International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Privacy-Centric Offline Chatbot Using Large Language Models

Dr.T.Prabakaran[1], Anjali K[2], Vipunsai K[3], Ruchitha K[4], Bhavani M[5]
[1]Professor, Computer Science & Engineering, Joginpally B.R.Engineering College
[2]Computer Science & Engineering, Joginpally B.R.Engineering College [3]Computer Science & Engineering, Joginpally B.R.Engineering College [4]Computer Science & Engineering, Joginpally B.R.Engineering College [5]Computer Science & Engineering, Joginpally B.R.Engineering College

Abstract

From ELIZA in the 1960s mimicking a psychotherapist to ChatGPT helping us to write code, chatbots have come a long way and are dominating the digital era. Chatbots, which come under Conversational AI make it possible for humans and machines to converse. They are developed significantly through the latest advances in AI, becoming more complex and intelligent but some downfalls exist. Many current solutions depend on cloud-based models which need internet connectivity. They also might collect and store the data leading to privacy breaches. This research paper focuses on these problems. This proposed methodology equips locally managed Large Language Models (LLMs) using tools like Ollama and bge-m3 model for embedding. Operating offline ensures that no data is collected and shared and it can also work in remote places with no internet connectivity. This integrates Retrieval- Augmented Generation (RAG) to generate context- aware, accurate answers and it also employs vector indexing search for visually related images and files to extract information.

Keywords*:* Artificial Intelligence (AI), Large Language Models (LLMs), Local Models, Offline Chatbot, Retrieval-Augmented Generation (RAG).

1. Introduction

Deep learning models such as Large Language Models (LLMs) are trained on a huge amount of data. They can be used for text generation, image classification, knowledge answering, dialog generation and translation. Chatbots built with the help of these LLMs can hold conversations and answer accordingly. Ollama models which Meta releases are powerful open- source language models and are efficient for data handling and privacy [1]. Chatbots can be domain- specific or general. Most of the methods include retrieval and identifying documents which are stored in vector databases that contain information for satisfying the user's query. Then they are provided to LLM to generate responses based on the retrieved information [2]. Vector embeddings are an integral part of LLMs which can provide accurate results when combined with Retrieval-Augmented Generation (RAG) [3]. Traditional methods like Optical Character Recognition (OCR) combined with RAG can be used for the extraction of multi- modal documents like images and text [4]. Users can upload any type of document ranging from .png to .ppt and OCR helps in retrieving the text from those images. In this digital era, effective summarization is in demand for almost every sector from businesses to education and from startups to multinational companies to provide instant support, 24/7 availability and personalized answers according to the specific user [5]. Although more research is needed in this area Llama models provide the advantage of running offline and implementing on low-cost devices like Raspberry Pi [6]. LLMs are also used in the analysis of source code where evaluations of errors, improvements in

performance, and specifications for library functions [7]. Integrating a Knowledge Base with Retrieval-Augmented Generation (RAG) will help in managing and reusing existing knowledge and help in finding accurate results satisfying the needs and demands of the user [8].

Despite their widespread usage, chatbots raise a few issues including privacy concerns due to potential data sharing and their reliance on internet connectivity. Uploading documents to chatbots for tasks such as analysis, coding, or modifications poses a risk of data collection as chatbots often collect and store this information potentially infringing on privacy and their dependence on internet access remains a significant constraint, particularly in offline or low- bandwidth environments as most of the chatbots require internet to process the queries and use cloud services [9]. In the text classification process evaluating semantic similarities among sentence pairs is very essential [10]. The area of Natural Language Processing (NLP) has undergone colossal development leaps that new technologies have influenced in the last few years [11]. It represents a high-level architecture for designing an AI- powered document-based query system.

This project mainly focuses on resolving these two challenges by focusing on local models. This research incorporates Llama.cpp which compiles models into single, generalizable CUDA “backend” that can run on a wide range of NVIDIA GPUs. Llama.cpp when evaluated against TensorRT provides better accessibility and cross- platform operability. It also supports 1.5 bit-8-bit quantization.

We employed the Unstructured.io library, which partitions documents into semantic units using document formats instead of relying solely on text-based features and helps in model fine- tuning.

