most tasks and even surpassed fine-tuned models in certain cases. However, its average success rate of 60.28 percent in reasoning problems could be improved. The study also found that ChatGPT demonstrated stronger deductive reasoning than inductive reasoning. One significant issue identified was "hallucinations," where the model generates incorrect or nonsensical information. The researchers suggested that closer collaboration and oversight could help mitigate these instances. However, Frieder *et al.,* [11] pointed out that ChatGPT may lag behind the mathematical abilities of a typical graduate student. Their research indicates that while ChatGPT generally understands the questions it is asked, it often requires assistance to provide accurate answers, particularly in complex mathematical contexts, highlighting an essential area for improvement as the technology progresses.

IV. CURRENT BEST PRACTICES

A. Leveraging ChatGPT for Debugging Code Errors

To rectify programming errors using ChatGPT, the model must be trained on an extensive dataset comprising code snippets, bug reports, and relevant information from software development. This training aims to equip the model with the ability to discern connections between code and bugs, enabling it to detect and correct errors in new code snippets efficiently.

B. Assisting with Debugging Processes

ChatGPT can be an invaluable tool for offering suggestions and corrections to code by leveraging its understanding of the intricate relationships between code and bugs. This capability can significantly streamline the debugging process, reducing the time and effort needed to identify and resolve issues. By utilizing its advanced natural language processing (NLP) skills, knowledge representation, and pattern recognition, ChatGPT can provide developers with automated debugging assistance.

For example, the model may use the vast amount of training data it has processed to suggest a fix when it identifies a flaw in the code. These suggestions are based on its deep knowledge of programming languages, common error scenarios, and software development best practices. Although its application is still in its early phases, research must continue to thoroughly evaluate the possibilities and limitations of ChatGPT for debugging. The quality of training data, system architecture, and the types of programming problems addressed are crucial factors determining ChatGPT's success in this context.

C. Predicting Bugs in Code

ChatGPT leverages its comprehensive understanding of the relationships between code and defects to predict and identify bugs accurately. This capability is invaluable for early bug detection in the development cycle, preventing the escalation of complex and costly issues in the future. ChatGPT's bug prediction relies on its ability to analyze and comprehend code snippets, utilizing its knowledge representation and pattern recognition skills to identify potential issues.

The success of ChatGPT in predicting programming bugs largely depends on the quality and diversity of the training data it has been exposed to. A well-rounded and high-quality dataset enables the model to develop a robust understanding of code-bug relationships, leading to more accurate predictions and early identification of potential issues during development.

D. Explaining the Root Cause of Bugs

ChatGPT's ability to provide detailed explanations for bugs-such as insights into why specific code is malfunctioning and how it can be corrected-is a highly user-friendly feature. This capability enhances developers' understanding of bugs and offers guidance on avoiding similar issues in the future. By utilizing its knowledge representation and natural language generation capabilities, ChatGPT can explain programming bugs in an informative and easy-to-understand manner. When the model identifies a bug, it can analyze the code and generate a clear explanation of the root cause, helping developers understand what went wrong and how to fix it. This explanatory process typically involves several stages: input analysis, bug identification, explanation generation, and output delivery.

E. Comparing ChatGPT with Traditional Debugging Tools

While both ChatGPT and conventional debugging tools have their advantages and disadvantages, the best method for fixing programming errors will vary depending on the situation and the developer. Traditional debugging tools, such as Integrated Development Environments (IDEs) and debuggers, offer a range of features like breakpoints, variable inspection, and trace analysis, which are essential for diagnosing and fixing bugs. However, a solid grasp of these tools' capabilities is necessary, as they may require careful use.

In contrast, ChatGPT provides a more user-friendly and intuitive approach to debugging. With its natural language processing and knowledge representation capabilities, ChatGPT can analyze code snippets and deliver explanations for bugs in a straightforward and accessible manner. This makes it particularly useful for identifying and addressing more complex bugs. Although the precise performance of ChatGPT and other tools will depend on their implementation, Table I compares ChatGPT's capabilities with those of other debugging tools.

TABLE I COMPARING CHATGPT WITH TRADITIONAL DEBUGGING TOOLS

Capability	Explanation
Expense	IDEs and debuggers, traditional debugging tools, often have high costs. In contrast, ChatGPT is generally available as a cloud service with a more adaptable and flexible pricing model.
Efficiency	ChatGPT can rapidly and accurately identify bugs and forecast issues, outperforming traditional debugging tools in speed and efficiency.
Precision	The efficacy of ChatGPT's bug identification and explanations is contingent on the quality of the training data. Conversely, conventional debugging tools may provide higher precision but demand a deeper understanding of the code.
Customization	Traditional debugging tools offer extensive customization options, whereas ChatGPT is built for immediate deployment and may offer a different level of customization.
User-Friendliness	The Natural Language Generation (NLG) capabilities of ChatGPT simplify outputs for developers, compared to the complexity and challenges associated with traditional debugging tools.
Compatibility with Existing Tools	It's important to note that conventional debugging tools are robustly integrated with various systems and tools, whereas ChatGPT may achieve a distinct level of compatibility.
Scalability	ChatGPT can efficiently handle large-scale debugging, making it well-suited for extensive and intricate codebases. This sets it apart from traditional debugging tools, which often face challenges when dealing with such high demands.

