2025 32nd Asia-Pacific Software Engineering Conference (APSEC) | 979-8-3315-6653-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/APSEC66846.2025.00079

Towards Lightweight LLM Software Solutions for InsurTech: A Framework for Scalable Question Answering Systems

Hi Kuen Yu[1], Jacky Keung[1], Yicheng Sun[1,*], Yihan Liao[1], and Richard Suen[2]
[1]Department of Computer Science, City University of Hong Kong, Hong Kong, China
[2]Faculty of Management, Shenzhen MSU-BIT University, Shenzhen, China
hikuenyu2-c@my.cityu.edu.hk, jacky.keung@cityu.edu.hk, yicsun2-c@my.cityu.edu.hk,
yihanliao4-c@my.cityu.edu.hk, 2120240147@smbu.edu.cn
*Corresponding author

***Abstract*—The integration of Large Language Models (LLMs) into software systems is transforming regulated sectors like insurance, where precision, compliance, and efficiency are essential. While proprietary LLMs like GPT-4 offer state-of-the-art performance, their closed-source nature and high computational demands constrain adoption in privacy-sensitive and cost-restricted InsurTech environments. In response, this paper investigates how lightweight, open-source LLMs can be effectively deployed for domain-specific question answering in insurance, emphasizing software engineering considerations such as modularity, inference stability, and prompt orchestration. We propose a software-engineered evaluation framework tailored to insurance-related tasks, featuring modular prompt management, automated rubric-based evaluation, and backend support for reproducibility and compliance tracking. A curated benchmark dataset derived from the Hong Kong Insurance Intermediaries Qualifying Examination (IIQE) is constructed to reflect real-world regulatory and operational challenges. Ten open-source models are systematically evaluated across four question types using both standard and Chain-of-Thought (CoT) prompting strategies. Our findings show that compact models such as DeepSeek-R1-1.5B achieve strong accuracy with minimal resource consumption, making them suitable for practical deployment. CoT prompting further enhances reasoning performance, particularly for models with 3B parameters or more. With proper prompt design and modular deployment, lightweight LLMs can support secure, efficient, and interpretable InsurTech applications, enabling trustworthy AI-driven software systems in regulated domains.**

***Index Terms*—Large Language Models, Prompt Engineering, InsurTech, Chain-of-Thought Reasoning, Software System Integration**

I. Introduction

The rise of Large Language Models (LLMs) such as GPT-4, LLaMA, and DeepSeek has significantly advanced the capabilities of natural language processing, driving progress in question answering, code generation, and conversational AI. As summarized by Hou et al. [1], LLMs are rapidly transforming various stages of the software development lifecycle, including design, implementation, testing, and maintenance. At the same time, Ozkaya [2] warns that while these tools are powerful, they pose risks related to hallucination, overconfidence, and the difficulty of distinguishing between human-written from AI-generated content. These developments increasingly influence software system design, particularly in the creation of intelligent components that support automation and user interaction.

Aghajani et al. [3] highlight that software documentation is often inadequate or outdated, hindering understanding and maintenance. This motivates the use of LLMs to generate or augment documentation in a task-aware and context-sensitive manner, especially valuable in regulated domains where transparency and clarity are critical. In such domains, including insurance, Cao et al. [4] emphasize that integrating LLMs into software systems offers promising opportunities to enhance operational efficiency.

However, it simultaneously raises challenges regarding compliance, reliability, and cost. Aryan et al. [5] note that organizations must carefully balance model accuracy, generalization, and deployment cost, while Jin et al. [6] emphasize that effective LLM adoption requires disciplined engineering practices, such as model monitoring, orchestration, and cost management, to ensure safe and efficient operation in real-world software environments.

The InsurTech sector exemplifies this transformation, as software engineers actively leverage LLM-powered solutions, demonstrated by Jin et al. [6], to automate claim processing, policy explanation, and customer support. Kemell et al. [7] observe that LLMs are increasingly employed as evaluators within software pipelines, supporting tasks such as scoring, correctness verification, and error detection. However, domain-specific constraints, demonstrated by Anwar et al. [8] and Fan et al. [9], including legal accuracy, data sensitivity, and regulatory compliance, require careful software design. The integration of LLMs into regulated industries, as highlighted by Kemell et al [7], underscores the need for explainable, secure and resource-efficient system architectures that ensure reliable outputs.

Furthermore, reliance on closed-source LLMs such as GPT-4 raises concerns about data privacy and system transparency. This paper investigates how LLMs can be effectively integrated into insurance-related question-answering systems, highlighting software engineering considerations such as deployment strategies, performance trade-offs, and alignment with industry standards.

979-8-3315-6653-1/25/$31.00 ©2025 IEEE
DOI 10.1109/APSEC66846.2025.00079

Wang et al. [10] emphasize that many organizations opt for self-hosted, fine-tuned open-source LLMs to safeguard sensitive code and data despite the performance gap relative to proprietary models. Chen et al. [11] finds that targeted language selection enhances performance while reducing training time in code-related tasks, reinforcing the importance of lightweight models, such as DeepSeek-R1-1.5B, in environments where privacy, cost control, and efficiency must be carefully balanced.

Given these constraints, lightweight and open-source LLMs offer a pragmatic alternative for the InsurTech applications. They balance computational efficiency, cost management, and data protection, making them well suited to deployment in regulated settings. Kang et al. [12] further suggest that with careful prompting and supporting tool, smaller or self-hosted models can handle a large portion of tasks typically assigned to expensive cloud-based systems.

To address the specific engineering demands of this domain, this paper proposes a modular integration framework that facilitates the adoption of lightweight LLMs for insurance-specific question-answering tasks. Yang et al. [13] emphasize the non-functional requirements of the framework such as efficiency, privacy, and explainability are essential to trustworthy AI system design. It supports plug-and-play compatibility with multiple open-source models. In this study, DeepSeek-R1-1.5B is adopted as the default engine due to its strong downstream performance. Alizadeh et al. [14] demonstrate that optimized model deployment enables both full-precision and quantized variants to balance inference accuracy with energy efficiency and latency in offline environments.

Our work focuses on insurance-related question answering, a task demanding precision, compliance awareness, and contextual reasoning. To support this, we construct a domain-aligned benchmark inspired by Hong Kong's Insurance Intermediaries Qualifying Examination (IIQE), encompassing real-world topics such as policy definitions, regulatory compliance, actuarial calculations, and case-based advisories. Yu et al. [15] show that instruction tuning can significantly improve LLM adaptability to domain-specific tasks when coupled with well-structured prompts. This supports our use of tailored prompt engineering and instruction formats to guide insurance-related question answering with high accuracy and relevance.

We evaluate multiple open-source LLMs using both standard prompting and Chain-of-Thought (CoT) prompting strategies across varied question types. To explore the effectiveness and limitations of LLMs in insurance-related software applications, this study defines three core objectives: (1) to evaluate and compare the performance of different LLMs on domain-specific insurance tasks, (2) to analyze performance variation across question typologies, including definition-based, calculation-based, scenario-based, and regulation-focused, and (3) to assess the impact of CoT prompting on accuracy in reasoning-intensive contexts.

