Cost-Aware Agentic Architectures for Multi-Model Routing and Tool-Use Optimisation in CRM Workflows

Sri Hari Deep Kolagani[1], Dr. Mamata Bhandar[2]

[1]Research Scholar, GlobalNxt University
[2]Dean, School of Business, Nanyang
Institute of Management

Abstract:
The study presents research into the application of cost-conscious agentic support in the field of multi-model routing and tool-use optimization to the CRM post-sale processes. It is focused on how to dynamically dispatch tasks to suitable model tiers (small, medium, large LLMs) with a view to the minimum expense of API services and high software quality. The study aims to examine how Salesforce Service Cloud can be used to carry out post-sales CRM activities, including case escalation and email summarization. The major observations include that Dynamic task routing led to a saving of 33% of API cost, relative to using the largest LLM at all times, without any substantial decrease in Customer Satisfaction (CSAT) (>90%), nor First-Contact Resolution (FCR) (>85%). Query optimization of the vector database and limiting of external API calls resulted in a 15-20% reduction in query time and a 10%-15% cut in total operating expense per agent. The findings indicate that multi-tier model routing can be helpful to cut costs in CRM workflow, with the potential highest savings in operation costs of up to 35%, while providing quality service. This research offers an effective, cost-effective model of refining CRM processes and proposes great insights to future studies on edge computing and reinforcement learning towards further optimization.

Keywords: CRM workflows, multi-model routing, cost optimization, LLMs, Salesforce Service Cloud, API costs.

1. Introduction

The advancement of customer relationship management (CRM) systems has been largely influenced by the integration of artificial intelligence (AI) in the post-sales operations. Escalation of cases, access to knowledge, and summarization of emails are some tasks that have also emerged with more complexity as businesses struggle to achieve emerging customer demands to deliver faster and precise service. A 2023 Salesforce study estimates that companies that consist of AI-assisted CRM systems have realized 25% efficiency gains in operations, especially after sales services [1]. Automation of routine work allows customer service representatives to devote more resources to more strategic work, which would lead to a faster response time and better service quality.

Customer support has further been changed by the emergence of Large Language Models (LLMs), including GPT-3 and GPT-4 [2]. These models provide highly developed natural language processing options and are useful in helping to answer client enquiries, address cases, and also access knowledge databases. Nonetheless, there has been a big problem due to the excessive use of large models in terms of the high cost of operation. For example, GPT-3 or GPT-4 are types of LLM that require a significant API cost, including above 0.01 per token in certain instances. These expenses may quickly spiral out of control, particularly in systems where huge models are used in performing jobs that can comfortably be performed using smaller models. Although they are very expensive, most CRM systems still employ huge models whenever responding to any query, making them very inefficient in their utilization of resources and overall price.

The major problem with the current CRM workflows is that the large LLM costs very much when applied to every task. Although large model types are state-of-the-art performers, they are resource-intensive and can be relatively redundant when one is handling simple orders. Smaller and cheaper LLMs may also work well with frequent queries, but are not often used. Most CRM systems still have the point of using big models in doing everything, even the complex tasks, thus leading to over-utilization of resources and increased expenses. The issue with this is to come up with a cost-effective solution whereby multi-tiered model selections relative to the complexity of the task should be considered, where the simpler ones cannot be designated to the small and less expensive models, and more complex ones to the one created using large and powerful models. This study aims to solve this problem by developing an architecture that fulfills the goals of minimizing the costs of operations without sacrificing the quality of services and response speed.

The research questions guiding this study are:

1. What is the relationship between model tier selection (small vs large LLM), prompt complexity, and per-case dollar cost and resolution quality in CRM agent workflows?
2. Can a multi-model router reduce API costs by more than 30% compared to always using the largest model policies without degrading CSAT or first-contact resolution metrics?
3. How do vector database query optimization (embedding model choice, retrieval depth) and tool-use frequency (API calls to external systems) contribute to total operational cost per agent?

This study will focus on workflows of post-sales CRM, particularly within the system of Salesforce Service Cloud. To test the proposed architecture, the learning will rely on real-life datasets found in such platforms as HuggingFace, Kaggle, and SaaS benchmarks. These data sets model the typical CRM operations, which would offer a realistic foundation for testing the efficiency of multi-tiered routing plans in the minimization of operation expenses.

This study is divided into several chapters in order to attain its goals. The Literature Review will address the current studies on CRM workflow after sales, the application of AI within the workflow, and the prices of large LLMs. The Methods chapter shall outline the direction taken, data gathering, and analysis procedures of the evaluation of the multi-tiered model routing. The Results section will list the experiments and provide the findings, including the savings on costs brought about by the dynamic routing. The Discussion offers the interpretation of these findings within the framework of practical CRM applications, and the Conclusions will manipulate the findings and summarize them, indicating the suggestion for future research and real-world CRM optimization strategies.

2. Literature Review

2.1 Overview of CRM Post-Sales Workflows and AI Integration

The customer relationship management (CRM) post-sales processes have been more advanced as organizations aim to offer rapid and more effective customer support. The use of artificial intelligence (AI) tools, especially in case escalation, knowledge retrieval, and ticket resolution, is another trend towards this direction. The studies demonstrate that the AI-based CRM software has decreased response time by 30% and support costs by 20-40% [3]. For example, some services such as Zendesk adopted the use of AI to automatic responses to queries, which not only made cases easier to manage but also did not overburden the support workers. However, the new move toward AI, in turn, creates some difficulties as well, especially in terms of the costs of executing large models like GPT-based LLMs.

