INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor : 5.725

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 09, September 2023, pp : 127-130

SYSTEMATIC MAPPING: ARTIFICIAL INTELLIGENCE TECHNIQUES IN SOFTWARE ENGINEERING

Mrs. S. Chandrakala[1], Ms. A. Eniya[2]

[1]Assistant Professor, Department Of MCA, Vivekanandha Institute of Information and Management Studies, Tiruchengode, Tamilnadu, India.

[2]Student, Department of MCA, Vivekanandha Institute of Information and Management Studies, Tiruchengode, Tamilnadu, India.

ABSTRACT

Artificial Intelligence (AI) has become a core feature of today's real-world applications, making it a trending topic within the software engineering (SE) community. The rise in the availability of AI techniques encompasses the capability to make rapid, automated, impactful decisions and predictions, leading to the adoption of AI techniques in SE. With industry revolution 4.0, the role of software engineering has become critical for developing productive, efficient, and quality software. Thus, there is a major need for AI techniques to be applied to enhance and improve the critical activities within the software engineering phases. Software is developed through intelligent software engineering phases. This paper concerns a systematic mapping study that aimed to characterize the publication landscape of AI techniques in softwe engineering. Gaps are identified and discussed by mapping these AI techniques against the SE phases to which they contributed. Many systematic mapping review papers have been produced only for a specific AI technique or a specific SE phase or activity. Hence, to our best of knowledge within the last decade, there is no systematic mapping review that has fully explored the overall trends in AI techniques and their application to all SE phases.

1. INTRODUCTION

As software products become pervasive in all areas of society, the productive building of high-quality software has become crucial to the software industry. The rise of artificial intelligence (AI) applications is potentially a game-changer in improving Software Engineering (SE) phases to ensure higher-quality software, accelerate productivity, and increase project success rates. AI has the capability to assist software teams in many aspects, from automating certain activities in an SE phase to providing project analytics and actionable recommendations, and even making decisions. AI techniques can support software engineers by detecting parts of the SE phases that are more likely to contain vulnerabilities and raising alerts about these issues. Such techniques can help to prioritize efforts and optimize inspection and testing costs.

2. CONCEPT

In general Several software practitioners claim that the performance of SFP models mostly depends on software metrics that we choose rather than modeling techniques, they also reported that selecting of prediction model offers a lesser impact over the performance on SFP. But according to a few latest articles the performance of the SFP model mostly depends on the prediction methods. There are few specific situations where deep learning (DL) based SFP can be employed, such as when the dataset doesn't have extracted features set and/or class labels. As DL algorithms require hyperparameters tuning of many control parameters, so they are implicitly complex in nature. Hyperparameters tunning and optimization mechanism are two critical aspects of the DL approaches, and if software practitioner is not properly taking care of these aspects, then the model may perform poorly than the ML approach, which is not recommended. DL architectures are hunger for training instances; the more the training set, the more can be the accuracy. The performance of SFP models depends on various factrs. We collected those factors and analyzed them accordingly

2.1 Objective

- A systematic mapping study that aimed to characterize the publication landscape of AI techniques in software engineering
- Gaps are identified and discussed by mapping these AI techniques against the SE phases to which they contributed.

2.2 Purpose of System

It will reduce the difficulty on the fault occur in motor while using in industry.

It will reduce the time consumption and increase the production growth.

By this application we can easily find out the problem facing on the frequently running industries for 24 hours.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor : 5.725

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 09, September 2023, pp : 127-130

SOFTWARE DESCRIPTION

- **Front End:** Python
- **Web Application:** Machine learning,Artificial intelligence
- **Back End:** Data set

3. MODULE DESCRIPTION

- Development Process Module
- Software Requirements Analysis
- System development
- Software Coding and Testing

3.1. Development, process module-

The major tasks of the development process based on the standard described above and then survey some of the AI techniques used in supporting the tasks of this process. In particular we focus on the tasks related to requirements analysis, architecture design, coding, and testing. The system and software architectures play a major role in driving the management activities during the development and maintenance of software systems. The standard provides a flow of the development process activities and associated documents. The system architecture design activity which produces the Software architecture and requirements allocation description (SARAD) document establishes the top-level architecture of the system and defines the software and hardware items of the system.

3.2. Software requirements analysis-

Requirements are first expressed in natural language within a set of documents. These documents usually represent “the unresolved views of a group of individuals and will, in most cases be fragmentary, inconsistent, contradictory, not prioritized and often be overstated, beyond actual needs”. The main activities of this phase are requirements elicitation, gathering and analysis and their transformation into a less ambiguous.

