Artificial Intelligence in Industry 4.0: The future that comes true: AI

Karabegović, Isak; editor

2024-09-17

https://bastina.anubih.ba/handle/123456789/791

DOI: 10.5644/PI2024.215.08

Prompt Engineering

Samir Lemeš[*1]

Abstract:*Prompt engineering is the process of designing, testing, and optimizing prompts that are sent to artificial intelligence (AI), especially large language models like GPT-4. The goal is to formulate the prompts in a way that allows the AI model to provide the most relevant, accurate and useful answers. This process involves understanding how the AI model interprets and responds to different query formulations, as well as tailoring those queries for different applications. Well-designed prompts can significantly improve the performance of AI systems, making them more useful and efficient for various applications. In the context of Industry 4.0, where AI is used for production optimization, equipment maintenance, data analytics and other critical functions, more efficient prompts can contribute to increased productivity and reduced costs. Prompt engineering is an interdisciplinary field that requires understanding of natural language, artificial intelligence, and domain-specific applications. As AI models become more sophisticated, the role of prompt engineering will become even more important in leveraging their full potential.*

Keywords: *Machine Learning, Large Language Models (LLM), Prompt, Industry 4.0*

1. Introduction

Prompt engineering is the process of designing, testing, and optimizing prompts that are sent to large language models such as GPT. It is a new technique that is interdisciplinary and requires a wide range of other skills that are not always necessarily limited to the field for which artificial intelligence is used. Prompt engineering is an increasingly popular topic for wide research in various fields. Gu et al. have attempted to provide a comprehensive overview of recent research on prompt engineering across three types of visual language models: multimodal-to-text generation models (e.g., Flamingo), image-to-text matching models (e.g., CLIP), and image-to-text generation models (e.g., Stable Diffusion) [1]. Although they refer to 216 literature sources, their research was limited to pre-trained models only.

Strobelt et al. have developed a workflow that allows users to focus on model feedback using small data before moving to a big data mode that allows for empirical grounding of prompts using quantitative measures of the task [2].

[*1]University of Zenica, Polytechnic Faculty, Zenica, Bosnia and Herzegovina
ORCID: 0000-0002-3596-645X, E-mail: samir.lemes@unze.ba

They have developed an open-source system (http://prompt.vizhub.ai/) that can work with any available language model backend.
Marvin et al. in [3] provided a thorough understanding of prompt engineering, with relevant exercises for applying these engineering techniques in practice, current and future LLM trends, and prompt engineering research, including the rise of automatic instruction generation and query selection methods.
Sahoo et al. have addressed the lack of systematic organization and understanding of various prompt engineering methods and techniques by providing a structured overview of recent advances in prompt engineering, categorized by application area with a description of the methodology, its application, the models involved, and the datasets used [4]. They have also addressed the strengths and limitations of each approach, including a taxonomy diagram and table summarizing the datasets, models, and critical points of each technique.
Ekin in [5] provided a comprehensive guide to mastering prompt engineering techniques, tips and best practices for achieving optimal results for ChatGPT. He also covered best practices, including iterative refinement, balancing user intent, leveraging external resources, ensuring ethical usage, and advanced strategies such as temperature and token control, fast chaining, domain-specific customizations, and handling ambiguous inputs.
Henao, Franco-Cardona and Cadavid-Higuita introduced a methodology to optimize interaction with language models of artificial intelligence, such as ChatGPT, through prompt engineering [6]. They called the approach GPEI, which consists of four steps: define the goal, design the prompt, evaluate the response, and repeat. Their proposal includes two key aspects: incorporating data into the design of prompts for engineering applications and integrating explainable artificial intelligence principles to evaluate responses, increasing transparency.
Shin et al. investigated the effectiveness of GPT-4 LLMs with three different prompt engineering techniques (basic prompting, learning in context, and task-specific prompting) against 18 fine-tuned LLMs on three typical automated software engineering (ASE) tasks: code generation, code compression and code translation [7]. They concluded that GPT-4 with conversational prompting (i.e. when a human provides feedback and instructions to the model to achieve the best results) showed a drastic improvement compared to GPT-4 with automatic prompting strategies. They observed that participants tended to seek improvements, add more context, or provide specific instructions as conversational prompts, which went beyond typical and generic prompting strategies.
Korzynski et al. in [8] aimed to create a theoretical framework that would highlight optimal approaches in the field of prompt engineering for AI. The study revealed the profound implications of prompt engineering for the

