

# **ChatGPT's code suggestion accuracy evaluation**

LAB University of Applied Sciences

Bachelor of Business Administration, Business Information Technology

2024

Uyen Lai

## Abstract

Author(s)	Publication type	Completion year
Uyen Lai	Thesis, UAS	2024
	Number of pages	
	37	
Title of the thesis		
<b>ChatGPT's code suggestion accuracy evaluation</b>		
Degree, Field of Study		
Bachelor of Business Administration, Business Information Technology		
Name of the thesis supervisor		
Tommi Kallonen		
Abstract		
<p>Generative AI is becoming increasingly prominent in the field of software development. For developers, AI-powered programming tools are essential. While many studies have been published on generative AI tools, such as their usage by people or the influence on developers, there has been limited research into the precision of AI's recommended code. This thesis presents an empirical evaluation of generative AI's accuracy in generating code using ChatGPT as the main tool.</p> <p>The research utilizes both quantitative and qualitative methods to gain a more comprehensive understanding of the subject and address the research questions. The literature review section employs a qualitative method to understand the raised research problem. The study's practical section applies the quantitative approach to evaluate the correctness of ChatGPT responses, through testing 30 randomly selected questions from the LeetCode platform.</p> <p>Research result shows that ChatGPT's solutions are effective for simple coding problems, but their effectiveness decreases as challenges become more complex. ChatGPT's solutions are often accurate, but the high memory usage and upper limit on memory indicate that they may not be optimal in many cases. The findings also suggest that ChatGPT's below-average performance is likely due to the quality of its training data rather than just semantic inaccuracies.</p>		
Keywords		
AI, machine learning, generative AI (GAI), ChatGPT		

## Contents

1	Introduction.....	1
1.1	Research background.....	1
1.2	Research problem .....	2
1.3	Research objectives .....	2
1.4	Research questions .....	4
1.5	Research methods.....	4
1.5.1	Research methodology .....	5
1.5.2	Research reasoning.....	6
1.6	Thesis structure .....	8
2.	Literature review .....	9
2.1	What is artificial intelligence?.....	9
2.2	Natural language processing .....	9
2.3	Machine learning.....	10
2.4	Generative AI (GAI) .....	11
2.5	The difference between AI and generative AI .....	12
2.6	ChatGPT .....	13
2.6.1	What is ChatGPT .....	13
2.6.2	Chatbots versus ChatGPT .....	14
2.6.3	How does ChatGPT work?.....	14
2.6.4	How important ChatGPT's transformers are? .....	14
2.6.5	Traning the model .....	15
2.7	Pros and cons of generative AI.....	16
2.8	Generative AI in software development .....	17
2.9	Generative AI's impacts on developers' productivity .....	18
2.9.1	How does generated code by GAI impact the productivity of developers? .....	18
2.9.2	Where generative AI was most striking? .....	19
2.9.3	Which tasks demand developer expertise?.....	19
3.	Empirical research.....	21
3.1	What is Leetcode? .....	21
3.2	Testing approach .....	22
3.3	Research result.....	25
3.4	Conduct more testing for better insights .....	28
4.	Conclusion.....	30
5.	Research limitations and recommendations.....	31

6. Summary .....	32
References .....	33

# 1 Introduction

## 1.1 Research background

In recent years, there has been an incredible technological revolution driven by the rapid growth of Artificial Intelligence (AI). This innovative breakthrough has impacted a wide range of industries, including healthcare, education, transportation, and technology. AI has quickly become an integral part of our daily lives, shaping economies, enhancing human experiences, and providing solutions to complex problems. Today's AI systems are incredibly advanced, capable of analyzing vast amounts of data, making intelligent decisions, and performing tasks that were previously only possible with human intelligence. From virtual assistants and self-driving cars to personalized healthcare diagnoses and recommendations, AI innovations continue to transform the way we live, work, and engage with the world around us.

The field of software development is witnessing a growing prominence of generative AI (GAI), a sub-branch of AI. This powerful technology is capable of producing realistic images, music, and coherent text within seconds. Executives from companies such as GitHub have declared that generative AI tools have revolutionized software development, significantly boosting developer's productivity and transforming the entire process (Lin 2023a). Furthermore, AI has the potential to impact employment, and company organization in significant ways, influencing the labor market and businesses.

For developers, AI-powered programming tools are essential. Among the most widely used are ChatGPT, GitHub Copilot, and TabNine. While ChatGPT offers a broad range of information beyond programming operations, GitHub Copilot and TabNine are primarily focused on code completion. Specifically, developers can request suggestions from scratch by describing the context where the code will be used using natural language. They will then receive code snippets and tests, or recommendations for the code they have already used (GitHub a).

Although AI has undeniably improved developer productivity, it is necessary to consider possible cybersecurity, biases, license issues, or the accuracy of code suggestions to ensure the effective integration of AI technologies in the development process (Lin 2023b).

## 1.2 Research problem

Research problems are defined as the educational issues, disputes, or concerns that provide a study conducting. Research can be performed on a variety of topics, including *transition rates, gender, performance in national examinations, strikes and violence, parental involvement, financing, leadership attributes, and affirmative action*. (Ayiro 2012a, 82.)

AI has the potential to enhance developer productivity by offering real-time suggestions, reducing manual code exploration, and providing intelligent code completion and recommendation systems. It can also assist in identifying and resolving software issues. However, it is essential to consider potential challenges that may arise while implementing AI in software development. For example, GitHub Copilot may include undesirable patterns such as insecure coding patterns, bugs, or references to outdated APIs since the tool was trained based on a vast amount of open-source code (GitHub b). As a result, the quality of the AI-based suggested code should be considered.

While there are numerous studies published on generative AI tools such as how people use them or the impact of them on developers there has been limited research on the accuracy of AI's suggested code. This thesis contributes an empirical assessment of generative AI's capabilities. The insights gained from the research may help developers understand how they can leverage AI tools as well as provide valuable information to other researchers in this area.

## 1.3 Research objectives

Ayiro's (2012b, 165-166) study describes that researchers often use general research questions as a starting point for formulating research objectives. This is due to the fact that research objectives are seen as a stronger indication of the researcher's specific intent and direction by the research community. The author also argues that research objectives tend to yield more precise results than research questions. This is because research objectives require stricter reasoning, which is reflected in the use of formal language. If these objectives met the well-known SMART criteria proposed by Maylor & Blackmon (2005), they would be improved.

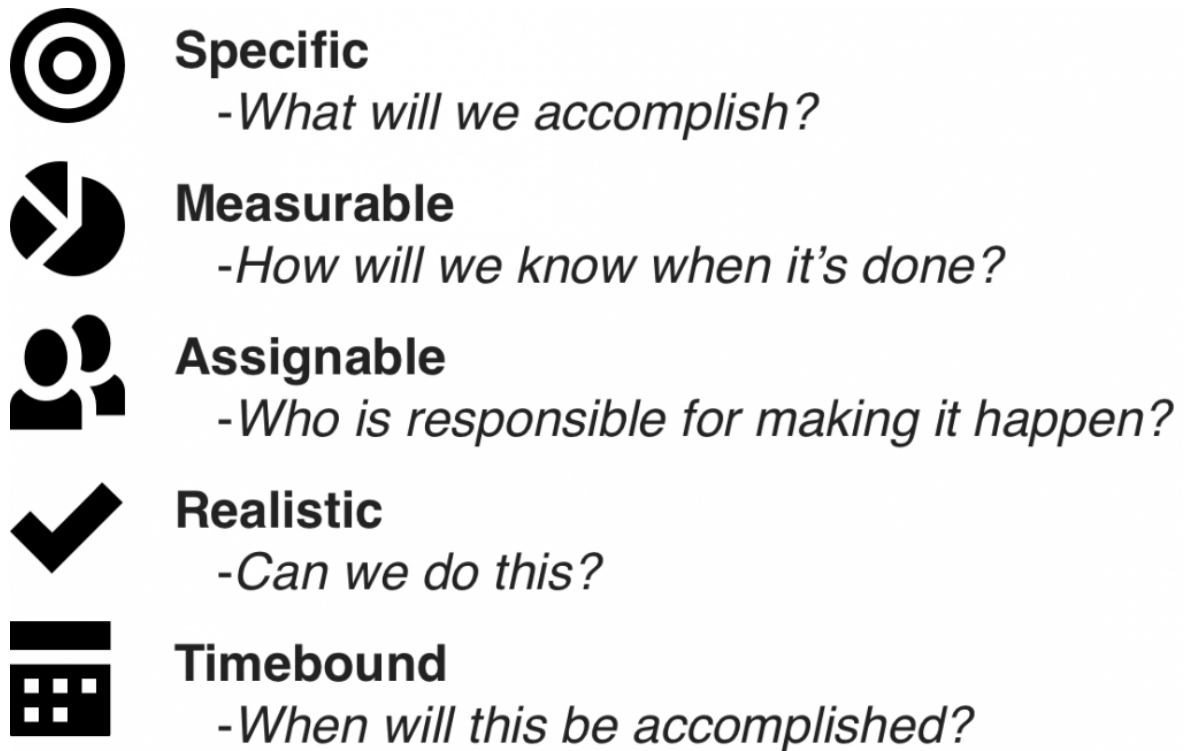


Figure 1. SMART criteria (Szatmary 2016)

In recent times, the field of artificial intelligence has witnessed remarkable advancements, and as a result, programmers have started relying on AI-powered code suggestion tools to enhance their productivity and speed up the software development process. These tools assist programmers by reducing coding errors and providing suggestions for code completion. Some of the most prominent tools in this field include GitHub Copilot, TabNine, and ChatGPT.

In this research, the author aims to:

- Summary of the pros and cons of generative AI tools based on previously published studies on the topic.
- Assess the accuracy of ChatGPT's code suggestion. By employing 30 LeetCode questions in JavaScript programming language and executing the tests provided by LeetCode, the result promises to shed light on the strengths and limitations of ChatGPT in the code suggestion domain.

