International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 02 | Feb 2021 www.irjet.net p-ISSN: 2395-0072

ANALYSIS OF AI BASED EDUCATIONAL TECHNOLOGICAL TOOLS & ITS IMPORTANCE

Divyansh Rana [1], Dr. Ashad Ullah Qureshi [2]

[1] *Research Scholar, UIT, Kurukshetra University*

[2] *Technical Officer, NIT, Kurukshetra*

Abstract – This study aims at comparison of the new age technologies used in the education field for the betterment of the school going students.

This research study focuses on the effectiveness of these tools and software used frequently in the Indian school systems nowadays.

Furthermore, the research asses the various parameters and their effect on students was assessed from both educational and technological perspectives.

Keywords: Artificial Intelligence, Ed-Tech Tools, Cloud Computing, etc.

1. INTRODUCTION

Edtech companies are using artificial intelligence and machine learning to solve many issues in education. Here we look at different ways some prominent Edtech companies are applying these technologies to develop their solutions.

Elemental Path

This is a super smart company making super smart toys for kids. When Elemental Path debuted on Kickstarter in 2016, it was with the help of IBM Watson, which enabled the company to make toys that kids can communicate with.

IBM Watson combines machine learning, natural language processing, computer vision, data mining, and human/computer interaction to mimic the workings of the human brain. The toy company, however, has switched to its own proprietary software for its new dino robot, the STEMosaur, that kids can build and talk to.

Querium Corporation

Querium uses AI in their company to help students with STEM skills so they can be ready for further studies at high schools, colleges, and universities.

The Querium platform delivers personalized, bite-sized lessons and step-by-step tutoring assistance. Its personalized program is called StepWise and it works on smartphones and computers.

The company uses artificial intelligence for STEM (math, engineering, technology, and science education) applications. The app's cognitive assistant helps students as they come across obstacles.

MobyMax

MobyMax is a free web-based learning tool that uses artificial intelligence to create a complete K-8 differentiated and adaptive curriculum for Math, Language, Vocab, and Fact Fluency. With the help of machine learning algorithms, MobyMax provides personalized lessons for each student based on their individual needs.

Hubert.ai

Hubert.ai is an award-winning EdTech company from Sweden. Hubert.ai brings something truly unique to the assessment system in education. The present assessment system can't evaluate important student skills and abilities such as creativity, imagination, ethical reflection, background reasoning and so on.
Hubert.ai can.

Hubert.ai has developed a cognitive computing assistant that can take test-taking a step further. The cognitive assistant effectively turns surveys into speaking tests by asking students follow-up questions. This effectively personalizes assessment as well.

Discovery Education

Discovery Education is a big player in EdTech. The company has established itself as the leading provider of digital curriculum resources, digital content, and professional development for K-12 classrooms. Around 5.6 million students across all 50 states and Canada are

currently using Discovery Education. These resources add up to a truly dynamic learning environment that engages students. With these innovations, the days of sitting behind a desk and trying to pay attention to what the teacher is saying are something of the past.

2. LITERATURE SURVEY

[1] Dr. *S.* Bennett, Professor D.A. Linkens, Dr. E.B. Tanyi and Mr. A. Scott ,"APPLICATION OF *AI* AND MODEL BUILDING TECHNIQUES TO SOFTWARE ENGINEERING " IEEE Xplore.

The fundamental ideas in KEMS arise from consideration of methodologies for applying AI to the construction of models and hence we have also considered how the techniques can be applied to model-based software engineering methodologies for specification and design of software.

[2] Derek Partridge, "THE RELATIONSHIPS OF AI TO SOFTWARE ENGINEERING ", University of Exeter.

As this last talk is supposed to provide a summary and overview of the day's proceedings, I can, with a clear conscience, excuse myself from providing the text of my talk before the event has taken place. However, I can provide a framework for the interactions of artificial intelligence (AI) and software engineering (SE) - mathematicians begin with definitions, computer scientists (especially AIoriented ones) begin by setting out the basic acronyms. And 1 can also provide some pointers into the growing conglomeration of literature that infringes on this area of overlap.

[3] Andrew Begel, "Best Practices for Engineering AI-Infused Applications: Lessons Learned from Microsoft Teams", 2019 IEEE/ACM Joint 7th International Workshop on Conducting Empirical Studies in Industry (CESI) and 6th International Workshop on Software Engineering Research and Industrial Practice (SER&IP)

Artificial intelligence and machine learning (AI/ML) are some of the newest trends to hit the software industry, compelling organizations to evolve their development processes to deliver novel products to their customers. In this talk, I describe a study in which we learned how Microsoft software teams develop AI/ML-based applications using a nine-stage AI workflow process informed by prior experiences developing early AI applications (e.g. search and NLP) and data science tools (e.g. application telemetry and bug reporting). Adapting this workflow into their pre-existing, well-evolved, Agile-like software engineering processes and job roles has resulted in a number of engineering challenges unique to the AI/ML domain, some universal to all teams, but others related to the amount of prior AI/ML experience and education the teams have. I tell you about some challenges and the solutions that teams have come up with. The lessons that Microsoft has learned can help other organizations embarking on their own path towards AI and ML