application of artificial intelligence in various domains such as entrepreneurship, art, science and health. They have shown how efficient prompting can significantly improve the performance of large language models (LLMs), generating more accurate and contextually relevant results.
López-Riobóo-Botana, Gallent-Iglesias and Gonzalez-Vázquez presented QUA4I (Question Answering for the Industry 4.0), a chatbot or IVA (Intelligent Virtual Assistant) oriented application for the Industry 4.0 domain, mixing NLU and NLG techniques and using Rasa chatbot framework [9]. They designed a custom demo for answering questions and extracting information about Industry 4.0, including a dialogue system that can generate automatic responses in natural language. They have also included ASR (Automatic Speech Recognition) and TTS (Text to Speech) modules, so one can interact with the bot using Spanish language speech.
Tinnes et al. have explored the use of largelanguage models for the automatic conversion of unstructured asset data into industry-relevant structures in the Asset Administration Shell (AAS) [10]. For some product categories, they showed that full automation is possible — in the worst case, 17% of the structures had to be corrected manually. They provided a new set of real-world industrial data as well as tools for comparing AAS structures for future studies.
Blasek et al. investigated the quality of LLM outputs to define requirements for Digital Twins [11]. The main contribution of their work is the results of an experiment that compared requests from DT air conditioners and ChatGPT and the prompt engineering conclusions that resulted from that experiment.
Howell investigated how artificial intelligence can be used in manufacturing processes [12]. He showed how prompt can help transform manufacturing processes, for example by improving predictive analytics to improve supply chain management through proactive risk detection and mitigation.

2. Large Language Models and Natural Language Processing

One of the most fascinating features of the technology that became freely and publicly available in November 2022 is its ability to understand different languages. Unlike previous search services, which required precise and correct entry of search terms, large language models, on which services such as OpenAI ChatGPT, Microsoft Copilot or Google (Bard) Gemini are based, can process even hybrid sentences in which different languages are mixed.
The terminology used by software engineers and other IT professionals is a kind of mixture of English and other languages, because some terms are simply untranslatable. Figure 1 shows a ChatGPT prompt made in such a hybrid language, and the response shows that the model understands such mixture of languages without any problems. The model recognizes even an incorrect transcription (the word "diplojamo" is derived from the English term "deploy",

which is not translated, but is already a common incorrect transcription that is perfectly understood by people involved in software development).

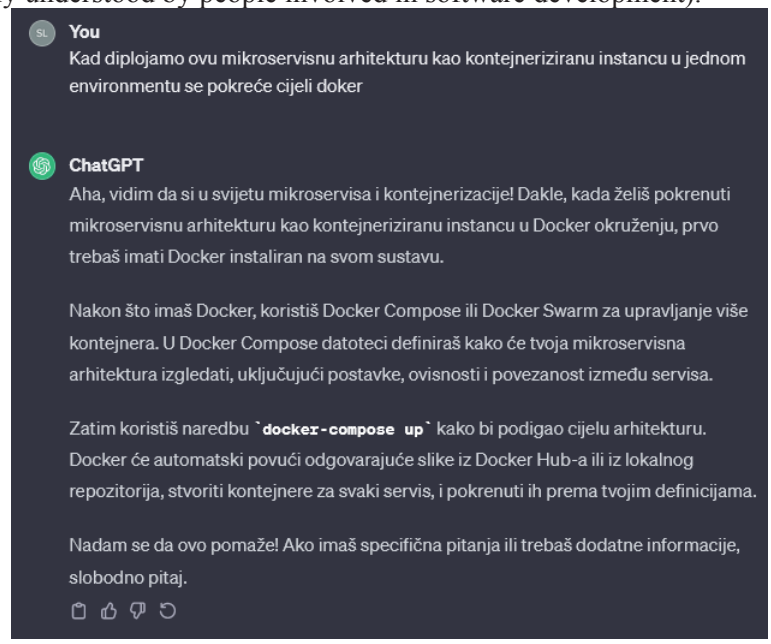

Figure 1. An example of a reply to a query made to ChatGPT with a mixture of incorrect English transcriptionand Bosnian language

