNN.

To verify the superiority of the proposed method, different neural networks are tested for comparison, including the Convolution Neural Network (CNN), an efficient recognition algorithm developed recently and has attracted widespread attention. The basic structure of CNN includes two layers. The first is the feature extraction layer. The input of each neuron is connected to the local receptive field of the previous layer, and the local features are extracted. The second is the feature mapping layer. Each computing layer of the network consists of multiple feature maps. Each feature map is a plane, and the weights of all neurons on the plane are equal. The feature mapping structure uses a Sigmoid function with a small influencing function core as the activation function of CNN; thus, the feature mapping has displacement invariance. Besides, since neurons on a mapping surface share weights, the free parameters of the network are reduced. A Recurrent Neural Network (RNN) is a recursive network that takes sequence data as input, recurses in the evolution direction of the sequence, and connects all nodes in a chain. RNN can process the connections among multiple inputs to shorten the processing time [19].

Attention mechanism: It allows the model to focus on, learn, and absorb vital information, which can be applied in any sequence model. As for machine translation, if the sentence is long, a C may not hold so much information, which will cause the translation accuracy to decrease. Hence, to solve the above problem, the attention mechanism has come into being [20]. The specific structure is shown in Figure 3:

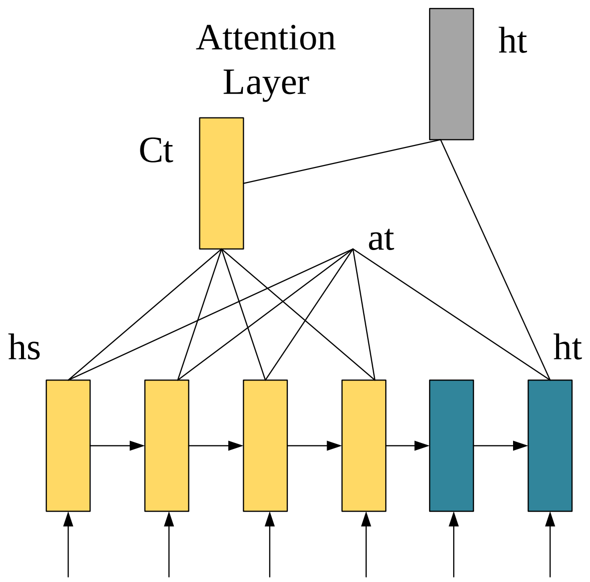

Figure 3. Structure of attention mechanism.

2.3. Semantic Repairing Algorithm

QIA: Since the proposal of the Shor and Grover algorithms, the unique calculation methods shown by quantum computing and the vast potential in information processing have attracted widespread attention from researchers. Intelligent algorithms have always been a hot topic in algorithm research. QIA combines quantum theory with intelligent computing and uses quantum parallelism to compensate for some of the shortcomings of intelligent algorithms, such as accelerating the convergence speed of the algorithm and avoiding premature [21]. Currently, common QIAs include quantum evolution algorithm, quantum immune algorithm, quantum annealing algorithm, quantum neural network, and quantum clustering algorithm. Quantum evolutionary algorithms and quantum neural networks have become current academic research hot spots. The neural network can extract the features and classify the algorithm but cannot modify the sentences. In contrast, QIA can optimize the performance of the original neural network and repair the wrong sentences. The unit vector of the 2D-bit space is:

$$|p >= a|0 > +b|1 > \tag{6}$$

In Eq. (6), a and b are all (x, y) values of 2D space, and p is the vector value. Its structure is shown in Figure 4:

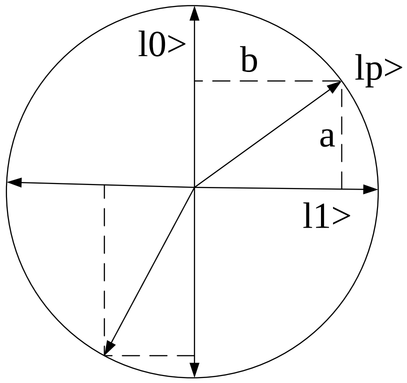

Figure 4. Computing structure of quantum state space.