V. FUTURE RESEARCH

The next steps in our research should focus on enhancing ChatGPT's capabilities specifically for software debugging. This involves improving the model's ability to accurately predict and explain bugs across various programming languages and environments. Additionally, we should explore integrating ChatGPT with other advanced debugging tools to create a more comprehensive and resilient debugging ecosystem. Validating the model in real-world scenarios and addressing issues such as overreliance on high-quality training data are crucial for maximizing its performance. Ongoing research in this domain is essential to fully harness AI's transformative potential in revolutionizing software development methodologies.

VI. CONCLUSION

ChatGPT demonstrates significant potential in aiding the resolution of programming bugs through debugging support, issue prediction, and bug explanations. It excels at analysing and understanding code snippets, representing information in a sophisticated manner, and generating natural language. However, while ChatGPT is useful, it is important to remember that more comprehensive alternatives exist. The effectiveness of its output is primarily influenced by the quality of its training data and the system's overall architecture. To accurately assess ChatGPT's predictions and explanations and to conduct thorough bug testing, it is essential to integrate its functionality with established debugging tools and protocols. Therefore, ChatGPT should be viewed as a component within a comprehensive debugging toolkit, to be used alongside other methodologies. Combining ChatGPT with standard debugging tools may enhance developers' understanding of their code and improve their ability to identify and resolve bugs. Given that this is a rapidly evolving field, further investigation into ChatGPT's role in bug resolution is continuously needed. The type of defects being addressed, the quality of the training data, and the system's overall architecture are all factors that will determine how well ChatGPT performs in resolving programming issues.

REFERENCES

[1] W. E. Wong, X. Li, P. A. Laplante, and M. Siok, "Be More Familiar with Our Enemies and Pave the Way Forward: A Review of the Roles Bugs Played in Software Failures," *J. Syst. Softw.*, vol. 133, pp. 68-94, 2017. [Online]. Available: https://doi.org/10.1016/j.jss.2017.06.069

[2] D. R. E. Cotton, P. A. Cotton, and J. R. Shipway, "The Benefits and Challenges of ChatGPT: An Overview," *Front. Comput. Intell. Syst.*, vol. 2, no. 2, pp. 81-83, 2022. [Online]. Available: https://www.researchgate.net/publication/367106604_The_Benefits_and_Challenges_of_ChatGPT_An_Overview

[3] Y. Xu, T. Zhang, and M. Li, "AI-Driven Automated Bug Detection: A Comparative Study," *J. Softw. Eng.*, vol. 15, no. 2, pp. 85-98, 2023.

[4] R. Sharma and V. Patel, "Enhancing Software Development with AI-Integrated Debugging Tools," *Int. J. Comput. Sci.*, vol. 22, no. 1, pp. 123-137, 2023.

[5] J. Liu and H. Wang, "Context-Aware NLP in AI-Driven Debugging: A New Approach to Software Quality," *J. Artif. Intell. Res.*, vol. 28, no. 1, pp. 45-61, 2024.

[6] J. Deng and Y. Lin, "The Benefits and Challenges of ChatGPT: An Overview," *Front. Comput. Intell. Syst.*, vol. 2, no. 2, pp. 81-83, 2023. [Online]. Available: https://doi.org/10.54097/fcis.v2i2.4465

[7] M. Aljanabi, "ChatGPT: Future Directions and Open Possibilities," *Mesopotamian J. Cybersecurity*, pp. 16-17, 2023. [Online]. Available: https://doi.org/10.58496/MJCS/2023/003

[8] D. Sobania, M. Briesch, and C. Hanna, "An Analysis of the Automatic Bug Fixing Performance of ChatGPT," Preprint, pp. 1-8, 2023. [Online]. Available: https://arxiv.org/pdf/2301.08653

[9] J. H. Choi, K. E. Hickman, and A. B. Monahan, "A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity," Preprint, pp. 1-16, 2022. [Online]. Available: https://www.researchgate.net/publication/368361643_A_Multitask_Multilingual_Multimodal_Evaluation_of_ChatGPT_on_Reasoning_Hallucination_and_Interactivity

[10] S. Cahyawijaya, N. Lee, W. Dai, D. Su, and B. Wilie, "A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity," *Comput. Lang.*, vol. 4, pp. 1-45, 2023. [Online]. Available: https://arxiv.org/pdf/2302.04023

[11] S. Frieder et al., "Mathematical Capabilities of ChatGPT," in *37th Conf. Neural Inf. Process. Syst. (NeurIPS 2023) Track on Datasets and Benchmarks*, pp. 1-46, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/58168e8a92994655d6da3939e7cc0918-Paper-Datasets_and_Benchmarks.pdf