Large Language Models (LLMs) are the result of decades of research in the field of Artificial Intelligence (AI) and Natural Language Processing (NLP). They were developed thanks to advances in machine learning, especially deep learning, and the vast amount of textual data available. Algorithms in early attempts (until the 1980s) of language processing using rules and small dictionaries were based on fixed rules (Rule-Based Systems). Statistical methods used in the 1990s, such as n-gram models, which used large amounts of data to learn the probabilities of word sequences, had limitations in understanding context. The introduction of neural networks, especially Recurrent Neural Networks (RNN) and Long-Term Memory Networks (LSTM), enabled a better understanding of sequences and contexts, or models based on deep learning. Transformer architecture, presented in 2017 [13] enabled parallel data processing, which significantly speeded up training and improved performance. The development of models such as OpenAI GPT-3 (Generative Pre-trained Transformer 3), with 175 billion parameters, is trained on huge collection of data from the Internet, which allowed them to generate high-quality text and answer complex queries.

Large language models use several techniques to recognize and to correct the misspelled text:
- Contextual understanding: Using transformers, models analyse the context of the entire text, which allows them to recognize irregularities and generate correct versions.
- Ability to generalize: They are trained on many different texts, including texts with errors. This helps them recognize and correct common mistakes.
- Correlation of sound and spelling: In poor transcription, models use phonetic similarities and context to infer which word was most likely intended to be used.
- Translation errors: When recognizing text that is the result of poor transcription or translation, the models use parallel sets of texts and their translations to identify and correct errors.

Today, large language models are used in numerous applications and can recognize and correct irregularities in text thanks to their ability to understand context and generalize from large bodies of data. These technologies continue to evolve, making interaction with machines more and more natural and efficient.
It is important to note that LLMs are not perfect and can sometimes make mistakes. However, they are constantly improving and getting better at recognizing and correcting errors in the text.

3. Possible Consequences of Using LLM and NLP

The use of large language models and NLP technologies in Industry 4.0 brings significant advantages, but also potential risks. Positive consequences include increasing productivity, improving user experience, improving quality and safety, and encouraging innovation. On the other hand, negative consequences include privacy concerns, ethical issues, labour substitution, and technical complexity. The role of prompt engineering is key to optimizing model performance and tailoring their responses to specific industry needs, which is essential for success and sustainability in the Industry 4.0 era.
The positive consequences of using LLM and NLP in Industry 4.0 include:
- Increased productivity through automation and faster decision-making: LLM and NLP enable the automation of routine and repetitive tasks, such as answering user queries, analysing data and generating reports. The integration of NLP technologies can speed up the analysis and decision-making process, enabling faster reactions to market changes.
- Improving user experience through personalization and ongoing support: LLMs enable personalized communication with users, providing responses tailored to individual needs and preferences. Chatbots and virtual assistants enable constant (24/7) support, improving customer satisfaction.

- Improving quality and safety: NLP can help analyse customer feedback and automatically identify product quality issues. Analysis of text data can identify potential security threats and risks.
- Innovation and research: Analysing large amounts of data can reveal market trends and needs, enabling the development of innovative products and services. NLP can speed up the process of searching and analysing scientific papers, facilitating research and development.

Of course, there are also potentially negative consequences of using these technologies:

- Threat to privacy: The use of LLM often requires the collection and analysis of large amounts of data, which may pose a risk to user privacy. There is a risk of misuse of collected data, which can lead to problems with user trust.
- Ethical issues reflected through bias and manipulation of information: LLMs can inherit biases from the data they are trained on, which can lead to discriminatory decisions or content. NLP technologies can be used to spread misinformation or manipulate public opinion.
- Labor replacement: One of the most frequently used arguments against the introduction of Industry 4.0 is the automation of jobs, which can lead to the loss of jobs, especially those involving routine tasks. Employees will need to adapt to new technologies, which may require additional training and education. Someone accepts it as a problem, and someone as an opportunity.
- Technical complexity as a challenge in implementation and maintenance: Integrating LLM and NLP technologies can be technically demanding and expensive, which can be a challenge for smaller companies. Complex technology systems can be prone to errors and breakdowns, which can affect operational efficiency.