In a quantum computer, n-bit quantum registers in a superposition state can simultaneously store all 2n numbers from 0 to 2n-l, which exist at the same time with a particular probability. Therefore, in a classical electronic computer, an n-bit register can only store one n-bit binary number; in a quantum computer, however, an n-bit quantum register can store 2n n-bit binary numbers simultaneously with a particular probability.

To verify the superiority of the proposed method, different neural networks are tested for comparison, including GGNN, a model of spatial information transmission based on GRU structure. A similar principle of RNN is adopted to transmit information among the graph. Compared to chain or tree structure data, such structures are often more flexible. Many traditional algorithms often compress the data of this structure into a chain structure or convert the data into a tree structure before adopting RNN for processing [22]. Genetic Algorithm (GA) is a search algorithm applied to solve optimization in computational mathematics, an evolutionary algorithm. It is usually realized through computer simulation. For an optimization problem, an exact number of candidate solutions can be abstractly represented as chromosomes so that the population can evolve to a better solution [23]. The fault localization method of the Genporn algorithm preprocesses each statement in the program code and executes the test cases to calculate the possibility that each statement has a bug. If the statement can only be executed using positive test cases, it is less likely to have bugs. If the semantics can only be executed using anti-test cases, the possibility of bugs is higher. The sk-p algorithm uses the Data-Driven Synthesis (DDS) mode to input the previous and subsequent lines of codes into the Seq neural network model. Then, it outputs the correct codes according to the distribution. This model can fix syntax and semantic errors simultaneously, but each line of the code needs to be predicted. As a result, the calculation amount is tremendous. Besides, this model is simple and uses less information hidden in the code segment; hence, the accuracy is low [24]. The specific process is shown in Figure 5.

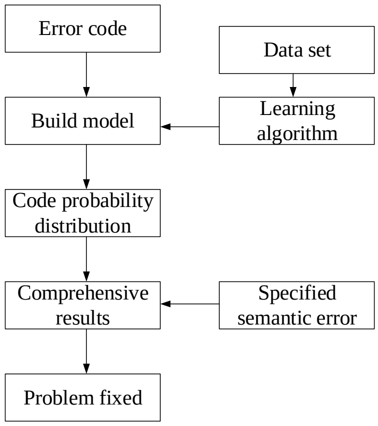

Figure 5. Structure of sk-p algorithm.

2.4. Construction of the Localization-Repairing Model for Semantic Errors Based on Deep Quantum Neural Network

The traditional semantic localization and repairing models have poor accuracy because their feature extraction ability is weak. Besides, the running algorithm does not employ a distributed network, so the requirements for equipment are higher. With the continuous development of technology, a semantic positioning and repairing model based on the deep quantum neural network is proposed. This model combines DNN and attention mechanisms to extract features of codes. First, the codes are segmented into lines. Then, each token in each row is encoded with the C Embedding method. After the encoding is completed, the RNN model inputs each token in each line of code and performs only the Encoder operation. As a result, the context vector, which hides the information of this line of code, is obtained. After the context vector of each line of code is obtained, these vectors are input into the pointer network model as a sequence. Then, the location of the line where the semantic error of the program code occurs is output. Hence, the semantic error localization of programs is achieved. It also uses a quantum neural network to optimize the algorithm and repair the semantics. The specific structure is shown in Figure 6.

