RiskRAG: Automating Financial Risk Control with Retrieval-Augmented LLMs

Mengmei Zhang
China Telecom Bestpay
Shanghai, China
zhangmengmei@bestpay.com.cn

Dehua Xu
China Telecom Bestpay
Shanghai, China
xudehua@bestpay.com.cn

Huajian Xu
China Telecom Bestpay
Shanghai, China
xuhuajian@bestpay.com.cn

Wenbing Cui
China Telecom Bestpay
Shanghai, China
cuiwenbing@bestpay.com.cn

Fuli Meng
China Telecom Bestpay
Shanghai, China
mengfuli@bestpay.com.cn

Minwei Tang
China Telecom Bestpay
Shanghai, China
tangminwei@bestpay.com.cn

Rongyan Zhang
China Telecom Bestpay
Shanghai, China
zhangrongyan@bestpay.com.cn

Zhen Li
China Telecom Bestpay
Shanghai, China
lizhen@bestpay.com.cn

ABSTRACT

The widespread application of online payment services has transformed financial transactions, while its accessibility and convenience greatly increases the vulnerabilities to financial crimes like money laundering, threatening economic integrity and societal stability. Traditional financial risk control mechanisms rely on two-stage processes: fraudster detection and expert-driven report writing. However, they struggle under the growing transactional data volume and lacks standardized writing protocol, leading to inefficiency and inconsistency. To address these challenges, we introduce RiskRAG, a data-centric framework that automates and standardizes financial risk control from detection to reporting, leveraging Large Language Models (LLMs). To ensure the quality of reports, our RiskRAG enhances LLM's generative capabilities by retrieving similar reports from the distilled risk knowledge base, RiskKB, serving as prompts for LLM. Deployed in a real-world anti-money laundering scenario, our model demonstrates a substantial reduction in report writing time, achieving 72% usage rate. Comprehensive experimental results demonstrate the effectiveness of our RiskRAG, significantly improving the analysis accuracy and reducing hallucinations of LLM.

KEYWORDS

Large Language Model, RAG, Text Generation

ACM Reference Format:
Mengmei Zhang, Dehua Xu, Huajian Xu, Wenbing Cui, Fuli Meng, Minwei Tang, Rongyan Zhang, and Zhen Li. 2024. RiskRAG: Automating Financial Risk Control with Retrieval-Augmented LLMs. In *Proceedings of Proceedings of the ACM Web Conference 2024 (WWW '24 Data-centric Artificial Intelligence Workshop).* ACM, New York, NY, USA, 6 pages.

WWW '24 Data-centric Artificial Intelligence Workshop, May 13–17, 2024, Singapore, Singapore

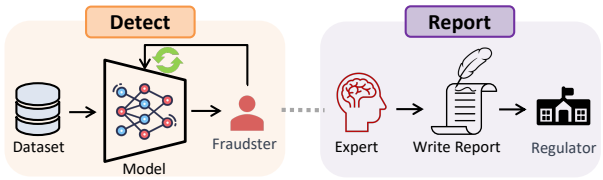
1 INTRODUCTION

Nowadays online financial services have transformed the economic landscape by offering significant convenience and efficiency in transactions. However, their rapid proliferation, convenience, and accessibility make them susceptible to financial crimes, such as money laundering [3, 8, 9], cash-out fraud [5, 18] and user default [11, 12]. This not only undermines the integrity of financial institutions but also poses huge risks to the overall economic system, facilitating criminal activities that threaten societal well-being and financial stability.

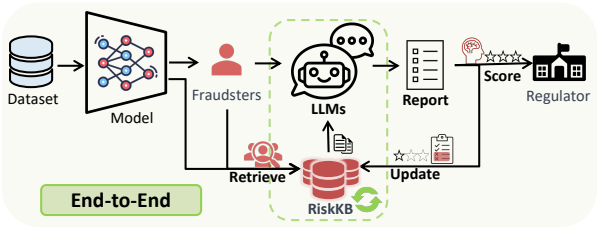
To combat the financial risks, as shown in Figure 1 (a), current transaction systems typically employ machine learning models to detect fraudsters based on their suspicious transaction patterns. For each fraudster, experts will conduct thorough analysis and write reports for regulatory authorities. In this process, previous researchers have primarily focused on the design of detection models, i.e., model-centric view, and often neglect the subsequent reporting phase. However, this two-stage workflow is becoming increasingly impractical due to two key reasons: First, with the surge in financial transaction data, manually analyzing and writing reports is not only time-consuming and labor-intensive, but also heavily reliant on expert knowledge, fail to meet the real-world requirement. Second, different experts have varying writing styles and protocols, leading to the inconsistency and unreliability of the reports. Overcoming above obstacles is crucial for improving automation, standardization, and transparency in financial risk control, thereby facilitating more effective collaboration across the industry.