[4] CHUANQI TAO, JERRY GAO AND TIEXIN WANG, "Testing and Quality Validation for AI Software–Perspectives, Issues, and Practices" IEEE Access 2019, SPECIAL SECTION ON INNOVATION AND APPLICATION OF INTELLIGENT PROCESSING OF DATA, INFORMATION AND KNOWLEDGE AS RESOURCES IN EDGE COMPUTING

With the fast growth of artificial intelligence and big data computing technologies, more and more software service systems have been developed using diverse machine learning models and technologies to make business and intelligent decisions based on their multimedia input to achieve intelligent features, such as image recognition, recommendation, decision making, prediction, etc. Nevertheless, there are increasing quality problems resulting in erroneous testing costs in enterprises and businesses. Existing work seldom discusses how to perform testing and quality validation for AI software. This paper focuses on quality validation for AI software function features. The paper provides our understanding of AI software testing for new features and requirements. In addition, current AI software testing categories are presented and different testing approaches are discussed. Moreover, test quality assessment and criteria analysis are illustrated. Furthermore, a practical study on quality validation for an image recognition system is performed through a metamorphic testing method. Study results show the feasibility and effectiveness of the approach

[5] Aarthi Prasad,E. K. Park "AI-based Classification and Retrieval of Reusable Software Components" IEEE Xplore 2019

The concept of software reusability is examined from the perspective of classifying and accessing reusable software. To improve the practicality of software reuse, one has to have a knowledge of its location, an understanding of the reusable components, and its adaptability to a particular need. The current state of-the-art methods are assessed, and a new system for performing classification-based reuse is proposed. The concepts of subsumption and closeness introduced in are used by the proposed reuse system to facilitate searching for reusable components and to provide capabilities for helping programmers to assess the worth of reusing particular packages. A description of the major algorithms required to compute these metrics is given. Also, domain analysis which helps in deciding whether a particular application is oriented towards software reuse is given.

3. RESEARCH METHODOLOGY

The method proposed in this paper mainly includes three parts, and they will be discussed in detail in this section.

A. Perspective: Network Visualization and Architecture Evaluation

This section aims to abstract the software structure into a network, with elements as nodes and relationships as edges, to evaluate the overall structural quality of software. A software network is a visual network diagram in which the software modules and the relationships between the modules are abstracted as nodes and edges based on the complex network theory. A large number of studies have shown that the software network conforms to complex network characteristics, such as "small world" and "scaleless" characteristics [1][2]. Therefore, this paper uses various parameters in complex networks to evaluate the reliability level of software. Human skeletal structure determines the strength of a person; thus, the structural characteristics of the software network can reflect the quality of the software to a certain extent. This section measures the quality of the software structure from four aspects: modularity, hierarchy, complexity, and fault propagation [3]. A better software modularity means a higher degree of cohesion of the program code structure and a lower degree of coupling.

The better the hierarchy is, the more obvious the hierarchy of the code is and the higher the stability of the software structure is. The more complex the software structure is, the more confused and bloated the internal code structure is. Fault propagation is the evaluation of the ability of faults to propagate in software. A higher fault propagation means that a single node has greater influences and impacts on software, which causes faults to spread more easily in the software. This method uses complex network parameters to calculate the values of the above four aspects of a software, and, for software with different functions and operations under different environments, the weights of the four aspects should be different to evaluate the quality of the software structure. According to different types of software, a large amount of well-known, well-structured and mature open source software is selected as the standard library to obtain the reference threshold of each parameter to relatively evaluate the structural quality level of a new software in the same type of software simultaneously.

B. Important Module Identification and Module Reliability Risk Assessment

The pathological change in vital organs can affect the human body more seriously. Therefore, this section aims to analyze the characteristics of software modules, taking into account the importance of the module as well as the reliability risks and to evaluate the overall reliability risk of software.

In this part, the software module is taken as the research object; thus, the internal elements of the software module are taken as nodes, and the relationship between the elements is taken as the edge to construct the software module subnetwork.

The evaluation of the importance of software modules is to find the key modules in the software, which have profound influences globally or locally [4]. For the global aspect, class location and the ability of the class to control the information flow of software are mainly considered. In contrast, the local aspect focuses on the interactions between classes and their neighbors as well as the complexity of the class itself. Modules are sorted according to the comprehension of global and local structural importance with a certain weight. The top ranked modules have a relatively important position in software in whole or in parts, so more attention should be paid to improve the reliability of these modules and, further, the software. Module reliability risk assessment comprehensively considers the complexity of the module itself and the historical changes in the module to measure the risk that the module may introduce defects. Module complexity includes structural complexity, code complexity and interface complexity, while the module change rate considers the frequency of changes in the module during its lifetime. The more complex the module, the more frequent the change is, and the more likely it is to introduce defects. Combining the above two points, identifying key modules in the software and measuring the reliability risks of these important modules, the reliability risk of the key modules and software can be evaluated. Using different module partitioning methods, the importance of different types of modules, such as class, file and package, can be ranked as well as their reliability risks. The reliability risk of the software can be assessed through multiple dimensions.