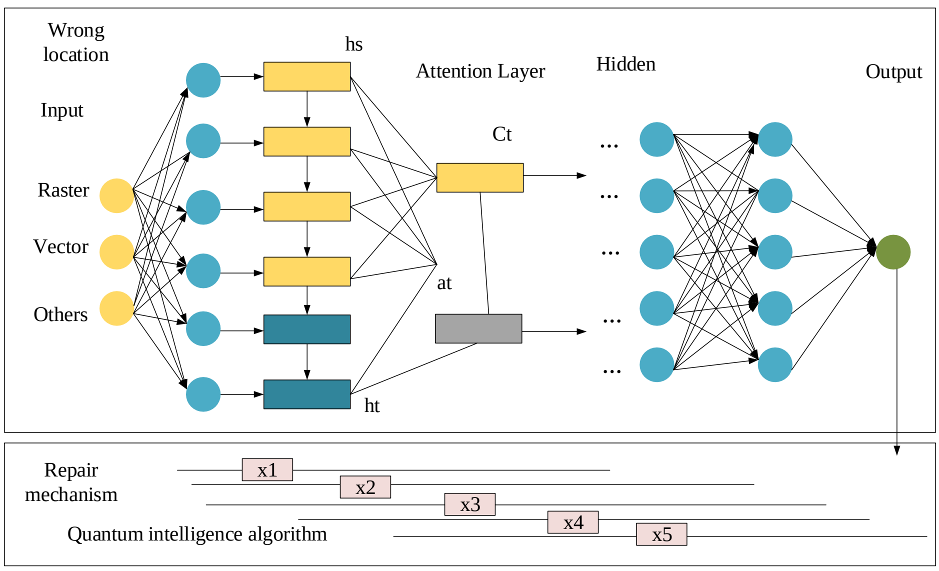

Figure 6. Localization-repairing model for semantic errors based on deep quantum neural network.

2.5. Data Collection and Model Performance Evaluation

(1) Data collection: in the experiment, crawlers are adopted to obtain the Python codes and test cases on the Codeforces platform of the OJ system. The correct codes that can pass all test cases and the error codes that can be compiled but failed to pass are collected as a dataset, with a total of 26,946 code segments. This dataset is divided into a training set, a validation set, and a test set. Of all the code segments, 20,524 are in the training set, 2,335 are in the validation set, and 2,354 are in the test set. It is ensured that the test focus and verification set codes have never appeared in the training set; hence, similar codes do not appear in the training set, verification set, and test set simultaneously, ensuring the accuracy of the model. There are five types of semantic errors in the generated conditional expressions (A1: adding wrong conditional expressions; A2: variable name errors; A3: operator errors; A4: deleting conditional expressions; A5: operator and variable name errors simultaneously). These are errors that programmers often make in the coding process. The random methods are utilized to generate errors; that is, the error of each sentence is entirely random. The distribution of error types generated is shown in Table 1.

Table 1. Number of various error semantic datasets.

Error type	A1	A2	A3	A4	A5
Number of collected data	7842	2132	13860	1002	2110

(2) Performance evaluation: to effectively evaluate the performance of the proposed model, Accuracy, Precision, Recall, and F-Measure are adopted as the experimental evaluation criteria. Accuracy represents the proportion of the semantic errors the system can identify to the total semantic errors. It is a crucial indicator to measure the accuracy of model predictions [25]. Precision indicates the proportion of positive classes in the samples identified as positive [26]. Recall indicates the proportion of all positive samples correctly identified as positive classes [27]. F-measure is the weighted harmonic average of Precision and Recall, often adopted to evaluate the quality of classification models [28]. The calculation is as follows:

$$Accuracy = \frac{A+B}{A+B+C+D} \tag{7}$$

$$Pr\,e\,cision = \frac{A}{A+B} \tag{8}$$

$$Re\,c\,\mathrm{all} = \frac{B}{B+D} \tag{9}$$

$$F - measure = \frac{2 * Pr\,ecision * Re\,call}{Pr\,ecision + Re\,call} \tag{10}$$

In Eq. (7) - Eq. (10), A is the number of correct positive semantics recognized by the model, B is the number of correct negative semantics recognized by the model, C is the number of unrecognized false-positive semantics, and D is the number of unrecognized false-negative semantics.

3. Results and Discussions

3.1. Comparison Results of Semantic Localization Algorithms

Figure 7 shows the performance comparison of semantic localization algorithms. A comparison of RNN, CNN, and DNN shows that the accuracy, recall, and comprehensive evaluation performance differences are not significantly different. After the attention mechanism is introduced, the performance of the models is much improved, which shows that the attention mechanism has good semantic discrimination ability. Besides, the A-DNN algorithm based on deep neural networks exhibits the best performance. Compared to the algorithms without attention mechanism, the accuracy increases by 21.91%; compared to A-RNN, the accuracy increases by 9.14%; compared to A-CNN, it increases by 12.89%. The above results show that the proposed semantic localization model not only accurately locates the error segment but also dramatically improves the model's performance.