3.3. Software architecture design-

One of the most important problems facing the software engineer is to develop quality architecture from the requirements model. In this section we describe recent work on software architecture design using AI techniques. Developing the software architecture starts by defining a hierarchy of subsystems and components with allocated responsibilities from the information provided by the requirements and analysis models. AI techniques uses quality attributes to define a goodness function over the space of possible architectures process

3.4. Software coding and testing:

Techniques learned from AI research make advanced programming much simpler, especially with regard to information flow and control as a result of advances in knowledge representation. In the following we focus on the AI techniques used in supporting the tasks of coding and testing

Coding:

Software engineers can apply AI techniques to help automate or assist the programming process.

Use of AI to help assist the programming process:

The main idea here is to create an expert system to assist software engineers during software development. This proposal is called the Programmer's Apprentice Project. The Programmer's Apprentice should have the capability of interacting with the human programmers exactly the same way as human assistants would, thereby hopefully increasing the productivity of the human programmers.

Testing:

Software testing remains an expensive task in the development process and one of the main challenges concerns its possible automation. AI techniques can play a vital role in this regard. One of these techniques are constraint solving techniques. Attention has been devoted to the use of constraint solving techniques in the automation of software testing (Constraint-based testing).

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

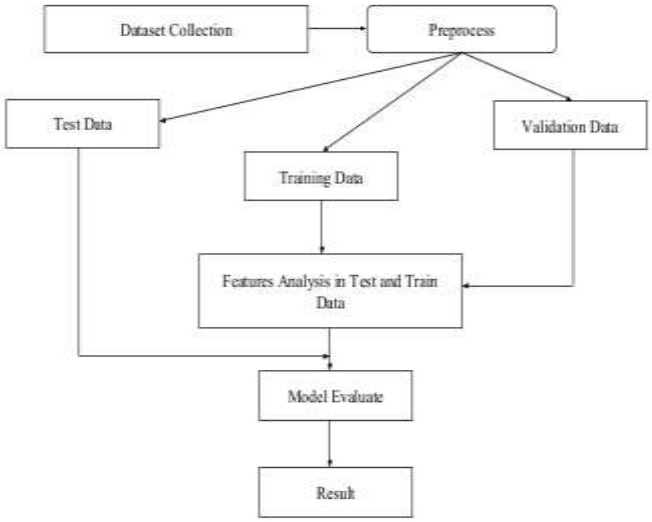
Vol. 03, Issue 09, September 2023, pp : 127-130