It also supports 27 file extensions like .bmp, .csv, .doc, .heic, .html, .xlsx, .jpeg, .pptx and many more. Additionally, we employed the BGE M3 embedding model, which supports over 100 languages and is designed for dense, sparse, and multi-vector retrieval to improve efficiency and tailor better responses and speed file processing. NLP comes under the domain of artificial intelligence (AI) it mainly focuses on interaction between the user and the system [12]. The remaining article is arranged as follows. Section 2 presents the materials and methods used in this system. Section 3 presents the results obtained. Section 4 presents the conclusions along with future work and finally.

2. Materials and Methods

2.1 Materials

These materials determine the workflow of a system that is designed to process user input, index documents, and generate responses as shown in Fig. 1. Tender documents are key in procurement management and provide very necessary information and guidelines for obtaining goods, services, or works [13]. Here it follows

User Input: The user interacts with the system through the Gradio interface which may be in the form of textbox, file uploads, etc.

File Processing: We use a tool called Unstructured.io to extract the text from unstructured data.

Document Indexing: Vector Store Index is used to create, store, and search document embedding from processed text.

Chat Engine: It integrates user queries and passes them to local models for generating responses.

Local Models: We use local models like Ollama Embed which creates embeddings for user queries and Ollama LLM which generates human-like responses.

Response Generator: It displays the responses through the Gradio Chatbot component.

Software requirements: Python 3.8 or higher, RAM-8 Gb or higher, GPU (recommended), Ollama (for local LLM and embedding models).

Python libraries: Gradio, nltk, llama index, unstructured, typing, logging, numpy.

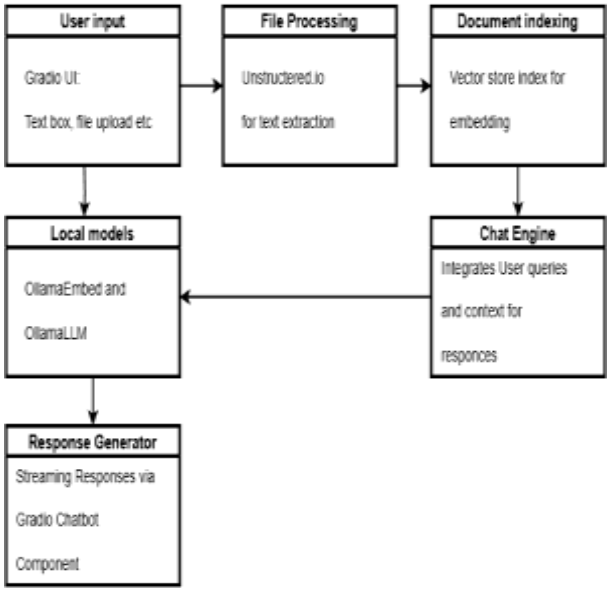

Fig.1: System Architecture

2.2 Methodology

The flow chart of the proposed methodology is shown in Fig.2. The user interacts with the system through the Gradio interface. The user can either select two options: Direct Query Handling and File Processing. The temperature settings are set in a way that the model can answer questions less predictably. If the user selects query handling, then they can directly type the query and the interface sends the query to the Query Processor which can stream the chat. The Query Processor converts it into an index-based query and uses a Large Language Model (LLM) for querying. It aims to construct a specialization base in the control field, probe the use of artificial intelligence LLM technology, and construct large knowledge models and responses with knowledge Q&A and data query functions [14]. The resultant response will be streamed to the user with the help of the Gradio interface. This project is built using LLM and Gradio Framework, initiates to develop an advanced and efficiency chatbot system using local models [15]. If user decides to upload a file (PDFs, images etc.) before asking a query then the file is processed by file processor and then partitioned into smaller chunks or segments for further processing. The partitioned files are processed in parallel to extract relevant information. The extracted text is further analysed by various mechanisms like tokenization, stop word removal etc. The response is generated taking into account the user's context and intent. And the chatbot also adapts to user needs in a quick and smooth manner.