Recently, Large Language Models (LLMs) like ChatGPT [1], have shown exceptional abilities of complex text understanding and generation [15], and have revolutionized diverse fields, such as

[1] https://openai.com/blog/chatgpt



(a) Typical two-stage (model-centric) framework.



(b) Our end-to-end (data-centric) framework.

Figure 1: Comparison of (a) traditional two-stage financial risk control process involving model detection then manual writing report, and (b) our proposed data-centric framework for automated financial risk control with LLM.

healthcare [10], law [2], and finance [16, 17]. These LLMs, leveraging extensive domain data, can be further enhanced by retrieving augmented generation (RAG) from external knowledge bases [4] or fine-tuning LLM [6]. Yet, their application in financial risk is yet to be explored. A question naturally arises: *Can we automate the whole financial risk control with LLM?* Specifically, we aim to generate reports with LLMs and transform the current two-stage, labor intensive financial risk control process into an end-to-end, automated framework. However, it is non-trivial to build such framework due to several challenges. (1) How can LLMs acquire the expert knowledge required for financial risk analysis? The limited quantity of risk reports is insufficient for fine-tuning LLMs, and manual reports fail to be used for RAG due to varied styles and protocols, leading to more hallucinations [7], especially for smaller, locally deployed LLMs. (2) How can we build a standardized and automated process for financial risk control? This process requires the creation of a transparent and efficient workflow, including data collection, report generation, report scoring, and model update. It involves the participation of detection models, LLMs, and human experts, and needs to support regular updates to new risk patterns to maintain performance over time.

To address the challenges and promote the automation, standardization, and transparency in financial risk control, we propose *RiskRAG* framework as shown in Figure 1 (b), featured for:

A data-centric framework: The framework begins with the identification of potential fraudsters from extensive data and extraction of their unusual transaction activities. Leveraging our constructed Risk Knowledge Base (RiskKB), LLM can analyze the potential risks within the extracted activities then generate reports. Before submitting, human experts will review and revise the low-score reports, then update them to RiskKB, enabling our *RiskRAG* to maintain its performance over time.

Distillation-based RAG method: To ensure the quality of reports, *RiskRAG* enhances our local LLM's generative capabilities by retrieving similar reports from RiskKB, which is a risk knowledge base distilled from teacher LLM and human expert.

Experimental results demonstrate that our *RiskRAG* can significantly improve the precision and recall of financial risk analysis on both macro-level content analysis and micro-level detail comparison, alleviating the hallucinations of LLM.

Upon online deployment in real-world anti-money laundering scenario, our framework has achieved an adoption rate of 72%, highlighting its considerable practical utility and broad acceptance in reality.

2 THE PROPOSED *RISKRAG*

In this section, we introduce the RiskRAG framework, a data-centric Retrieval-Augmented Generation approach that transforms the financial risk control process into an end-to-end paradigm.

2.1 Formalization of Problem

To integrate the two-stage process, we redefine the financial risk detection and reporting phases as follows.

Financial Risk Detection. In detection phase, the goal of financial institutions is to identify fraudsters from the extensive transaction data, represented as $\mathbf{D} = \{D_1, \ldots, D_N\}$ for all N users. The detection scope covers various tasks, such as money laundering, gambling, and cash-out fraud, with each employing different models. Taking the widely utilized rule engine as an example, for user i, the rule engine $\mathbf{F}$ predict whether user is financial fraudster from the user's transaction history $D_i = \{d_1, \ldots, d_n\}$. Rule engine $\mathbf{F}^{\circ}$ consists of a set of rules $\{f_1, f_2, \ldots, f_m\}$, where each rule f_j is a function that evaluates to either true or false based on certain conditions applied to user behaviors:

$$f_j : D_i \rightarrow \{\text{True}, \text{False}\}. \tag{1}$$

Taking money laundering as an example, a rule could be "single transaction amount exceeding a threshold", or more complex conditions, such as "the proportion of overnight transactions that exceed a certain percentage threshold". Finally, the rule engine outputs a set of financial fraudsters whose activities violate multiple rules.

