Proceedings of the International Conference on Sustainable Expert Systems (ICSES-2024)
IEEE Xplore Part Number: CFP24VS6-ART; ISBN: 979-8-3315-4036-4

# Enhancing Communication and Data Transmission Security in RAG using Large Language Models

Venkata Gummadi
Software Engineer
Expedia Group
Austin, United States
gummadic@gmail.com

Pamula Udayaraju
Department of CSE
School of Engineering and Sciences
SRM University-AP, India
udayaraju.p@srmap.edu.in

Venkata Rahul Sarabu
Senior Software Engineer
Wallmart Global Tech
Texas, USA.
rahulvenkata.sarabu@gmail.com

Chaitanya Ravulu
Senior Data Engineer
Rocket Companies
Michigan, United States
chaitanya.karunya@gmail.com

Dhanunjay Reddy Seelam
Software Quality Engineer
Wallmart Global Tech
Bentonville, Arkansas
dhanunjayseelam@gmail.com

Dr. S. Venkataramana
Professor, Department of IT
SRKR Engineering College
Bhimavaram
svramana@srkrec.ac.in

2024 4th International Conference on Sustainable Expert Systems (ICSES) | 979-8-3315-4036-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICSES63445.2024.10763024

*Abstract*— **Retrieval-augmented generation (RAG) enhances large language models (LLMs) by integrating external knowledge sources, enabling more useful information and generating accurate responses. This paper explores RAG's architecture and applications, combining generator and retriever models to access and utilize vast external data repositories. While RAG holds significant promise for various Natural Language Processing (NLP) processes like dialogue generation, summarization, and question answering, it also presents unique security challenges that must be addressed to ensure system integrity and reliability. RAG systems face several security threats, including data poisoning, model manipulation, privacy leakage, biased information retrieval, and harmful outputs generation. Generally, in the traditional RAG application, security threat is one of the major concerns. To tighten the security system and enhance the efficiency of the model on processing more complex data this paper outlines key strategies for securing RAG-based applications to mitigate these risks paper outlines key strategies for securing RAG-based applications to mitigate these risks. Ensuring data security through filtering, sanitization, and provenance tracking can prevent data poisoning and enhance the quality of external knowledge sources. Strengthening model security via adversarial training, input validation, and anomaly detection improves resilience against manipulative attacks. Implementing output monitoring and filtering techniques, such as factual verification, language moderation, and bias detection, ensures the accuracy and safety of generated responses. Additionally, robust infrastructure and access control measures, including secure data storage, secure APIs, and regulated model access, protect against unauthorized access and manipulation. Moreover, this study analyzes various use cases for LLMs enhanced by RAG, including personalized recommendations, customer support automation, content creation, and advanced search functionalities. The role of vector databases in optimizing RAG-driven generative AI is also discussed, highlighting their ability to efficiently manage and retrieve large-scale data for improved response generation. By adhering to these security measures and leveraging best practices from leading industry sources such as Databricks, AWS, and Milvus, developers can ensure the robustness and trustworthiness of RAG-based systems across diverse applications.**

***Keywords— RAG, LLM, Query Analysis, Security Enhancement, Data Privacy.***

## I. Introduction

Retrieval-augmented generation (RAG) is an advanced technique for developing language and logic models (LLMs) that effectively utilize extensive knowledge sources to produce comprehensive responses. RAG finds numerous applications in NLP, including dialogue generation, question-answering, and summarization. However, as RAG technology gains traction and becomes more prevalent, it also encounters growing security challenges and risks. This blog post will explore the security surrounding RAG-based applications and offer key strategies and best practices for safeguarding RAG systems. To produce actual output for tasks such as translating languages, answering queries, and completing incomplete sentences, In this paper, a Large Language Model (LLM) is used to train the model. It uses billions of parameters to produce various outputs like completing sentences, question-answering, and translating language. LLM with RAG enhances the model's performance and increases the model's capacity on particular domains or internal sources/ information of the association without retaining the model. Compared to traditional approaches, the LLM is more cost-effective and improves the performance and output of the RAG model to be more reliable, useful, and accurate.