Through this empirical investigation, we demonstrate the feasibility of building efficient, secure, and regulation-compliant InsurTech systems using lightweight LLMs. The key contributions of this paper are as follows:

- We present a comparative benchmark of several open-source Large Language Models (LLMs) on a domain-specific insurance question answering task, using a dataset aligned with real-world regulatory and operational requirements.
- We conduct a typology-driven analysis of LLM performance across four distinct question categories: definition-based, calculation-based, scenario-based, and regulation-focused items, highlighting the strengths and weaknesses of each model.
- We evaluate the effectiveness of Chain-of-Thought (CoT) prompting in enhancing reasoning accuracy for complex, context-sensitive insurance questions.
- We provide practical engineering insights and design recommendations for integrating LLMs into InsurTech systems, focusing on compliance-aware architecture, model selection strategies, and deployment considerations for regulated environments.

II. Methodology

We adopt a hybrid evaluation framework that combines model performance benchmarking with software-oriented integration analysis. This dual-layer approach not only measures the accuracy of large language models (LLMs) on insurance-specific content, but also evaluates their practicality for deployment in real-world InsurTech systems. From a software engineering perspective, this includes factors such as domain alignment, input/output reliability, and model adaptability within regulated application environments.

To assess the language models' effectiveness in handling insurance-related questions, we employ the Insurance Intermediaries Qualifying Examination (IIQE) as our benchmark. The IIQE serves as an authoritative and standardized certification for individual licensees in Hong Kong, encompassing diverse regulatory and operational aspects of insurance practice. Its well-structured design provides a robust foundation for constructing evaluation tasks that mirror real-world insurance workflows and compliance requirements, making it ideally suited for benchmarking domain-specific LLM performance within a software-integrated context.

A. Dataset

For this study, we constructed five examination sets, each consisting of 255 multiple-choice questions randomly sampled from the official IIQE question bank. All questions are written in Traditional Chinese, consistent with the language used in the actual IIQE examination. Each set includes 75 questions from the Principles and Practice of Insurance (P&P) Paper, 50 from the General Insurance (GI) Paper, 50 from the Long Term Insurance (LT) Paper, and 80 from the Investment-linked Long Term Insurance (IL) Paper. Every question is annotated with a reference answer and metadata (e.g. question type and source paper) to support granular evaluation. Table I summarizes the distribution of questions across the five sets.

TABLE I: Distribution of questions across the five examination sets.

Label	IIQE	Set 1	Set 2	Set 3	Set 4	Set 5	Total
PP	Principles and Practice of Insurance Examination	75	75	75	75	75	375
GI	General Insurance Examination	50	50	50	50	50	250
LT	Long Term Insurance Examination	50	50	50	50	50	250
IL	Investment-linked Long Term Insurance Examination	80	80	80	80	80	400
Total		255	255	255	255	255	1275

The dataset comprises four distinct question types, each targeting a specific dimension of insurance knowledge. Definition-based questions evaluate the model's understanding of foundational concepts, such as terminology, principles, and policy structures. Compliance-based questions test regulatory comprehension, including legal obligations, insurance legislation, and compliance standards. Scenario-based questions assess contextual reasoning and real-world decision-making. Calculation-based questions examine numerical reasoning and the application of actuarial or financial formulas relevant to insurance practice.

This typology establishes a structured basis for evaluating LLM capabilities across varied cognitive dimensions. By systematically categorizing questions, we enable fine-grained performance analysis, identifying each model's strengths and weaknesses in interpreting, reasoning about, and applying insurance knowledge. The distribution of these question types across the five examination sets is detailed in Table II.

B. Evaluated Language Models

Ten open-source LLMs of varying architectures and parameter scales were evaluated: DeepSeek-R1 model (1.5B), Gemma 2 models (2B and 9B), Llama 3.1 (8B), Llama 3.2 models (1B and 3B), and Qwen 2.5 models (0.5B, 1.5B, 3B, and 7B). Table III summarizes the models and their parameter sizes.

The selected models represent different trade-offs between reasoning ability, computational efficiency, and deployment scalability, key considerations for InsurTech software systems. **DeepSeek-R1** (Guo et al. [16]) features a 1.5B-parameter architecture optimized for reasoning via reinforcement learning, producing chain-of-thought (CoT) output efficiently. **Gemma 2** (Gemma Team et al. [17]) employs Transformer-based architectures with reinforcement learning from human feedback (RLHF), demonstrating strong performance across 2B and 9B scales, particularly in compliance-sensitive or edge-deployable settings.

The **Llama** series (Dubey et al. [18]) offers multilingual capability and flexible deployment. Both the 8B model from Llama 3.1 and the 1B/3B models from Llama 3.2 are tested for latency-sensitive environments. **Qwen 2.5** (Bai et al. [19]) provides domain-adaptive reasoning across models ranging from 0.5B to 7B parameters.

All models incorporate modern instruction-tuning methods such as Direct Preference Optimization (DPO) and support long-context inference, making them well suited for complex insurance tasks. Collectively, they demonstrate the growing maturity of open-source LLM ecosystems as viable alternatives to proprietary systems for compliant, cost-effective InsurTech deployment.

C. Prompting Strategies

To evaluate the reasoning and decision-making capabilities of LLMs in insurance-related question answering, we designed two prompting strategies: *Standard Prompting* and *CoT Prompting*. Both follow a zero-shot format, requiring the model to analyze a multiple-choice question and select the most appropriate answer from the given options.

The goal of this comparison is to determine whether explicit reasoning instructions improve model performance, especially in tasks involving domain-specific logic, regulatory interpretation, actuarial computation, and scenario-based judgment. In both setups, the models are also required to provide justifications for their answers, allowing simultaneous evaluation of answer accuracy and explanation quality, critical for interpretability and trustworthiness in regulated domains such as insurance.

From a software engineering perspective, prompt design serves as a lightweight but powerful mechanism for guiding model behavior. In compliance-sensitive environments like InsurTech, prompts function as structured interaction modules that ensure LLM outputs remain auditable, interpretable, and aligned with regulatory requirements. Properly engineered prompts thus enhance both trust and system accountability.

1) Standard Prompting: The model is instructed to read the question and answer options, select the best choice, and provide a brief justification. This setup is most effective for definition-based or factually grounded questions. The standard prompt is designed as follows:

> Analyze the following question and options, then select the best answer. Provide a brief reasoning for your choice.
> `Input: [Question and Options]`

2) CoT Prompting: The CoT strategy extends the standard prompt by instructing the model to "think step by step", encouraging the generation of intermediate reasoning steps before reaching a conclusion. This is particularly useful for tasks requiring contextual interpretation, policy analysis, or numerical reasoning. The CoT prompt is designed as follows:

> Analyze the following question and options, then select the best answer. Provide a brief reasoning for your choice.
> `[Input: Question and Options]`
> Let's think it step by step.