Financial Risk Reporting. In traditional reporting phase, given the transaction behaviors $D_i = \{d_1, d_2, \ldots, d_n\}$ for fraudster i, human experts typically manually write a report T_i, which can be viewed as a narrative text $T_i = \{w_1, w_2, \ldots, w_t\}$. However, for the same fraudster i, reports written by different experts exhibit significant differences in style and content, i.e., considerable variance of T_i. This variance hinders the standardization and stability of the process, and also not supports the accumulation of deeper knowledge.

End-to-End Standard Pipeline. To integrate the two stages, for the detected fraudster i, we combine the suspicious behaviors $\tilde{D}_i \subseteq D_i$, with the corresponding rules $\mathbf{F}$, resulting in the behavior description Q_i. Then Q_i is used to query the large language models to generates the corresponding report T_i, i.e., $T_i = LLM(Q_i)$. To further standardize the whole process, we constrain the generated reports to involve risk points R and evidences E as follows: For an fraudster user i, the report T_i is a narrative text that synthesizes the analysis derived from Q_i, integrating identified risk points $R_i \subseteq R$ with

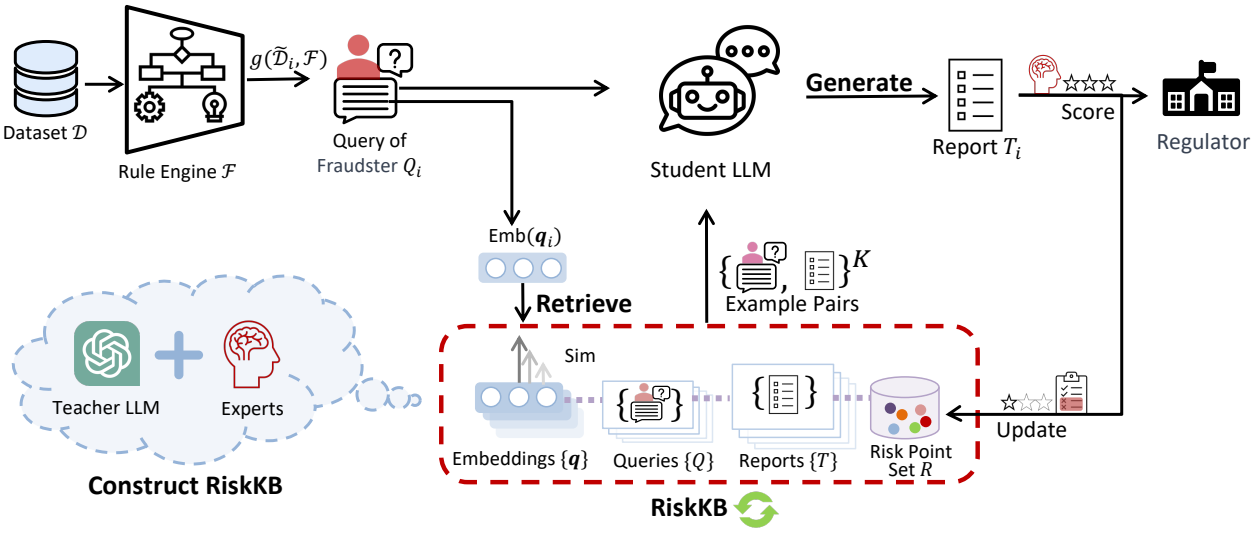


Figure 2: The overall framework of the proposed *RiskRAG*.

their corresponding evidential transaction descriptions E_i E. The set of all risk points, denoted as $R = \{r_1, \ldots, r_n\}$, encapsulates the underlying risks associated with suspicious transaction behaviors, such as "Large Withdrawals" and "Consistent Transaction Pattern". The evidence set E is subdivided into subsets corresponding to key transactional elements: $\{E_{\text{Time}}, E_{\text{Type}}, E_{\text{Amt}}, E_{\text{Qty}}\}$, which represent the time of the transaction, its type (such as Transfer, Refund, Recharge, Consumption, etc.), the transaction amount, and other quantities (like the number of credit cards), respectively.

