UDC: 004.93 https://doi.org/10.15407/jai2024.02.107

M. Shash
State University of Information and Communication Technology, Ukraine
7, Solomyanska Str., Kyiv, 03110
max.shash@gmail.com
https://orcid.org/0009-0009-3274-5318

LEVERAGING LANGCHAIN AGENTS TO AUTOMATE DATA ANALYSIS FOR SAAS

Abstract. This paper investigates the implementation of LangChain, a language model-powered framework, in automating data analysis within the SaaS sector.The approach included setting up LangChain agents for exploratory, univariate, and bivariate analyses, as well as hypothesis testing, transforming extensive data into human language text answers. Experiments confirmed the effectiveness of the proposed method using GPT-3.5 LLM agents, tested on the Amazon AWS SaaS Sales Dataset. Identified deficiencies need to be addressed for complex queries and comprehensive reports. Future research prospects include improving the method for complex queries, providing more detailed information about companies and business models, creating report templates, and training the model to solve complex questions. To automate data analysis, the method of using LangChain agents was proposed. A software implementation was developed, and data analysis indicators were studied using SaaS sales data as a case study. The study demonstrated LangChain agents' capability to automate data analysis processes in the SaaS industry. Future research will aim to expand its application across more complex data, larger number of data questions, and pre-trained LLMs.

Keywords: LangChain, LangChain agent, AI, LLM, artificial intelligence, data analysis, data analysis automation.

Abbreviations

Agent is a LangChain agent;
AI is an Artificial Intelligence;
ChatGPT is an instance of the GPT model by OpenAI;
Dataframe is a data structure in Python;
EDA is an Exploratory Data Analysis;
GPT is a Generative Pre-trained Transformer;
gpt-3.5-turbo-0125 is a version of GPT-3.5 model
LangChain is a framework for LLMs;
LLM is a Large Language Model;
OpenAI is an AI research and deployment company;
Pandas is a library in Python
RAG is a Retrieval-Augmented Generation;
SaaS is a Software as a Service;
Temperature is a hyperparameter for LLM.

Nomenclature

A_i is a vector created by embedding mechanism;
B_i is a vector created by embedding mechanism;
Θ_i is an angle between vectors A_i and B_i.

Introduction

In the rapidly changing Software as a Service (SaaS) industry, effectively using advanced data analytics is crucial for guiding business strategies and maintaining competitiveness [1]. Traditional data processing in this sector depends heavily on the expertise of data analysts. This reliance can strain budgets and lead to operational slowdowns and scaling difficulties, issues that are intensified by the rapid increase in data volume and complexity. The introduction of new technologies like LangChain and Retrieval-Augmented Generation (RAG) heralds a significant transition towards more sustainable and streamlined approaches to data management using LLMs.

Object of Study: The object of this study is the automation of data analysis processes within the SaaS sector, aimed at automating traditional, manual methods of data examination and interpretation by data analysis with AI chatbots using LangChain agents.

Subject of Study: The subjects of this study include AI-driven technologies—specifically LangChain and RAG—that facilitate the automation of data analytics. These technologies enable the execution of complex tasks such as exploratory data analysis, univariate and bivariate analysis, and hypothesis testing through AI chatbots, reducing the need for extensive human intervention.

Purpose of the Work

The primary aim of this paper is to explore the potential of LangChain agents and other related AI technologies to automate data analysis in the SaaS industry.

Problem statement

The conventional approach to data analysis in SaaS sector is deeply dependent on the expertise of data analysts.