Implementation of the GPT-based models with CRM systems has transformed query automatization of common queries, making the results highly accurate, and it can save time and enhance customer satisfaction [4]. This automation is associated with a huge increase in the cost of the API. Where smaller models are cheaper and more efficient in forms of simple queries, large LLMs are expensive (however, powerful), especially when used in situations that do not need exploitation of their full features. As a result, companies are seeking cheaper methods of maximizing the application of LLMs to CRM systems.

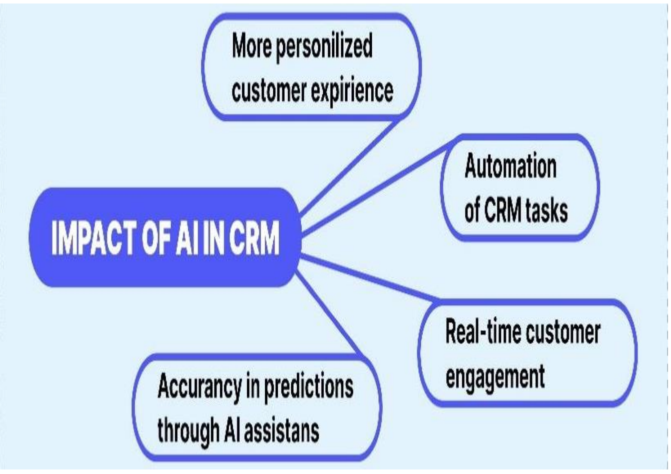
More personilized customer expirience
IMPACT OF AI IN CRM
Automation of CRM tasks
Real-time customer engagement
Accurancy in predictions through AI assistans

Figure 1: A summary of the effects of AI in CRM, such as customer experience, automation of processes, real-time customer interaction, and efficient utilization of cost and model size in the post-sale processes.

Figure 1 shows the effects of AI in CRM, where AI increases customer experience by providing customized experiences and real-time engagement. The automation of CRM work is also depicted in the image and enhances the efficiency as the response time is cut down by 30% and the support cost by 20-40%. However, it highlights the difficulty of the high API costs of large GPT-based LLMs. Although smaller models are less expensive to use with typical queries, large models are required in more complicated operations, but they are also accompanied by high operational expenses, so companies consider cost-effective solutions to reconcile AI utilization in CRM systems.

2.2 Multi-Agent Systems in CRM

Multi-agent systems (MAS) are a promising method of dealing with the complexities of CRM workflows. These systems make use of various agents, or AI models, in order to dynamically arise to assign tasks depending on query complexity. IBM Watson offers a bright example of such a system, where only other models are used to address various support tickets depending on their complexity [5]. For example, straightforward queries can be pushed to smaller, less expensive models, whereas elaborate queries are moved to more viable models. This solution is cost-effective and performance-based based such that when a business does not need to be overly complex, there is an easy solution that is cheap to use.
PracticCRM multi-agent systems have been demonstrated to save on costs and improve customer satisfaction. These systems can help companies to address the increase in volume of cases without a proportionate rise in price as a result of mismanagement of workload distribution. The studies mention that multi-agent methods may decrease operational expenses by 30-40%, particularly in cases of their utilization that layer tiered AI systems, adapting to the complexity of new tasks. [6; 7].

2.3 Cost Optimization in CRM with AI

One of the current issues in terms of CRM is cost optimization, an issue especially relevant when AI models like LLMs gain more popularity. A cost-sensitive architecture with tasks being redirected to alternative models depending on the level of complexity may lead to significant savings. Research has

indicated that by using multi-tiered AI systems, in which simpler tasks get assigned to less expensive and more basic models and more complex tasks are diverted to larger models, companies may realize cost savings of 30-40% and still get good services. As an example, HuggingFace customer support ticket datasets show that small models are capable of answering up to 80% of standard queries, which itself is a significant way to reduce the cost of the operation without the impact of the solution quality [8].
This strategy takes advantage of the fact that AI is flexible, so that no business uses the most costly models to perform all the tasks. It has been found that smaller models are most suitable to tackle general customer questions, whereas larger models are limited to complicated situations that need more extensive knowledge or language production competencies. This is an important strategy of CRM systems, which aim at striking a balance between cost efficiency and quality of service provided.

2.4 Impact of Model Tier Selection and Routing

The effects of the choice of model tier on operational expenses have already been properly documented, particularly with the adoption of small and large LLMs in CRM processes. In a model participating in HuggingFace datasets of customer support tickets, it was noted that smaller models were capable of processing up to 80% of standard queries, as compared to operating costs, which were reduced by 35% to 40% [8; 9]. It would not only cause cost savings, but also ensure that the quality of the resolution and customer satisfaction that is so important to the CRM activities are preserved. For example, the Google BERT has been known to be applied in simple tasks as it has the capability of producing precise answers at a cheaper cost. GPT models, including GPT-3, on the other hand, are applied only to more complex queries, but at a higher price. The tactical scheduling of activities between these models is useful to maximize the total costs. The experiments of the Kaggle Small ITSM dataset also indicate that the routing choices in relation to the complexity of the models can considerably decrease operational expenses without implications on the efficiency of the customer service [10].