TABLE II: Distribution of question types across the five examination sets

Label	Type	Set 1	Set 2	Set 3	Set 4	Set 5	Total
CAL	Calculation-based Question	2	4	1	3	2	12
COM	Compliance-based Question	64	67	60	63	64	318
DEF	Definition-based Question	125	119	123	119	121	607
SCE	Scenario-based Question	64	65	71	70	72	338
Total		255	255	255	255	255	1275

TABLE III: List of evaluated models

No.	Label	Model	Parameter	Size
1	Deepseek-R1-1.5B	DeepSeek-R1	1.5B	Small
2	Gemma-2-2B	Gemma 2	2B	Small
3	Gemma-2-9B	Gemma 2	9B	Medium
4	Llama-3.2-1B	Llama 3.2	1B	Small
5	Llama-3.2-3B	Llama 3.2	3B	Small
6	Llama-3.1-8B	Llama 3.1	8B	Medium
7	Qwen-2.5-0.5B	Qwen 2.5	0.5B	Small
8	Qwen-2.5-1.5B	Qwen 2.5	1.5B	Small
9	Qwen-2.5-3B	Qwen 2.5	3B	Small
10	Qwen-2.5-7B	Qwen 2.5	7B	Medium

This step-by-step design enhances traceability, transparency, and post-decision auditability, critical features for regulated domains. CoT responses can also be logged and reviewed, making them suitable for explanation-aware interfaces or human-in-the-loop validation.

By comparing both strategies across our benchmark dataset, we provide empirical insights into how prompt engineering influences model performance and system interpretability in insurance applications.

III. Empirical Setup

A. System Architecture

To enable structured experimentation and reproducible benchmarking, we developed a modular software evaluation system tailored for LLM-based insurance question answering. The system architecture prioritizes flexibility, scalability, and traceability, supporting prompt selection, model inference, and automated evaluation. The workflow, shown in Fig. 1, comprises of two key stages: *Examination* and *Evaluation*.

In the **Examination** stage, each insurance question is routed through a *Prompt Manager*, which formats the question into either a *Standard Prompt* or a *CoT Prompt*, depending on the experimental configuration. The generated prompt is then processed by the *Model Execution Engine*, which interfaces with all ten open-source models. Each model generates a predicted answer and, where applicable, its reasoning.

In the **Evaluation** stage, the system captures the question, ground-truth answer, and model-generated output. This information is passed to a *Rubric-based Evaluator*, which scores both answer accuracy and explanation quality according to predefined criteria. All metadata (model name, prompt type, question type, and scores) are stored in a MongoDB database for reproducibility, visualization, and subsequent analysis.

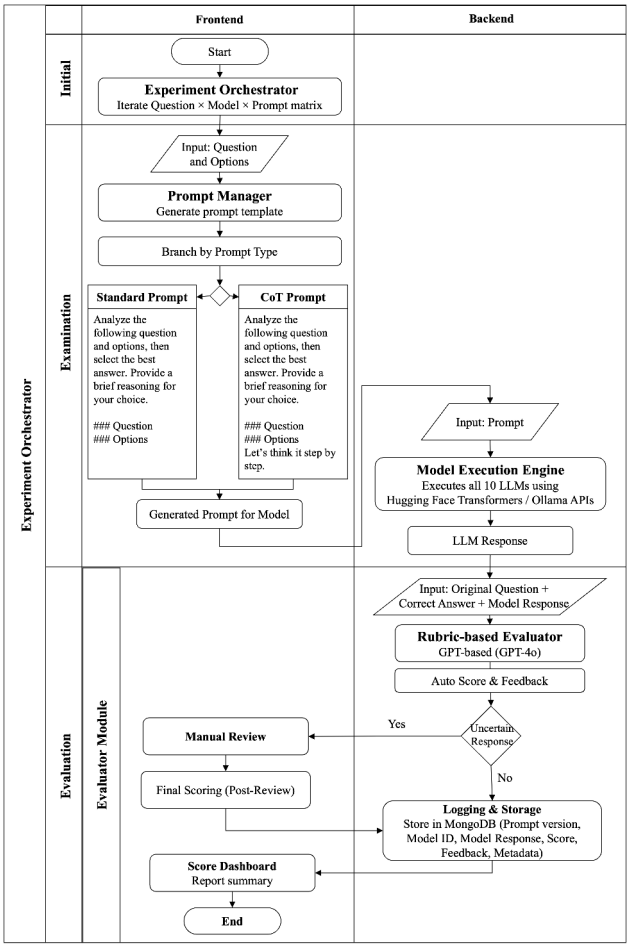

Fig. 1: LLM Evaluation Workflow: Insurance questions are processed through standard or CoT prompt templates, executed by the model engine, and scored using a rubric-based evaluator. Results are stored in MongoDB for analysis.

This architecture provides end-to-end automation for domain-specific evaluation while satisfying software-engineering standards for modularity and audit-ability, principles emphasized in trust-worthy AI system design.

B. Core Components and Experimental Control

The experimental infrastructure consists of five modular components, each designed for reproducibility, fairness, and integration within InsurTech platforms.

The **Prompt Manager** handles the dynamic selection and formatting of prompt templates. It maintains version control for both *standard* and *CoT* prompts, ensuring consistent and repeatable input generation across experiments.

The **Model Execution Engine** serves as the inference backend, interfacing with models via Hugging Face Transformers and Ollama APIs. The engine supports batch execution, caching, and configurable token limits, optimizing throughput while maintaining compatibility with varying model architectures.

The **Evaluator Module** employs GPT-4o as an external judgment model to assess both answer correctness and explanation quality. A rubric-based assessment is applied, following the interpretability and auditability practices proposed by Chundi et al. [20] and Kang et al. [21]. The rubric evaluates correctness, relevance, and clarity, with human review applied in ambiguous cases to ensure integrity.

All outputs, including model responses, prompts, scores, and logs, are stored in the **MongoDB-based Database Layer**, enabling complete traceability and longitudinal analysis.

Finally, an **Experiment Orchestrator** automates the full experimental cycle, ensuring consistent testing across all models, question types (definition, scenario, compliance, calculation), and prompting strategies. It also manages scheduling, error handling, and performance logging to maintain reliability.

This modular design supports scalable and transparent benchmarking, key requirements for evaluating AI systems in regulated industries.

C. Dataset Formatting and Inference Workflow

A standardized inference workflow ensures consistency across all evaluated models. The IIQE dataset comprises five sets of 255 multiple-choice questions, covering four papers: (P&P, GI, LT, and IL. Each question contains four options and one correct answer. The preprocessing and inference pipeline proceeds as follows:

1) **Input Formatting**: Each question is embedded into either a standard or CoT prompt template according to the experimental setting.
2) **Response Generation**: The model generates an answer and, where applicable, an accompanying explanation.
3) **Answer Extraction**: The system parses the model's output to extract the selected answer (e.g., A, B, C, or D).
4) **Result Logging**: All outputs, including responses, prompt type, and metadata, are stored in the MongoDB database.

To account for stochastic variability in LLM outputs, each model is executed five times per exam set and per prompting strategy. The final accuracy scores are computed as the mean across runs, providing a more reliable estimate of model performance. This repetition-based control method aligns with best practices in empirical AI evaluation.