e-ISSN : 2583-1062

Impact Factor : 5.725

4. SYSTEM ANALYSIS AND DESING

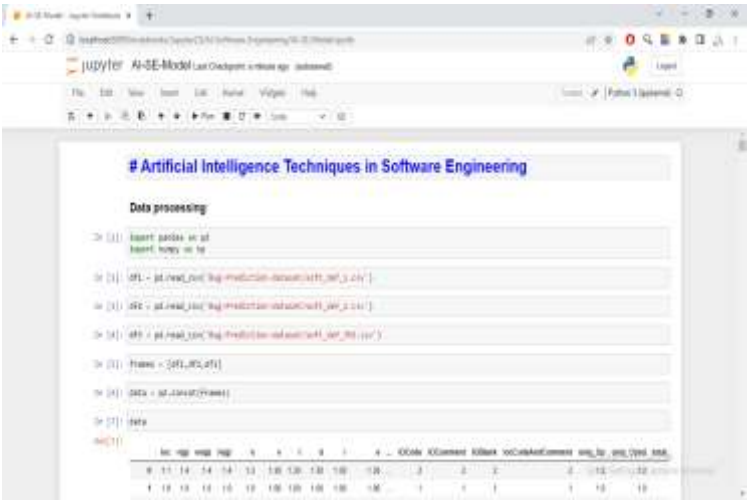
4.1. Use Case Diagram



5. SYSTEM TESTING

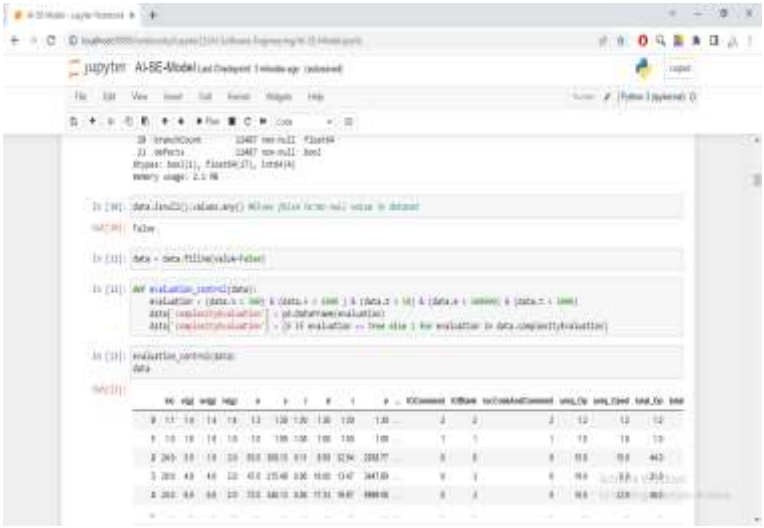
5.1. Unit Testing

Load Dataset



5.2. Integration Testing

Check Null Value



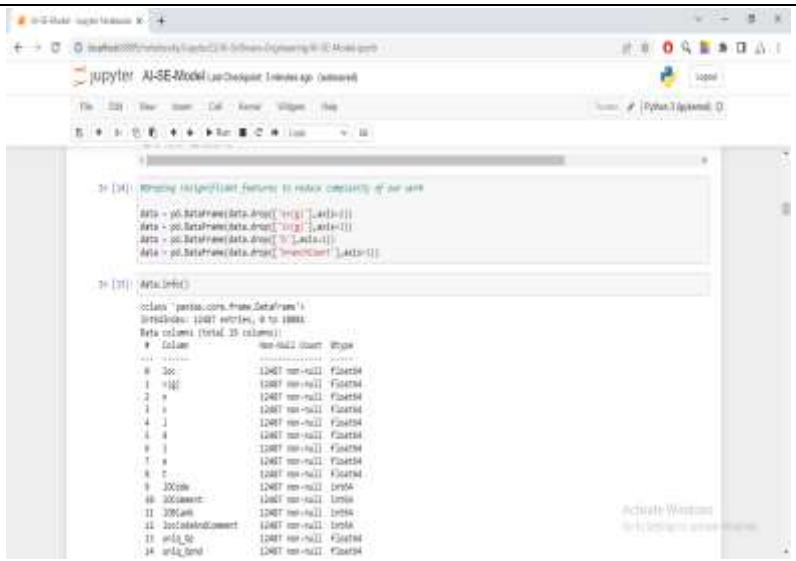
INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor : 5.725

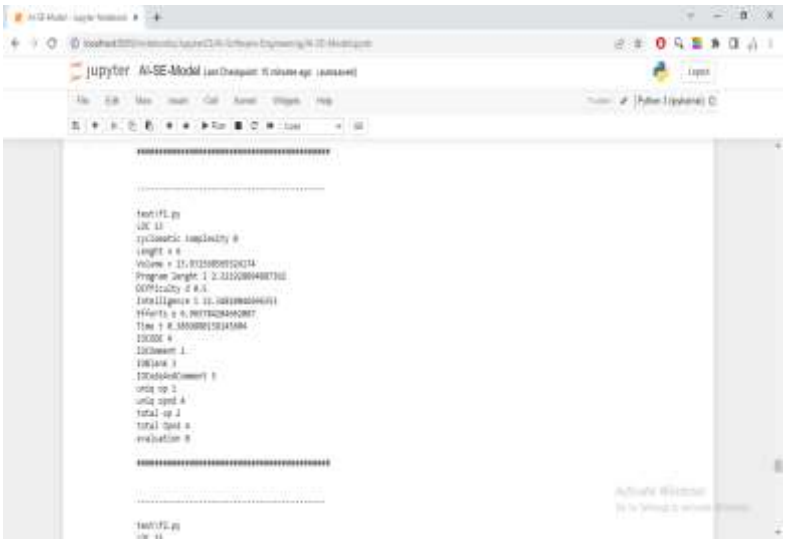
www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 09, September 2023, pp : 127-130



5.3. Validation Testing

In Register Form if we give User id as a string it Shows error.



6. CONCLUSION

This systematic mapping study provides an overview of the contributions and trends involving AI techniques in the SE literature. This study shows that the literature on this topic is heterogeneous, presenting several AI techniques used in SE. A significant proportion of the literature focuses mainly on SE phases, AI techniques, and performance measurement. To conduct this study, the authors used the mostly identified eight SE phases to assess the AI techniques environment. This study found that ML is used the most for automating the process in various SE phases by using a designation algorithm. ML is seen continually evolving and enhancing its efficiency and accuracy.

7. REFERENCES

[1] Technical debt as an indicator of software security risk: a machine learning approach for software development enterprise. Authors miltiadis siavvas, dimitrios tsoukalas and dimitrios tzovaras

[2] Machine learning based methods for software fault prediction: a survey Authors: sushant kumar pandey, ravi bhushan mishra and anil kumar tripathi

[3] An empirical framework to investigate the impact of bug fixing on internal quality attributes Authors: lov kumar, sahithi tummalapalli and lalita bhanu murthy