2.2 Risk Knowledge Base Construction

To guide our local student LLM towards generating reports based on risk points R and evidence E, we propose the construction of a risk knowledge base, referred to as *RiskKB*. This knowledge base will archive exemplary query-report pairs $\{(Q_i, T_i)\}_{i=1}^{N_p}$, serving as contextual prompt for student LLM during deployment to augment the generation of standardized reports. Here, each query Q_i is derived from data $\tilde{D}_i$ and rule engine F, and the corresponding T_i represents the desired report involving risk points R and transaction elements E.

Query Construction. For a fraudster i in RiskKB, query Q_i describing the suspicious transaction $\tilde{D}_i$, can be formally represented as a sequence of words, denoted as $Q_i = \{w_1, w_2, \ldots, w_{q_i}\}$. The generation process of query q_i is non-parametric, primarily involving the matching of predefined rules F with specific transaction values $\tilde{D}_i$. Formally, this can be represented as a function $g : \mathsf{F} \quad D \; ! \; Q$. Each rule f_j in F is associated with some specific transaction values, and the function g will generate query by combine these rules to their respective values.

Report Construction. Initially, experts are engaged to manually analyze risk points based on the query and extract key elements E to construct evidence, subsequently crafting several reports T. This process ensures consistency in the style of manually written reports. The anonymized pairs of query and T are then utilized as examples for teacher LLM ChatGPT in generating the remainder of the reports. In the final step, the produced reports are manually reviewed, and the correct (Q, T) pairs are archived in RiskKB, enriching the knowledge base.

Global Risk Point Set R. From the reports stored in RiskKB, we systematically extract all risk points to establish the comprehensive risk point set R. This set serves to encapsulate the knowledge of domain experts, which will facilitate model updates and maintenance by directly incorporating emerging risk points into R.

2.3 Retrieval Augmented Generation

With the constructed RiskKB, we can retrieve similar suspicious users for new user through similarity retrieval and use their query-report pairs (Q, T) to guide the student LLM in generating reports for the new user.

Indexing and Embedding. An offline indexing procedure is conducted to enhance the retrieval efficiency. Queries within the RiskKB are transformed into high-dimensional vector spaces via an embedding model, thus facilitating nuanced similarity assessments.

$$\boldsymbol{q}_j = \text{Emb}(Q_j), \quad 8Q_j \; 2 \; \text{RiskKB}. \tag{2}$$

These query embeddings, alongside their corresponding report texts, are stored as key-value pairs, such as $(\boldsymbol{q}_j, T_j)$, establishing a foundation for efficient and scalable search functionalities.

RiskKB Retrieval. For a new fraudster i, we intially construct query by $Q_i = g(\tilde{D}_i, \mathsf{F})$, then employ the embedding model to generate its vector representation $\boldsymbol{q}_i$. The retrieval process, driven by similarity metrics, then selects the top K most similar queries from RiskKB, formalized as:

$$\{Q\}^K = \underset{Q_j 2 \text{RiskKB}}{\arg \text{Top} \quad K} \; \text{sim}(\boldsymbol{q}_i, \boldsymbol{q}_j), \tag{3}$$

where the employed similarity metric $\text{sim}^{1\ 0}$ is defined as the cosine similarity. Upon identifying the top K similar queries $\mathfrak{f}Q\mathfrak{g}^K$ from RiskKB, the system retrieves their corresponding query-report pairs, denoted as $\mathfrak{f}^1Q, T^0\mathfrak{g}^K$.