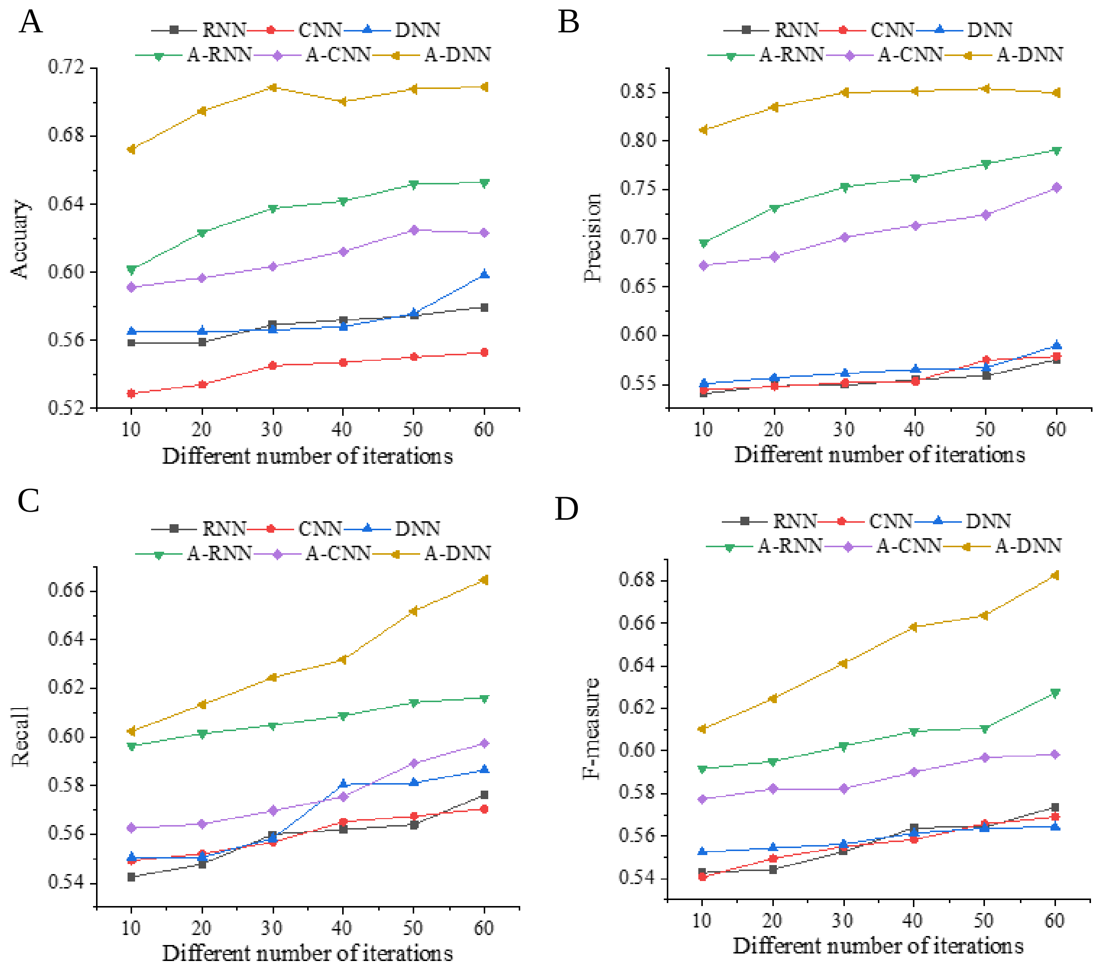

Figure 7. Performance comparison results of semantic localization algorithms.

Note: A-RNN, A-CNN, and A-DNN indicate adding the attention mechanism to the three neural networks.

3.2. Comparison Results of Semantic Repairing Algorithms

Figure 8 shows the comparison results of semantic repairing algorithms. As the number of training increases, the model's performance continues to improve. A comparison of all algorithms reveals that if QIA is adopted, the model performance is improved, and the performance of GGNN is also increased. Genporn has the worst performance, and the performance difference between various algorithms is significant. QIA has the best performance. Compared to the Genporn algorithm, QIA improves the model accuracy by 32.25, and the highest semantic repairing accuracy can reach 85.24%. The above results show that introducing QIA into semantic repairing is booming, and the model performance is better.