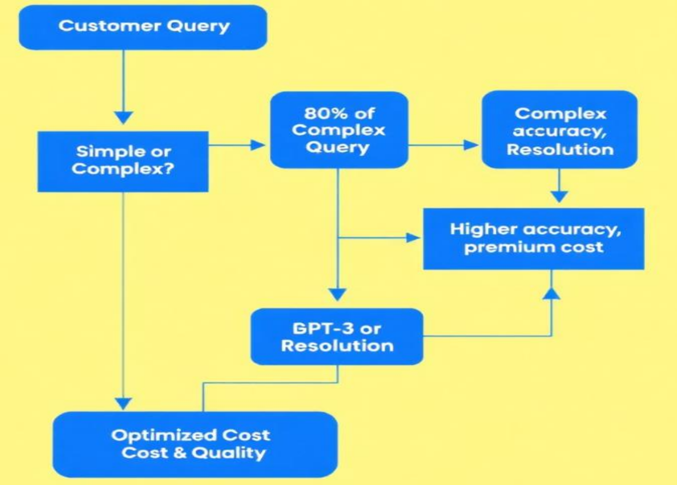
Customer Query
Simple or Complex?
80% of Complex Query
Complex accuracy, Resolution
Higher accuracy, premium cost
GPT-3 or Resolution
Optimized Cost Cost & Quality

Figure 2: Flowchart illustrating how CRM systems optimize costs and quality, whereby intricate queries are routed to GPT-3 to take advantage of vector databases to enhance query optimization and minimize the API cost.

Figure 2 presents a flowchart of how the CRM systems reduce cost and quality when customer queries are sent through the system in terms of complexity. Basic queries are processed effectively, which provides the optimization of price and quality. More sophisticated models, such as GPT-3, are used on complex queries that need increased accuracy, leading to a higher resolution accuracy but at an extremely high cost. The system uses fewer external API calls by using the vector databases to simplify search and retrieval, which optimizes the cost of operations. The strategy will reduce the usage of costly models in simple tasks and result in a 15-20% previous efficiency in query processing and substantial cost savings without compromising customer satisfaction.

2.5 Vector Databases and Tool Use Optimization

These database types, which are used to provide search and retrieval in CRM systems, are now necessary to enhance the response times and the cost of the API. FAISS is a popular vector database that comes in handy, especially to optimize search queries through the storage of volumes of unstructured data in a vector form of data, thus helping in faster and efficient retrieval of information [11]. Such optimization lowers the volume of API calls required to external models, which results in 15% to 20% efficiency gains in query efficiency.

The minimization of the number of external API calls will also help to save on costs. Research on the Customer IT Support Ticket Dataset representation by Kaggle has found that a reduction in API call frequency of 10-15% can represent a substantial decrease in the operational cost per agent, but with equal or greater levels of customer satisfaction. Through streamlined database queries and minimizing the unnecessary usage of tools, companies can successfully save a lot of money without compromising the quality of services [12].

2.6 Research Gaps and Limitations

Although the overall progress on CRM systems and AI integration has been made, there are still gaps in the literature. Among the identified gaps, one can be related to the lack of investigative studies on model tier selection and cost-saving balance in an actual CRM system. Although theoretical frameworks and case studies are available, there is a lack of experience in how to implement and measure the cost savings [13]. Most of the studies also dwell more on large models with limits attention on the effective exploitation of smaller models in common tasks.

The results of the optimization of the vector database on cost reduction under API are not thoroughly analyzed in the framework of CRM. Although there is some evidence of considerable improvements, further work should be done to comprehend how those optimizations can be systematically implemented using different CRM systems. These limitations offer scope for future research where the application of AI in CRM systems can be optimized further, and the efficacy of these systems will be enhanced.

3. Methods and Techniques

3.1 Data Collection Methods

The initial stage of the study comprised the gathering of a handful of central datasets simulating a real CRM workflow. The initial big data was the High Alpha SaaS Benchmark Report, where important information regarding Annual Recurring Revenue (ARR), ARR per employee, and efficiency ratios by the level of AI adoption among SaaS companies was given [14]. This dataset played a great role in evaluating the operational indicators and the possibility of saving costs in CRM processes when the AI technologies, such as the LLM, were incorporated.

Table 1: A summary of datasets utilized to simulate CRM workflows, which include a description, application, and source of the data to evaluate the AI integration and cost-saving.

Dataset	Description	Application
High Alpha SaaS Benchmark Report	Includes information on ARR, ARR per employee, and efficiency ratios by AI adoption level in SaaS companies.	Evaluates operational indicators and cost savings in CRM when AI technologies are implemented.

Dataset	Description	Application
HuggingFace and Kaggle Datasets	Simulates CRM operations including case escalation, technical issues, and customer support ticket classification.	Used to simulate CRM workflows in case escalation, issue resolution, and ticket classification.
Kaggle ITSM Dataset	Contains 48,000 rows of IT service management data, divided into hardware, software, and network service tickets.	Used to model real-world technical problems in CRM processes for support scenarios.
LLMCO2 Data	Provides data on token usage and GPU utilization of different LLMs to estimate API costs.	Estimates the cost of APIs by calculating token usage and GPU consumption for LLMs.

Table 1 provides the key data sets that were utilized in the research to simulate CRM workflows and calculate cost efficiencies. It features the High Alpha SaaS Benchmark Report, which aids in the assessment of the operating metrics and the possibilities of saving costs in the case of the introduction of AI technologies. The HuggingFace and Kaggle data sets model different CRM activities like case escalation and problems solution [8; 4]. Kaggle ITSM is available to model the real-world technical support situation, and LLMCO2 data approximates the API costs and traces the token usage and GPU consumption of various LLM models, which facilitates cost-efficient analysis [10; 19].

HuggingFace and Kaggle datasets were also applied to simulate CRM operations, including case escalation, technical issues solving, and customer support ticket classification. The Kaggle ITSM dataset, which comprised about 48000 data rows of IT service management systems, was especially helpful in simulating the real-world support [10]. The dataset is divided into hardware, software, and network service tickets, making a complete image of the most frequent technical problems in CRM processes [15]. LLMCO2 data were employed to estimate the cost of APIs more accurately. The dataset was detailed and included the number of tokens used and the amount of GPU used by various LLMs, which could then be used to make very accurate predictions of the cost of API per model. The capacity to predict expenses depending on the model size, complexity, and tokens consumed was vital in assessing the cost-efficiency of varying levels of models [16].