## 1.4 Research questions

In research, either quantitative or qualitative, research questions are used to narrow down the broad topic to specific questions that the researchers aim to address. These questions are formulated before the precise research methodologies such as data collecting, analysis, and interpretation are chosen. In qualitative research, the questions focus on the fundamental topic under investigation, whereas in quantitative research, the inquiries revolve around the features or characteristics (variables) of persons or organizations. (Ayiro 2012c, 157.)

This thesis aims to explore the realm of ChatGPT and its potential use in the code suggestion field. The main objective of this research is to evaluate the accuracy of ChatGPT's code suggestion capabilities. Through providing valuable insights and analysis, this study seeks to contribute to the development of better code suggestion systems.

## 1.5 Research methods

After identifying the research question and objectives, it is essential to choose appropriate research methods that align with the goals of the study.

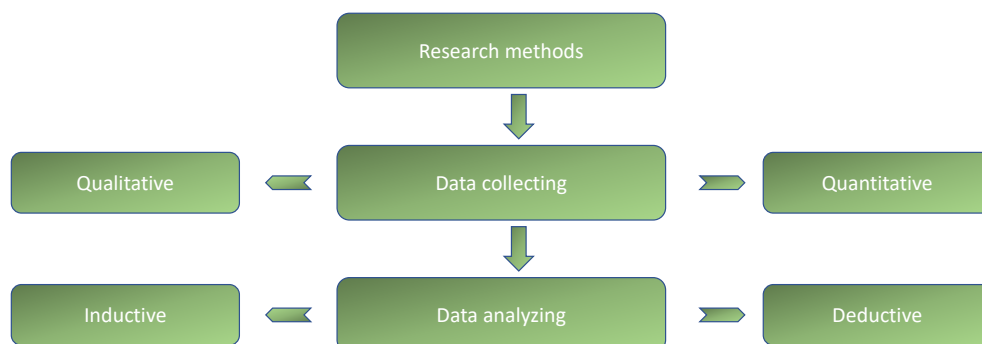


Figure 2. Research methods map (adapted from O’Gorman & MacIntosh 2015, 51)



### 1.5.1 Research methodology

Data collection is an essential step in research and analysis since it provides the foundation for deriving useful insights and making wise decisions. There are several types of commonly used techniques, including *experimental research*, *questionnaire surveys*, *ethnographic research*, *in-depth interviews*, *content analysis*, *secondary analysis of existing data*, *historical research*, and *comparative analysis*. These techniques for conducting research are often divided into quantitative and qualitative categories. (Wang & Park 2016a, 86.)

The quantitative method is used widely across various fields such as medicine, psychology, and education. Quantitative research presents its findings in the form of numerical descriptions. While collecting empirical data, quantitative researchers barely find conclusive proof for their theories. This outcome might discourage policymakers or students who begin their thesis journey to prove a specific point. (Blair 2016a, 52.) Furthermore, quantitative research methods include numerous techniques such as questionnaire surveys, content analysis, experimental tests, and other approaches used to quantify data for numerical analysis. Statistical tools are applied by quantitative researchers to assess and summarize the gathered data. (Wang & Park 2016b, 86.)

According to Blair (2016b, 56-57), qualitative research encompasses a variety of methodologies and should not be perceived as the opposite of quantitative research. Qualitative research is often associated with constructivism, as it acknowledges that knowledge is socially constructed. Qualitative studies typically involve a smaller number of participants and are carried out in real-life settings. Qualitative researchers employ a comprehensive method of inquiry, actively engaging in data gathering and analysis. The researcher serves as the primary research instrument during one-on-one interviews with open-ended questions or observational studies. The interviewer has the flexibility to ask insightful questions, allowing the interview to go in various directions. While qualitative researchers strive to understand phenomena from the perspectives of the participants, they acknowledge that they also have socially constructed worldviews that influence their research approach. As a result, researchers need to assess personal biases and properly identify their relationship to the phenomena being studied.

In certain research studies, researchers utilize both quantitative and qualitative methods to collect data. This approach is commonly known as the mixed method. Rather than relying on a single dataset, the researcher combines both to obtain a more thorough understanding of the issue being studied (Ayiro 2012d, 493).

In this research, the author utilizes both quantitative and qualitative approaches to gain a comprehensive understanding of the topic and to answer the research question. The literature review section employs qualitative methods to grasp the current research problem, while secondary research method is used to gather data from previous studies or literature sources. This helps the author to obtain a more profound insight into how generative AI technologies are enhancing developers' productivity.

In the empirical part of the research, a quantitative method is employed by the author to evaluate the correctness of ChatGPT solutions. The author conducts research using 30 random questions from the LeetCode platform. These test results will be used to draw a conclusion that answers the research question.

### 1.5.2 Research reasoning

There are two prominent types of research reasoning which are deductive and inductive respectively.

The deductive approach is associated with the hypothesis testing approach, where the researcher starts with broad principles and moves to specific instances (Williamson & Bow 2002a, 26-27).

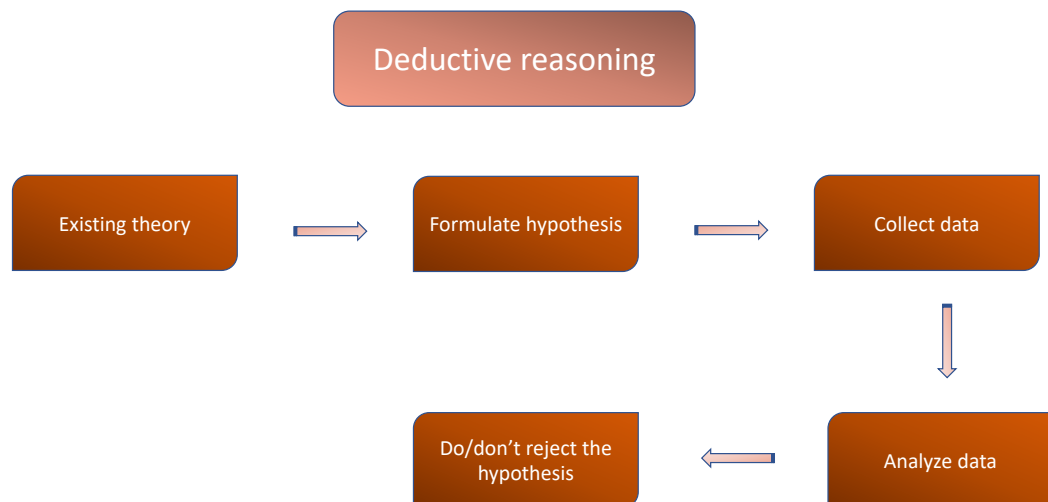


Figure 3. Deductive research approach (adapted from Streefkerk 2023a)

On the other hand, inductive reasoning follows the hypothesis-generating approach, in which the fieldwork and observations are conducted first, and hypotheses are formulated from the analysis of the collected data (Williamson & Bow 2002b, 26-27).

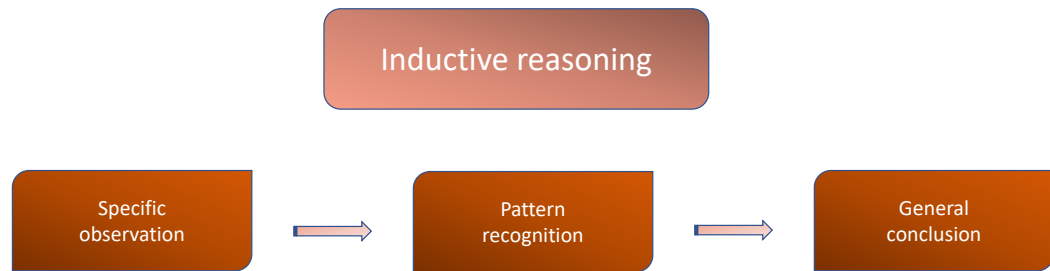


Figure 4. Inductive research approach (adapted from Streefkerk 2023b)

This thesis uses deductive reasoning, starting with broad theoretical concepts including AI, machine learning, generative AI, and ChatGPT. The empirical section then narrows the focus to evaluate ChatGPT's code solutions for accuracy.