D. Research Questions

The study is structured around three research questions:

RQ1: Which LLMs perform best on domain-specific insurance-related questions in terms of accuracy and clarity?

Ten open-source LLMs (0.5B–9B parameters) were benchmarked on 1,275 multiple-choice questions from the Insurance Intermediaries Qualifying Examination (IIQE), covering (P&P, GI, LT, and IL papers. All models were tested under Standard Prompting, which instructed them to select the best answer and provide a short justification. Each model was run five times to reduce stochastic variation, and average accuracy was computed across runs. This setup provides a controlled comparison of factual accuracy and robustness in applying LLMs to insurance-specific knowledge.

RQ2: Does CoT prompting improve LLM performance on reasoning-intensive insurance questions?

All ten models were further evaluated using Zero-shot CoT Prompting, where an additional instruction to "think step by step" encouraged explicit reasoning before answering. Models generated both an answer and explanation, which were scored for accuracy and explanation quality using a rubric-based assessment with GPT-4o. Each experiment was repeated five times per exam set, and results were averaged. This design enables a controlled comparison of how structured reasoning instructions affect model performance, particularly in compliance interpretation and client-case analysis.

RQ3: How does performance vary across different question types (e.g., definition-based, calculation-based, scenario-based, compliance-based)?

All IIQE questions were annotated into four categories, Calculation, Compliance, Definition, and Scenario, and each model was evaluated under both Standard and CoT prompting. This typology-based analysis reveals model strengths and weaknesses across reasoning styles and knowledge types, highlighting where targeted fine-tuning or domain adaptation may further enhance InsurTech applications.

E. Evaluation Metrics and Scoring Procedure

To evaluate model performance on the IIQE benchmark, we employed a structured, multi-dimensional scoring pipeline. The evaluation framework measures both answer correctness and explanation quality, using automated model-based scoring and statistical aggregation across multiple runs to ensure reliability and interpretability.

All model responses were assessed by GPT-4o, configured with a rubric prompt to evaluate four aspects:

- **Correctness (1 or 0)**: Whether the selected answer matches the ground truth.
- **Relevance (1 or 0)**: Whether the explanation references key insurance concepts or reasoning paths.
- **Clarity (1 or 0)**: Whether the explanation is concise, coherent, and free of unnecessary content.
- **Overall Comment**: A qualitative summary generated by GPT-4o to support human review.

Overly long outputs were truncated to a predefined token limit for consistency. All responses, evaluation scores, and related metadata were logged in a MongoDB database for detailed and reproducible analysis. The pipeline, fully automated

with integrated logging, captured essential information such as model predictions, correctness, and prompt type (Standard or CoT). This structured logging enabled efficient post-hoc statistical analysis and visualization across models, prompting methods, and question categories.

We adopted several quantitative metrics to compare models under different prompting strategies and question types:

- **Accuracy:** Ratio of correct answers to total questions:

$$\text{Accuracy} = \frac{\text{Number of correct answers}}{\text{Total number of questions}}$$

- **Stability via Averaged Scoring:** Each model was evaluated five times per exam set under each prompting strategy; final scores were averaged to reduce variance from stochastic generation.
- **Prompting Effectiveness:** Comparison of performance under Standard and CoT prompting to quantify the impact of structured reasoning cues.
- **Model-wise and Size-wise Comparison:** Cross-family evaluation (e.g., Llama, Gemma, Qwen, DeepSeek) and parameter-scale comparison (small 0.5B–3B vs. medium 7B–9B) to assess how architecture and scale affect performance.
- **Component-Level Analysis:** Results broken down by IIQE sections - (P&P, GI, LT, and IL, to examine model sensitivity to different regulatory subdomains.
- **Question-Type Analysis:** Performance categorized by question type - definition-, compliance-, scenario-, and calculation-based, to identify model strengths and weaknesses across reasoning styles.

This comprehensive, rubric-guided evaluation framework provides fine-grained, reproducible performance assessment across linguistic, conceptual, and regulatory dimensions, enabling a clearer understanding of LLM capabilities in domain-specific insurance question answering.

IV. Study Results

A. Comparative Accuracy and Clarity Across LLMs (RQ1)

To address RQ1, we evaluated the accuracy and explanatory clarity of ten open-source LLMs across five IIQE examination sets under both Standard and CoT prompting strategies. Tables IV reports detailed accuracy rates per model and examination set under the standard prompting condition.

Under the Standard Prompting, **DeepSeek-R1-1.5B** consistently outperformed all other models, achieving the highest average accuracy of **51.11%** across the five sets. Despite its relatively small parameter size (1.5B), the model exhibited remarkable consistency across all IIQE components and question types, demonstrating an exceptional balance between performance and efficiency.

Qwen-2.5-7B also achieved strong results, with an average accuracy of **39.31%** under standard prompting. Notably, this model exhibited the largest improvement when transitioning from Standard to CoT prompting, especially in compliance-based and scenario-based questions—highlighting its capability for complex reasoning and policy interpretation.

Gemma-2-9B showed solid performance on regulation-focused questions, underscoring its strength in rule-based reasoning.

In contrast, smaller-capacity models such as **Qwen-2.5-0.5B** and **Gemma-2-2B** recorded substantially lower accuracies (below 16%), reflecting the limitations of ultra-lightweight models in handling knowledge-intensive and context-dependent tasks.

Findings: For InsurTech applications emphasizing both accuracy and reasoning reliability, DeepSeek-R1-1.5B and Qwen-2.5-7B emerge as leading candidates. Their balance of model size, reasoning capacity, and computational efficiency makes them suitable for real-world deployment. Mid-scale models such as Qwen-2.5-3B and Llama-3.1-8B offer moderate performance and may be used within hybrid routing strategies—delegating complex queries to larger models while assigning simpler tasks to smaller, latency-optimized ones.

B. Effectiveness of CoT Prompting (RQ2)

To examine the effect of CoT prompting on reasoning-intensive insurance tasks, all ten LLMs were re-evaluated using the IIQE dataset under the Zero-shot CoT prompting condition. As shown in Table V and Fig. 2, CoT prompting consistently improved accuracy across all models, though the magnitude varied with model architecture and size.

DeepSeek-R1-1.5B maintained its leading position, achieving an average accuracy of **52.88%**, a modest yet stable gain over its standard baseline. **Qwen-2.5-7B** recorded a significant increase from **39.31%** to **50.21%**, confirming the effectiveness of CoT prompting for reasoning-heavy contexts.

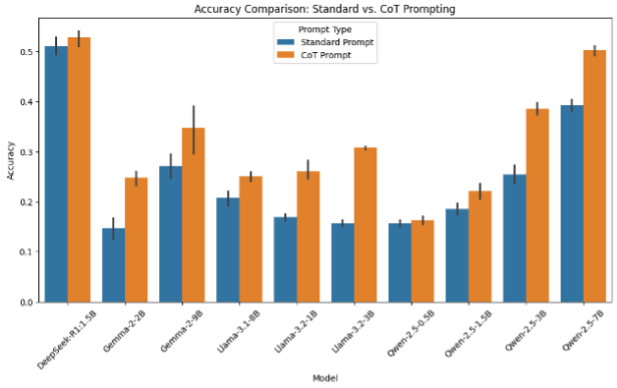

Fig. 2: Comparison of accuracy between Standard and CoT prompting across models.