C. Case Analysis: High Risk Module Identification

The historical disease condition of a human has an important reference value to the current health evaluation of the human body. Therefore, this section aims to analyze the characteristics of the nodes that have discovered defects in the history of software development to identify the modules with high defect risks in the current version.

First, the defect information of the software historical versions is collected and sorted. Then, taking the parameters of the software modules and their defect conditions as input, machine learning methods are used to predict the risk of defects in each module of the current version to find the module that is most likely to have defects, and more attention should be paid to improve the quality of the software system more efficiently.

As software is highly individualized, various machine learning methods are used for predictions. The results of

all machine learning methods are analyzed together to find the best prediction result

.

4. CONCLUSION

In order to meet the different needs of the diverse fields and enterprises, reconfigurable application System must provide an adequate number of functional components. However, to rapidly find suited components from component libraries whose functions are distinct, the effective algorithm of search and matching is all-important, while the definition of the algorithm of search and match depends on the method of component description. It solves the key problems of description, classification and storage of components to take AI frame representation as the method of components description.

The method mainly evaluates the software reliability level through the overall software structural quality, reliability risk and defect risk. From a plurality of angles, the qualitative evaluation of the factors that affect software reliability is given, as well as the weaknesses that affect software reliability, which can guide software improvement.

This paper provides perspective views on AI software validation, including the tutorial concepts, test features and focuses, as well as validation process and test modeling for AI classification function testing. Moreover, the primary types of AI software testing and existing validation approaches are analyzed and discussed. Moreover, the paper also discusses The primary challenges, issues and needs are presented in the end.

REFERENCES

I. Dr. *S.* Bennett, Professor D.A. Linkens, Dr. E.B. Tanyi and Mr. A. Scott ,"APPLICATION OF *AI* AND MODEL BUILDING TECHNIQUES TO SOFTWARE ENGINEERING " IEEE Xplore.

II. Derek Partridge, "THE RELATIONSHIPS OF AI TO SOFTWARE ENGINEERING ", University of Exeter.

III. Andrew Begel, "Best Practices for Engineering AI-Infused Applications: Lessons Learned from Microsoft Teams", 2019 IEEE/ACM Joint 7th International Workshop on Conducting Empirical Studies in Industry (CESI) and 6th International Workshop on Software Engineering Research and Industrial Practice (SER&IP)

IV. CHUANQI TAO, JERRY GAO AND TIEXIN WANG, "Testing and Quality Validation for AI Software–Perspectives, Issues, and Practices" IEEE Access 2019, SPECIAL SECTION ON INNOVATION AND APPLICATION OF INTELLIGENT PROCESSING OF DATA, INFORMATION AND KNOWLEDGE AS RESOURCES IN EDGE COMPUTING

V. Aarthi Prasad,E. K. Park "AI-based Classification and Retrieval of Reusable Software Components" IEEE Xplore 2019

VI. Lan-Hsin Wang1, Fu-Ren Hsiao1, Yu-Xuan Zhang2 , and Tsung-Wen Chiu2 1 ,"AI-based antenna technology for 5G software defined radio platform" , Advanced Wireless and Antenna Inc.

VII. Jerry Gao③, Chuanqi Tao①②, Dou Jie Í, Shenqiang LuÍ, "What is AI Software Testing? and Why" 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)

AUTHOR'S PROFILES

Dr. Ashad Ullah Qureshi Ph.D, M.Tech, B.Tech, BEC Cambridge University, who received his Ph.D from OPJS University, Jhunjhunu, Rajasthan & M.Tech from Rajiv Gandhi Technical University Bhopal. He has contributed extensively to the betterment of students with his research & development. Ile has published many research papers in International Journals. He is the editor of "Creative Musings of an Army wife", a unique out of waste coffee table book. He is recognized nationally as an elite academician in the field of computer science & engineering. In 2014 invented Student Welfare Smart Machine a Student KIOSK for printing Solution. He is the recipient of the 2011 Young Scientist Award from Ministry of Electronics & Information Technology, New Delhi.

Mr. Divyansh Rana completed his Bachelor of Technology in Electronics and Communication Engineering from the University Institute of Engineering and Technology, Kurukshetra University in 2019. For the last 2+ years, he has been working as a freelance Full-Stack Developer and delivered successful projects to his clients from all over the world. Being a research enthusiast, his main research work focuses on Artificial Intelligence, Machine Learning, Cloud Computing, and Advanced Algorithms.