## 1.6 Thesis structure

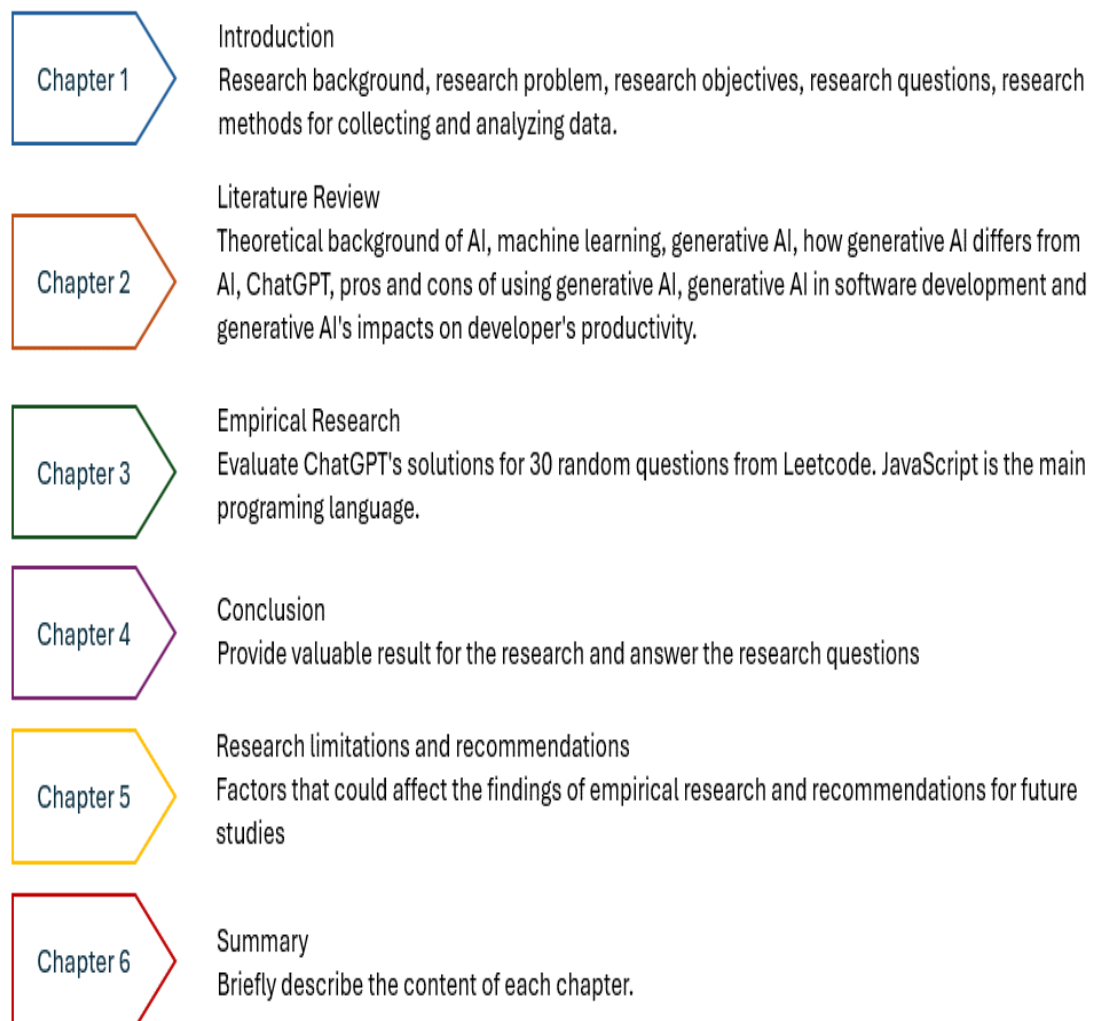


Figure 5. Thesis structure

## 2. Literature review

### 2.1 What is artificial intelligence?

From Apple's launch of SIRI and its use in smartphones like the iPhone to the current development of autonomous or self-driving cars, Artificial Intelligence (AI) is now an essential part of our daily lives, deeply embedded in various sectors including industry, finance, economy, homeland security, and governance. While AI is frequently portrayed in science fiction as robots with human-like characteristics, AI can refer to anything from autonomous weapons to IBM's Watson to Google's search algorithms. The seamless integration emphasizes the deliberate utilization of advanced artificial intelligence technologies to enhance and collaborate with humans, thus changing and augmenting various aspects of contemporary life. (Zohuri & Zadeh 2020a, 1 – 3.)

The term Artificial Intelligence was introduced in 1956 by John McCarthy. Since then, numerous definitions for AI have been proposed by scientists and researchers such as:

- AI is about to make computers do tasks that humans currently perform better.
- AI is a part of computer science focused on creating smart computer systems that resemble human intelligence.
- It is the branch of computer science that represents knowledge using symbols instead of numbers and using practical, heuristic techniques for information processing.

(Gupta & Mangla 2020a, 4.)

Similarly, Zohuri and Zadeh (2020b, 5) assert that Artificial Intelligence involves instructing computers to imitate human behavior using the Neural Network technique. They emphasize the primary objective of AI is to make computers capable of executing tasks that have historically been associated with human intelligence. The authors further categorize AI into weak artificial intelligence (WAI) also known as narrow AI and strong artificial intelligence (SAI) or super artificial intelligence. WAI is designed to focus on a specific task, while SAI, with the machine learning techniques running in the background can perform multiple tasks or tackle any problem with ease.

### 2.2 Natural language processing

Natural language processing (NLP) is a subset of artificial intelligence that enables machines to comprehend, interpret, and manipulate human language. NLP aims to fill in the gap between human-machine communication and a variety of areas, including computer

science and computational linguistics. Machine code, also referred to as machine language, is the primary language used by computers. Most people are unable to comprehend it. Devices communicate through vast sequences of zeros and ones rather than through words. Neural language processing becomes crucial when handling extensive textual data, enabling computers to engage with humans using natural language and facilitating various language-related tasks. With the assistance of natural language processing, computers can read text, understand spoken language, evaluate emotion, and recognize crucial information. In conclusion, NLP is a part of artificial intelligence and refers to the capacity of a computer program to understand human language. (Zohuri & Zadeh 2020c, 13 – 14.)

### 2.3 Machine learning

With the increasing prevalence of automation and AI across various economic sectors and job functions, machine learning (ML) is emerging as a rapidly developing field of new technology and cutting-edge science. According to Gupta and Mangla (2020b, 95), machine learning refers to the capacity of an artificial intelligence system to improve its performance progressively. This relies on the system's ability to learn new knowledge and abilities as well as its recognition of prior knowledge from the absorbed material. The initiation of machine learning typically requires a foundational set of knowledge and the appropriate structuring of that knowledge, which allows the system to understand, evaluate, and confirm the acquired information.

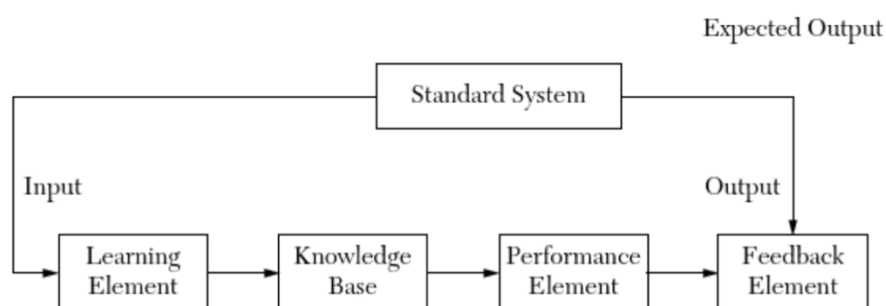


Figure 6. Learning System Model (Gupta & Mangla 2020c, 96)

The depicted learning system model comprises five key components:

- Learning element processes input from a teacher or external sources.

- Knowledge base acts like a database, initially containing basic knowledge and receiving updates.
- Performance element utilizes the updated data to carry out tasks or resolve issues.
- Feedback element compares inputs from the learning element and the standard system to direct corrections.
- Standard system, serving as an idealized benchmark, aids in evaluating the machine learning system's proficiency by comparing its output with that of the standard system.

(Zohuri & Zadeh 2020d, 95 – 96.)

The field of machine learning is supported by significant scientific research on pattern rearrangement, mathematical optimization, computational learning theory, self-optimizations, and nature-inspired algorithms. These advancements, although often unnoticed, have become a vital component of our daily lives. Machine learning has revolutionized various applications, including web searches, speech recognition (such as Apple's Siri), online entertainment (like Netflix), and even our understanding of the human genome. Its ability to create intelligent software and self-learning computers is fundamental in enhancing human capabilities. (Zohuri & Zadeh 2020e, 87.)

## 2.4 Generative AI (GAI)

In contrast to traditional AI, which primarily focuses on analyzing and classifying existing data, generative AI actively creates new content, including text, images, and audio. As proven by GPT-4, Copilot, and Dall-E 2, the widespread adoption of this technology is currently transforming work processes and collaboration. Generative AI systems like GPT-4 show remarkable performance in answering questions, demonstrating their intelligence, and broadening their utility beyond creative applications. Their use is expanding, ranging from assisting with transitional knowledge work activities at IT help desks to providing general needs such as recipe generation or medical assistance. OpenAI's GPT-4, an advanced language prediction model, is an outstanding example of generative AI. It uses vast amounts of internet training data to generate human-like text that closely resembles content created by humans. The primary drawbacks associated with advanced generative AI tools include fairness and bias, potential copyright violations, and environmental issues. It is apparent that the accuracy of generative AI models depends on the quality of training data and learning methods used. While the results from generative AI models may appear reliable, they may lack accuracy in syntax or meaning. Additionally, the content produced

by these models is often biased, relying on assumptions rather than factual information. (Feuerriegel et al. 2024a, 111 – 126.)

According to Feuerriegel et al. (2024b, 111 – 126), the significance of the alignment procedure and the input of training data is important to ensure unbiased generative AI. The utilization of biased data to train deep learning models can lead to the reinforcement of negative language, amplification of human biases, and preservation of preconceptions related to gender, sexual orientation, politics, and religion. While the academic community has given considerable attention to the issues of bias and fairness in AI, further research is necessary to gain a complete understanding of the concept of fair AI.

Concerning copyright infringement, generative AI models, systems, and applications have the potential to violate copyright laws by generating outputs that closely resemble or replicate existing works without obtaining permission or providing compensation to the original creators. This creates legal questions about the delicate balance that generative AI systems must maintain between creativity and originality. In addition, concerns over patent ownership and other intellectual property rights arise concerning works produced by generative AI. (Smits & Borghuis 2022, according to Feuerriegel et al. 2024c, 111 – 126.)

Finally, Schwartz et al. (2020) emphasize to consider the environmental impact of generative AI systems, which are typically built using complex neural networks. The creation and operation of these systems can result in significant electricity consumption and a negative carbon footprint. However, the AI research community is actively working to address these issues by using optimized hardware, compressing neural network architectures, and developing more effective training algorithms. These efforts aim to make the development and application of AI algorithms more environmentally friendly. (Feuerriegel et al. 2024d, 111 – 126.)

## 2.5 The difference between AI and generative AI

In recent years, artificial intelligence (AI) has become a ubiquitous term in various industries, driving notable technological advancements and increased efficiency. One of the latest developments in this field is generative AI, which has the potential to transform entire industries. The crucial question is: What sets generative AI apart from traditional AI?

The main differences between generative AI and conventional AI are in their functions and applications. While the main goals of traditional AI systems are data analysis and prediction, generative AI goes one step further by producing new data that is comparable to its training resources. Traditional AI is highly skilled at identifying patterns, whereas generative AI demonstrates excellence in generating new patterns. The potential of generative AI is



extensive and presents exciting opportunities for creativity and innovation. In design, generative AI may produce a large number of prototypes quickly, cutting down on the amount of time required for brainstorming. It might assist with writing scripts, making new songs, or even generating deepfakes in the entertainment industry. Generative AI could be used in journalism to produce reports or narratives. In contrast, traditional AI performs effectively in task-specific applications, supporting a wide range of features including chatbots, recommendation engines, and predictive analytics. (Marr 2023.)