Mid-sized models, such as *Llama-3.2-3B* and *Qwen-2.5-3B*, experienced the largest relative improvements (+15.11 and +13.15 percentage points, respectively). This suggests that CoT prompting is particularly beneficial for models with moderate capacity, as it helps activate latent reasoning pathways and improves performance in domain-specific reasoning.

By contrast, high-performing models under the standard prompting (e.g., *Deepseek-R1-1.5B* and *Qwen-2.5-7B*) exhibited smaller but consistent improvements (+1.77% and +10.90%, respectively), reflecting strong baseline reasoning

TABLE IV: Accuracy Rates of Models Using Standard Prompting

Model	Set 1	Set 2	Set 3	Set 4	Set 5	Average
Deepseek-R1-1.5B	48.31%	50.51%	53.33%	53.33%	50.04%	**51.11%**
Gemma-2-2B	18.12%	14.59%	14.20%	11.06%	15.61%	14.71%
Gemma-2-9B	28.55%	23.06%	27.06%	30.98%	26.04%	27.14%
Llama-3.2-1B	18.12%	16.78%	16.16%	16.31%	17.02%	16.88%
Llama-3.2-3B	16.55%	15.61%	16.47%	15.14%	14.82%	15.72%
Llama-3.1-8B	22.35%	18.59%	19.37%	22.51%	20.63%	20.69%
Qwen-2.5-0.5B	16.16%	16.71%	15.53%	15.14%	14.75%	15.65%
Qwen-2.5-1.5B	19.29%	16.78%	17.80%	18.04%	20.71%	18.53%
Qwen-2.5-3B	27.45%	24.24%	23.22%	24.00%	28.31%	25.44%
Qwen-2.5-7B	40.86%	37.10%	38.75%	40.24%	39.61%	39.31%

TABLE V: Accuracy Rates of Models Using CoT Prompting

Model	Set 1	Set 2	Set 3	Set 4	Set 5	Average
Deepseek-R1-1.5B	54.12%	52.78%	53.73%	54.27%	49.49%	**52.88%**
Gemma-2-2B	23.22%	22.75%	24.55%	26.98%	26.27%	24.75%
Gemma-2-9B	41.57%	26.20%	32.63%	37.65%	35.84%	34.78%
Llama-3.2-1B	24.86%	24.55%	30.35%	24.94%	25.33%	26.01%
Llama-3.2-3B	30.90%	30.20%	30.82%	31.45%	30.75%	30.82%
Llama-3.1-8B	23.37%	24.63%	25.65%	25.25%	26.35%	25.05%
Qwen-2.5-0.5B	16.00%	15.69%	16.94%	17.65%	15.14%	16.28%
Qwen-2.5-1.5B	24.78%	19.61%	22.27%	20.86%	22.90%	22.09%
Qwen-2.5-3B	36.47%	38.27%	40.24%	39.84%	38.12%	38.59%
Qwen-2.5-7B	51.29%	48.55%	50.35%	49.57%	51.29%	50.21%

stability. Smaller models, including *Qwen-2.5-0.5B* and *Qwen-2.5-1.5B*, showed minimal change, suggesting limited reasoning depth.

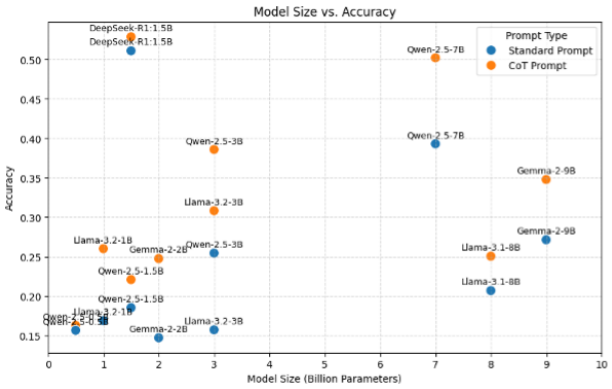

Fig. 3: Comparison of accuracy between Standard and CoT prompting across models.

The relationship between model size and accuracy is visualized in Fig. 3, which presents a scatter plot comparing each model's performance under both Standard and CoT prompting strategies. The figure reveals a clear trend: as model size increases, the relative performance gains from CoT prompting become more pronounced. This pattern suggests that larger models are more capable of leveraging structured reasoning cues embedded in CoT prompts to enhance domain-specific question answering. For example, models such as `Qwen-2.5-7B` and `Gemma-2-9B` exhibited significant accuracy improvements under CoT prompting, likely due to their enhanced parameter capacity that enables deeper contextual representation and multi-step reasoning. In contrast, smaller models like `Qwen-2.5-0.5B` and `Llama-3.2-1B` showed only marginal gains, indicating limited capacity to utilize the additional reasoning path provided by CoT.

These results suggest that while CoT prompting is a promising software-level strategy to improve LLM interpretability and performance, its effectiveness is closely tied to model architecture and parameter scale. Therefore, in designing intelligent InsurTech solutions, the selection of prompt strategies must consider model size to maximize reasoning efficiency and cost-performance trade-offs.

Findings: CoT prompting generally enhances reasoning and interpretability, especially for models with $\geq$ 3B parameters. However, for deterministic tasks such as numerical or actuarial calculations, Standard Prompting may remain preferable to minimize verbosity and prevent reasoning-induced errors.

C. Performance Variation Across Insurance Question Types (RQ3)

To analyze how prompting strategies influence specific question categories, we conducted a comparative assessment across four types: Definition-based, Compliance-based, Scenario-based, and Calculation-based questions, details as shown in Table VI.

TABLE VI: Average Accuracy per Question Type (in percentage)

Model	Calculation			Compliance			Definition			Scenario		
	S	CoT	+	S	CoT	+	S	CoT	+	S	CoT	+
DeepSeek-R1:1.5B	70.00	60.00	-10.00	53.02	56.54	3.52	48.57	50.58	2.01	53.20	53.31	0.12
Gemma-2-2B	8.33	16.67	8.33	14.72	25.16	10.44	14.76	25.14	10.38	14.85	23.96	9.11
Gemma-2-9B	26.67	21.67	-5.00	26.29	35.47	9.18	29.36	35.16	5.80	23.96	33.91	9.94
Llama-3.1-8B	21.67	26.67	5.00	18.36	23.21	4.84	22.08	26.36	4.28	20.36	24.38	4.02
Llama-3.2-1B	10.00	21.67	11.67	16.79	25.72	8.93	17.66	26.92	9.26	15.80	24.79	8.99
Llama-3.2-3B	6.67	16.67	10.00	15.91	29.56	13.65	15.91	31.24	15.32	15.50	31.78	16.27
Qwen-2.5-0.5B	10.00	8.33	-1.67	14.28	15.72	1.45	16.38	16.34	-0.03	15.86	16.98	1.12
Qwen-2.5-1.5B	11.67	10.00	-1.67	17.74	19.87	2.14	20.23	23.00	2.77	16.45	22.96	6.51
Qwen-2.5-3B	18.33	23.33	5.00	22.58	38.43	15.85	26.79	40.23	13.44	25.98	36.33	10.36
Qwen-2.5-7B	26.67	31.67	5.00	35.72	49.06	13.33	40.79	51.57	10.77	40.47	49.53	9.05

Definition-based questions benefited consistently from CoT prompting. Models such as `Qwen-2.5-7B` (40.79% $\rightarrow$ 51.57%) and `Gemma-2-9B` demonstrated improved lexical and conceptual precision. Mid-sized models like `Llama-3.2-3B` and `Qwen-2.5-3B` gained substantially (+15.33% and +13.44%, respectively), indicating that CoT enhances conceptual articulation.