Artificial intelligence (AI) utilizes powerful, intelligent NLP and chatbot-based applications of the LLM is one of the key models. The main goal is to develop a chatbot that can answer the user's questions in every aspect by referring to previous existing data sources. However, the model has some challenges in generating the output. Challenges in LLMs: It provides false information when it doesn’t know the correct answer. When the user expects a particular current response, if it doesn’t know, it gives the past data or information. It also produces accurate source information for the users. Technology confusion creates inaccurate results, whereas other training software uses the same technologies to produce results.

### A. *Uses of LLM in RAG*

Based on the existing known data, the LLM model creates input for the user and generates a response. Meanwhile, instead of analyzing previous data from the new user input data source, the RAG application fetches all the information to create a response. Then, the retrieved information is transferred to the LLM model. Now, the LLM model compares the obtained and existing data and produces accurate results. To create a knowledge library from the external data, the LMM model trains and stores various types of data such as database, document, or API repositories gathered from outside sources. To improve the model's performance, all the input data are converted into uniform format because the format of each data is varied, such as file, database, numeric, etc. So, an AI-based technique embedding language model is applied to converter the external data in LLM into a uniform format and store it in a vector database. This will make it easier for the model to understand the data in the library. The next step in LMM is the relevancy search, which checks the current and previous data in the vector database to generate the response. For example, let's take an intelligent chatbot that can answer all the human doubts that AI knows. The accurate result was calculated based on the previous calculation and information. Next, prompt evaluation is performed by the RAG model by adding the relevant data into the context. In this phase, a prompt engineering technique is applied to create communication with LLM. This technique allowed the LMM model to produce more accurate results for the user questions or queries. The next phase is updating external data. This step is mainly performed to update external data and retrieve useful information. It is achieved through periodic batching or automated real-time processing techniques.

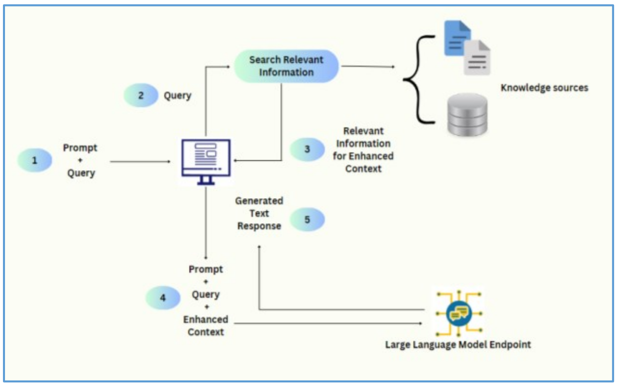

Search Relevant Information
2 Query
Knowledge sources
1 Prompt + Query
3 Relevant Information for Enhanced Context
Generated Text Response 5
4 Prompt + Query + Enhanced Context
Large Language Model Endpoint

FIGURE I. LLM-Based Data Retrieval

### B. *Benefits of RAG with LLM*

The RAG-based application provides various benefits to AI-based organizations or applications. The function of the chatbot is performed based on the foundation model (FM). It is an API of LLM trained using unlabelled or generalized data. However, the computational cost of FM in a particular domain or organization is high. Implementing RAG with LLM is more cost-effective and makes the AI technique easier to access. Even if the original data suits the user's queries, maintaining and checking relevancy is challenging. It is achieved by implementing RAG with LLM. Combining RAG with LLM produces more accurate results and essential user information. This can gain more trust and hope for the AI results. With this deep research, LLM can provide users with the latest and most relatable information. Due to these benefits, in this paper, the LLM model is proposed and experimented with to tighten the security system and protect the input data of the RAG application. The efficiency of the model is verified using various simulation results. The following section discusses earlier research work, the workflow of the proposed model, the model result, and the conclusion.