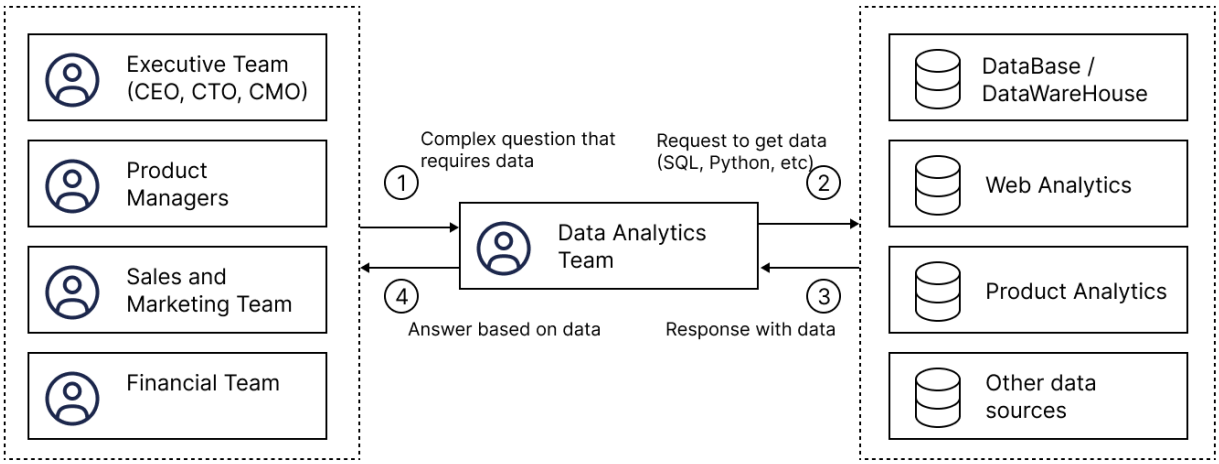


Fig. 1. The common process of data analysis interactions with humans and data

This dependence introduces substantial obstacles: it leads to bottlenecks in the operational workflow, increases operational expenditures, and constrains SaaS company ability to scale their data analytics functions effectively. Such challenges are intensifying as the volume and complexity of data expand exponentially.

The advent of AI and associated technologies like Retrieval-Augmented Generation (RAG) and LangChain offers a transformative solution to these bottlenecks. By integrating these technologies, businesses can leverage AI-driven chatbots to undertake a broad spectrum of data analyses, ranging from basic exploratory data analysis (EDA) to more sophisticated univariate and bivariate analyses, as well as hypothesis testing. The implementation of these tools challenges traditional methodologies by enabling automated, efficient data processing that does not compromise on depth or accuracy.

Review of the literature

LangChain is a versatile and powerful framework that facilitates the development of applications powered by large language models (LLMs), such as ChatGPT, especially within the context of automating data analysis for SaaS or other sectors [9,10]. Various resources highlight how LangChain can significantly enhance data analysis processes through its ability to integrate LLMs for natural language understanding and generation, offering a new level of interaction and automation in data-related tasks.

LLMs can effectively perform market sentiment analysis on Reddit posts, with potential for competitive performance against existing supervised models [5].

The new method using large language models (LLMs) improves qualitative data analysis by automating keypoints extraction and relevance evaluation, achieving higher accuracy and reducing time and effort in various application settings [6].

Mergen, an R package, leverages Large Language Models (LLMs) for data analysis code generation and execution, enabling humans to conduct data analysis by describing objectives and desired analyses through clear text [7].

For instance, LangChain simplifies the creation of applications that leverage LLMs for tasks like text summarization, question answering, and chatbot development. It allows these applications to interact with various data sources, enabling more context-

aware, responsive, and intelligent systems. Developers can define use cases, build logic with flexible prompts and chains, and set context to guide application behavior, which enhances performance and user experience [3].

LangChain can be used to analyze various data input sources such as PDF documents including text, images, tables, and other embedded elements [2] as well as summarize documents and provide answers based on the document content [8].

In terms of automation, LangChain's applicability extends to specific data analysis tasks such as Exploratory Data Analysis (EDA), hypothesis testing, and bivariate analysis using agents created to interface with data frames directly, simplifying the workflow and improving the efficiency of data-driven decisions [4].

These capabilities demonstrate LangChain's potential to revolutionize data analysis, providing a robust tool that can accommodate a wide array of analytical tasks, fostering innovation, and enabling automated data analysis.

Materials and methods

In the article we suggest to replace data analyst with AI Chatbot based on LangChain agent. The agent receives data-related questions that needs data to be analyzed in order to answer the question.

The suggestion is to use LangChain agent that involves leveraging an LLM in tandem with a series of predefined actions. The agent employs a reasoning engine to select the most appropriate actions to answer the questions. The agent can manage multiple tasks such as receiving data from the database, run such tasks in loops, and use additional context to provide contextually-aware answer.

Firstly, based on the description of tools, the agent decides which tool should be used to get relevant information.