Compliance-based questions, requiring contextual interpretation of regulatory frameworks, also saw strong gains. `Qwen-2.5-7B` improved from 35.72% to 49.06%, and `Qwen-2.5-3B` from 22.58% to 38.43%, whereas smaller models displayed marginal gains due to capacity constraints.

Scenario-based questions, demanding multi-step reasoning, exhibited the highest relative improvements. `Deepseek-R1-1.5B` remained stable (53.20% $\rightarrow$ 53.31%), while `Qwen-2.5-7B` achieved a substantial jump (40.47% $\rightarrow$ 49.53%).

Calculation-based questions produced mixed results. CoT prompting sometimes degraded numerical accuracy by introducing extraneous reasoning. For example, `Deepseek-R1-1.5B` dropped from (70.00% $\rightarrow$ 60.00%), and `Gemma-2-9B` declined by 5 points. Although some smaller models improved slightly, the overall trend suggests CoT may not suit arithmetic-intensive questions.

Findings: CoT prompting enhances reasoning-intensive categories in definition, compliance, and scenario-based questions, but may reduce precision in calculation tasks. Therefore, prompt strategy should be adapted to question type to optimize LLM performance in InsurTech applications.

V. Implications

A. Model Selection and Suitability

The empirical findings highlight that model selection in InsurTech applications must balance both technical constraints and task-specific demands. While larger models such as `Qwen-2.5-7B` deliver superior reasoning accuracy, compact models like `DeepSeek-R1-1.5B` achieve competitive performance at a fraction of the computational cost. This makes them ideal for edge deployment or cost-sensitive systems where latency, privacy, and energy efficiency are critical.

For mission-critical applications, such as client suitability assessments, policy interpretation, and regulatory compliance, mid- to large-scale models ($\geq$ 3B parameters) are recommended due to their robust reasoning under CoT prompting. In contrast, sub-2 B models are suitable mainly for retrieval-based or fact-centric tasks with limited reasoning complexity.

From a software engineering perspective, prompt engineering, particularly zero-shot CoT, emerges as a cost-effective alternative to fine-tuning. The results demonstrate consistent gains in scenario-based and compliance-driven questions when structured prompts are applied. Nevertheless, CoT is not universally optimal; in calculation-oriented or deterministic problems, it may introduce verbosity and arithmetic noise. Accordingly, a dynamic prompt-routing mechanism should be adopted, automatically selecting prompt structures based on question classification, versioning, and runtime logic detection to ensure both efficiency and precision.

B. Modular and Scalable Deployment Architecture

For real-world deployment, InsurTech platforms should adopt modular and microservice-based architectures, where LLMs function as interchangeable service components with scalable interfaces. A backend routing layer can dynamically assign incoming queries to models based on task complexity. Lightweight models handle frequent, low-risk, or routine queries. Mid- or large-scale models (e.g., `Qwen-2.5-7B`) handle reasoning-intensive or compliance-critical inquiries. Retrieval-augmented backends process context-rich or high-stakes decisions.

All model outputs should pass through a post-processing pipeline that performs answer verification, log storage, and fallback management. Such a design ensures traceability, explainability, and fault tolerance, satisfying both technical reliability and regulatory auditability standards essential in the insurance sector.

C. Practical Integration Pathways and System Robustness

As illustrated in Fig. 4, we propose a hybrid LLM integration architecture for insurance applications. Incoming queries are first classified and routed to one of several back-ends: a

lightweight base model; a CoT-enhanced model for reasoning-intensive questions; a retrieval-augmented generator (RAG) for compliance-heavy or document-referenced tasks.

The system's output then flows through a multi-stage verification layer, encompassing scoring, reliability assessment, and optional human-in-the-loop review. This layered approach minimizes latency for simple inquiries while ensuring robustness and accuracy for complex, high-risk decisions.

The proposed design aligns with modern software engineering best practices, including microservice orchestration, fault-tolerant pipelines, and explainable AI governance. It offers a scalable, auditable, and adaptive foundation for embedding LLMs into regulated InsurTech environments—bridging the gap between innovation and compliance.

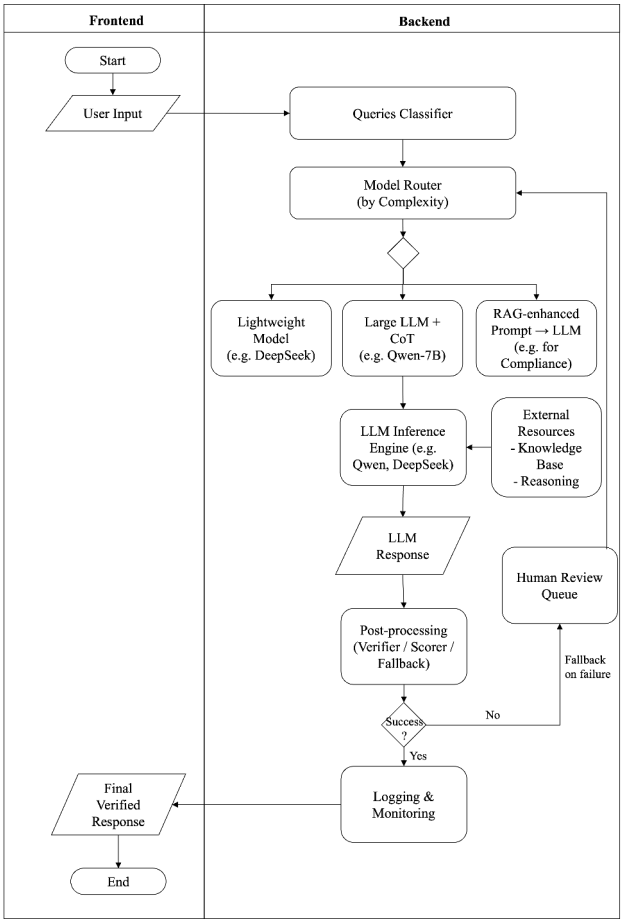

Fig. 4: Proposed InsurTech LLM integration architecture.

VI. Threats to Validity

A. Construct Validity

Construct validity may be affected by how question difficulty and domain specificity were operationalized. Although this study utilized 1,275 real-world insurance questions and manually categorized them into four distinct types, definition, compliance, scenario, and calculation, in the annotation process may contain elements of subjective bias.