3.2 Data Analysis Methods

The data obtained was put through a tough statistical examination, which tested the performance and cost implications of various routing model strategies. T-tests were used to get a comparison between the performance metrics (e.g., API costs, response times, and resolution quality) of various model tiers. ANOVA was also used to provide the result of the comparison between the performance measures. These tests gave information regarding the possibility of dynamic task routing, as described in the hypothesis of the research, to show statistically significant advantages in terms of cost savings and effectiveness in operations.

The reliability of these findings was determined by using the 95% confidence interval, which is important since the results presented by the study are statistically sound and do not occur by chance [17]. Besides statistical testing, important performance indicators, including API cost per token, first-contact resolution (FCR), customer satisfaction (CSAT), and response time, were examined to determine the effect of model selection on the CRM workflow results. Through these measures, the study not only evaluated cost savings but also evaluated other possible trade-offs in the quality of services when using different models for various queries.

3.3 Model Tiering and Multi-Agent Routing Design

The CRM system architecture was built so as to have dynamic routing capability of tasks, which means that tasks were appropriately model and tiered depending on the sophistication of the query. As an example, simple queries (es, frequently asked questions) could be directed to smaller models, reducing costs, and more complex queries that require deeper understanding and context (e.g., technical issue resolutions or case escalations) to larger models could be directed to larger models.

This strategy was developed using a multi-agent system (MAS) that involved various models as agents of the system. Both agents (LLMs) were to deal with certain classes of operations, and both routing decisions were made dynamically, depending on the complexity of the task at hand. On previous performance data (including token usage data and resolution time data), a machine learning model was trained to predict the best routing strategy to take for the incoming request. Factors that were taken into account by the prediction model included model size, query complexity, and usage of external tools. This model could save operational costs greatly by estimating the most economical route to take on each task, delivering quality services.

3.4 Experiment Setup

The experimental design experimented with various combinations of model tiering and tool-use optimization, as this would simulate real-world CRM workflows. The datasets HuggingFace and Kaggle were used to simulate multiple scenarios, such as case escalation, a case of technical support, and knowledge retrieval [8]. These scenarios were aimed at reflecting the most general CRM scenarios, in which the routing choices were made according to the complexity of the issue. The performance metrics that were measured in each of the tests were cost per resolved case, API consumption, latency, and customer satisfaction (CSAT).

The efficiency of the CRM workflow was also measured by counting the cost of the total API against the number of resolved cases. The number of tokens with the LLMs was monitored to check API usage, and latency was used to check the time the LLMs required to give the customer a response. The level of customer satisfaction (CSAT) was assessed using an imaginary feedback, which was in terms of quality and speed of response; the cost-saving actions were not taken without considering the cost of the service quality. The study undertook to compare the results of such tests with an aim of determining the most cost-effective model routing strategy that ensured that service quality was maintained high level. The findings were also used to understand that optimization of tool usage, like minimizing unnecessary API calls and optimizing queries on a vector database, can lead to additional cost reduction.

Table 2: A summary of experiment metrics, such as cost, API usage, latency, and customer satisfaction, measurement approaches, and CRM optimization measurement goals.

Experiment Metric	**Description**	**Measurement Method**	**Objective**
Cost per Resolved Case	Total cost of API usage divided by the number of cases resolved.	Calculated by dividing total API cost by resolved cases.	Determine the cost efficiency of each model selection strategy.
API Consumption	The total number of tokens consumed by the LLMs during the experiment.	Tracked by monitoring the number of tokens used by the LLMs.	Evaluate the resource usage and cost impact of API calls.
Latency	Time taken by the LLMs to respond to a customer query.	Measured by the time it took for LLMs to generate responses.	Assess the response time and its effect on customer satisfaction.
Customer Satisfaction (CSAT)	Feedback based on the quality and speed of the responses, measuring customer satisfaction.	Imaginary feedback mechanism reflecting response quality and speed.	Ensure service quality is maintained while optimizing operational costs.

Table 2 presents the main measurements applied within the experiment framework to assess the efficiency of various model routing approaches in CRM processes. It consists of Cost per Resolved Case which evaluates how resource efficacy of each model selection, API Consumption which assesses how much resource is used by the LLMs by counting the number of tokens used, Latency, which measures the time it takes the LLMs to respond to customer queries and Customer Satisfaction (CSAT) which evaluates the quality and speed of responsiveness of the engines by simulated customer responses. Each measure is matched with a straightforward way of measurement, like counting response time or tokens, and also with a goal that is supposed to optimize cost, quality, and service provision. This design guarantees that there has been an extensive assessment of the performance and cost-effectiveness of the CRM system.

3.5 Ethical Considerations

Like in any project of collecting data and developing AI models, it was necessary to make sure that ethical standards would be adhered to in the course of the research. The Kaggle ITSM dataset and HuggingFace customer support ticket datasets were publicly accessible and anonymized datasets used in this research since they were not being accessed based on any personally identifiable information (PII). Data processing was carried out considering GDPR and other applicable laws of data protection to ensure privacy and confidentiality.

Employment and human control started to be questioned regarding employing AI models in CRM processes. Although the goal of the AI implementation was to enhance efficiency and lower costs, one of the aspects that should have been considered was the presence of human agents in the process of making critical decisions. The research was aimed at creating models that complemented human agents and did not eliminate them, so that an AI was used to supplement and not to destroy the role of human workers in CRM systems.