## 2.6 ChatGPT

### 2.6.1 What is ChatGPT

ChatGPT represents a notable technological development that marks a significant change in the constantly evolving tech environment. As a large language model (LLM) classified under the generative AI subset, it possesses the capability to generate novel content instead of merely analyzing pre-existing data. Categorized as a large language model within the generative AI subset, it is capable of creating new content instead of just analyzing existing data. Furthermore, ChatGPT has the capability to interact with users in their native language, leading to a more natural and human-like conversation. ChatGPT is a distinctive form of AI-powered software that sets itself apart from other categories due to its dialogue-based structure. Unlike conventional chatbots, which utilize pre-written responses to produce answers based on natural language queries, ChatGPT follows a more dynamic approach. When a user engages with ChatGPT, it independently generates responses without requiring predefined answers. This user-friendly method makes the system more accessible and simplifies interaction with it. (Baker 2023a, 5 – 8.)

In order to reduce the potential risks of ChatGPT generating unusual or erroneous responses, or adopting an offensive tone, it may be advisable to consider limiting the number of responses allowed within a single conversation. This measure may prove to be a necessary precaution to ensure that the interactions maintain a high level of quality and align with the intended purpose of the conversation. Thus, implementing this approach may prove to be a useful strategy to maintain the integrity of the conversation and safeguard against any unintended outcomes. For example, Microsoft has placed a limit on the number of responses that ChatGPT can provide in a conversation. Users are allowed to initiate new conversations, but ongoing exchanges are limited to a maximum of five responses per user. While this feature increases the probability of generating unique responses, it also raises the possibility of providing incorrect or false information or even hallucinations. Therefore, it is essential to keep in mind that ChatGPT, like any other generative AI model, is not a

reliable source of information, a trustworthy narrator, or an authority on any topic. (Baker 2023b, 9.)

### 2.6.2 Chatbots versus ChatGPT

AI-driven assistants like Siri, Alexa, Google Assistant, and Cortana are widely used. While AI chatbots have been in existence for a while, earlier versions had limitations that prevented them from effectively engaging in conversations with users. Some of the major drawbacks of these chatbots included their inability to make decisions based on user inputs, a lack of contextual understanding that made it hard for them to respond appropriately, heavy reliance on pre-programmed responses that made them less flexible, and memory limitations that prevented them from engaging in longer and more complex dialogues. These limitations often made it difficult for users to get the information or assistance they needed from chatbots, leading to frustration and dissatisfaction with these tools. ChatGPT stands out by overcoming the limitations of other AI-powered assistants. It can understand the context of conversations, make decisions, and sustain lengthy and human-like conversations. Moreover, ChatGPT's responses are customized for each prompt and its variations, avoiding the use of pre-written responses. This results in a flexible and dynamic engagement experience, regardless of the specific keywords used. (Baker 2023c, 15.)

### 2.6.3 How does ChatGPT work?

ChatGPT is an advanced AI assistant that responds to user inputs without revealing its sources. It employs cutting-edge technology, such as GPT-3 or GPT-4 models, to analyze input prompts and predict the most probable sequence of words in context. ChatGPT leverages GPT-3 or GPT-4 transformers to produce coherent and natural language in response to prompts. This impressive language generation capability offers a superior user experience. This process involves analyzing the context, assigning weight to words based on their likelihood to follow the prompt, and generating an appropriate response (Baker 2023d, 28.)

### 2.6.4 How important ChatGPT's transformers are?

ChatGPT produces responses to user inputs using a multilayered transformer network. In the field of artificial intelligence (AI), a neural network consists of processing nodes that use algorithms to simulate the functions of the human brain. Transformative networks are a distinct form of neural network architecture. ChatGPT utilizes transformers that are specifically designed for language processing. These transformers come in different forms that are tailored to handle various types of data, such as text or images. Transformers excel in

parallelization, making them highly efficient for scaling and accommodating large AI models and their training needs. The use of transformers was essential for GPT in general, and specifically for ChatGPT to generate outputs that closely resemble human-like responses. (Baker 2023e, 31.)

#### 2.6.5 Training the model

The ChatGPT model underwent training using an extensive database, incorporating text gathered from nearly the entirety of the internet up until 2021. Additionally, human instructors played a role in refining the initial model by assuming both human and machine roles. They provided guidance on distinguishing appropriate from inappropriate responses to prompts, contributing to the model's understanding and responsiveness. The newly constructed dialogue data set was then combined with the human feedback data set and converted into a dialogue format. Reward learning from human feedback (RLHF) is the term used to describe this procedure. For each model training instance, different approaches may be used. In other words, RLHF can be adjusted to meet the specific training requirements of a specific model. The process's reinforcement component is derived from the gathering of comparison data and human feedback via a thumbs-up/down ranking system. AI trainers used to rank the responses from previous conversations they had with the training model by selecting them randomly as a part of the training process. To further refine the model, reward models like the thumbs-up and thumbs-down rating system on ChatGPT can be utilized through a technique called proximal policy optimization. (Baker 2023f, 32 – 33.)

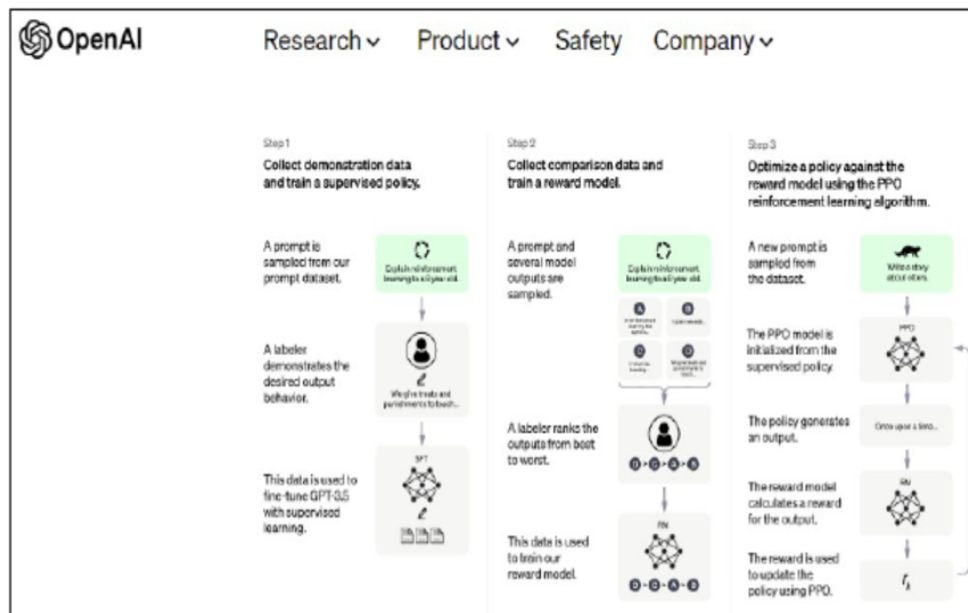


Figure 7. OpenAI's process of training the ChatGPT model (Baker 2023g, 34)

## 2.7 Pros and cons of generative AI

While generative AI holds promising prospects for businesses, it is crucial to carefully assess its advantages in comparison to its drawbacks. This technology introduces exciting possibilities for businesses, offering enhancements in creativity, efficiency, personalization, and predictive analytics.

Some advantages of generative AI include:

- **Enhanced creativity:** Generative AI can generate new and original content, providing businesses with the opportunity to explore innovative ideas and creative solutions.
- **Time and cost efficiency:** GAI enables businesses to automate various tasks that would otherwise require significant time and resources, leading to increased efficiency and cost savings.
- **Personalization and customer experience:** GAI helps organizations tailor their offerings by analyzing large volumes of data to understand consumer preferences and behavior, resulting in higher customer loyalty, improved client satisfaction, and better experiences.

- Predictive analytics: Generative AI can provide valuable insights and accurate predictions based on past data.

(Ewell 2023.)

In contrast, the disadvantages of generative AI must also be taken into account. These can be described as follows:

- Hallucination: The term refers to the irregularity of certain models in producing outputs that do not accurately reflect real-world facts or common-sense logic.
- Legal and regulatory issues such as:
  - ◆ Copyright issues: Due to the extensive training on large datasets, verifying whether generative AI models, including the materials within their data and the generated outputs, adhere to copyright laws can be challenging.
  - ◆ Privacy issues: Generative AI introduces concerns regarding the collection, storage, utilization, and security of data, encompassing both personal and business-related information.
- Energy consumption: AI models have a significant impact on the environment since they consume a lot of electricity. As these technologies become more prevalent, there will be an increase in demand for energy, which will further impact the environment.

(Bell 2023.)

## 2.8 Generative AI in software development

The substantial increase in GAI has resulted in software development becoming one of the most advantageous fields. Throughout the software engineering lifecycle, generative AI can assist developers in various aspects such as code generation, test case writing, and code maintainability (Robert & Schmidt 2023).

In terms of code generation, users can effortlessly create a text prompt describing their desired code output, and the advanced generative AI technology can generate code that precisely meets their requirements (IBM 2023). Regarding test case generation, generative AI tools can automatically provide comprehensive and efficient test cases by analyzing software requirements and user stories. Previously, developers had to write test cases manually to detect application defects and errors, which is a crucial step in software development.

However, manual testing is time-consuming and can lead to human errors. With the advancements of GAI, software testing has become more effective in the software development lifecycle. (Pattam 2023.) Finally, generative AI's code maintainability feature can detect complex code, eliminate redundancy, and improve code readability. It uses training standard input and past code modifications to analyze the code and provide detailed suggestions for improvement. This can save developers time and effort on debugging while also improving the overall quality of their codebase. (Ong 2023.)

Despite its enormous capabilities in every aspect of software development, this study focuses on the code generation field of generative AI to identify areas for improvement and contribute to technology development.