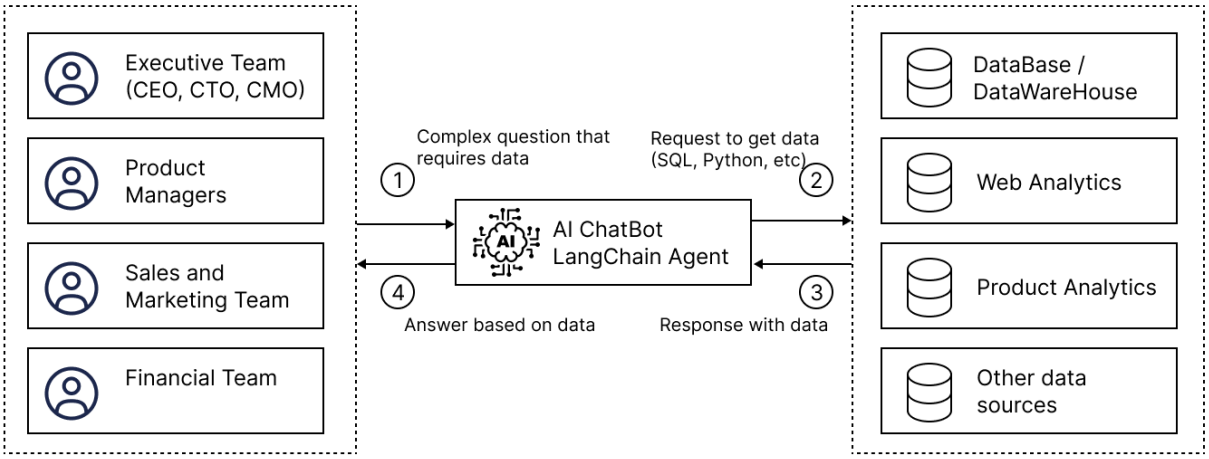


Fig. 2. Replacing the data analysts with AI ChatBot

Secondly, the agent performs actions, such as generating Python code to extract required data from a Pandas dataframe, and considers the context of the obtained results. It also seeks additional information from other sources, like conducting a Google search or referencing descriptions of SaaS domains.

Lastly, the agent checks the results and repeats the process to get the desired information to answer the question in human language to be understanded by the person who asked the question.

Recognizing the likelihood of absence of domain knowledge and company-specific data within LLM, the decision was made to create a second agent enhanced with the incorporation of such information.

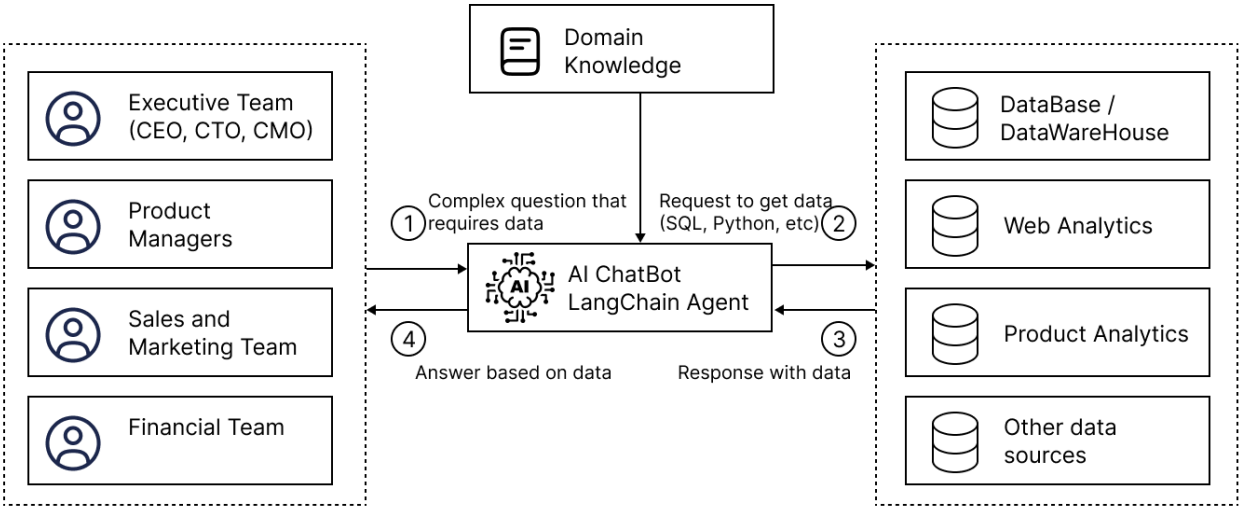


Fig. 3. Providing additional domain knowledge to AI agent

The domain knowledge data is split into the discrete chunks, each representing a distinct unit of information. Subsequently, an embedding algorithm processes these segmented data pieces, generating embeddings that are then inserted into a vector database. Each vector is an n-dimensional space and is represented by numerical values.