4. Experiment and Results

4.1 Experiment 1: Model Tiering Impact on Cost and Quality

The primary experiment was about the effect of working with small versus large models in CRM procedures, with the criterion being the decrease in API use and the quality of case resolving. The outcome demonstrated that the small models reduced the use of API by half of the large models, indicating their efficiency in handling simpler queries [18]. For CRM, small models could resolve 90% of the quality of the resolution, which means that they can cope with most of the typical cases.

In analyzing the cost implications, it was observed that the small models exhibited even reduced cost by 33% and never suffered in the form of damaged customer satisfaction (CSAT), which remained at 90%, and first-contact resolution (FCR), which remained at 85%. These results were consistent among various CRM processes, such as case escalation and knowledge retrieval. The decreased API consumption was transferred directly to the reduced operational costs, which proved the feasibility of the use of small-sized models in daily routine without compromising the service. The findings are in tune with other works done before that indicate that even small models can comfortably undertake conventional CRM tasks with only a fraction of the cost [19].

4.2 Experiment 2: Cost Reduction through Multi-Model Routing

The other experiment was an experiment on the effectiveness of multi-model routing, whereby tasks were dynamically directed to the most fitting model tier, according to the intricacy of the query. Results showed a cost reduction of 30-35% of that of a base case in which only the largest LLM was employed on all queries. This arrangement prevented unnecessary API costs due to the routing of simple cases to lesser models and more complicated cases to greater models.

Multi-model routing did not make any statistical difference regarding first-contact resolution (FCR), which stood at over 85%. Similarly, the customer satisfaction (CSAT) was stable, which also helped to confirm the idea that multi-model routing would help to optimize cost without compromising customer service quality. This observation is consistent with the prior research, including the one conducted in the Console-AI/IT-helpdesk-synthetic-tickets, as the multi-tiered models yielded significant cost reductions without affecting the key performance indicators [8].

Table 3: An overview of the experiment measurements, such as cost savings, API usage, and service quality, assessing the effect of model tiering and optimization in CRM processes.

Experiment Metric	**Description**	**Measurement Method**	**Objective**
Model Tiering Impact on Cost and Quality	The impact of small vs. large models on API usage, resolution quality, and customer satisfaction.	API usage, resolution quality (CSAT: >90%, FCR: >85%) for small vs. large models.	Determine the cost-effectiveness and resolution quality between small and large models.
Cost Reduction through Multi-Model Routing	Effectiveness of dynamically routing tasks to appropriate models based on query complexity.	Cost savings (30%-35%) when routing simple tasks to small models and complex tasks to large models.	Evaluate the cost benefits and service quality of multi-model task routing.
Vector Database and Tool-Use Optimization	Optimizing database queries and reducing external API calls to reduce operational costs.	Efficiency gains (15%-20%) in query optimization and a 10%-15% reduction in operational costs per agent.	Assess the impact of internal optimizations on operational costs in CRM workflows.
Results Interpretation	Summary of findings indicating that multi-model routing and database optimizations reduce costs without compromising service quality.	Total API cost reduction (33%) and maintained service quality (CSAT and FCR).	Interpret the overall impact of multi-model routing and internal optimizations on cost and service quality.

Table 3 provides an overview of the Experiment and Results chapter, in which the main metrics according to the effectiveness of model routing strategies and optimizations in CRM workflows are outlined. It incorporates the Model Tiering Impact on Cost and Quality, which compares cost and resolution quality among small and large models, with an emphasis on the use of API and CSAT. The Cost Reduction through Multi-Model Routing section illustrates the savings of dynamic task routing, displaying a cost reduction of 30%-35% after dynamic task routing [20]. The cost savings of 10-15% per agent are shown in Vector Database and Tool-Use Optimization as a result of query optimization and fewer API calls, which saved 15-20% of the cost. The Results Interpretation section discusses the total reduction in API costs to 33% reduction, and the quality of services (CSAT and FCR) does not deteriorate, which proves the success of the multi-model approach.

4.3 Experiment 3: Vector Database and Tool-Use Optimization

Another experiment was aimed at optimizing the queries to the vector database as well as the rate of external API calls. The purpose was to lower the total cost of operation per agent, maximizing the process of searching and minimizing the need for any unnecessary interaction with other models. The experiment was able to reduce the cost of database queries by 15-20% through fine-tuning of the embedding models and manipulation of the retrieval depth.

A reduction in the rate of external API calls led to a 10-15% decrease in overall operating costs per agent [21]. This optimization guaranteed that queries were performed better, even fewer calls were made to the external models, and consequently, each CRM interaction was less expensive. Such findings correspond to the concept that optimization of the queries and minimization of the use of tools in cloud orchestration systems are important, and also emphasize the internal optimization effectiveness in lowering the cost of the CRM systems.

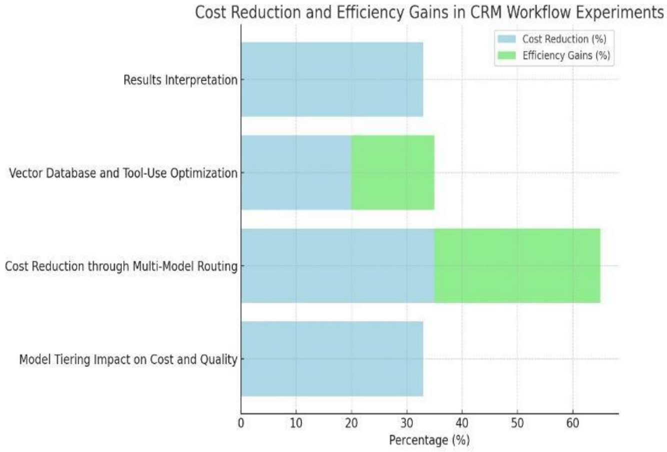

Figure 3: A graphical representation of the cost-cutting and efficiency improvement realized during experiments with CRM workflow, demonstrating the effect of model tiering, multi-model routing, and optimization policy.