During the generation phase, the system utilizes the retrieved query-report pairs $\mathfrak{f}^1Q, T^0\mathfrak{g}^K$ as contextual input in a multi-turn dialogue setting for the student LLM. This approach facilitates the construction of a detailed report T_i for a new user i:

$$T_i = \text{LLM}^1\mathfrak{f}^1Q, T^0\mathfrak{g}^K \mathfrak{j}\mathfrak{j}Q_i{}^0, \tag{4}$$

where the LLM implicitly follows the constrained generative paradigm observed within $\mathfrak{f}^1Q, T^0\mathfrak{g}^K$, generate report for new user i that align with established criteria.

In this way, the integration of RiskKB not only ensures consistency in report generation but also imbues the automated reports with a level of detail and specificity previously attainable only through manual analysis.

2.4 RiskKB Update

In the field of financial risk control, where security is paramount, reports generated by LLMs inevitably need human verification before submission to regulatory. From a data-centric perspective, we integrate above human review and revision process into the lifecycle of the project as a critical component of maintaining and updating the RiskKB. Specifically, the reports with low scores, after human revision, can be incorporated into RiskKB. Obviously, such framework facilitates the accumulation of substantial financial risk knowledge, which can be leveraged for a variety of tasks, including question-answering systems, further enhancing the utility and scalability of the framework.

3 EXPERIMENT

3.1 Experimental settings

Datasets. We evaluate our *RiskRAG* on a real-world Anti-Money Laundering (AML) scenario, where money laundering refers to the process by which individuals conceal the origins of illegally obtained money. Our AML dataset, derived from real online payment services, is coupled with a corresponding rule engine model. Utilizing data from 279 identified fraudsters, we construct a comprehensive Risk Knowledge Base (RiskKB). Additionally, another set of data from 547 fraudsters is employed for the offline testing of our model.

Baselines. To evaluate the effectiveness of our *RiskRAG* in generating Anti-Money Laundering (AML) reports, we compare with several baselines:

Zeroshot: This baseline tests the LLM's intrinsic ability to generate AML reports by appending suspicious transaction data to a prompt without any examples.

Fewshot with Original Reports (Fewshot_Org): This model randomly select five original AML reports (written by human expert) as prompt.

RAG with Original Reports (RAG_Org): Leveraging the Retrieval-Augmented Generation framework, this method enhances the LLM's context with five similar original reports.

Few-shot Learning with RiskKB (Fewshot_RiskKB): This model randomly select five examples from our RiskKB as prompt.

LLMs. For our LLM implementation, we utilize the GPT-4, a 175B-parameter model, as the teacher model to distill RiskKB for our tasks. The student model is Baichuan2-13B [1], a smaller yet powerful model.

Evaluation. Inspired by [14], our evaluation metrics are designed to quantify the quality of generated reports, which can be divided into two categories: micro evaluation and macro evaluation, each addressing different aspects of the report quality. For the micro evaluation, we calculate the precision, recall, and F1 scores of essential elements such as time, transaction type, amount, and quantity in AML reports. For the macro evaluation, human experts will assess the accuracy of the generated content.

3.2 Micro Evaluation Results

We compute the element-aware precision and recall, which separately reflect the accuracy of the key elements included in the report and the hit rate of the key elements in the query.

Element Precision: It measures the proportion of correctly generated elements to the total elements generated by LLM. If the elements in generated report T_i do not present in the input query Q_i, it suggests the existence of hallucinations within the report, yielding a lower precision score.

Element Recall: It measures the proportion of correctly generated elements to the total elements in query. Lower recall indicates possible omissions in the T's generation.

Element F1 Score: The harmonic mean of precision and recall for elements, penalizing unbalanced performance.

Taking element Date as an example, if E_T represents the set of elements in the generated report and E_Q represents the set of elements in the corresponding query:

$$\text{Precision}_{\text{Date}} = \frac{\mathrm{j}E_T \setminus E_Q\mathrm{j}}{\mathrm{j}E_T\mathrm{j}},$$

$$\text{Recall}_{\text{Date}} = \frac{\mathrm{j}E_T \setminus E_Q\mathrm{j}}{\mathrm{j}E_Q\mathrm{j}},$$