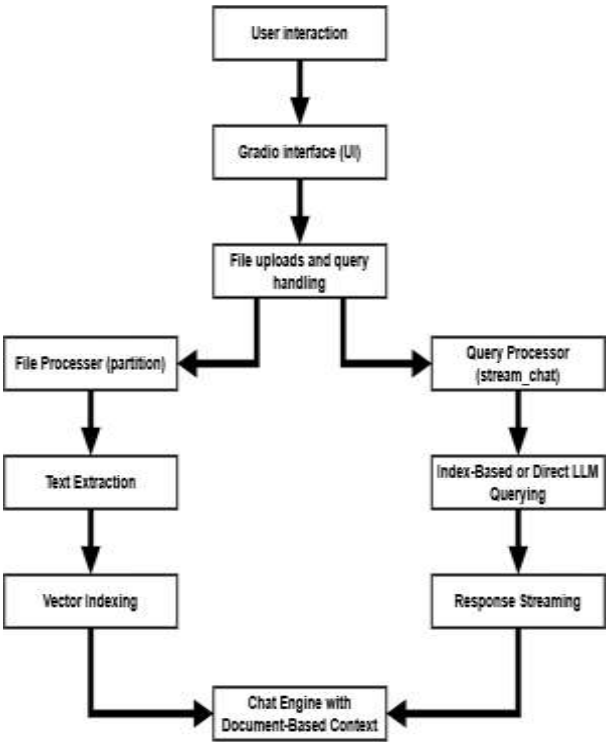

Fig.2: Chat or Document-Based Chatbot Design

The system also has an option of reset which resets the whole chat history and can also process multiple files simultaneously. This enables the system to maintain data privacy by not collecting data for any external sources and works offline with the help of local models. Considering Large Language Models (LLMs), we explain how they could support and helps this task [16]. There is a huge demand to analyse and understand the formation and growth factors influencing solder voids [17].

2.2.3 Procedure:

We examine LLMs' cognitive abilities and interaction pattern in MNP task, by testing various prompts and temperature parameters [18].

The working procedure is explained in the below steps from Fig. 3 to Fig. 6.

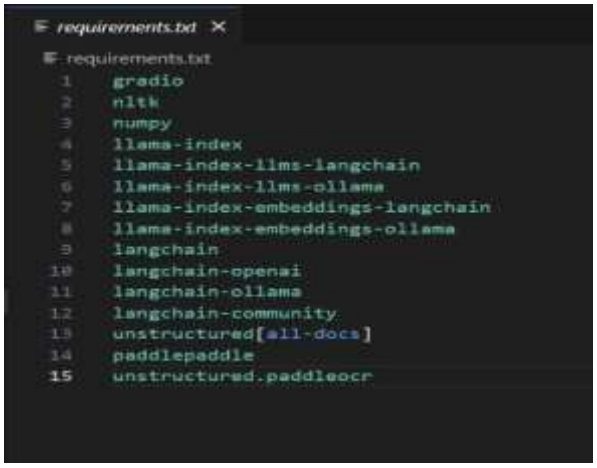

Fig.3 Listing Libraries required in a text file

Fig.3 shows a requirements.txt file, which we usually used in Python projects to list the libraries required for the project. This file defines the packages needed for installation using package manager called pip.

2.2.3.1 Prerequisites

Ensure that you have installed latest version of Python and Ollama.Fig.4 shows a terminal window where user can run a python command in a environment called myenv. To install this requirements.txt file use the command. The requirements.txt file contains list of libraries for python project. The command being executed is:

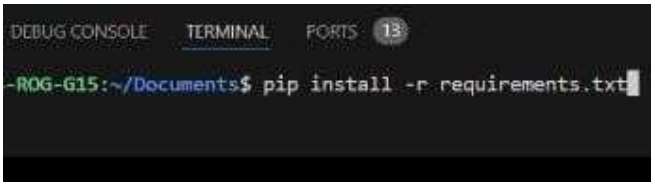

Fig.4 Installing the Libraries

2.2.3.2 Setup and Installation

Fig .5 shows a terminal window in VS code where the user works with Python virtual environment. Here environment preparation is done, and the command follows:

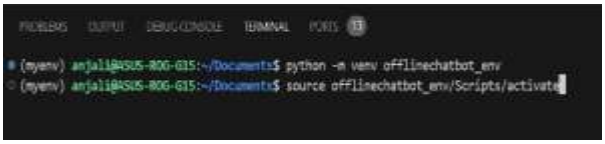

Fig.5 Enabling a virtual environment on windows

Fig.6 shows where user can execute the commands in a Bash shell. It creates a new Python virtual environment. Install Ollama from the official website and pull required models. The command is executed as follows:

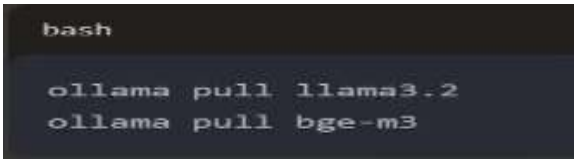

Fig.6 pulling models on Ollama (llama3.2, bge- m3)

2.2.3.3 Configuration

Adjust the Ollama base URL if different from http://127.0.0.1:11434/

3. RESULTS

3.1 Running the Application

Fig.7 shows where the user executes the system of Python script named offlinechatbot.py. That is being executed in a virtual environment (myenv). This output shows that a locally hosted chatbot interface, is built using Gradio, and runs successfully as shows in Fig. 7.

Fig.7 Executing the system

3.2 Outputs

Fig.8 shows a chatbot interface where a query about the differences between syntax
analysis and lexical analysis has been answered. Here the user uploads a file and query the chatbot. The chatbot provides a response to the query of the user. This how here interface works.

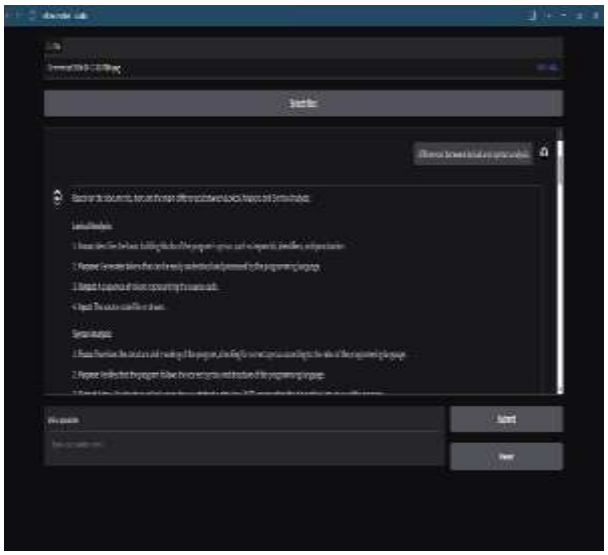

Fig.8: UI of the system

This project uses Llama 3.2 and compares with those of Claude3-Haiku and GPT-4o mini:

Table.1 "Benchmark Performance Comparison of AI models Across Modalities"

Modality	Category Benchmark	Llama 3.2 90B	GPT-4o- mini	Category benchmark	Llama 3.2 90B	GPT-4o-mini
Image	Mathematical reasoning	0.3	9.4	General MLU	6.0	2.0
	MMU (micro avg accuracy)	5.2	2.3	0-shot, CoT)	8.0	0.2
	MMU-Pro, Standard (10 opts,test)	7.3	6.7	ATH 0-shot, CoT)	6.7	0.2
	MathVista (testmini)			reasoning		
	Diagram Understanding ChartQA (test, 0-shot CoT)	5.5	-	PQA o-shot, CoT)	-	

This Table. 1 shows the performance of various large language models (LLMs) using different benchmarks like mathematical reasoning, image-based tasks, and text-based tasks. Models like Llama (3.2 versions with 11B and (90B parameters), GPT-4-mini and Claude3-Haiku are evaluated for solving college-level problems, analysing charts and diagrams and reasoning. This represents models' accuracy and efficiency.

Table.2 Evaluation of Llama Variants on General Knowledge and Reading Tasks

Category	Benchmark	Llama3 8B	Llama2 7B	Llama2 13B	Llama3 70B
General	MMLU (S-Shot)	6.6	5.7	3.8	9.5
	AGIEval English (3-5 shot)	5.9	8.8	8.7	3.0
	CommonSense QA (7-shot)	2.6	7.6	7.6	3.8
	Wingrande (5-shot)	6.1	3.3	5.4	3.1
	BIG-Bench Hard (3-shot,CoT)	1.1	8.1	7.0	1.3
	ARC-Challenge (25-shot)	8.6	3.7	7.6	3.0
Knowledge Reasoning	TriviaQA-Wiki (5-shot)	8.5	2.1	9.6	9.7
Reading comprehension	SQuAD (1-shot)	6.4	2.1	2.1	5.6
	QuAC (1-shot,F1)	4.4	9.6	4.9	1.1
	BoolQ (0-shot)	5.7	5.5	6.9	9.0
	DROP (3-shot,F1)	8.4	7.9	9.8	9.7