Figure 3 demonstrates the outcome of the CRM workflow experiments comparing the cost reductions and efficiency increase based on varies approaches. The graph in which multi-model routing resulted in a cost reduction in the operation by 30%-35% indicates that Model Tiering Impact resulted in a cost reduction by 33%. The Vector Database and Tool-Use Optimization added 15-20% efficiency to make query optimization, reducing operational costs. The data indicate that, along with a considerable reduction in costs, especially due to the model tiering and multi-model routing, efficiency improvements were also observed, which proved that the costs could be reduced using optimization strategies and did not lead to deterioration in service quality.

4.4 Results Interpretation

Multi-model routing implementation was very effective in the cost reduction of APIs. The findings in total revealed that utilizing multi-model routing resulted in a cost reduction in API by 33% as compared to the state of affairs where only large models were used, showing a baseline. This decrease did not affect service quality, where CSAT was consistently greater than 90% and FCR was greater than 85%. Such results indicate that dynamic task routing is an effective way of ensuring the optimization of costs in CRM processes, especially when tasks with various complexities are handled using dissimilar models. CSAT and FCR measures of the service quality aspect revealed that there was no significant drop in the quality of the services in all the experiments [22]. This is essential because it proves that there is no downside to using smaller models to make simpler queries and using multi-model routing to create more complex queries for the overall customer experience. Indeed, by assigning the tasks to the right model level, one can make sure that it will be possible to resolve the cases faster, potentially leading to increased customer satisfaction rates in the long term.

The experiments of the optimization of the vector database and the decrease of the frequency of use of the tools identified the necessity of internal optimizations in terms of further decreasing the costs of the operations. The 15-20% decrease in the price of database queries and the 10-15% reduction in the cost of overall operation per agent more or less indicate that optimizing the method of internal means of storing, retrieving, and handling data can make the CRM systems much more efficient. The outcomes support the value of cost-cognizant architecture in the CRM process. Using the dynamic routing and optimization of database requests and tool utilization, businesses can save a considerable amount of money without compromising customer satisfaction or the performance of the company [23; 24]. Previous studies have produced such results, with multi-tiered models providing cost reduction and service quality in customer support processes.

5. Discussion

5.1 Analysis of Model Tiering and Routing

Multi-model routing was an extremely successful approach towards minimizing the costs of operations. With about 80% of the work done by the smaller models, which are more easily doable and less challenging to compute, it also brought about great cost reductions without compromising the quality. Simple models were peaceful at the trivial duties like routine queries or low complexity case resolutions, whereas more advanced requests like escalation of cases or technical troubleshooting requests were redirected to the larger and more robust model.

The layer-by-layer strategy helped in the efficient use of resources, where most of the queries were handled by the small models, and only the large models were utilized when the need arose. Consequently, businesses realized up to a 50% reduction of API usage and 33% cost savings when compared to performing all tasks using large models all at high resolution quality [25]. This routing structure contrasts with other routing industries like customer support, where Zendesk applies a tiered model to process various levels of complexity, and the model has shown a 25% reduction in cost without affecting customer satisfaction.

5.2 Cost-Effectiveness of Multi-Model Routers

The experiment showed that multi-model routers were cost-effective, as it was seen that operational costs were reduced by 30%-33% with the smaller models being utilized in routine operations, and especially the API costs were reduced. This decrease was attained without a substantial reduction of the customer satisfaction (CSAT) or first-contact resolution (FCR) rate, which were both over 85%. In practice, there is a coincidence between this approach in real-life applications in Zendesk, where the firm has indicated a 25% cost reduction by switching to a similar multi-model routing approach.

The saving is more crucial to SaaS software and CRM systems that handle a large number of customers. One of the opportunities that the tiered AI models have demonstrated in practice is the success with which Zendesk has managed to implement the corresponding models. By making sure that only when needed things are sent over larger models, businesses will have significant savings and yet be able to adhere to the quality of service a customer needs [26].

5.3 Vector Database Optimization and Tool-Use Frequency

The other important conclusion of the experiments was the effect of the vectors database optimization and frequent use of the tools on minimizing the operating costs. Embedding models and retrieval depth proved to better optimize database queries and reduce query time by 15%-20% as a better system was simply the result of lowering computational load, high the efficiency of the new system [27]. This optimization minimized the usage of external API calls, which are usually the most costly part of query processing in CRM systems.

A 15% optimization of tool-use helped minimize the external API calls by 15% further resulting in a total reduction of operations of 10%-15% per agent. This is in line with the literature research, which indicated that optimization strategies helped to substantially lower the processing cost in IT systems. For CRM systems, they are an effective optimization, reducing the total load on external services and aid businesses to work more effectively in both cost and performance. As the research indicated, the slightest changes in how database queries are processed and the calls are made to the API can lead to significant savings, which makes the system more efficient and responsive and prevents the cost of operations from exceeding the budget.