User's question is processed by the embeddings mechanism and embedding is computed. Then the cosine similarity between the user's embedding vector and every database vector is calculated using the following formula [4]:

$$\text{cosine similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

A few examples of vector similarity are shown in Figure 4 below.

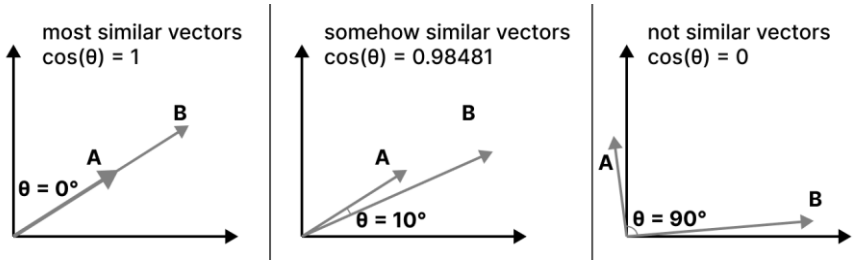


Fig. 4. An illustration of vector similarity for two vectors A and B with an angle θ

The most similar vectors are returned as the ones that contain the most relevant data. These vectors form the context for the subsequent input to the LangChain agent.

LangChain agent takes into account this information when answering the question.

Experiments

Two LangChain agents were developed: the first one had access to data in a Pandas dataframe, while the second one had access not only to the dataframe but also to contextual text describing the dataset, company, and domain knowledge. The contextual information was provided as text input.

The dataset comprised SaaS sales data sourced from AWS, consisting of a single table with 18 columns and 9994 rows. This

data was imported from a CSV file into a Pandas dataframe.

Both LangChain agents were evaluated in addressing various data analysis tasks, including exploratory data analysis (EDA), univariate and bivariate analysis, complex inquiries, hypothesis generation based on the data, and hypothesis testing using the t-test.

Twenty analytical questions covering diverse analysis types were formulated and presented to both LangChain agents in natural language. The responses were recorded, compared against each other, and analyzed.

The OpenAI GPT-3.5-turbo-0125 served as the Language Model (LLM) with specified parameters: temperature=0 and max_tokens=1000.

The langchain_experimental.agents. agent_toolkits.-pandas.base. create_pandas_dataframe_agent API was employed to create the dataframe agent.

To enhance the second LangChain agent with context, hardcoded text input was provided. Subsequently, the 20 questions were posed to each LangChain agent, and the resulting responses were evaluated across various metrics by comparing them to a set of pre-validated answers.

These metrics were utilized for the analysis and evaluation of the LangChain agents' performance:

Correctness – Assess whether the agent delivered correct response or solution.

Comprehensiveness – Evaluate whether the agent's responses covers all the details of the question.

Clarity – Assess whether the response is clear. Responses that are overly verbose or unclear may hinder user understanding.

Novelty – Measure the degree of novelty in each response compared to existing information. This metric is particularly relevant for hypothesis generation and explanations.

Relevance – Rate the relevance of the answer to the dataset and inquiries.

For each question, results are categorized as follows: 2 – fully corresponds, 1 – partially corresponds, 0 – does not correspond, N/A – result not provided due to LangChain agent issue.

Results

The results of conducted experiments is presented in the Table 1. Here we use the following notation: Agent 1 is a LangChain agent without additional context, Agent 2 is a LangChain agent with additional context. The metrics (Correctness, Relevance, Comprehensiveness, Clarity, Novelty) were calculated as average values.

The table 1 shows that for several metrics (Correctness and Relevance) agent 2 showed higher results. While for the other metrics the results are similar. It’s also seen that the use of the proposed method of instance significance determining allows in practice to select a subsample of smaller volume from of the original sample, enough to construct neural network models with the required accuracy, reducing the time to build models.