Prompt engineering refers to the design and optimization of queries sent to large language models to obtain the desired responses. In the context of Industry 4.0, prompt engineering plays a key role in several aspects:

- Model performance optimization: Well-formed queries can significantly improve the accuracy and relevance of the answers generated by the model, which is crucial for automated decision systems.
- Improving user experience: Through prompt engineering, models can be tuned to provide responses tailored to specific user needs or industry requirements.
- More efficient data analysis: It enables the creation of specific focused queries to extract relevant information from large data sets, facilitating analysis and making informed decisions.

- Innovation and research: Prompt engineering can encourage innovative and creative ways of using language models to explore new ideas and develop new products and services.
- Customized training and development of employees: Using prompt engineering, language models can provide personalized training and education programs tailored to the specific needs of employees.

Prompt engineering has the potential to become not only a technique, but even one of the most sought-after professions in the future.
After the initial enthusiasm for the possibilities provided by generative models of artificial intelligence, the first concerns also appeared, especially in IT industry. Some occupations will surely die out, as generative AI models can not only generate new but also optimize existing code for software. Some predictions [14] claim that the job of writing software code will be taken over by generative models of artificial intelligence in the next few years.

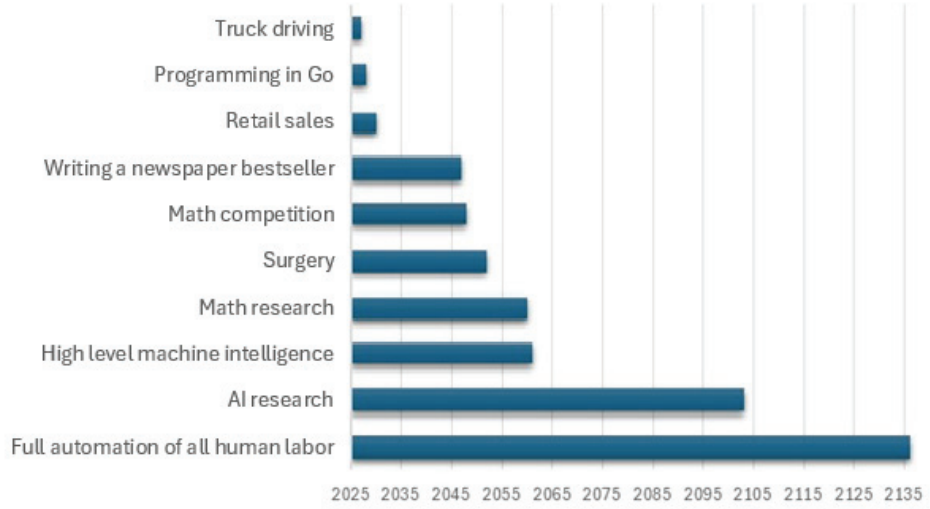

Figure 2. Predictions of when certain professions will be replaced by artificial intelligence [14]

4. Structure and Syntax of the Prompt

The results generated by AI depend not only on the size of the database on which language models are trained, but even more so on the formulation of the task. In short, prompt engineering is the skill of asking questions to a chatbot. It is a multidisciplinary skill, combining linguistics, logic, philosophy, engineering, etc. to achieve the best possible results from data generators based

on artificial intelligence. The precision and accuracy of querying with as much information and specific instructions as possible helps the model to generate the expected output, and to avoid generating irrelevant answers.

When defining a query (prompt), the rule "Garbage In = Garbage Out" applies, that is, a bad input usually generates a bad output. At the same time, the term "bad" does not refer to correct writing (spelling, grammar, language rules, terminology) at all, but rather to the formulation of the problem. This can be illustrated by an old programmer's joke, in which the wife instructs the programmer husband: "Go to the store, buy a butter, and if there are eggs, buy ten". The programmer came home with ten butters and the sentence "There were eggs". In this case, it is an insufficiently precisely formulated instruction that the programmer (and in all cases the software) does not critically analyse the instruction, but blindly listens to it in the form in which it was given. Mechanical response to insufficiently precise instructions cannot always be expected to produce the desired result.
The term "Garbage In = Garbage Out" in the case of applying generative AI models means that the quality of the output largely dependson the quality of the data corpus on which the model was trained. If that data is wrong, of course the result will also contain the error it inherited from the bad input.
However, the quality of the results can be influenced by knowing the correct structure of the prompt (Figure 3). In the introduction, a context should be set, which helps to give the AI model an imaginary 'role' to think about itself.