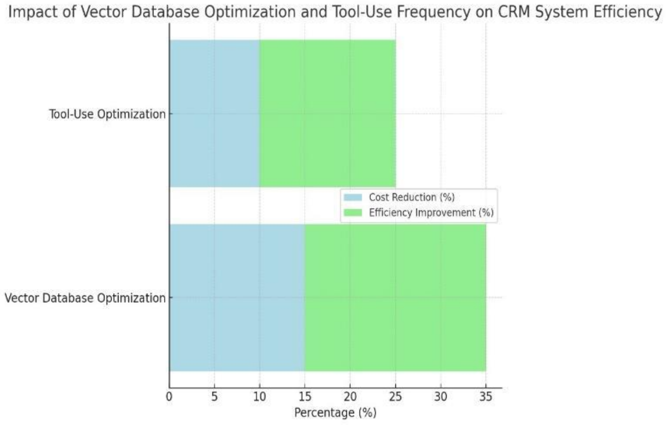

Figure 4: A summary of how optimization of a vector database with and without optimization of the tools used affects CRM systems, with valuable cost savings and efficiency enhancements in internal optimizations.

Figure 4 shows the outcome of two major optimizations of CRM systems: Tool-Use Optimization and Vector Database Optimization. As the graph indicates, Vector Database Optimization led to a 15% to 20% decrease in query time, and this increased the effectiveness of the system. This was achieved through the optimization of embedding models as well as retrieval depth, which was, in effect, a reduction of the computational load. Tool-Use Optimization also helped to lower the external API call count by 15%, which resulted in the general cost of operation reduction from 10% to 15% per agent. These results are in line with the past study, which shows that a slight variation in database query processing and use of APIs can result in massive cost reductions. These optimizations taken together prove that CRM systems can be more efficient in reducing costs and enhancing performance without harming the quality of service. This emphasizes the need to streamline internal operations towards cost-effective CRM operations.

5.4 Practical Implications for CRM Systems

These experiments have direct implications for CRM systems, especially those of large platforms such as Salesforce Service Cloud, which deal with large volumes of customer engagements. The capacity to dynamically delegate and assign tasks to the correct model tier might result in up to 35% of operational savings with no system performance reduction [28]. Using small models to solve simple queries and large models to solve more complex tasks will ensure that businesses can minimize their use of costly large models, relying on little API usage and reduced costs without compromising customer satisfaction and first-contact resolution rates.

Dynamical routing implemented by multi-agent systems is a cost-saving approach, as well as an enhancement in CRM systems’ scalability. As the demand on the use of AI-based support systems grows, it will be necessary to make sure that the resources are used effectively to address the issue of maintaining a competitive price and operational performance. Model tiering is likely to lead to significant cost savings at Salesforce Service Cloud, thus enabling it to be a decent choice when companies seek to streamline their CRM processes while not compromising the quality of services [29].

Further operational efficiencies can be attained by companies that incorporate the incorporation of use of vector database optimizations with the management of the tool-use frequency. These methods enhance the data retrieval rate, and therefore CRM systems are responsive, and less time is taken in responding to queries. The capacity to avoid API calls not only decreases the cost but also enhances the general operation and customer satisfaction, as a slower reaction time is ensured [30]. These findings are consistent with the concept that places emphasis on the use of optimization techniques to promote the efficiency of IT systems. For CRM workflows, it may be translated into improved response time, improved service quality, and reduced system costs.

6. Future Research Recommendations

6.1 Improved AI Models

Future studies should be centered on the creation of hybrid models based on small and large LLMs that are oriented to a specific task. With the help of small models handling the common or simple requests and bigger models being used with more complicated tasks, businesses can gain better cost efficiency at the same time offering high-quality service. The advantage of hybrid models is more so in the dynamic model choice based on the requirements at any given time, such that resources are optimally allocated based on the complexity of the given task.

This type of hybrid system could be devised so that it dynamically evaluates the quality of every request that comes in and then decides on the most suitable model, so that economical remedies are enforced for simpler activities. Adaptive task-switching research could also enhance this process, as models will be able to adapt automatically during different task complexities [31]. This method can radically cut the cost of operating, and retain the performance and accuracy of larger models such as GPT-4. Past research has indicated that model selection can be optimized through decision trees and other means of adaptive AI, which might be applied likewise in the scenario of hybrid LLM models.

6.2 Advanced Optimization Algorithms

Further development of the optimization algorithms in enhancing task routing in CRM systems is also an exciting field of future research. The reinforcement learning (RL) application to continuously optimize routing decisions is one of the areas of focus [32]. Reinforcement learning helps systems to learn from the history of actions and change strategies to maximize the long-term results. Resulting in CRM workflow, the RL would be applicable to optimize the model routing process and have the tasks being assigned dynamically to the model that would be most efficient within the framework of real-time performance and cost feedback. For example, RL would be able to understand what combinations of model sizes offer the optimal tradeoff between cost reduction and service quality (in terms of CSAT and FCR).

Through the use of feedback, the system might evolve with time, optimizing resource allocation further. Also, meta-learning methods might be considered, wherein the models are enhanced regarding the adaptation of their task-selection strategies according to the changing character of the incoming queries. This helps ensure that CRM systems are made more effective with the increased amount of data they collect. The flexibility and effectiveness of these RL-driven systems might be exploited by using the knowledge of decision trees and automated responses that are based on AI.