Furthermore, while accuracy served as the primary evaluation metric, other critical dimensions such as fairness, response calibration, and hallucination tendency were not directly quantified. These unmeasured factors may influence interpretability and limit the comprehensiveness of the evaluation. Future replications should consider multi-dimensional metrics that incorporate reliability, explanation consistency, and regulatory adherence.

B. External Validity

External validity may be constrained by the study's reliance on an IIQE-based benchmark, which reflects the specific linguistic and regulatory context of Hong Kong's insurance industry. While the dataset captures authentic reasoning patterns and regulatory logic relevant to InsurTech applications, its generalizability to other jurisdictions, languages, or regulatory systems remains uncertain.

To broaden applicability, future research should extend this framework to multilingual datasets, localized insurance standards, and cross-border regulatory corpora. Such extensions would enhance the robustness and transferability of the findings across different insurance markets and governance environments.

C. Internal and Methodological Validity

Although multiple evaluation safeguards were applied, such as rubric-based scoring, five-run averaging, and human-in-the-loop review, methodological threats remain. Differences in model architecture, tokenization schemes, and inference randomness could still introduce variance in results. While averaging mitigates this, it does not fully eliminate model stochasticity.

Additionally, the rubric-based evaluator (GPT-4o), despite showing strong alignment with human judgment, may itself carry latent evaluation bias depending on prompt phrasing or rubric sensitivity. Incorporating ensemble evaluators or expert-reviewed subsamples could further validate scoring integrity in future studies.

VII. Related Work

A. Prompt Engineering and CoT Reasoning

Prompt engineering has become a lightweight yet powerful alternative to full model fine-tuning, particularly when adapting Large Language Models (LLMs) for domain-specific tasks. Techniques such as zero-shot and few-shot prompting enable practitioners to elicit specialized behaviors from general-purpose models without retraining. Among these approaches, Chain-of-Thought (CoT) prompting, introduced by Wei et al. [22] has gained wide attention for improving multi-step reasoning in arithmetic, logical, and commonsense inference.

According to Vatsal and Dubey [23], prompt engineering methods can substantially influence performance across natural-language-processing (NLP) tasks by refining contextual understanding and reasoning depth. Similarly, Dakhel et al. [24] found that structured prompts, especially step-by-step CoT instructions, significantly enhance semantic accuracy and

reduce hallucination in complex code-generation scenarios. These results underscore the importance of well-structured prompting in precision-critical and traceable domains such as InsurTech.

CoT prompting encourages an LLM to decompose complex problems into intermediate reasoning stages before reaching a conclusion. Although the method has proven effective on general benchmarks such as GSM8K, SVAMP, and MATH, its application in high-stakes, regulated domains in law, medicine, or insurance, remains underexplored. Domain-specific question-answering (QA) requires precise legal interpretation and regulatory reasoning that may not benefit uniformly from CoT strategies.

Recent research has begun to address this limitation. For instance, Shi et al. [25] applied CoT prompting to legal reasoning, highlighting the need for domain-tailored adaptation to achieve interpretability and compliance alignment. Building on this foundation, our study applies CoT prompting to insurance QA and analyzes its impact across four regulatory question types: compliance, definition, scenario, and calculation-based items.

From a software-engineering standpoint, integrating CoT into production systems requires robust prompt versioning, intent classification, and fallback handling. The modular architecture developed in this study includes a Prompt Manager and an Experiment Orchestrator to ensure reproducibility and traceability of CoT performance across heterogeneous model deployments.

B. LLMs in Regulated Domains: The Case for InsurTech

The adoption of LLMs in regulated industries such as insurance, banking, and healthcare presents unique challenges. Beyond achieving raw accuracy, these systems must produce explainable, auditable, and legally defensible outputs.

Ahlgren et al. [26] observed that LLMs are increasingly leveraged for evaluative assistance in software engineering, complementing or replacing human reviewers in code validation and quality assessment. Their findings demonstrate the feasibility of rubric-based LLM evaluation pipelines in domains where accountability is critical, including insurance.

Traditional benchmarks like MMLU or HELM inadequately represent the domain-specific reasoning, terminology, and numerical rigor required in insurance QA. Examinations such as the IIQE test nuanced regulatory interpretation and premium calculation, areas poorly covered by general datasets.

To address this, Chen et al. [27] introduced FinQA, a financial QA dataset grounded in banking scenarios, while Shi et al. [25] examined LLM reliability in legal text generation. Nevertheless, such studies often lack multi-type question evaluation and comparative analysis of prompt strategies. Our work fills this gap by introducing an IIQE-aligned dataset and a modular, rubric-based evaluation framework capable of assessing both factual correctness and explanation quality across diverse insurance question types.

C. LLM-as-Evaluator: Scalable Assessment of Answer Quality

Manual evaluation of LLM outputs is time-consuming and inconsistent, particularly when scaling to multiple prompt formats. Recent studies propose using LLMs as automated evaluators, exploiting their ability to apply structured rubrics to assess output quality.

OpenAI and various academic groups have shown that GPT-4 can approximate human judgment in tasks assessing writing quality, dialogue coherence, and factual accuracy. Fan et al. [28] advanced this approach through Sedareval, a self-adaptive rubric framework that decomposes evaluation into interpretable dimensions in correctness, relevance, and clarity, in which achieve strong correlation with human ratings.

In parallel, open-source LLMs such as DeepSeek, Gemma, Llama, and Qwen offer cost-effective and privacy-preserving alternatives to proprietary APIs. Gao et al. [29] demonstrated that compact models like DeepSeek can replicate expert-level QA evaluation when aligned with domain-specific rubrics, enabling trustworthy offline assessment pipelines.

Consistent with Thode et al. [30], our framework employs structured rubric prompts to evaluate both factual accuracy and explanatory clarity, ensuring interpretability and auditability. Moreover, compact models (1–7 billion parameters) show competitive performance when combined with optimized prompting and reasoning strategies, underscoring their practicality for privacy-sensitive and resource-constrained InsurTech applications.

VIII. Conclusion

This study presents a comprehensive framework for evaluating open-source LLMs on insurance-related, domain-specific tasks. Leveraging a curated dataset inspired by Hong Kong's Insurance Intermediaries Qualifying Examination (IIQE), we systematically benchmarked ten open-source models across four question types and two prompting strategies — Standard Prompting and CoT Prompting.

The proposed modular architecture supports scalable and reproducible experimentation, integrating a rubric-based evaluation pipeline powered by GPT-4o for assessing both answer accuracy and explanation quality. Our findings show that compact models such as DeepSeek-R1-1.5B and Qwen-2.5-7B deliver strong performance relative to their size, demonstrating that lightweight models can be practically deployed in privacy-sensitive and cost-constrained InsurTech environments.

We further find that the effectiveness of CoT prompting depends on model scale and question type: larger or mid-sized models ($\geq$ 3B parameters) benefit most from structured reasoning instructions, while calculation-oriented tasks may still favor standard prompting. These results provide actionable guidance for model selection, prompt design, and system integration within regulated industries that demand accuracy, transparency, and compliance.