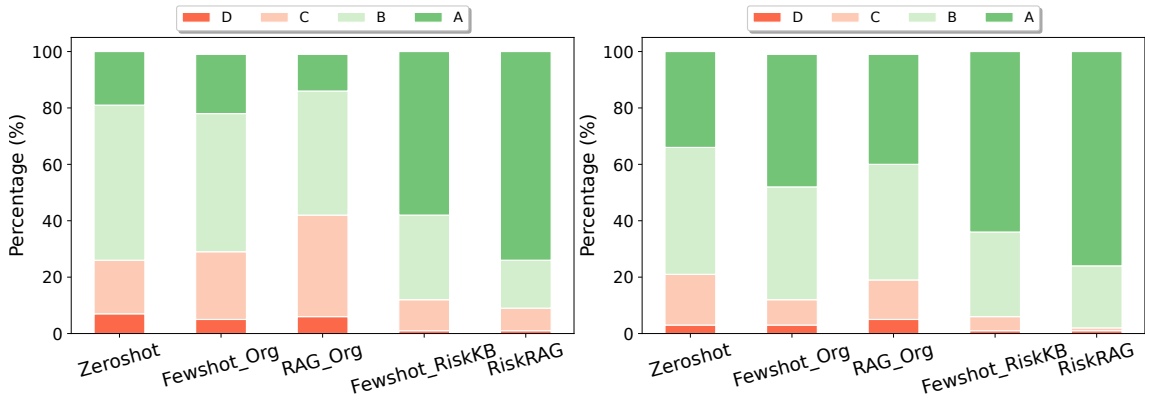
$$F1_{\text{Date}} = 2 \ \frac{\text{Precision}_{\text{Date}} \ \text{Recall}_{\text{Date}}}{\text{Precision}_{\text{Date}} \ , \ \text{Recall}_{\text{Date}}}.$$

The results of our experiments, as shown in Table 1, lead to several key observations:

(1) Our *RiskRAG* can outperform all baselines in most metrics. This success can be attributed to the effective use of the distilled RiskKB and Retrieval-Augmented Generation (RAG) in our approach.
(2) In terms of precision, our *RiskRAG* consistently achieves scores above 95%. This indicates that, with finely distilled data and retrieval mechanisms, our approach can significantly reduce the instances of hallucination typically associated with large language models.
(3) It is clear that methods based on original reports, Fewshot_Org and RAG_Org, suffer a significant drop in performance, often underperforming even Zeroshot method. This is because

Table 1: The results on Precision (P), Recall (R), and F1 score of four elements.

Model	Date			Amount			Quantity			Type		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Zeroshot	**1.000**	0.457	0.627	0.965	0.330	0.492	0.878	0.426	0.574	0.936	0.882	0.908
Fewshot_Org	0.174	0.129	0.148	0.636	0.116	0.196	0.428	0.113	0.179	0.830	0.752	0.789
RAG_Org	0.400	0.194	0.261	0.736	0.162	0.265	0.414	0.183	0.253	0.824	0.779	0.801
Fewshot_RiskKB	0.941	**0.642**	**0.763**	0.994	0.688	0.814	0.936	0.694	0.797	0.985	0.778	0.869
RiskRAG	0.966	0.629	0.762	**0.995**	**0.698**	**0.821**	**0.959**	**0.705**	**0.813**	**0.989**	**0.782**	**0.873**



(a) Precision Evaluation. **(b) Recall Evaluation.**

Figure 3: Quantitative analysis by human evaluator. The order of report quality ranking is as follows: A > B > C > D.

that the high variance within the original reports will lead to serious hallucinations when serving as contextual prompt.

(4) While the zero-shot approach may appear to have high precision, the quality of its output upon closer inspection is poor. Moreover, their formats is highly variable, and sometimes the model even fails to acknowledge the risk of money laundering, as further detailed in our macro-level experiments.

3.3 Macro Evaluation Results

To evaluate the utility and reliability of the generated reports, we conducted a quantitative analysis with four human evaluators. These evaluators reviewed a mixed set of 500 reports generated for 100 fraudsters across five models, following a blind evaluation protocol to ensure objectivity. Inspired by [13], we evaluate reports using a four-level rating system for precision and recall in risk analysis: A (75%-100%), B (50%-75%), C (25%-50%), and D (0%-25%). For example, a report achieves a "B" precision score if the evaluator judges that 50%-75% of the risk points and their corresponding analyses are correct. Similarly, a report receives an "A" recall score if 75%-100% of the risk points mentioned in the query are accurately analyzed within the report. This grading system quantifies the report's accuracy in identifying and analyzing risk points.