Phase	**Example**
Introduction	Act like a software engineer. You are an expert in Python and…
Task	I want you to develop software to manage your DVD collection.
Contextual information	I want it to be a web application written in Python.
Instructions	I want you to generate the source code.
Closing	I want it to be an AWS Lambda function.

Figure 3. Structure of a good prompt for generative AI models with examples

Separating instructions from context in a chatbot is an important step to achieve accuracy and relevance in responses. When asking questions or giving instructions, the user must be precise and clear. Specific language should be used, and ambiguity should be avoided. Key words in the instructions that are relevant to the desired task should be identified. The chatbot needs to analyse the context before executing the instructions.

Instructions are placed at the beginning of the prompt, denominators ### or """ can be used to separate the instructions from the context, and the context is separated by an extra blank line. For example:

```
### Instructions ###
Write code to sort an array of names

""" A series of names: Sarajevo, Zenica, Tuzla,
Banja Luka, Mostar """
```

When formulating the prompt, user should be as precise as possible. Instead of "Write a poem about OpenAI" one should write "Write a short inspirational poem about OpenAI, focusing on the recent launch of DALL-E products in the style of {famous poet}". The desired output can be articulated through several examples. Giving examples can improve the quality of results. A positive prompt should be used (instead of "don't use a negative prompt"). Instructions should be clear and concise, avoiding ambiguity and unnecessary complexity. Context and specifics should always be considered. For example, the expressions "I'm sorry" and "My bad" do not sound the same in real life and at the funeral. A conversational style should be used instead of giving instructions. Rhetorical questions should be used and the active instead of the passive, for example instead of "An essay on the benefits of exercise should be written." (object + verb + subject) should be written "Can you write a persuasive essay on the benefits of exercise?" (subject + verb + object). For certain purposes, such as writing software code, there are generic queries that give good results (Figure 4).

Scan the following code for potential problems	Even if the code executes successfully, potential problems can be discovered
Evaluate the following code and look for performance issues	Improving code performance
Write a test for the following {language} code	Generating tests to examine the code
Explain how {something} works in {language}.	It explains how the borrowed code logic works
Translate the following {first language} code into {second language}.	I'm not learning a new programming language, translate C# to Java for me
What is the correct syntax to {do something} in {language}?	How do I send an HTTP header in Python?
Write a function to {do something} in {language}.	Write me a function to connect to a MongoDB database in Rust

Figure 4. Examples of well-worded software coding prompts

5. What is next?

The European Commission proposed the first regulatory framework for artificial intelligence in April 2021 [15]. This framework, known as the "Artificial Intelligence Act", sets the rules for the development, deployment and use of AI in the European Union to make AI systems safe, transparent, traceable and non-discriminatory, to ensure safety and fundamental human rights, and at the same time to encourage innovation.

ChatGPT introduced a completely new challenge to the education system – the paradigm of education must be changed. Writing essays and term papers can no longer be used to evaluate students, because it is practically impossible to determine the originality of such works, that is, such works do not prove the competence of those who claim to be the authors of those works.

The wide availability and prevalence of multiple generic AI models multiply the problems of plagiarism, wrong data and privacy violations. Some schools and states have tried to ban the use of new technologies in education, attempting to protect against misuse. The ban is not only ineffective, but even counterproductive. Instead of banning, it is necessary to learn how to use AI for personal and general benefit.

The development of technology is unstoppable, so we just need to learn how to use technology properly. The fear of losing jobs should be used to strengthen skills that help achieve better results at work with the support of modern technologies such as LLM, NLP and AI. It is essential to develop critical thinking and logic, which are disciplines that are almost unknown and even undesirable in our educational system. Without critical thinking and knowing the advantages and disadvantages of technology, there is no progress and catching up with developed countries.