This Table.2 describes the performance of different Llama versions, including Llama 3 8B, Llama 2(7B, 13B and 70B), and Llama 3 70B, across various benchmarks that are general tasks, knowledge reasoning, and reading comprehension. This Table.2 highlights the progression in performance of the model with newer model versions and increasing parameter sizes.

4. Conclusion

This project provides an offline AI chatbot that handles uploaded documents, indexes their content, and engages in interactive, context-aware responses through a user-friendly interface. To specify the issues that arise when handling vector embeddings in applications, vector databases were specifically created. It is very optimal for document-based query handling, utilizing advanced NLP and text extraction technologies for efficient use offline. This offline AI chatbot is designed for queries-based documents and enables real-time, context-aware interactions. By evaluating Large Language Models, we can accurately evaluate its capabilities and the Multi-choice Questions (MCQ) benchmark is widely adopted for its efficient answers. This is achieved by integrating advanced tools like LlamaIndex for query handling and indexing documents, Ollama models for quality language generation, and Gradio for creating a user- friendly interface.

REFERENCES

[1] K. Ko, T. Y. Nyein, K. K. Oo, T. Z. Oo and T. T. Zin, "Retrieval Augmented Generation for Document Query Automation using Open source LLMs," 2024 5th *International Conference on Advanced Information Technologies (ICAIT)*, Yangon, Myanmar, 2024, pp. 1-6, doi: 10.1109/ICAIT65209.2024.10754919. (Conference)

[2] M. Maryamah, M. M. Irfani, E. B. Tri Raharjo, N. A. Rahmi, M. Ghani and I. K. Raharjana, "Chatbots in Academia: A Retrieval-Augmented Generation Approach for Improved Efficient Information Access," 2024 16th International Conference on *Knowledge and Smart Technology (KST)*, Krabi, Thailand, 2024, pp. 259-264, doi: 10.1109/KST61284.2024.10499652. (Conference)

[3] S. Kukreja, T. Kumar, V. Bharate, A. Purohit, A. Dasgupta and D. Guha "Performance Evaluation of Vector Embeddings with Retrieval-Augmented Generation," 2024 9th *International Conference on Computer and Communication Systems (ICCCS)*, Xi'an, China, 2024, pp. 333-340, doi: 10.1109/ICCCS61882.2024.10603291. (Conference)

[4] S. Bag, A. Gupta, R. Kaushik and C. Jain, "RAG Beyond Text: Enhancing Image Retrieval in RAG Systems," *2024 International Conference on Electrical, Computer and Energy Technologies (ICECET*, Sydney, Australia, 2024, pp. 1-6, doi: 10.1109/ICECET61485.2024.10698598. (Conference)

[5] A. Ramprasad and P. Sivakumar, "Context-Aware Summarization for PDF Documents using Large Language Models", 2024 *International Conference on Expert Clouds and Applications (ICOECA)*, Bengaluru, India, 2024, pp. 186-191, doi: 10.1109/ICOECA62351.2024.00044. (Conference)

[6] K.Sakai, Y.Uehara and S.Kashihara, *"Implementation and Evaluation of LLM-Based Conversational Systems on a Low-Cost Device,"* 2024 IEEE *Global Humanitarian Technology Conference (GHTC)*, Radnor, PA, USA, 2024, pp. 392-399,doi:10.1109/GHTC62424.2024.10771565. (Conference)

[7] V. N. Ignatyev, N. V. Shimchik, D. D. Panov and A. A. Mitrofanov, "Large language models in source code static analysis," 2024 *Ivannikov Memorial Workshop (IVMEM)*, Velikiy Novgorod, Russian Federation, 2024, pp. 28-35, doi: 10.1109/IVMEM63006.2024.10659715. (Conference)