## II. Literature Survey

In this section, various earlier research works have been proposed to tighten the security system of the RAG application in detail. For example, Mackay, A. (2024, June) has displayed LLM and RAG techniques to experiment with the set augmentation and also applied the power of GPT-4, a customized RAG framework to display the efficiency of automating test case generation to improve the strong verification in safety-related software applications. This paper used two main open-source projects, PX4 Autopilot, and Apollo Auto, which realized substantial benefits in test coverage, defect detection, and compliance, aligned with industry benchmarks like DO-178C and ISO 26262. This proposed method enhances the high potential of tests compared to human-authored test suites, and they are reviewing a wide array of test scenarios with high-severity defects. These two methods can potentially improve software reliability and safety with automated testing. Hu et al. (2024) have introduced the RAG-based LLMs and are finding a small prefix insertion, which mainly improves output accuracy. It can implement the effective technique known as gradient-guided prompt Perturbation (GGPP), and it can provide a more accurate rate in optimizing outputs of RAG-based LLMs to focus on incorrect answers, the prompts filtering out irrelevant details. The main difference in LLMs neuron activation patterns between with prompts and without GGPP perturbations is to develop a highly effective detection method. This method will enhance the strength of RAG-based LLMs with trained neuron activation triggered by prompts generated via Guided Gradient Perturbations.

Du et al. (2024) have discussed the LLM-based vulnerability detection technique Vul-RAG, with a retrieval-augmented generation (RAG) framework to perform a three-stage security to detect code vulnerabilities. The first stage of Vul-RAG is to build vulnerability knowledge by dividing the multi-dimensional knowledge through LLMs from previous CVE cases. Second, the Vul-RAG retrieves relevant vulnerability data on task semantics from the knowledge base for a specific code snippet. The last one is Vul-RAG-based LLMs, which estimate the vulnerability of a snippet code by logically analyzing vulnerability triggers and fixing strategies. The overall result of Vul-RAG shows its effectiveness by outperforming all baselines with an improvement of 12.96% and 110% in accuracy and pairwise accuracy on the PairVul Standard, respectively, which also increases the accuracy of manual detection from 0.60 to 0.77. Elsharef et al. (2024) have developed an LLM-based threat modeling system (TMS) with NLP techniques to reduce manual labor. In this paper, two fundamental queries of threat modeling are included in the process for Task-1 and Task-2, where the NLP techniques are used to analyze and understand the threats and documents, along with LLM and synthesized volumes of documentation are provided responses based on threat modeling questions. It will respond beyond the human evaluation expectations, which means their initial findings indicate more than 75%. A combination of RAG technology, which gives LLM the ability to outperform its initial version by providing more concise and informative responses, offers a more accurate solution.

Tilwani et al. (2024) have proposed large language models (LLMs) that can perform the relevancy search based on two queries such as direct and indirect queries. Finding public and proprietary LLMs such as Anthropomorphic GPT-4 and GPT-

3.5 that impact will maximize and minimize the pass percentage (PP) and hallucination rate (HR), unnaturally make some more errors, respectively. Augmenting relevant metadata can minimize the hallucination rate (HR) and reduce the pass percentage (PP). The RAG using Mistral shows they can offer reliable and robust citations for supporting indirect queries, including their GPT-3.5 and GPT-4 performance. Xia et al. (2024) have introduced a method for realizing semantic interoperability in digital twins and enhancing Asset Administration Shell's (AAS) development as part of Industry 4.0. Due to this, a "semantic node" is introduced for the data structure to record the semantic summary of the text. It can be based on LLM, created and implemented in this semantic node process, and develops the structure of the digital twin models from raw textual data.

The result of this paper has a high generation rate of 62-79%, with a substantial proportion of the data in the source text. Then, it is translated to the target digital twin instance model with the help of LLM to validate the accuracy and perfect conversion. Hennekeuser et al. (2024) have developed LLM-based RAG to construct based on capabilities and instructional resources. For this instance, this LLM-based RAG implemented a user-centered design approach. This paper initiates the need for LLMs with RAG in higher education by lecturers with four key criteria: reliability, explainability, controllability, and trustworthiness.