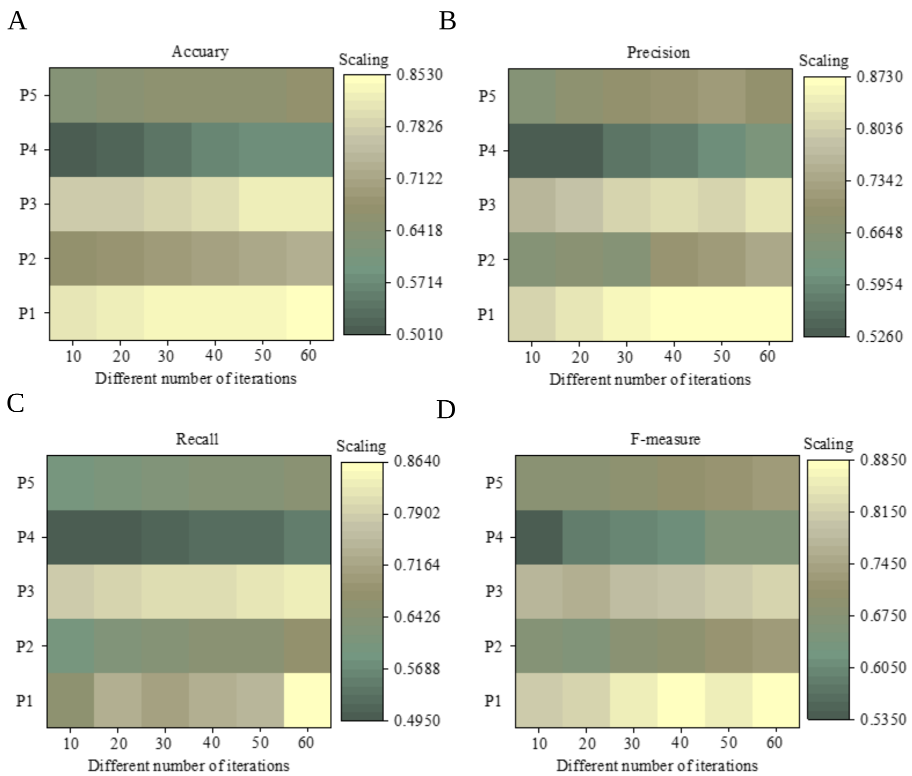

Figure 8. Result comparison of semantic repairing algorithms.

Note: P1-P5 are QIA, GA, GGNN, Genporn, and sk-p algorithm models, respectively.

Figure 9 illustrates the convergence performance of the semantic repairing algorithms. Models without QIA will converge when the dataset is 300, while models with QIA can converge in advance when the dataset is 200. Therefore, the performance loss can be reduced, and the result is consistent with Accuracy and Precision.

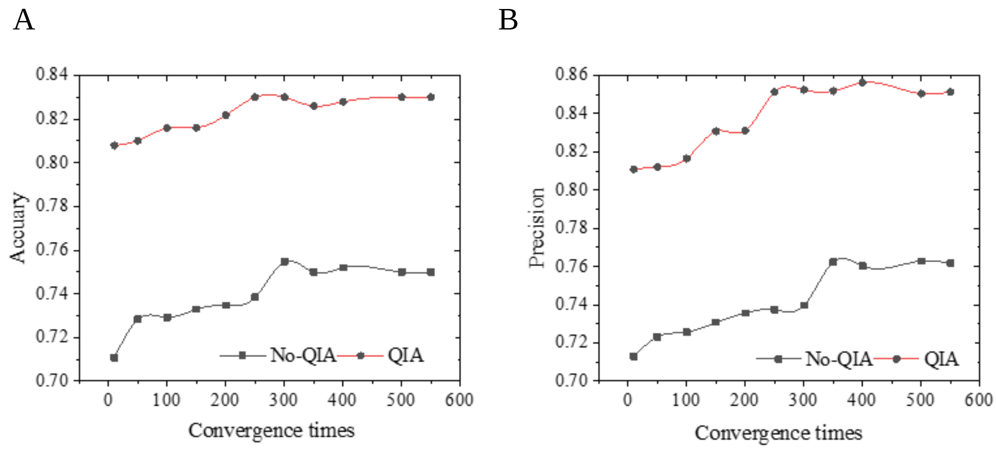

Figure 9. Comparison of convergence performance of semantic repairing algorithms.