[8] S. Knollmeyer, M. U. Akmal, L. Koval, S. Asif, S. G. Mathias and D. Groβmann, "Document Knowledge Graph to Enhance Question Answering with Retrieval Augmented Generation," 2024 IEEE *29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, Padova, Italy, 2024, pp. 1-4, doi: 10.1109/ETFA61755.2024.10711054. (Conference)

[9] L. Ruhländer, E. Popp, M. Stylidou, S. Khan and D. Svetinovic, "On the Security and Privacy Implications of Large Language Models: In-Depth Threat Analysis," 2024 IEEE *International Conferences on Internet of Things (iThings)* and IEEE *Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom)* and IEEE *Smart Data (SmartData)* and IEEE Congress on Cybermatics,

Copenhagen, Denmark, 2024, pp. 543-550, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics62450.2024.00102. (Conference)

[10] V. V. Mayil and T. R. Jeyalakshmi, "Pretrained Sentence Embedding and Semantic Sentence Similarity Language Model for Text Classification in NLP," 2023 3rd *International conference on Artificial Intelligence and Signal Processing (AISP)*, VIJAYAWADA, India, 2023,pp.1-5,doi: 10.1109/AISP57993.2023.10134937. (Conference)

[11] K. Kalaiselvi, R. Shanmugam, S. Tamilselvan, S. P. Manikandan and P. Rajasekar, "Innovations in Natural Language Processing through Enhanced Linguistic Model Accuracy and Efficiency Using Advanced Reinforcement Learning Techniques," 2024 *Second International Conference on Advances in Information Technology (ICAIT)*, Chikkamagaluru, Karnataka, India, 2024, pp. 1-5, doi:10.1109/ICAIT61638.2024.10690717. (Conference)

[12] P. Omrani, A. Hosseini, K. Hooshanfar, Z. Ebrahimian, R. Toosi and M. Ali Akhaee, "Hybrid Retrieval-Augmented Generation Approach for LLMs Query Response Enhancement," 2024 10th *International Conference on Web Research (ICWR)*, Tehran, Iran, Islamic Republic of, 2024, pp.22-26,doi: 10.1109/ICWR61162.2024.10533345. (Conference)

[13] Y. Zhao and D. Li, "Semi-Structured Tender Document Retrieval-Augmented Generation: A Framework Based on Large Language Model," 2024 IEEE 2nd *International Conference on Sensors, Electronics and Computer Engineering (ICSECE)*,Jinzhou,China,2024,pp.177-182,doi: 10.1109/ICSECE61636.2024.10729522. (Conference)

[14] M. Ni, J. Zhang, C. Fu, J. Wang, X. Ning and S. Li, "ChatGrid: Intelligent Knowledge Q&A for Power Dispatching Control Based on Large Language Models and Retrieval-augmented Generation," 2024 IEEE 7th *Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chongqing, China, 2024, pp. 921-925, doi: 10.1109/ITNEC60942.2024.10733337. (Conference)

[15] A. Sawant, S. Phadol, S. Mehere, P. Divekar, S. Khedekar and R. Dound, "ChatWhiz: Chatbot Built Using Transformers and Gradio Framework," 2024 5th *International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, Tirunelveli, India, 2024, pp. 454-461, doi: 10.1109/ICICV62344.2024.00077. (Conference)

[16] S. Arulmohan, M. -J. Meurs and S. Mosser, "Extracting Domain Models from Textual Requirements in the Era of Large Language Models," 2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS- C), Västerås, Sweden, 2023, pp. 580-587, doi: 10.1109/MODELS-C59198.2023.00096. (Conference)

[17] Z. Zhou and H. Fan, "Investigation of Power Device Solder Void Formation by Two-Phase Flow Simulation," 2024 25th *International Conference on Electronic Packaging Technology (ICEPT)*, Tianjin, China, 2024, pp. 1-6, doi: 10.1109/ICEPT63120.2024.10667663. (Conference)

[18] H. Qian, T. Xu, Z. Ding, W. Liu and S. Zhu, "Exploring Large Language Models for Method Name Prediction," 2024 IEEE 24th International Conference on Software *Quality, Reliability and Security (QRS)*, Cambridge, United Kingdom, 2024, pp. 192-203, doi: 10.1109/QRS62785.2024.00028. (Conference)