For third-party analytics providers handling conversational data from multiple sources, this strategy is very relevant. Using outside data analytics providers, hypothetical case studies demonstrate the method's effectiveness and practical use in real-world healthcare circumstances. M. Khoje et.al. (2024). The method's effectiveness in protecting privacy and retaining data usefulness for analysis is assessed by the dual assessment. The approach's usefulness in common third-party analytics service contexts is further demonstrated by experimental findings using artificially generated healthcare conversational data sets.

M.Fasha.et al. (2024) has proposed a unique method that, by integrating, can help reduce the OWASP Top 10 security vulnerabilities in Large Language Model (LLM) applications. The AutoGen framework and Recovery Augmented Generation (RAG) technologies should be used, according to the concept presented in this paper, to add more security layers to LLM implementations. Working inside the framework, the intelligent agents improve security while simultaneously introducing efficiency and flexibility. By utilizing offline resources, these agents can expand their expertise, ensuring that LLMs continue to be relevant and responsive in the face of constantly changing security threats.

## III. Limitation and Motivation

In the above survey, various techniques and methods of earlier research works are discussed in detail. Most of the authors have used the LLM model to improve the performance of the various search engines. The LLM will minimize the manual intervention and increases the data analysis speed. And flexible to any type of application. The traditional RAG model has limits on generating token for each prompts. This will limits the LLM to learn the entire data in the application. And also limits on token allowance will create a impact on RAG based system on processing with lengthy or complex queries in prompt.

Especially for RAG applications, most of the researchers have proposed the LLM model to tighten the model queries. However, the major drawbacks of those models are data privacy and computational time. To overcome these issues, this paper applies the LLM model with limited parameters and tightens the security system by performing various data protection steps. This will improve the model performance and strengthen the security system of RAG applications.

## IV. Proposed Methodology

This paper proposes a large language model (LLM) to secure data in RAG applications. The overall proposed work is performed according to the following steps: data collection, data pre-processing, data labeling, LLM-based evaluation, and output. The structure of the proposed approach is shown in Figure 2.

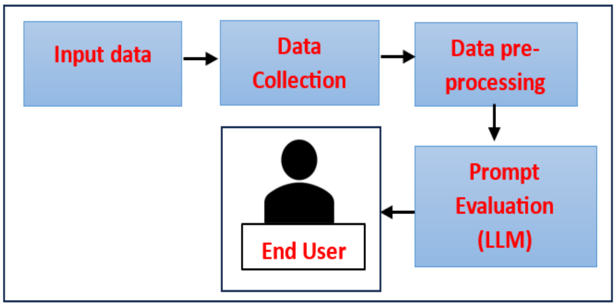


FIGURE II. Structure Of Proposed Approach

## V. Data Collection

Data collection is the initial process of measuring and gathering information from the input dataset, including structured and unstructured data. In RAG, the data collection process is achieved using LLMs, which is more suitable for data analysis. Through LMM, the input textual, numerical, or other format of data are converted into machine understandable format to simplify the decision-making process. Processing high-quality, up-to-date, and accurate data is essential to improving the performance of the LLMs.

## VI. Data pre-processing

Data preprocessing improves the quality and reduces the dimensionality of the input data. The most important steps in RAG applications are data cleaning, tokenization, normalization, and anonymization.

Data cleaning: The input data contains various artifacts and irrelevant data. Removing unwanted HTML tags, special characters, and data is essential. The data cleaning step rectifies these errors, improving the model's performance.

Tokenization: In a tokenization-based system, it is important to break down the input passages into multiple words, sentences, or units. Tokenization simplifies the LLM process and makes the input data easier for the models to understand.

Normalization: This step converts all types of input data into a uniform format and reduces the complexity of the model. The normalization process includes lemmatization and stemming, which reduce the dimension and length of the passage.

Anonymization: In data preprocessing, this step tightens the security of the input data. Its primary function is to remove personal information from the input passages. It creates user trust and data protection in RAG applications.