3.3. Performance Evaluation of Semantic Localization and Repairing System Based on Deep Quantum Neural Network

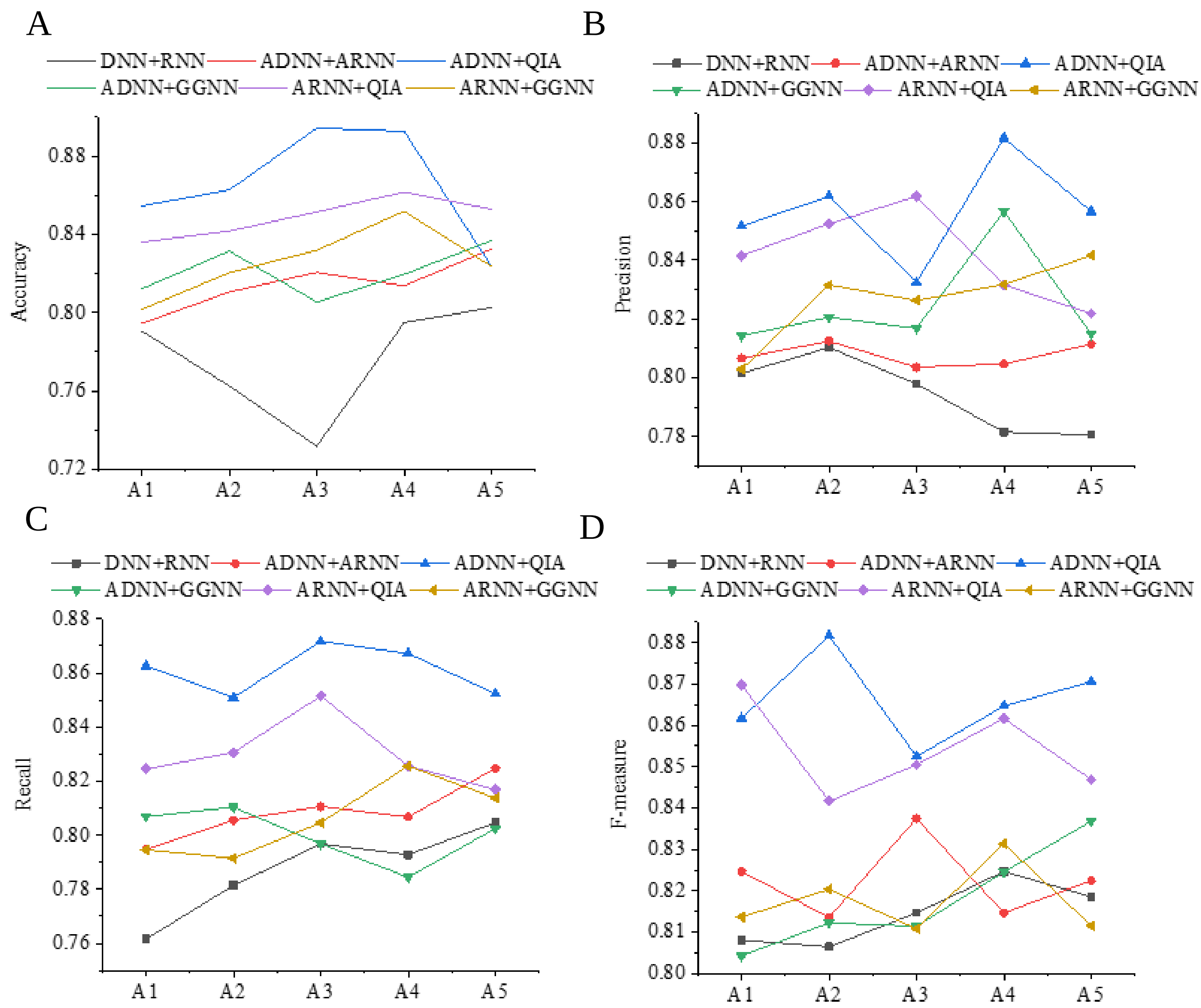

Figure 10. Performance evaluation of semantic localization and repairing system based on deep quantum neural network.