The experimental results, as depicted in Figure 3, reveal several key insights:

(1) Our RiskRAG model outperforms all baseline models, notably achieving more A ratings, with the majority of reports being classified within the A and B categories. This performance is attributed to the integration of RiskKB and the utilization of retrieval-augmented generation techniques, which collectively enhance the precision and relevance of the generated reports.

(2) Obviously, models lack of a high-quality knowledge base, such as the zero-shot approach and those rely on original reports, exhibit poor performance. These approaches obtain few A in precision evaluation, indicative of hallucinations within the content, which falls short of the requirements for practical financial risk control.

(3) Additionally, the Fewshot model, despite only randomly drawing five report examples from our RiskKB, still demonstrates a capacity to guide LLM towards producing standardized reports. However, its performance marginally lags behind our similarity retrieval-based approach, underscoring the efficacy of our method in facilitating more accurate and contextually relevant report generation.

3.4 Case Study

Our deployed model in an online payment service for anti-money laundering (AML) task is showcased in Figure 4. Initially, a rule engine is employed to detect money laundering suspects through vast transaction data, then their suspicious transaction histories are transformed into queries. Upon clicking the "Analyze" button, our system retrieves the top five most similar (query, report) pairs from RiskKB using query similarity, which, along with the query, will be fed into a local LLM. The LLM then returns a potential

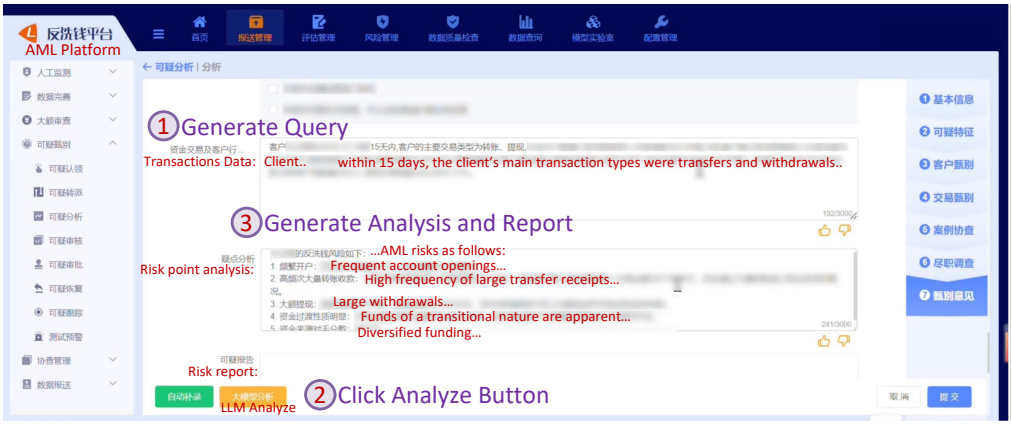


Figure 4: The online system of our RiskRAG.

risk analysis. Users can edit the report directly in the text box and evaluate its quality using the "like" or "dislike" buttons. The scores and the revised reports are recorded to update our RiskKB. For now, our system has achieved 72% usage rate, substantially reducing the time taken to write reports and streamlining the entire process with transparency and standardization.

4 CONCLUSION

In this work, we introduce *RiskRAG* a data-centric framework that seamlessly unites financial risk detection and report generation into a standard and automated process, leveraging the capabilities of Large Language Models. To ensure the quality of reports, our RiskRAG enhances LLM's generative capabilities by retrieving similar reports from the distilled risk knowledge base, RiskKB, serving as prompts for the model. Deployed in real-world anti-money laundering (AML) scenario, *RiskRAG* has proven its superiority by significantly reducing report generation times and achieving notable adoption rates. In future work, we will dedicate to extending our RiskRAG to broader scenarios, such as complex detection models and multi-modal data, enriching its analytical depth and adaptability to diverse financial crime.