## VII. Data Verification Using Large Language Model (LLM)

Compared to traditional approaches, the RAG application used external sources like web applications or Wikipedia to retrieve input data. This will be more useful to the RAG in generating informative data and accurate output. Also, the RAG application is affected by new attacks and vulnerabilities. This will minimize the security and performance of the RAG system. RAG-based applications face some security issues: model manipulation, data poisoning, privacy leakage, harmful output, and biased data retrieval. Various techniques and models have traditionally been proposed to overcome this issue, but the efficiency of those methods is insufficient. This paper applies a large language model (LLM) to RAG to enhance data privacy. The goal of the proposed model is to verify and establish model security and data security, ensure access control and data infrastructure, and verify the quality and safety of the model output.

### (A). Proposed Pipeline

The proposed LLM model in the RAG application is denoted as M, the data retriever is represented as R, and the retrieval dataset is denoted as D. From the given input query Q, the predicted answer a is identified using LLM. In the RAG application, the LLM model is applied as a retriever R to perform the data classification process from the Top-K queries(q) in the input dataset D. It is expressed using the following formula.

$$R(q,d) = \{DC_1, DC_2, \dots \dots . DC_K\} \subseteq D \quad (1)$$

Here, $R(q,d)$ represent the retrieved query and data, $DC_K$ denotes the data in top-K queries. This formula is mainly used to find similar and dissimilar data for query embedding $E_q$ and compared with stored data in the database $E_D$. This formula is mainly used to find similar and dissimilar data for query embedding. $E_q$ and compared with stored data in the database $E_D$. The LMM-based retrieval step is expressed using the formula,

$$R(q,d) = \left\{DC_i \in D \mid distance\left(E_q, E_D\right) in\ the\ top - K\ relevant \right\} \quad (2)$$

Using equation (2), the distance between the predicted embedding query $\boldsymbol{E_q}$ and stored query $\boldsymbol{E_D}$ in the dataset is detected.

This process continues until the final query in the input is verified. Once all the content from D is retrieved, the clean and malicious data are verified. The result of equation (2) shows that the matched or similar queries are classified as normal, and dissimilar queries are classified as malicious and rejected from the model. The final output (A) of model M is expressed as,

$$A = M(R(q,D) \parallel q) \quad (3)$$

Here, A denotes the final output, M represented the matched content in the retrieved data and query in the $R(q,D)$ input dataset.

## VIII. Performance Evaluation

Once the input clean and malicious queries are detected from the passage, the model's performance is predicted using different metrics such as rejection rate, accuracy, and F1-score. Then, the obtained result is compared with the existing approach to find the model's efficiency.

## IX. Result and discussion

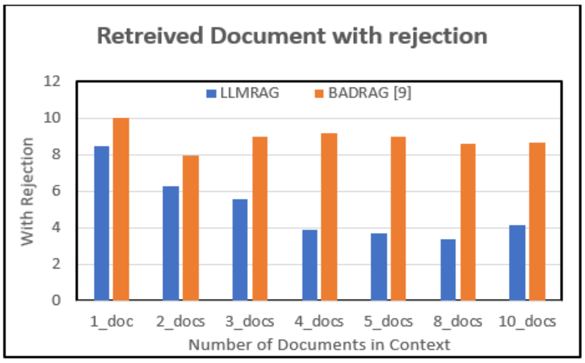
This paper proposes a large language model to detect vulnerabilities in the Retrieval Augmented Generation (RAG) application. This section discusses various simulation results of the proposed approach to detecting normal and abnormal quires in the RAG data structure.