Figure 10 shows the performance evaluation of semantic localization and repairing systems based on the deep quantum neural network. A combination of DNN and RNN is employed, which has improved performance compared to single models. Similar to the results of introducing the attention mechanism, the comprehensive performance of DNN and RNN has increased notably. However, the experiment reveals that such a combination fails to fully utilize the intrinsic performance of DNN and RNN. The performance is improved based on a massive amount of calculations. Hence, such a combination is not adopted. Another two model combinations are tested for comparison. The performance of the ADNN+QIA algorithm is the best, and the highest accuracy is 89.27%, which is 0.89% higher than that of ARNN+QIA. The difference is insignificant; however, the overall performance is improved by 1.4%. The above results show that the semantic localization and repairing model based on deep quantum neural network and QIA is useful. This model can significantly improve the localization of program semantic errors and realize automatic, efficient, and fast error-repairing.

4. Conclusion

The shortcomings in the current localization and repairing methods of semantic errors in programs are analyzed. DNN and attention mechanism are introduced to improve the precision of localizing program semantics. The system model is optimized through QIA, which can automatically repair the semantic program errors. Consequently, several algorithms are combined to construct a localization and repairing model for semantic errors based on a deep quantum neural network. The proposed model can precisely localize semantic errors in the OJ system. QIA significantly improves the convergence and repair capabilities of the algorithm. The research results can provide a theoretical reference for program error localization and repair. Although the issues described are more comprehensive, due to objective limitations, some deficiencies are found. First, the number of datasets is insufficient. Data collection is only performed on the OJ system; hence, resources are limited, which will influence the model's accuracy. Second, the types of errors identified by the proposed algorithm are not rich enough, including common errors. An error corpus should be established, supplemented, and improved through continuous learning of algorithms. Finally, the repairing algorithms all aim at sentences, which will undoubtedly increase the operating load of the system. Later, the neural networks of physical sign extraction can put the repaired sentence only in the error location. In the future, in-depth research on these aspects will be conducted to find a better combination of program semantic repairing.

References

[1] Shank DB, Graves C, Gott A, Gamez P, Rodriguez S. Feeling our way to machine minds: People's emotions when perceiving mind in artificial intelligence. Computers in Human Behavior. 2019, 98, pp. 256-266.

[2] Faisal FE, Newaz K, Chaney JL, Li J, Emrich SJ, Clark PL, Milenković T. GRAFENE: Graphlet-based alignment-free network approach integrates 3D structural and sequence (residue order) data to improve protein structural comparison. Scientific reports. 2017, 7(1), pp. 1-15.

[3] Wongranu P. Errors in translation made by English major students: A study on types and causes. Kasetsart Journal of Social Sciences. 2017, 38(2), pp. 117-122.

[4] Wang K, Singh R, Su Z. Dynamic neural program embedding for program repair. arXiv preprint arXiv:171107163. 2017, pp. 36-41.

[5] Schwartz MF, Middleton EL, Brecher A, Gagliardi M, Garvey K. Does naming accuracy improve through self-monitoring of errors? Neuropsychologia. 2016, 84, pp. 272-281.

[6] Xie G, Shen J, Yang H. Generalized RDP Code Based Concurrent Repair Scheme in Cloud Storage Nodes. Information. 2019, 10(1), pp. 20-28.

[7] Lyubetsky V, Gershgorin R, Gorbunov K. Chromosome structures: reduction of certain problems with unequal gene content and gene paralogs to integer linear programming. BMC bioinformatics. 2017, 18(1), pp. 537-542.

[8] Yu W, Gao F, Wang L, Yu T, Zhao J, Li X. Automatic Detection, Validation and Repair of Race Conditions in Interrupt-Driven Embedded Software. IEEE Computer Architecture Letters. 2020, (01), pp. 1-9.