## 2.9 Generative AI's impacts on developers' productivity

### 2.9.1 How does generated code by GAI impact the productivity of developers?

According to a study conducted by McKinsey, software developers can complete their coding tasks almost twice as fast by utilizing generative AI. This means that tasks such as documenting code functionality for maintainability, writing new code, and optimizing existing code through refactoring can be completed in significantly shorter time frames. Specifically, documenting code functionality can be completed in half the time, writing new code in nearly half the time, and optimizing existing code in almost two-thirds of the time. With the proper training and support from businesses, these increased speeds can result in higher productivity, exceeding the improvements seen from previous advances in engineering efficiency achieved through new tools and processes. (Deniz et al. 2023a.)

Despite the potential for a substantial increase in productivity, the research emphasizes that time savings can vary significantly based on task complexity and the experience level of developers. Based on the study, the time savings reduced to less than 10% for tasks that developers found to be extremely complicated, mainly because they were not familiar with the required programming framework. A similar trend appeared for developers with less than a year of experience. In some cases, inexperienced developers required 7 to 10 percent more time to complete tasks using the tools than they did without them. Developers working in collaboration with AI-powered tools were able to maintain quality while also accelerating their work. The integration of AI in code development led to slight enhancements in reducing bugs, ensuring maintainability, and improving readability. Developers need to understand the characteristics that define quality code and be able to effectively communicate with the tool to achieve the desired results. (Deniz et al. 2023b.)

### 2.9.2 Where generative AI was most striking?

According to the research (Deniz et al. 2023c), generative AI-based solutions provide enormous productivity benefits in four major fields. Firstly, expediting manual and repetitive work. In detail, generative AI simplifies coding tasks by automatically filling in standard functions, completing code statements as developers type, and documenting code functionality based on prompts. By automating these tasks, it allows developers to focus on more complex challenges and expedite the development of new software capabilities. Secondly, jump-starting the first draft of new code. This means developers utilizing generative AI-based technologies can ask for suggestions when confronted with a blank screen by typing prompts in an integrated development environment (IDE) or in a separate window. Thirdly, generative AI tools can boost developers' productivity by accelerating updates to existing code. According to the participants in the research, they were able to make significant changes to their existing code with ease by using appropriate prompting tools. For example, developers could copy and paste pre-written code into a prompt and submit iterative queries to the tool, asking it to modify the code based on the inputs they provided. This saved them time in tweaking code from an online coding library and refining it. Lastly, these tools increase developers' ability to tackle new challenges. Generative AI-based tools may not provide significant time savings for complex tasks but they offer valuable benefits. These tools help developers quickly understand unfamiliar code bases, languages, or frameworks. Furthermore, developers can rely on these tools for assistance when facing new challenges, which is comparable to seeking guidance from an experienced colleague. This results in a 25-30% higher likelihood of completing complex tasks within the given timeframe compared to those who don't use such tools.

### 2.9.3 Which tasks demand developer expertise?

The study findings indicate that generative AI tools deliver effective results only when users possess strong skills. Research participants highlighted three areas where human oversight and examination are necessary when utilizing these tools. First, regarding examining code for bugs and errors researchers have observed instances where generative AI-powered technologies have produced errors in the code and provided inaccurate coding recommendations. To ensure precise code debugging, some developers had to send multiple prompts to correct the tool's assumptions, while others had to manually direct the tool by "spoon-feeding" it to achieve the desired outcome. Second, developers need to contribute more organizational context. Specifically, prebuilt generative AI tools have a lot of coding knowledge but they do not have an understanding of specific project and organizational

requirements. To make sure that the final software product is seamlessly integrated, performance and security standards are complied with and the end-user needs are effectively addressed, developers need to provide contextual information. This information includes details about the intended use of the code, user interactions, system interfaces, data usage, and more. Finally, developers should pay attention more to navigating tricky code requirements area as generative AI-powered tools work better on smaller, simpler tasks like code optimization than they would on larger, more difficult ones like combining multiple frameworks with diverse code logic. (Deniz et al. 2023d.)



### 3. Empirical research

#### 3.1 What is Leetcode?

LeetCode is an online platform for honing one's programming skills through practicing interview questions. With a diverse range of coding problems spanning various areas like data structures, algorithms, SQL, and decision trees, LeetCode provides a comprehensive learning experience. The platform presents each question with a detailed problem description that includes function specifications, parameters, and expected output. Moreover, the testing component offers multiple cases to evaluate the accuracy of user responses. For developers seeking to enhance their programming proficiency, LeetCode is an invaluable resource. It is a platform that offers support to developers of all linguistic proficiencies. LeetCode presents a diverse range of programming languages along with questions and associated test cases to ensure accessibility to users.

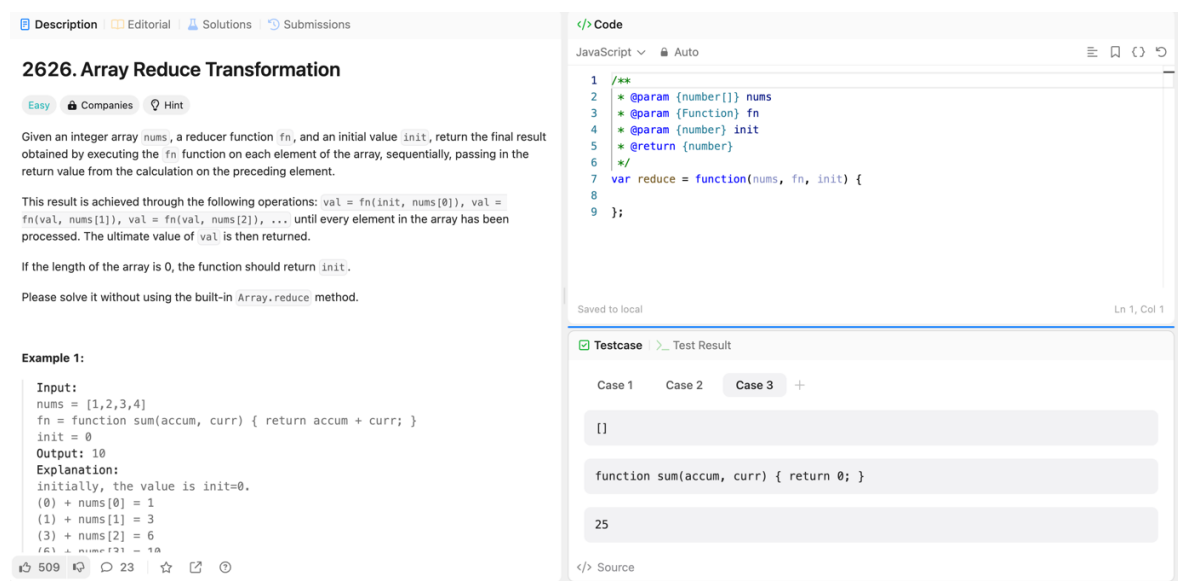


Figure 8. Leetcode's user interface for submitting problem solution

Additionally, the website provides a submission history that includes temporal data of submissions, particulars of attempted questions, submission status, runtime metrics, and the programming language employed. This feature enables users to keep track of their progress and identify areas of improvement.

All My Submissions

Time Submitted	Question	Status	Runtime	Language
1 day, 1 hour ago	Product of Array Except Self	Accepted	94 ms	javascript
1 day, 2 hours ago	Median of Two Sorted Arrays	Runtime Error	N/A	python

Newer
Older

Figure 9. Submission history tab on Leetcode

LeetCode offers five potential submission statuses that give users insights into the performance of their code. The first status, "Accepted," indicates that the provided solutions successfully pass all test cases. The second status, "Runtime error," implies that the submitted code encounters at least one test case failure due to execution errors. "Wrong answer", the third status, indicates that the submitted code is free of errors, yet produces output different from the expected output for at least one test case. The fourth status, "Compile error," demonstrates that the submitted code is incapable of compilation. The last status, "Time limit exceeded," suggests that the submitted code surpasses the allowable execution time for at least one test case while being free of errors.

### 3.2 Testing approach

In order to evaluate ChatGPT solutions, the author followed the steps below:

1. Randomly selected 30 questions with varying difficulty levels from Leetcode. To reduce any bias during the selection process, the author used the 'Pick one' option offered by the platform to randomly pick 5 easy, 15 medium, and 10 hard questions respectively.
2. The necessary information such as the problem description, function name, and parameters provided by Leetcode in the coding environment was extracted, which served as the prompts for ChatGPT in the next step.
3. For each problem, the author manually passed its description to ChatGPT as a prompt with the following format: "Write the code to solve the below question using JavaScript"

[description from Leetcode].

Sometimes, additional interaction with the model was required to provide the parameters for ChatGPT.

## 2624. Snail Traversal

Medium

Companies

Hint

Write code that enhances all arrays such that you can call the `snail(rowsCount, colsCount)` method that transforms the 1D array into a 2D array organised in the pattern known as **snail traversal order**. Invalid input values should output an empty array. If `rowsCount * colsCount !== nums.length`, the input is considered invalid.

**Snail traversal order** starts at the top left cell with the first value of the current array. It then moves through the entire first column from top to bottom, followed by moving to the next column on the right and traversing it from bottom to top. This pattern continues, alternating the direction of traversal with each column, until the entire current array is covered. For example, when given the input array `[19, 10, 3, 7, 9, 8, 5, 2, 1, 17, 16, 14, 12, 18, 6, 13, 11, 20, 4, 15]` with `rowsCount = 5` and `colsCount = 4`, the desired output matrix is shown below. Note that iterating the matrix following the arrows corresponds to the order of numbers in the original array.