## X. Dataset

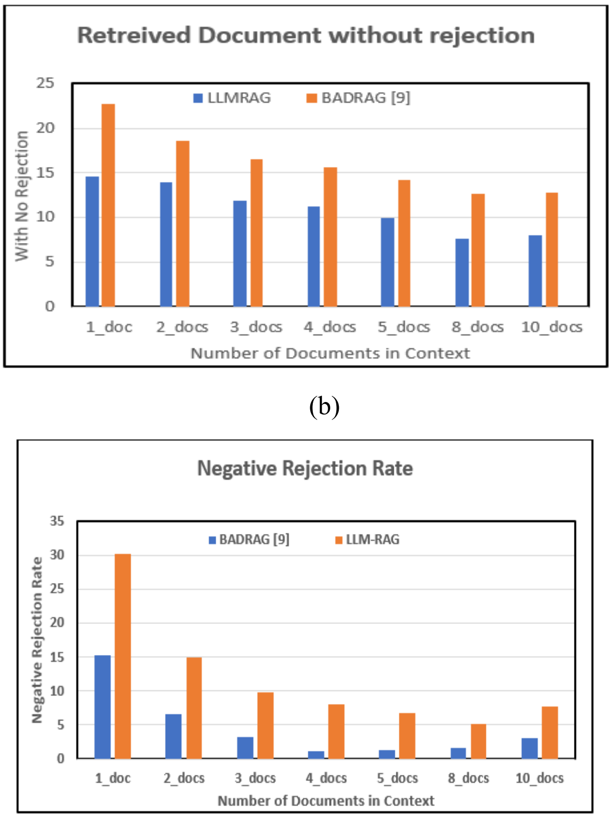
In this paper, two benchmark datasets, TriviaQA and Natural Question (NQ), are used to evaluate the efficiency of the proposed approach. The TriviaQA dataset [10] includes a pair of 950K question-answer collected from 662000 documents from various web applications and Wikipedia. It includes both machine-generated and human-verified questions. The NQ dataset [11] contains 307373, 7,803, and 7,842 train, development, and test examples. Each example comprises various queries collected from web pages and Wikipedia. Each query has a long answer or passage related to the questions.

## XI. Experimental Result

The performance of the proposed LLM approach on detecting normal and abnormal data in the input dataset is evaluated using different metrics such as rejection rate, accuracy, F1-score, and retrieval success rate. Figure-3 (a), (b), and (c) depict the total number of verses retrieved documents with rejection, without rejection, and negative rejection rates, respectively.



(a)



(b)

(c)

FIGURE III. Data Retrieval Rate (A) With Rejection (B) Without Rejection (C) Negative Rejection Rate

The model assesses the total number of documents submitted and provides the number of rejections and acceptance rates. The estimated output is compared with the BADRAG [9] method and given in Figure-3. It shows that the positive rate of rejection obtained by the proposed model is higher than the existing models. Similarly, the False negative rate of the rejection and non-rejection rate obtained using the proposed model is lower than that of the existing model.

The obtained result is compared with the existing model and is graphically depicted in Figure-3. Figure-3 (a) represents the total number of data retrieved from the input passage with rejection. That is, when the obtained result is not matched with the content in the dataset, the predicted query is rejected from the input. Figure-3(b) depicts the total number of contents retrieved without rejection compared to the existing model, and the proposed model accurately predicts the clean data from the dataset. To further prove the proposed model's efficiency, the model's negative rejection rate is detected and compared with the existing model. This will show how the proposed model responds to the retrieved queries that are not in the passage. It is noticed that when the number of documents increases, the efficiency model gets variations. However, the proposed model more effectively handles the queries and identifies the clean and malicious data in the documents.

From the two datasets, the queries are analyzed, and the cleaned and uncleaned categories are predicted. The obtained output of the proposed model is compared with the existing model and given in Table-1, which depicts the proposed LLM-RAG model's efficiency in retrieving content from the input dataset's top 50 passages or quires. The queries in each dataset have two types: clean and trigger. Then, the result is compared with the existing model's contriver [3] and Bad RAG [4]. The comparison result indicates that the proposed model more efficiently retrieves the queries from the top 50 passages. The proposed LLM technique has retrieved the clean and trigger quired on both datasets with 99.1% efficiency.

**Table I. The Percentage of Queries That Retrieve At Least One Adversary**

| Models | Queries | NQ | | | TriviaQA | | |
|---|---|---|---|---|---|---|---|
| | | Top 1 | Top 10 | Top 50 | Top 1 | Top 10 | Top 50 |
| Contriver [8] | Clean | 0.26 | 0.48 | 1.97 | 0.09 | 0.17 | 1.39 |
| | Trigger | 98.7 | 100.4 | 100.5 | 99.2 | 99.6 | 100.5 |
| BadRAG [9] | Clean | 0.19 | 0.23 | 1.07 | 0.07 | 0.4 | 0.24 |
| | Trigger | 62 | 75.4 | 86.0 | 16.8 | 30.1 | 41.9 |
| LLM-RAG | Clean | 0.35 | 90.1 | 100 | 87 | 100 | 98 |
| | Trigger | 80 | 89.9 | 99.1 | 95 | 84 | 99.2 |