6. Conclusion

Artificial intelligence has the potential to replace boring and repetitive jobs, just as robots have replaced manual labour. The role of the software developer is changing drastically with the advent of AI: instead of coding, the programmer will control artificially generated code, he/she will have to develop creative thinking, intuition, and strengthen expertise in a certain domain. All of these are new skills that must be included in the list of competencies for modern engineers on which Industry 4.0 rests. AI will not replace but complement software developers. It should be noted that LLM is not a real artificial intelligence, it is just an excellent autocomplete tool or a top search engine. True AI (Strong AI) emerges only when an autonomous system can surpass human abilities in most economically valuable tasks.

7. References

[1] Gu, J., Han, Z., Chen, S., Beirami, A., He, B., Zhang, G., ... & Torr, P. (2023). A systematic survey of prompt engineering on vision-language foundation models. arXiv preprint arXiv: 2307.12980 https://doi.org/10.48550/arXiv.2307.12980

[2] Strobelt, H., Webson, A., Sanh, V., Hoover, B., Beyer, J., Pfister, H., & Rush, A. M. (2022). Interactive and visual prompt engineering for ad-hoc task adaptation with large language models. IEEE transactions on visualization and computer graphics, 29(1), 1146-1156. https://doi.org/10.48550/arXiv.2208.07852

[3] Marvin, G., Hellen, N., Jjingo, D., Nakatumba-Nabende, J. (2024). Prompt Engineering in Large Language Models. In: Jacob, I.J., Piramuthu, S., Falkowski-Gilski, P. (eds) Data Intelligence and Cognitive Informatics. ICDICI 2023. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-99-7962-2_30

[4] Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. arXiv preprint arXiv: 2402.07927. https://doi.org/10.48550/arXiv.2402.07927

[5] Ekin, S. (2023). Prompt engineering for ChatGPT: a quick guide to techniques, tips, and best practices. Authorea Preprints. http://dx.doi.org/10.36227/techrxiv.22683919

[6] Velásquez-Henao, J. D., Franco-Cardona, C. J., & Cadavid-Higuita, L. (2023). Prompt Engineering: a methodology for optimizing interactions with AI-Language Models in the field of engineering. Dyna, 90(230), 9-17. https://doi.org/10.15446/dyna.v90n230.111700

[7] Shin, J., Tang, C., Mohati, T., Nayebi, M., Wang, S., & Hemmati, H. (2023). Prompt Engineering or Fine Tuning: An Empirical Assessment of Large Language Models in Automated Software Engineering Tasks. arXiv preprint arXiv:2310.10508. https://doi.org/10.48550/arXiv.2310.10508

[8] Korzynski, P., Mazurek, G., Krzypkowska, P., &Kurasinski, A. (2023). Artificial intelligence prompt engineering as a new digital competence: Analysis of generative AI technologies such as ChatGPT. Entrepreneurial Business and Economics Review, 11(3), 25-37. https://doi.org/10.15678/EBER.2023.110302

[9] López-Riobóo-Botana, I., Gallent-Iglesias, D., & Gonzalez-Vázquez, S. (2023). QUA4I: Question Answering for the Industry 4.0 Domain. An Application of Intelligent Virtual Assistants.https://ceur-ws.org/Vol-3516/paper17.pdf

[10] Tinnes, C., Ristin, M., Hohenstein, U., Fathi, K., & van de Venn, H. W. (2024).From Unstructured Product Descriptions to Structured Data for Industry 4.0 with ChatGPT. International Conference on Industrial Cyber-Physical Systems (ICPS)

[11] Blasek, N., Eichenmüller, K., Ernst, B., Götz, N., Nast, B., &Sandkuhl, K. (2023). Large language models in requirements engineering for digital twins. 2nd International Workshop on Digital Twin Engineering (DTE), Technical University Vienna, Vienna, Austria

[12] Howell, J. (2024) How Prompt Engineering is Revolutionizing the Manufacturing Industry?, https://101blockchains.com/prompt-engineering-in-manufacturing/[Dostupno: 11.06.2024]

[13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30. https://doi.org/10.48550/arXiv.1706.03762

[14] Grace, K., Salvatier, J., Dafoe, A., Zhang, B., & Evans, O. (2018). Viewpoint: When will AI exceed human performance. Evidence from AI experts. JArtifIntellRes62, 729-754. https://doi.org/10.1613/jair.1.11222

[15] European Commission (2021). AI Act https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai [Dostupno: 14.06.2024]