Figure 10. A problem description on Leetcode

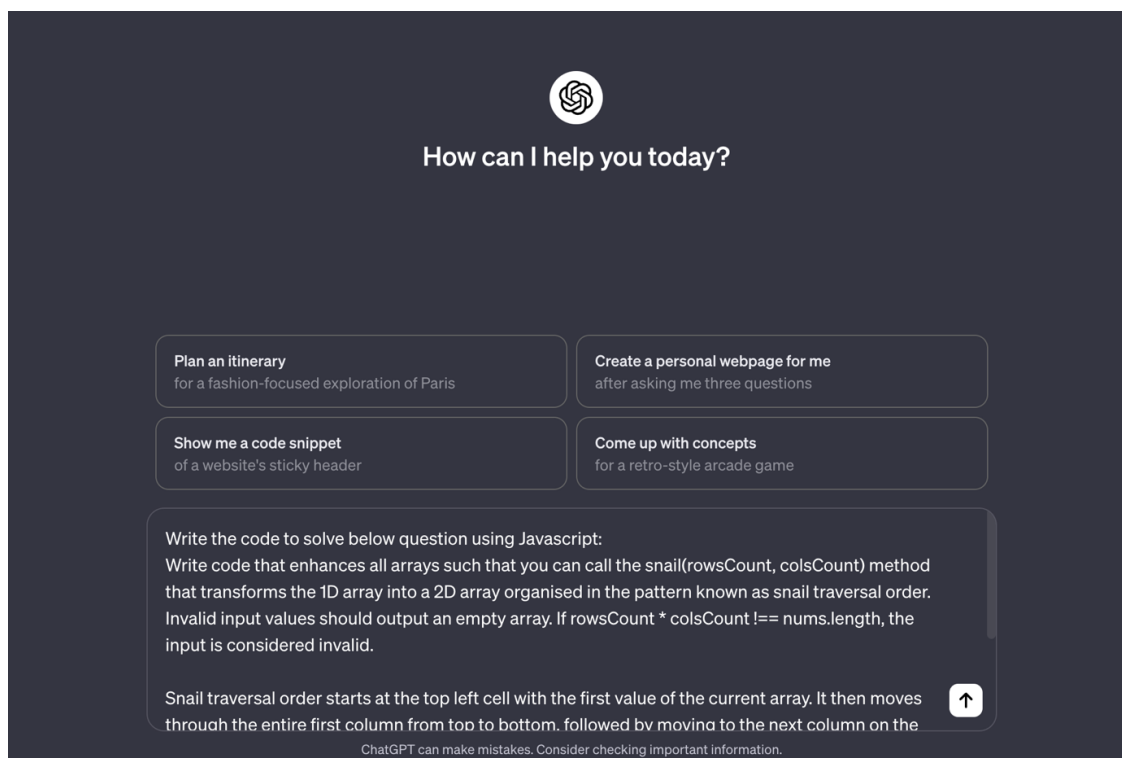


Figure 11. A prompt is entered into ChatGPT

Once receiving solutions from ChatGPT, the author proceeded to manually enter these solutions into Leetcode's coding environment. After entering the code, the author submitted it to Leetcode and carefully examined if it passed all the test cases provided by Leetcode. This process ensures that the solutions obtained from ChatGPT are accurate and reliable.

Wrong Answer

27 / 84 testcases passed

Editorial

Input

Use Testcase

[19, 10, 3, 7, 9, 8, 5, 2, 1, 17, 16, 14, 12, 18, 6, 13, 11, 20, 4, 15]

5

4

Output

[ [19,10,3,7], [18,6,13,9], [12,15,11,8], [14,4,20,5], [16,17,1,2]]

Expected

[ [19,17,16,15], [10,1,14,4], [3,2,12,20], [7,5,18,11], [9,8,6,13]]

Figure 12. Submission status for a solution

Figure 12 is the Leetcode submission tab which illustrates the number of test cases that a ChatGPT's solution has passed. Then, the author repeated the process for the next 29 questions. After submitting a total of thirty questions on Leetcode, the author accessed the submission history tab. This tab provides a valuable means by which to record the status of submitted answers, as well as any associated error messages received when a submission fails to satisfy the given test cases. Submission history allows the submitter to both monitor their progress and identify areas of their problem-solving skills in need of improvement. By

leveraging this functionality, the author can establish a comprehensive and informative record of submissions.

### 3.3 Research result

Question	Number of Tests	Test passed (%)	Runtime (ms) - (The submission's runtime beats [%] of JavaScript submissions)	Memory (MB) - (The submission's memory usage beats [%] of JavaScript submissions)
<b>Easy</b>				
2727. Is Object Empty	41	41 (100%)	58 - (26,24%)	49,2 - (11,20%)
2703. Return Length of Arguments Passed	104	104 (100%)	61 - (15,16%)	48,9 - (8,75%)
2666. Allow One Function Call	37	37 (100%)	52 - (61,75%)	48,7 - (10,21%)
2695. Array Wrapper	70	34 (48,57%)		
2619. Array Prototype Last	27	27 (100%)	37 - (98,72%)	48 - (16,71%)
<b>Medium</b>				
2649. Nested Array Generator	34	34 (100%)	227 - (12,92%)	79,2 - (56,86%)
2618. Check if Object Instance of Class	311	41 (13,18%)		
2637. Promise Time Limit	16	0 (0%)		
2693. Call Function with Custom Context	105	105 (100%)	64 - (16,70%)	48 - (12,94%)
2623. Memoize	20	20 (100%)	313 - (28,94%)	91 - (86,03%)
2627. Debounce	9	9 (100%)	49 - (91,86%)	49,8
2705. Compact Object	156	65 (41,67%)		
2694. Event Emitter	112	112 (100%)	43 - (96,80%)	50,7 - (8,66%)
2625. Flatten Deeply Nested Array	131	131 (100%)	101 - (70,66%)	68 - (64,91%)
2721. Execute Asynchronous Functions in Parallel	30	0 (0%)		
2622. Cache With Time Limit	9	9 (100%)	66 - (18,76%)	48,6 - (13,73%)
2631. Group By	86	86 (100%)	109 - (93,06%)	71,3 - (10,32%)
2722. Join Two Arrays by ID	42	42 (100%)	367 - (24,11%)	94,63 - (74,80%)
2653. Sliding Subarray Beauty	717	150 (20,92%)		
2747. Count Zero Request Servers	3129	16 (0,51%)		
<b>Hard</b>				
2650. Design Cancellable Function	23	2 (8,70%)		
2630. Memoize II	94	75 (79,79%)		
2584. Split the Array to Make Coprime Products	75	2 (2,67%)		
2751. Robot Collisions	2433	128 (5,26%)		
2781. Length of the Longest Valid Substring	763	0 (0%)		
2846. Minimum Edge Weight Equilibrium Queries in a Tree	733	1 (0,14%)		
2876. Count Visited Nodes in a Directed Graph	941	903 (95,96%)		
2926. Maximum Balanced Subsequence Sum	346	9 (2,60%)		
3008. Find Beautiful Indices in the Given Array II	237	225 (94,94%)		
2902. Count of Sub-Multisets With Bounded Sum	643	167 (25,97%)		

Accepted Wrong answer Runtime error Time limit exceeded

Figure 13. Research result

A comprehensive overview of all the LeetCode questions, along with the number of test cases for each question is shown in Figure 13. Additionally, it displays the number of tests

that the ChatGPT solution has passed, along with detailed information on the runtime and memory usage for that particular solution.

The pie chart below illustrates the distribution of ChatGPT's solutions for Leetcode problems. The data is based on 30 submissions.

- Accepted (13/30): Among all the submissions, the largest portion of the pie chart, amounting to 43.33%, is dedicated to the accepted solutions.
- Wrong Answers (12/30): Out of all the submissions received, 40% of them were marked as incorrect, making it the second-largest segment.
- Time Limit Exceeded (3/30): 10% of submitted answers have exceeded the allowed time limits according to data analysis.
- Runtime Errors (2/30): The chart shows that only a small portion of submissions, 6.67%, faced runtime errors.

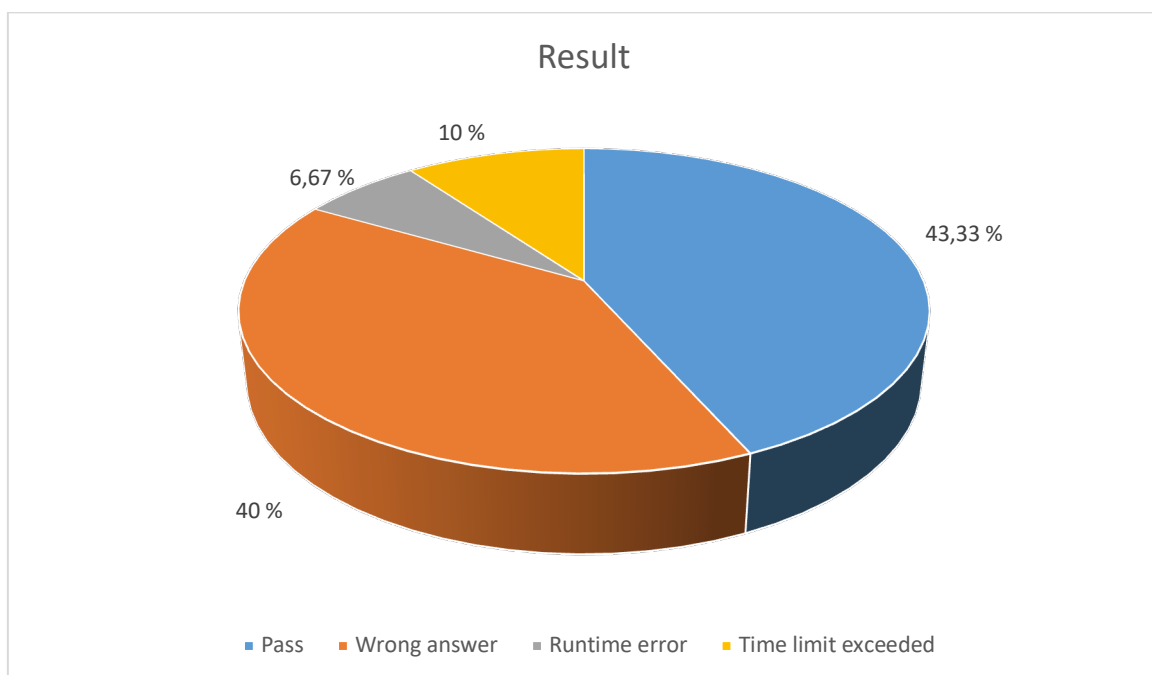


Figure 14. ChatGPT's results distribution

The pie chart in Figure 14 shows the research distribution of ChatGPT's solutions for 30 Leetcode problems. In summary, numerous submissions have effectively tackled problems of different levels of difficulty. However, the majority of successful solutions are in the easy

or medium complexity range. It appears that there is a lack of established solutions to address challenging coding problems present in ChatGPT training input data. This problem needs more attention and a stronger emphasis on addressing more complicated scenarios.