REFERENCES

[1] Baichuan. 2023. Baichuan 2: Open Large-scale Language Models. *arXiv preprint arXiv:2309.10305* (2023). https://arxiv.org/abs/2309.10305

[2] Jiaxi Cui, Zongjia Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. ChatLaw: Open-Source Legal Large Language Model with Integrated External Knowledge Bases. *ArXiv* abs/2306.16092 (2023). https://api.semanticscholar.org/CorpusID:259274889

[3] Rafał Drężewski, Jan Sepielak, and Wojciech Filipkowski. 2015. The application of social network analysis algorithms in a system supporting money laundering detection. *Inf. Sci.* 295 (2015).

[4] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-Augmented Generation for Large Language Models: A Survey. *ArXiv* abs/2312.10997 (2023). https://api.semanticscholar.org/CorpusID:266359151

[5] Binbin Hu, Zhiqiang Zhang, Chuan Shi, Jun Zhou, Xiaolong Li, and Yuan Qi. 2019. Cash-Out User Detection Based on Attributed Heterogeneous Information Network with a Hierarchical Attention Mechanism. In *AAAI Conference on Artificial Intelligence*.

[6] J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *ArXiv* abs/2106.09685 (2021). https://api.semanticscholar.org/CorpusID:235458009

[7] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ArXiv* abs/2311.05232 (2023). https://api.semanticscholar.org/CorpusID:265067168

[8] Xujia Li, Yuan Li, Xueying Mo, Hebing Xiao, Yanyan Shen, and Lei Chen. 2023. Diga: Guided Diffusion Model for Graph Recovery in Anti-Money Laundering. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2023).

[9] Xiangfeng Li, Shenghua Liu, Zifeng Li, Xiaotian Han, Chuan Shi, Bryan Hooi, He Huang, and Xueqi Cheng. 2020. FlowScope: Spotting Money Laundering Based on Graphs. In *AAAI Conference on Artificial Intelligence*.

[10] K. Singhal, Shekoofeh Azizi, Tao Tu, Said Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Kumar Tanwani, Heather J. Cole-Lewis, Stephen J. Pfohl, P A Payne, Martin G. Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Scharli, Aakanksha Chowdhery, P. A. Mansfield, Blaise Agüera y Arcas, Dale R. Webster, Greg S. Corrado, Yossi Matias, Katherine Hui-Ling Chou, Juraj Gottweis, Nenad Tomaev, Yun Liu, Alvin Rajkomar, Joëlle K. Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. 2022. Large language models encode clinical knowledge. *Nature* 620 (2022), 172 – 180.

[11] Daixin Wang, Yuan Qi, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, and Shuang Yang. 2019. A Semi-Supervised Graph Attentive Network for Financial Fraud Detection. *2019 IEEE International Conference on Data Mining (ICDM)* (2019), 598–607.

[12] Daixin Wang, Zhiqiang Zhang, Yeyu Zhao, Kai Huang, Yulin Kang, and Jun Zhou. 2023. Financial Default Prediction via Motif-preserving Graph Neural Network with Curriculum Learning. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2023).

[13] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *Annual Meeting of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:254877310

[14] Yiming Wang, Zhuosheng Zhang, and Rui Wang. 2023. Element-aware Summarization with Large Language Models: Expert-aligned Evaluation and Chain-of-Thought Method. In *Annual Meeting of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:258841145

[15] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed Huai hsin Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent Abilities of Large Language Models. *Trans. Mach. Learn. Res.* 2022 (2022).

[16] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. BloombergGPT: A Large Language Model for Finance. *ArXiv* abs/2303.17564 (2023). https://api.semanticscholar.org/CorpusID:257833842

[17] Hongyang Yang, Xiao-Yang Liu, and Chris Wang. 2023. FinGPT: Open-Source Financial Large Language Models. *ArXiv* abs/2306.06031 (2023). https://api.semanticscholar.org/CorpusID:259129734

[18] Ya-Lin Zhang, Jun Zhou, Wenhao Zheng, Ji Feng, Longfei Li, Ziqi Liu, Ming Li, Zhiqiang Zhang, Chaochao Chen, Xiaolong Li, and Zhi-Hua Zhou. 2018. Distributed Deep Forest and its Application to Automatic Detection of Cash-Out Fraud. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10 (2018), 1 – 19. https://api.semanticscholar.org/CorpusID:21677807