Table 1. The experiment results on agent performance by analysis types

Metric	Correctness, avg		Relevance, avg		Comprehensiveness, avg		Clarity, avg		Novelty, avg	
Analysis type	Agent 1	Agent 2	Agent 1	Agent 2	Agent 1	Agent 2	Agent 1	Agent 2	Agent 1	Agent 2
Bivariate analysis	2.00	2.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Complex analysis	0.33	0.33	0.67	0.67	0.00	0.00	0.33	0.33	0.00	0.00
Exploratory Data Analysis	1.60	1.80	1.20	1.60	1.00	1.00	1.60	1.60	0.80	0.80
Hypothesis testing	2.00	2.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Univariate analysis	2.00	2.00	1.00	1.25	1.00	1.00	1.00	1.00	1.00	1.00
Average	1.59	1.63	0.97	1.10	0.80	0.80	0.99	0.99	0.76	0.76

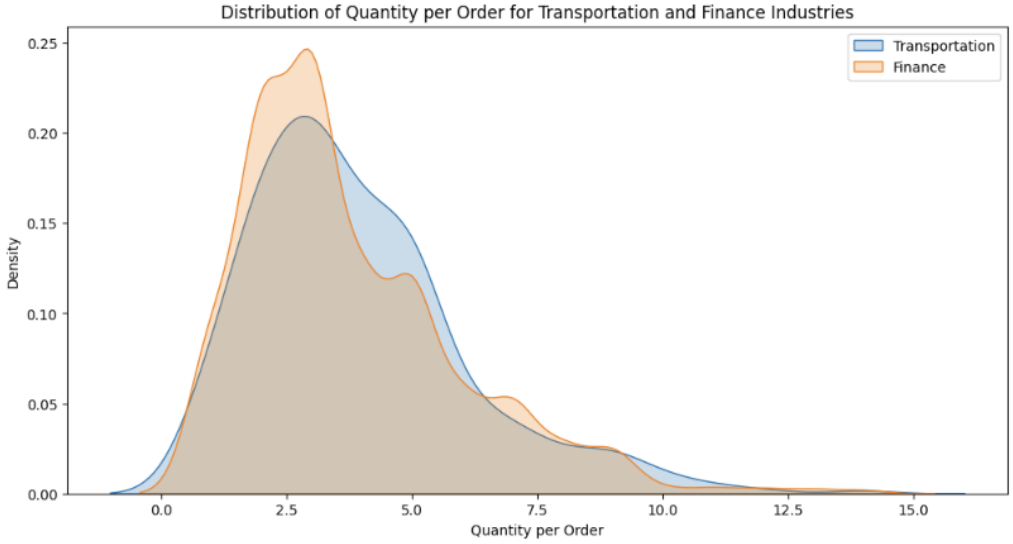
```
# Question 14 – Hypotheses.  How is the distribution of Quantity per Order across various industries
agent.invoke("""As a helpful data analyst, Plot the distribution of Quantity per Order for the Transportation
industries using kde plot. Add the distribution of Quantity per Order for the Finance industry on the same plot.
Also provide a legend and label the x and y axises. """)
```

```
<string>:6: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

<string>:7: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.
```




```
Here is the KDE plot showing the distribution of Quantity per Order for the Transportation and Finance industries. The plot displays the d

- The blue curve represents the Transportation industry.
- The orange curve represents the Finance industry.

The x-axis is labeled as 'Quantity per Order' and the y-axis is labeled as 'Density'. The legend indicates which curve corresponds to each

> Finished chain.
{'input': 'As a helpful data analyst, Plot the distribution of Quantity per Order for the Transportation industries using kde plot. Add
the distribution of Quantity per Order for the Finance industry on the same plot. Also provide a legend and\nlabel the x and y axises. ',
 'output': "Here is the KDE plot showing the distribution of Quantity per Order for the Transportation and Finance industries. The plot
displays the density of quantity values per order for each industry.\n\n- The blue curve represents the Transportation industry.\n- The
orange curve represents the Finance industry.\n\nThe x-axis is labeled as 'Quantity per Order' and the y-axis is labeled as 'Density'.
The legend indicates which curve corresponds to each industry."}
```

Fig. 5. The example of interactions with LangChain agent that received a human-text input and produces plain-text output as well visual plot to support the answer

Figure 1 graphically illustrates an example of the output generated by LangChain agents instance. It shows that LangChain agents are capable to receive user question, analyze data, provide text output as well as visual charts using Python packages such as Seaborn.

Discussion

As it is evident from the Figure 1, in general, the LangChain agents are capable to receive data-related answers, access the data to make calculations and return plain-text answers based on data as well as provide supportive visual charts.

At the same time we can see from the table 1 that not all answer were correct, clear, comprehensive, relevant, and containing novelty.