Furthermore, out of 13 passed solutions, ChatGPT's runtime is generally favorable, indicating that its recommended code is efficient. However, the high memory usage implies that ChatGPT's solutions are often not optimal. Figure 15 demonstrates the memory usage for a ChatGPT solution ranks 10.21% among JavaScript submissions, meaning a suboptimal performance. In certain scenarios, ChatGPT's solutions even surpass the memory limit. This suggests that while ChatGPT may solve the problem in most instances, its solution is not always ideal and requires improvement in the quality of the code.

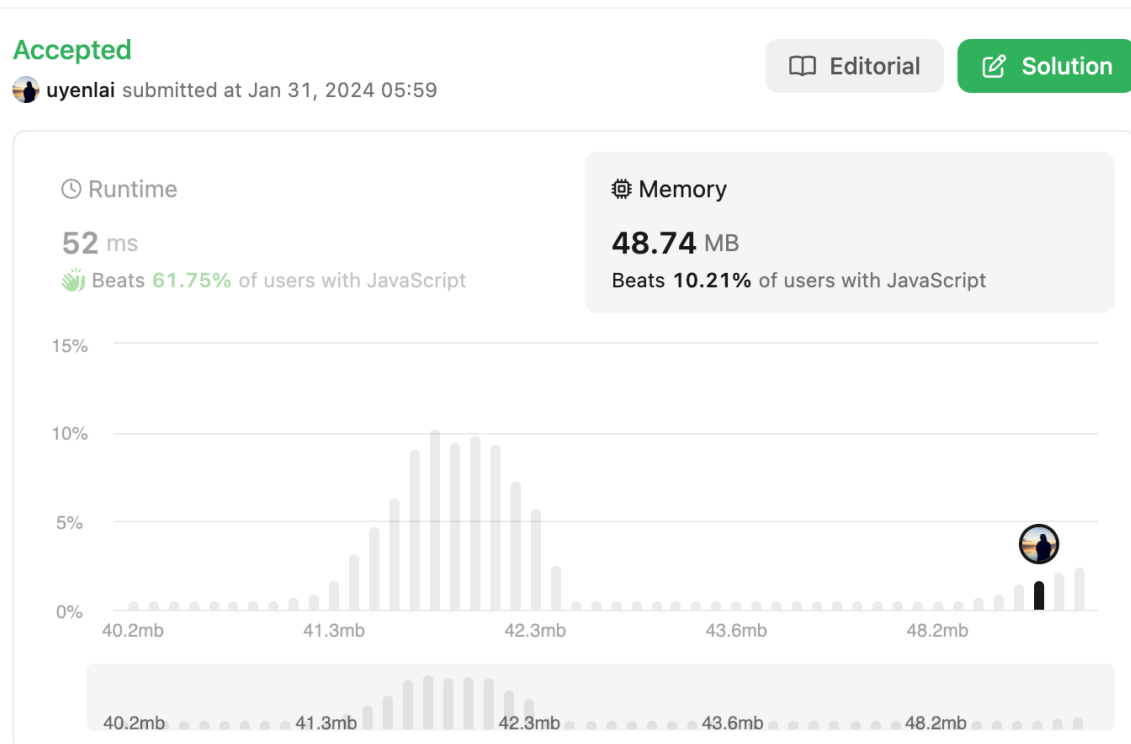


Figure 15. Memory usage of a solution

### 3.4 Conduct more testing for better insights

In order to gain more insights into ChatGPT's generated code, the author conducted more tests with one of the incorrect solutions. To achieve this, the author randomly picked a problem called "Count of Sub-multisets with bounded sum" which previously received the wrong



answer status. This process was carried out to determine if semantic errors were the primary reason for ChatGPT's incorrect solution. The author repeated the testing steps twice to ensure the accuracy of the findings.

In the first test, ChatGPT's suggested code was able to pass 27 out of 643 tests. In the second test, ChatGPT only passed 22 out of the 643 tests. The results show that ChatGPT's solution consistently scored lower, even though the author tried to improve the prompt box by adding more context and asking for an alternative solution if the first one was incorrect. Thus, ChatGPT's below-average performance is probably due to the quality of its training data rather than just semantic inaccuracies. Promptly addressing this issue is essential to enhance the reliability and effectiveness of ChatGPT's solutions.

## 4. Conclusion

The emergence of Artificial Intelligence (AI) has led to a significant transformation in our interaction with technology and has had a profound impact on various industries. AI applications have become ubiquitous in digital experiences, encompassing personalized recommendations on streaming services and voice-activated virtual assistants. The integration of AI into diverse industries, including healthcare, finance, and manufacturing, has been transformative. As artificial intelligence continues to advance, it will increasingly impact society and industry, creating new opportunities for innovation.

Generative AI is a rapidly growing field of artificial intelligence that has the potential to revolutionize technology. Unlike traditional AI, which analyzes existing data, generative AI has the remarkable ability to create new content in various forms such as text, images, and audio. The remarkable ability of generative AI is to produce content that is indistinguishable from that of a human. Generative AI is a highly beneficial tool that can be utilized across various domains. However, the use of GAI also poses certain risks, including the potential for the production of misleading or hallucinatory content. Additionally, deploying generative AI requires caution due to legal considerations such as copyright and privacy issues.

ChatGPT is a generative AI tool that has gained recognition for its ability to generate text across different fields. One of its most significant advantages is its capacity to assist developers by creating code snippets, which has proven to be a game-changer in the software development industry. This feature not only improves developer productivity but also significantly enhances their efficiency. Numerous research studies have demonstrated the significant advantages of integrating ChatGPT into coding practices. This AI-powered assistant greatly simplifies the process of writing and comprehending code, ultimately promoting faster and more efficient development.

Although ChatGPT is a robust platform, this study has brought to light some of its limitations. Research shows that ChatGPT's solutions are effective for simple coding problems, but their effectiveness decreases as challenges become more complex. It appears that ChatGPT may have certain limitations in handling complex programming issues, which raises concerns regarding its reliability in situations that require in-depth knowledge of complicated coding.

## **5. Research limitations and recommendations**

It is crucial to recognize and consider the various limitations of research and factors that impact the interpretation of findings. Firstly, the study has a limited scope as it evaluates ChatGPT's performance in only one programming language and uses a restricted set of 30 questions. Secondly, the author attempted each problem only once, except for one question that was tested three times to gain deeper insight. Consequently, only a relatively small number of solutions passed the tests. Finally, the study did not evaluate the specific error status of failed solutions, which could have contributed to an understanding of why ChatGPT generated inaccurate responses.

To summarize, this study highlights the importance of expanding the range of evaluation in ChatGPT's training data to improve its ability to solve problems. Additionally, it is important to prioritize natural language processing in order to strengthen ChatGPT's understanding of complex issues and enhance its performance in different coding scenarios.

## 6. Summary

Chapter 1 of the study presents a comprehensive overview of the emergence and impact of Artificial Intelligence (AI) in our daily lives. The thesis describes how artificial intelligence is transforming the way we live, work, and interact with the world around us. Furthermore, it identifies the research problems, objectives, questions, and methods used in the study.

Chapter 2 of the thesis explores theoretical concepts such as artificial intelligence (AI), machine learning, and generative AI. This section provides a detailed overview of the advantages and disadvantages of generative AI, the power of generative AI in the software development lifecycle, and how it can enhance developers' productivity. Furthermore, this chapter introduces the core idea of the thesis which is ChatGPT.

Chapter 3 presents the empirical research conducted to evaluate the accuracy of ChatGPT's solution for programming problems extracted from the Leetcode platform. The author also introduces the research tools, testing approaches, and research results.

Chapter 4 summarizes the key concepts discussed in the study, including AI, generative AI, and the central tool – ChatGPT. It illustrates ChatGPT's effectiveness in aiding developers' performance while also outlining its limitations.

Chapter 5 of the research explores the study's limitations and provides recommendations for future research. This chapter identifies areas for improvement that may guide other developers to use it with caution.