Table-2 depicts the rejection, F1, and accuracy scores of the proposed LLM-RAG and existing models such as BADRAG, LLMS, and GPT-4. The comparison result depicts that on dataset NQ, the proposed model has achieved 0.01, 90.2, 98.9 rejection, F1-score, and accuracy rate, respectively, on retrieving clean queries from the NQ dataset and detecting malicious queries 99.7, 90.2, 99.1 scores achieved as rejection, F1-score, and accuracy rate, respectively. Similarly, on retrieving clean and malicious queries from the TriviaQA, the proposed LLM has achieved 0.01, 17.2, 99 rejection, F1-score, and accuracy rate, respectively, on retrieving clean queries and 99.9, 91.4, 99.5 rejection, F1-score, and accuracy rate, respectively on retrieving malicious queries.

**Table II. Performance Comparison**

| Models | Queries | NQ | | | TriviaQA | | |
|---|---|---|---|---|---|---|---|
| | | Rej. | F1 | Acc | Rej. | F1 | Acc |
| Contriver [8] | Clean | 0.4 | 8.27 | 64.6 | 0.32 | 7.88 | 76.4 |
| | Poison | 83.4 | 4.2 | 6.0 | 84.6 | 4.0 | 5.7 |
| BadRAG [9] | Clean | 0.06 | 24.2 | 93.1 | 0.05 | 19.6 | 92.1 |
| | Poison | 75.1 | 6.99 | 19.6 | 73.4 | 6.3 | 23.3 |
| LLM-RAG | Clean | 0.01 | 90.2 | 98.9 | 0.01 | 17.2 | 99 |
| | Poison | **100.0** | **90.2** | **99.1** | **99.9** | **91.4** | **99.5** |

The paper's main objective is to identify and detect the attacks present in the dataset. The dataset comprises two different attack models the data are predicted using the proposed model and compared with the existing model, given in Tabble-3, where it depicts the efficiency of the proposed and existing BadRAG model in detecting two types of attacks, such as Sentiment Steer and DOS attacks, in the input dataset. The analysis results indicate that the proposed LLM-RAG model rejected the attached file with a 90.2% rejection ratio. It achieved F1 scores and accuracy of 97 and 99% in detecting DOS attacks and 78.7, 93.8%, and 99.1% in detecting sentiment steer attacks, respectively.

**Table III. Comparison Result On Attack Detection**

| Model | Dos attack | | | Sentiment Steer attack | | |
|---|---|---|---|---|---|---|
| | **Rej.** | **F1** | **Acc** | **Rej.** | **F1** | **Acc** |
| BadRAG [9] | 83.4 | 4.2 | 6.0 | 84.6 | 4.0 | 5.7 |
| LLM-RAG | 90.2 | 9.7 | 9.9 | 78.7 | 9.3 | 9.9 |

In the internet-based applications the traffic data is analysed using the LLM model and labelled with respect to their activity such as benign, hulk, slowloris, etc. It analysis the protocol, the IP address of the source an destination, total number of packet size, and average time taken to transmit the data packet. The traffic labels describe the behaviour of the DoS attacks. Using the devices such as router and IPS. In order to identify and mitigate the attacks the LLM model follows a template for analysing and characterizing the traffic flow data. It act like a cyber security expert which gathers and analyse the information of the traffic statistics and attack behaviour.