For example, the performance of LangChain agents on the complex analysis questions is low for all metrics. The hypothesis is that LLM is not trained to answer complex data analysis questions and the provided query was not detailed enough. We probably should provide a more complex prompt covering all details as well as examples of desired output.

However, for more simpler questions such as EDA and univariate analysis the results are matching the expected quality.

As of the difference between agents 1 and 2, we see a small improvement in the correctness and relevance. The result is not statistically significant, but it can be treated as

a signal that providing additional context can help LangChain agents to produce more correct and more relevant results. Based on the small number of questions and data in the experiment, we do not calculate possible uplift, but rather treat this as signal that there is likely an improvement.

We noticed that providing additional context haven't made any impact on comprehensiveness, clarity, and novelty. Possible reasons for not having the impact on these metrics likely lies in the small amount of information passed in the additional context, prompts and the pandas specific LangChain agent that was used.

Conclusions

The problem of data analysis automation using LangChain agents was examined in this article. The author created and experiments with 2 LangChain agents to analyze typical data analysis questions in SaaS.

The scientific novelty lies in the development of the method for automating data analysis for SaaS using LangChain agents.

The practical significance of obtained results is that the proposed method can be used for automating simple data analysis questions such as EDA.

Prospects for further research are to study the advanced usage of LangChain agents on a broader datasets, larger number of questions, and pre-trained LLMs.

References

1. A Madhuri, S. Phani Praveen, D Lokesh Sai Kumar, S Sindhura, Sai Srinivas Vellela. (2021). Challenges and Issues of Data Analytics in Emerging Scenarios for Big Data, Cloud and Image Mining. Annals of the Romanian Society for Cell Biology, 412–423. Retrieved from http://annalsofrscb.ro/index.php/journal/article/view/128

2. Holkar A, Bhosale S, Harpale A, Pachangane VH. Unlocking the depth analysis of PDF using artificial intelligence, large language model, LangChain. Third Year, Information Technology, Jaywantrao Sawant Polytechnic, Pune, Maharashtra, India. *International Research Journal of Modernization in Engineering Technology and Science*. 2024;06(02):682. DOI: 10.56726/IRJMETS49113

3. Bayer, S., Gimpel, H., & Markgraf, M. (2022). The role of domain expertise in trusting and following explainable AI decision support systems. Journal of Decision Systems, 32(1), 110–138. DOI: 10.1080/12460125.2021.1958505

4. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

5. Deng, X., Bashlovkina, V., Han, F., Baumgartner, S., & Bendersky, M. (2022). What do LLMs Know about Financial Markets? A Case Study on Reddit Market Sentiment Analysis. Companion Proceedings of the ACM Web Conference 2023. DOI: 10.1145/3543873.3587324.

6. Zhao, F., Yu, F., Trull, T., & Shang, Y. (2023). A New Method Using LLMs for Keypoints Generation in Qualitative Data Analysis. 2023 IEEE Conference on Artificial Intelligence (CAI), 333-334. DOI: 10.1109/CAI54212.2023.00147.

7. Jansen, J., Manukyan, A., Khoury, N., & Akalin, A. (2023). Leveraging large language models for data analysis automation. DOI: 10.1101/2023.12.11.571140.

8. Pokhrel, Sangita, Ganesan, Swathi, Akther, Tasnim, & Karunarathne, Lakmali. (2024). Building Customized Chatbots for Document Summarization and Question Answering using Large Language Models using a Framework with OpenAI, LangChain, and Streamlit. Journal of Information Technology and Digital World, 6(1), 70-86. DOI: 10.36548/jitdw.2024.1.006

9. Z. Cui, X. Jing, P. Zhao, W. Zhang and J. Chen, "A New Subspace Clustering Strategy for AI-Based Data Analysis in IoT System," in IEEE Internet of Things Journal, vol. 8, no. 16, pp. 12540-12549, 15 Aug.15, 2021, DOI: 10.1109 / JIOT.- 2021.3056578

10. R. Asyrofi, M. R. Dewi, M. I. Lutfhi and P. Wibowo, "Systematic Literature Review Langchain Proposed," 2023 International Electronics Symposium (IES), Denpasar, Indonesia, 2023, pp. 533-537, DOI: 10.1109/IES59143.2023.10242497.

The article has been sent to the editors 28.05.24.
After processing 15.06.24.
Submitted for printing 28.06.24.