[9] Yang C, Yin XC, Pei WY, Tian S, Zuo ZY, Zhu C, Yan J. Tracking based multi-orientation scene text detection: A unified framework with dynamic programming. IEEE Transactions on Image Processing. 2017, 26(7), pp. 3235-3248.

[10] Moghimi M, Varjani AY. New rule-based phishing detection method. Expert systems with applications. 2016, 53, pp. 231-242.

[11] Li Q, Li S, Zhang S, Hu J, Hu J. A Review of Text Corpus-Based Tourism Big Data Mining. Applied Sciences. 2019, 9(16), pp. 3300-3309.

[12] Qin X, Jiang J, Yuan C-A, Qiao S, Fan W. Arbitrary Shape Natural Scene Text Detection Method Based on Soft Attention Mechanism and Dilated Convolution. IEEE Access. 2020, 8, pp. 122685-122694.

[13] Devlin J, Uesato J, Singh R, Kohli P. Semantic code repair using neuro-symbolic transformation networks. arXiv preprint arXiv:171011054. 2017, pp. 1325-1330.

[14] Mohan VT, Jannesari A. Automatic Repair and Type Binding of Undeclared Variables using Neural Networks. arXiv preprint arXiv:190706205. 2019, pp. 368-374.

[15] Devlin J. Sharp models on dull hardware: Fast and accurate neural machine translation decoding on the cpu. arXiv preprint arXiv:170501991. 2017, pp. 875-882.

[16] Gulwani S, Radiček I, Zuleger F. Automated clustering and program repair for introductory programming assignments. ACM SIGPLAN Notices. 2018, 53(4), pp. 465-480.

[17] Xu T, Chen L, Pei Y, Zhang T, Pan M, Furia CA. Restore: Retrospective fault localization enhancing automated program repair. IEEE Transactions on Software Engineering. 2020, pp. 39-43.

[18] Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE. A survey of deep neural network architectures and their applications. Neurocomputing. 2017, 234, pp. 11-26.

[19] Kong W, Dong ZY, Jia Y, Hill DJ, Xu Y, Zhang Y. Short-term residential load forecasting based on LSTM recurrent neural network. IEEE Transactions on Smart Grid. 2017, 10(1), pp. 841-851.

[20] Yan C, Tu Y, Wang X, Zhang Y, Hao X, Zhang Y, Dai Q. Stat: spatial-temporal attention mechanism for video captioning. IEEE transactions on multimedia. 2019, 22(1), pp. 229-241.

[21] Zouache D, Abdelaziz FB. A cooperative swarm intelligence algorithm based on quantum-inspired and rough sets for feature selection. Computers & Industrial Engineering. 2018, 115, pp. 26-36.

[22] Sorokin D, Gurevych I. Modeling semantics with gated graph neural networks for knowledge base question answering. arXiv preprint arXiv:180804126. 2018, pp. 812-823.

[23] Metawa N, Hassan MK, Elhoseny M. Genetic algorithm based model for optimizing bank lending decisions. Expert Systems with Applications. 2017, 80, pp. 75-82.

[24] Bellavia S, Gurioli G, Morini B, Toint PL. Adaptive regularization algorithms with inexact evaluations for nonconvex optimization. SIAM Journal on Optimization. 2019, 29(4), pp. 2881-915.

[25] Wang Q, Wang J, Zhao P, Kang J, Yan F, Du C. Correlation between the model accuracy and model-based SOC estimation. Electrochimica Acta. 2017, 228, pp. 146-159.

[26] Mets KD, Armenteras D, Dávalos LM. Spatial autocorrelation reduces model precision and predictive power in deforestation analyses. Ecosphere. 2017, 8(5), pp. e01824-e01831.

[27] Poll S, Mittag M, Musacchio F, Justus LC, Giovannetti EA, Steffen J, Wagner J, Zohren L, Schoch S, Schmidt B. Memory trace interference impairs recall in a mouse model of Alzheimer's disease. Nature Neuroscience. 2020, pp. 1-7.

[28] Hung DN, Ha HTV, Binh DT. Application of F-score in predicting fraud, errors: Experimental research in Vietnam. International Journal of Accounting and Financial Reporting. 2017, 7(2), pp. 303-322.