## References

Ayiro, P. 2012a. A functional approach to educational research methods and statistics: Qualitative, quantitative, and mixed methods approaches. Ebook. Lewiston: Mellen Press. Limited availability at <http://primo.lut.fi/lab>

Ayiro, P. 2012b. A functional approach to educational research methods and statistics: Qualitative, quantitative, and mixed methods approaches. Ebook. Lewiston: Mellen Press. Limited availability at <http://primo.lut.fi/lab>

Ayiro, P. 2012c. A functional approach to educational research methods and statistics: Qualitative, quantitative, and mixed methods approaches. Ebook. Lewiston: Mellen Press. Limited availability at <http://primo.lut.fi/lab>

Ayiro, P. 2012d. A functional approach to educational research methods and statistics: Qualitative, quantitative, and mixed methods approaches. Ebook. Lewiston: Mellen Press. Limited availability at <http://primo.lut.fi/lab>

Baker, P. 2023a. ChatGPT for Dummies. Ebook. Newark: John Wiley & Sons, Incorporated. Limited availability at <http://primo.lut.fi/lab>

Baker, P. 2023b. ChatGPT for Dummies. Ebook. Newark: John Wiley & Sons, Incorporated. Limited availability at <http://primo.lut.fi/lab>

Baker, P. 2023c. ChatGPT for Dummies. Ebook. Newark: John Wiley & Sons, Incorporated. Limited availability at <http://primo.lut.fi/lab>

Baker, P. 2023d. ChatGPT for Dummies. Ebook. Newark: John Wiley & Sons, Incorporated. Limited availability at <http://primo.lut.fi/lab>

Baker, P. 2023e. ChatGPT for Dummies. Ebook. Newark: John Wiley & Sons, Incorporated. Limited availability at <http://primo.lut.fi/lab>

Baker, P. 2023f. ChatGPT for Dummies. Ebook. Newark: John Wiley & Sons, Incorporated. Limited availability at <http://primo.lut.fi/lab>

Baker, P. 2023g. ChatGPT for Dummies. Ebook. Newark: John Wiley & Sons, Incorporated. Limited availability at <http://primo.lut.fi/lab>

Bell, E. 2023. Generative AI: How It Works, History, and Pros and Cons. Blog. Retrieved on 1 January 2024. Available at <https://www.investopedia.com/generative-ai-7497939> - toc-the-pros-and-cons-of-generative-ai

Blair, L. 2016a. Writing a Graduate Thesis or Dissertation. Ebook. Rotterdam: SensePublishers: Imprint: SensePublishers. Limited availability at <http://primo.lut.fi/lab>

Blair, L. 2016b. Writing a Graduate Thesis or Dissertation. Ebook. Rotterdam: SensePublishers: Imprint: SensePublishers. Limited availability at <http://primo.lut.fi/lab>

Deniz, K., B., Gnanasambandam, C., Harrysson, M., Hussin, A., & Srivastava, S. 2023a. Unleashing developer productivity with generative AI. Blog. Retrieved on 20 January 2024. Available at <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/unleashing-developer-productivity-with-generative-ai> - /

Deniz, K., B., Gnanasambandam, C., Harrysson, M., Hussin, A., & Srivastava, S. 2023b. Unleashing developer productivity with generative AI. Blog. Retrieved on 20 January 2024. Available at <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/unleashing-developer-productivity-with-generative-ai> - /

Deniz, K., B., Gnanasambandam, C., Harrysson, M., Hussin, A., & Srivastava, S. 2023c. Unleashing developer productivity with generative AI. Blog. Retrieved on 20 January 2024. Available at <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/unleashing-developer-productivity-with-generative-ai> - /

Deniz, K., B., Gnanasambandam, C., Harrysson, M., Hussin, A., & Srivastava, S. 2023d. Unleashing developer productivity with generative AI. Blog. Retrieved on 20 January 2024. Available at <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/unleashing-developer-productivity-with-generative-ai> - /

Ewell, R. 2023. The Advantages Of Disadvantages Of Using Generative AI In Your Business. Blog. Retrieved on 30 December 2023. Available at <https://russewell.medium.com/the-advantages-of-disadvantages-of-using-generative-ai-in-your-business-be3950b7cac7> - ~:text=Generative AI presents exciting opportunities,complexity must be carefully considered

Feuerriegel, S., Hartmann, J., Janiesch, C. & Zschech, P. 2024a. Generative AI. Business & information systems engineering 66, no. 1 (111–126). Retrieved on 4 January 2024. Available at DOI: 10.1007/s12599-023-00834-7

Feuerriegel, S., Hartmann, J., Janiesch, C. & Zschech, P. 2024b. Generative AI. Business & information systems engineering 66, no. 1 (111–126). Retrieved on 4 January 2024. Available at DOI: 10.1007/s12599-023-00834-7

Feuerriegel, S., Hartmann, J., Janiesch, C. & Zschech, P. 2024c. Generative AI. Business & information systems engineering 66, no. 1 (111–126). Retrieved on 4 January 2024. Available at DOI: 10.1007/s12599-023-00834-7

Feuerriegel, S., Hartmann, J., Janiesch, C. & Zschech, P. 2024d. Generative AI. Business & information systems engineering 66, no. 1 (111–126). Retrieved on 4 January 2024. Available at DOI: 10.1007/s12599-023-00834-7

GitHub a. About GitHub Copilot for Individuals. Blog. Retrieved on 15 May 2023. Available at <https://docs.github.com/en/copilot/overview-of-github-copilot/about-github-copilot-for-individuals>

GitHub b. About GitHub Copilot for Individuals. Blog. Retrieved on 15 May 2023. Available at <https://docs.github.com/en/copilot/overview-of-github-copilot/about-github-copilot-for-individuals>

Gupta, N. & Mangla, R. 2020a. Artificial Intelligence Basics. Ebook. Dulles, Virginia; Boston, Massachusetts; New Delhi: Mercury Learning and Information. Limited availability at <http://primo.lut.fi/lab>

Gupta, N. & Mangla, R. 2020b. Artificial Intelligence Basics. Ebook. Dulles, Virginia; Boston, Massachusetts; New Delhi: Mercury Learning and Information. Limited availability at <http://primo.lut.fi/lab>

Gupta, N. & Mangla, R. 2020c. Artificial Intelligence Basics. Ebook. Dulles, Virginia; Boston, Massachusetts; New Delhi: Mercury Learning and Information. Limited availability at <http://primo.lut.fi/lab>

IBM 2023. AI code-generation software: What it is and how it works. Blog. Retrieved on 5 February 2024. Available at <https://www.ibm.com/blog/ai-code-generation/>

Lin, B. 2023a. Generative AI Helping Boost Productivity of Some Software Developers; Enterprise technology chiefs still have questions about the security and governance of the code-writing programs. The Wall Street journal. Retrieved on 4 January 2024. Available at <https://www.proquest.com/docview/2778460488?parentSessionId=jspwHlyvduxs9%2FIiGzQ3jN7kDpJ9oZ5B5cMJrClhfE=&pq-origsite=primo&accountid=202350&sourcetype=Newspapers>

Lin, B. 2023b. Generative AI Helping Boost Productivity of Some Software Developers; Enterprise technology chiefs still have questions about the security and governance of the code-writing programs. The Wall Street journal. Retrieved on 4 January 2024. Available at <https://www.proquest.com/docview/2778460488?parentSessionId=jspwHlyvduxs9%2FIiG>

zQ3jN7kDpJ9oZ5B5cMJrClhfE=&pq-  
origsite=primo&accountid=202350&sourcetype=Newspapers

Marr, B. 2023. The Difference Between Generative AI And Traditional AI: An Easy Explanation For Anyone. Blog. Retrieved on 27 December 2023. Available at <https://www.forbes.com/sites/bernardmarr/2023/07/24/the-difference-between-generative-ai-and-traditional-ai-an-easy-explanation-for-anyone/>

O’Gorman, K. & MacIntosh, R. 2015. Research Methods for Business and Management: A Guide to Writing Your Dissertation. Ebook. Oxford: Goodfellow Publishers, Limited. Limited availability at <http://primo.lut.fi/lab>

Ong, TJ. 2023. Debugging and Refactoring with Generative AI. Blog. Retrieved on 8 February 2024. Available at <https://medium.com/@teejayoh/debugging-and-refactoring-with-generative-ai-4ff7f0b82975>

Pattam, A. 2023. Generative AI in Software Engineering. Blog. Retrieved on 6 February 2024. Available at <https://arunapattam.medium.com/generative-ai-in-software-engineering-fc7da0710050>

Robert, E. & Schmidt, D. 2023. Generative AI Q&A: Applications in Software Engineering. Blog. Retrieved on 4 February 2024. Available at DOI: <https://doi.org/10.58012/m5kp-ma05>

Streefkerk, R. 2023a. Inductive vs. Deductive Research Approach | Steps & Examples. Scribbr. Retrieved on 12 February 2024. Available at <https://www.scribbr.com/methodology/inductive-deductive-reasoning/>

Streefkerk, R. 2023b. Inductive vs. Deductive Research Approach | Steps & Examples. Scribbr. Retrieved on 12 February 2024. Available at <https://www.scribbr.com/methodology/inductive-deductive-reasoning/>

Szatmary, A. 2016. Using SMART Goals to Make Scientific Progress. Blog. Retrieved on 14 February 2024. Available at <https://irp.nih.gov/blog/post/2016/07/using-smart-goals-to-make-scientific-progress>

Wang, G. & Park, K. 2016a. Student research and report writing: from topic selection to the complete paper. Ebook. Chichester, England: Wiley Blackwell. Limited availability at <http://primo.lut.fi/lab>

Wang, G. & Park, K. 2016b. Student research and report writing: from topic selection to the complete paper. Ebook. Chichester, England: Wiley Blackwell. Limited availability at <http://primo.lut.fi/lab>



Williamson, K. & Bow, A. 2002a. Research Methods for Students, Academics and Professionals: Information Management and Systems. Ebook. Wagga Wagga, New South Wales: Centre for Information Studies. Limited availability at <http://primo.lut.fi/lab>

Williamson, K. & Bow, A. 2002b. Research Methods for Students, Academics and Professionals: Information Management and Systems. Ebook. Wagga Wagga, New South Wales: Centre for Information Studies. Limited availability at <http://primo.lut.fi/lab>

Zohuri, M. & Zadeh, S. 2020a. Artificial Intelligence Driven by Machine Learning and Deep Learning. Ebook. New York: Nova Science Publishers. Limited availability at <http://primo.lut.fi/lab>

Zohuri, M. & Zadeh, S. 2020b. Artificial Intelligence Driven by Machine Learning and Deep Learning. Ebook. New York: Nova Science Publishers. Limited availability at <http://primo.lut.fi/lab>

Zohuri, M. & Zadeh, S. 2020c. Artificial Intelligence Driven by Machine Learning and Deep Learning. Ebook. New York: Nova Science Publishers. Limited availability at <http://primo.lut.fi/lab>

Zohuri, M. & Zadeh, S. 2020d. Artificial Intelligence Driven by Machine Learning and Deep Learning. Ebook. New York: Nova Science Publishers. Limited availability at <http://primo.lut.fi/lab>

Zohuri, M. & Zadeh, S. 2020e. Artificial Intelligence Driven by Machine Learning and Deep Learning. Ebook. New York: Nova Science Publishers. Limited availability at <http://primo.lut.fi/lab>