**Table IV Accuracy rate of Attack detection**

| Dataset | Attack | #Flow | Accuracy |
|---|---|---|---|
| CIC-DoS2017 | Benign | 115,572 | 0.99 |
| | Hulk | 656 | 0.99 |
| | Slowloris | 1,027 | 0.99 |
| CIC-DoS2019 | Benign | 1,578 | 0.99 |
| | Hulk | 621 | 1.00 |
| | Slowloris | 517 | 0.99 |

## XII. Conclusion

RAG systems are advanced LLM technologies that excel in retrieving and generating information across various domains. However, they come with significant security risks that must be actively managed. To protect these systems from malicious attacks and harmful outputs, developers and users should employ security techniques such as data filtering, adversarial training, input validation, and secure APIs. Securing RAG systems prevents errors and unethical behavior and promotes a quality, safety, and ethics culture in AI development. We hope this post has offered valuable insights into securing RAG systems and encourages ongoing research and collaboration in this evolving field.

## References

[1] Mackay, A. (2024, June). Test Suite Augmentation using Language Models-Applying RAG to Improve Robustness Verification. In ERTS2024.

[2] Hu, Z., Wang, C., Shu, Y., & Zhu, L. (2024). Prompt perturbation in retrieval-augmented generation-based large language models. arXiv preprint arXiv:2402.07179.

[3] Du, X., Zheng, G., Wang, K., Feng, J., Deng, W., Liu, M., ... & Lou, Y. (2024). Vul-RAG: Enhancing LLM-based Vulnerability Detection via Knowledge-level RAG. arXiv preprint arXiv:2406.11147.

[4] Elsharef, I., Zeng, Z., & Gu, Z. Facilitating Threat Modeling by Leveraging Large Language Models.

[5] Tilwani, D., Saxena, Y., Mohammadi, A., Raff, E., Sheth, A., Parthasarathy, S., & Gaur, M. (2024). REASONS: A benchmark for REtrieval and Automated citations Of scientific Sentences using Public and Proprietary LLMs. arXiv preprint arXiv:2405.02228.

[6] Xia, Y., Xiao, Z., Jazdi, N., & Weyrich, M. (2024). Generation of Asset Administration Shell with Large Language Model Agents: Interoperability in Digital Twins with Semantic Node. arXiv preprint arXiv:2403.17209.

[7] Hennekeuser, D., Vaziri, D. D., Golchinfar, D., Schreiber, D., & Stevens, G. (2024). Enlarged Education–Exploring the Use of Generative AI to Support Lecturing in Higher Education. International Journal of Artificial Intelligence in Education, 1-33.

[8] Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., & Grave, E. (2021). Unsupervised dense information retrieval with contrastive learning. arXiv preprint arXiv:2112.09118.

[9] Xue, J., Zheng, M., Hu, Y., Liu, F., Chen, X., & Lou, Q. (2024). BadRAG: Identifying Vulnerabilities in Retrieval Augmented Generation of Large Language Models. arXiv preprint arXiv:2406.00083.

[10] https://nlp.cs.washington.edu/triviaqa/

[11] https://ai.google.com/research/NaturalQuestions

[12] M. Khoje, "Navigating Data Privacy and Analytics: The Role of Large Language Models in Masking conversational data in data platforms," *2024 IEEE 3rd International Conference on AI in Cybersecurity (ICAIC)*, Houston, TX, USA, 2024, pp. 1-5, doi: 10.1109/ICAIC60265.2024.10433801.

[13] M. Fasha, F. A. Rub, N. Matar, B. Sowan, M. Al Khaldy and H. Barham, "Mitigating the OWASP Top 10 For Large Language Models Applications using Intelligent Agents," *2024 2nd International Conference on Cyber Resilience (ICCR)*, Dubai, United Arab Emirates, 2024, pp. 1-9, doi: 10.1109/ICCR61006.2024.10532874.

[14] T. Huang, L. You, N. Cai and T. Huang, "Large Language Model Firewall for AIGC Protection with Intelligent Detection Policy," *2024 2nd International Conference On Mobile Internet, Cloud Computing and Information Security (MICCIS)*, Changsha City, China, 2024, pp. 247-252, doi: 10.1109/MICCIS63508.2024.00047.

[15] W. Huang, Y. Wang, A. Cheng, A. Zhou, C. Yu and L. Wang, "A Fast, Performant, Secure Distributed Training Framework For LLM," *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Seoul, Korea, Republic of, 2024, pp. 4800-4804, doi: 10.1109/ICASSP48485.2024.10446717.