Building on this foundation, future research may explore the integration of retrieval-augmented generation (RAG) pipelines linked to domain-specific knowledge bases to enhance factual grounding and compliance alignment. Further work could

also involve fine-tuning open-source LLMs on annotated insurance corpora to improve performance in scenario-based and calculation-heavy contexts. Expanding to multilingual datasets, incorporating real-time user feedback, and conducting adversarial robustness testing would strengthen both reliability and practical deployment readiness for production-grade InsurTech solutions.

REFERENCES

[1] X. Hou, Y. Zhao, Y. Liu, Z. Yang, K. Wang, L. Li, X. Luo, D. Lo, J. Grundy, and H. Wang, "Large language models for software engineering: A systematic literature review," *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 8, pp. 1–79, 2024.

[2] I. Ozkaya, "Application of large language models to software engineering tasks: Opportunities, risks, and implications," *IEEE Software*, vol. 40, no. 3, pp. 4–8, 2023.

[3] E. Aghajani, C. Nagy, M. Linares-Vásquez, L. Moreno, G. Bavota, M. Lanza, and D. C. Shepherd, "Software documentation: the practitioners' perspective," in *Proceedings of the acm/ieee 42nd international conference on software engineering*, 2020, pp. 590–601.

[4] C. Cao, X. Yao, W. Gong, Y. Li, Y. Pan, Y. Yang, and C. Zhao, "Llms for insurance: Opportunities, challenges and concerns," 2024.

[5] A. Aryan, A. K. Nain, A. McMahon, L. A. Meyer, and H. S. Sahota, "The costly dilemma: generalization, evaluation and cost-optimal deployment of large language models," *arXiv preprint arXiv:2308.08061*, 2023.

[6] H. Jin, L. Huang, H. Cai, J. Yan, B. Li, and H. Chen, "From llms to llm-based agents for software engineering: A survey of current, challenges and future," *arXiv preprint arXiv:2408.02479*, 2024.

[7] K.-K. Kemell, M. Saarikallio, A. Nguyen-Duc, and P. Abrahamsson, "Still just personal assistants?–a multiple case study of generative ai adoption in software organizations," *Information and Software Technology*, p. 107805, 2025.

[8] U. Anwar, A. Saparov, J. Rando, D. Paleka, M. Turpin, P. Hase, E. S. Lubana, E. Jenner, S. Casper, O. Sourbut *et al.*, "Foundational challenges in assuring alignment and safety of large language models," *arXiv preprint arXiv:2404.09932*, 2024.

[9] A. Fan, B. Gokkaya, M. Harman, M. Lyubarskiy, S. Sengupta, S. Yoo, and J. M. Zhang, "Large language models for software engineering: Survey and open problems," in *2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE)*. IEEE, 2023, pp. 31–53.

[10] J. Wang, Y. Huang, C. Chen, Z. Liu, S. Wang, and Q. Wang, "Software testing with large language models: Survey, landscape, and vision," *IEEE Transactions on Software Engineering*, vol. 50, no. 4, pp. 911–936, 2024.

[11] F. Chen, F. H. Fard, D. Lo, and T. Bryksin, "On the transferability of pre-trained language models for low-resource programming languages," in *Proceedings of the 30th IEEE/ACM international conference on program comprehension*, 2022, pp. 401–412.

[12] S. Kang, J. Yoon, N. Askarbekkyzy, and S. Yoo, "Evaluating diverse large language models for automatic and general bug reproduction," *IEEE Transactions on Software Engineering*, 2024.

[13] Z. Yang, Z. Sun, T. Z. Yue, P. Devanbu, and D. Lo, "Robustness, security, privacy, explainability, efficiency, and usability of large language models for code," *arXiv preprint arXiv:2403.07506*, 2024.

[14] N. Alizadeh, B. Belchev, N. Saurabh, P. Kelbert, and F. Castor, "Language models in software development tasks: An experimental analysis of energy and accuracy," in *2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR)*. IEEE, 2025, pp. 725–736.

[15] L. Yu, E. Alégroth, P. Chatzipetrou, and T. Gorschek, "Measuring the quality of generative ai systems: Mapping metrics to quality characteristics-snowballing literature review," *Information and Software Technology*, p. 107802, 2025.

[16] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[17] G. Team, M. Riviere, S. Pathak, P. G. Sessa, C. Hardin, S. Bhupatiraju, L. Hussenot, T. Mesnard, B. Shahriari, A. Ramé *et al.*, "Gemma 2: Improving open language models at a practical size," *arXiv preprint arXiv:2408.00118*, 2024.

[18] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv e-prints*, pp. arXiv–2407, 2024.

[19] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang *et al.*, "Qwen2. 5-vl technical report," *arXiv preprint arXiv:2502.13923*, 2025.

[20] G. Chundi, A. Dawar, S. Sarwar, S. Prasad, M. Vosbikian, and I. Ahmed, "Comparative evaluation of llms in orthopedic surgery," *Journal of Orthopaedic Reports*, p. 100728, 2025.

[21] H. Kang and X.-Y. Liu, "Deficiency of large language models in finance: An empirical examination of hallucination," *arXiv preprint arXiv:2311.15548*, 2023.

[22] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[23] S. Vatsal and H. Dubey, "A survey of prompt engineering methods in large language models for different nlp tasks," *arXiv preprint arXiv:2407.12994*, 2024.

[24] A. M. Dakhel, A. Nikanjam, V. Majdinasab, F. Khomh, and M. C. Desmarais, "Effective test generation using pre-trained large language models and mutation testing," *Information and Software Technology*, vol. 171, p. 107468, 2024.

[25] J. Shi, Q. Guo, Y. Liao, and S. Liang, "Legalgpt: legal chain of thought for the legal large language model multi-agent framework," in *International Conference on Intelligent Computing*. Springer, 2024, pp. 25–37.

[26] T. L. Ahlgren, H. F. Sunde, K.-K. Kemell, and A. Nguyen-Duc, "Assisting early-stage software startups with llms: Effective prompt engineering and system instruction design," *Information and Software Technology*, p. 107832, 2025.

[27] Z. Chen, W. Chen, C. Smiley, S. Shah, I. Borova, D. Langdon, R. Moussa, M. Beane, T.-H. Huang, B. Routledge *et al.*, "Finqa: A dataset of numerical reasoning over financial data," *arXiv preprint arXiv:2109.00122*, 2021.

[28] Z. Fan, W. Wang, X. Wu, and D. Zhang, "Sedareval: Automated evaluation using self-adaptive rubrics," *arXiv preprint arXiv:2501.15595*, 2025.

[29] C. Gao, X. Chen, and G. Zhang, "Sva-icl: Improving llm-based software vulnerability assessment via in-context learning and information fusion," *Information and Software Technology*, p. 107803, 2025.

[30] L. Thode, U. Iftikhar, and D. Mendez, "Exploring the use of llms for the selection phase in systematic literature studies," *Information and Software Technology*, p. 107757, 2025.