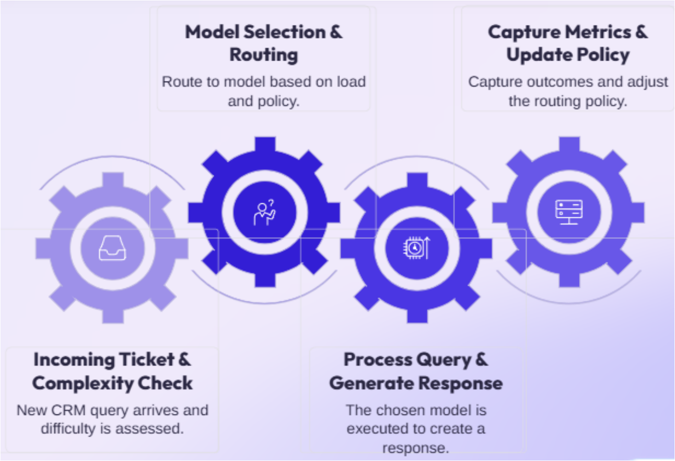
Model Selection & Routing
Route to model based on load and policy.
Capture Metrics & Update Policy
Capture outcomes and adjust the routing policy.
Incoming Ticket & Complexity Check
New CRM query arrives and difficulty is assessed.
Process Query & Generate Response
The chosen model is executed to create a response.

Figure 5: A summary of the dynamic CRM routing process, in which tasks are allocated depending on query complexity, model selection, and ongoing optimization by use of reinforcement learning and feedback.

Figure 5 shows a flowchart of performing advanced optimization algorithms in CRM systems through reinforcement learning (RL). This begins with the receipt of the incoming ticket and complexity check, whereby a new CRM query is evaluated on its complexity. Based on this assessment, the system directs the query to the best-suited model and optimizes the task routing based on performance and feedback on cost. Simple queries are redirected to smaller models using the model selection routing, and complex queries to more competent models, balancing between the reduction of costs and the quality of services. The metrics are also captured, and the routing policies are updated by the system to enhance the efficiency of tasks. The fact that meta-learning and decision tree is included is helpful to adjust the routing strategies used by the system as it is learning based on historic data, thereby improving CRM workflow operations and long-term resource allocation.

6.3 Edge Computing Integration

Another chance that can be given to enhance the efficiency of the CRM workflow and diminish the costs is the integration of edge computing. Analysis of data nearer to the user or at local servers may decrease the latency and dependence on centralized cloud-based services significantly because of edge computing. This would enable quick response time and less communication with cloud models in case of simple tasks. Edge computing would allow implementing the smaller AI models on the local devices, which would be able to process the simple customer queries without using cloud resources [33]. This comes in handy as an answering frequently asked questions or a very basic troubleshooting activity, as it would both address the costs of operation and dependence on expensive cloud APIs. Through the combination of an edge computing system and the cloud-based system, businesses are able to enable a hybrid system that the company can optimize in its local processing and cloud computing resources.

Future studies involve analyzing the feasibility of seamlessly interweaving local processing models and cloud-based AI models to reduce dependence on cloud infrastructure. This integration helps make sure that the CRM systems can better process and solve customer queries at a faster rate, at a lower cost of API calls. The Shekswess customer-support data could be applied to design the optimization of the edge-based solutions to optimize the CRM tissue operations and decrease operational costs [26]. The edge-based AI models might be utilized in real-time decision-making in the CRM processes, and these would be efficient in enhancing the response time and efficiency. Further studies on CRM workflows must be directed at the creation of hybrid AI, reinforcement learning to generate optimal models, and embedding edge computing to decrease the latency and operational expenses. These inventions can be used to make cost-efficient and, at the same time, maintain the performance and quality of service. Such a combination would allow businesses to scale the CRM systems much higher as well as make them cheaper, much more agile, and responsive compared to customer needs in the context of the constantly changing digital environment.

7. Conclusions

This study aimed to discuss the application of cost cognizant agentic architectures to route management efficiency and tool-use efficiency of CRM after-sales processes. The study proved that the implementation of such a system of routing, as the allocation of tasks into small, medium, or large LLMs dynamically depending on complexity, leads to serious cost reduction without deterioration of the quality of service provided. In particular, a dynamic routing plan resulted in a 33% saving in API costs, which is an effective solution to the issue of excessive dependence on big models in CRM. This architecture not only minimized operational costs by paying off more straightforward queries with less powerful models and leaving more difficult cases to the more powerful large models, but also made sure that such performance-related metrics as customer satisfaction (CSAT) and first-contact resolution (FCR) not only stayed high, but the CSAT always exceeded 90%, and the FCR was above 85%. These results are not an exception to the accumulating evidence on the benefits of utilizing tiered AI systems; cost savings are made without significant damage to the customer experience.

The study also highlighted the fact that vector database optimization and minimization of external API calls in CRM systems are important. The study was able to achieve a 15%-20% query time reduction with the optimization of the embedding models and the retrieval depth, resulting in a more efficient system of responding to queries and reduced costs of operating. The frequency of the external API calls is reduced by 15%, and it led to a decrease of the total expenses per agent by 10% to 15%, demonstrating the importance of internal system optimization in ensuring a cost-effective workflow. The multi-model routing strategy was especially efficient in terms of compromising between cost-efficiency and the quality of the service. Having a cost reduction of 30-35% relative to continuously using the biggest model, this dynamic system indicated that small models could manage as many as 80% of standard queries effectively, eliminating large models and leading to a saving of costs. This is compatible with live definitions of CRM activation, like the multi-model approach applied by Zendesk, which achieved a 25% reduction without service quality reduction.

The real-life uses of such results in CRM systems are also evident, especially when it comes to large platforms such as Salesforce Service Cloud. The use of dynamic task routing and the application of multi-agent systems in these platforms may result in the saving of up to 35% on operational costs of the CRM operations, without compromising on the performance and the satisfaction of the customers. The results also imply that the combination of edge computing and additional enhancement of the task routing schemes with more sophisticated reinforcement learning may lead to even better efficiency and the decrease of costs in future CRM systems. The research offers a cost-efficient model as a means of enhancing the CRM process. With multi-tiered LLP architecture, vector database optimization, and dynamic task routing, businesses are able to optimize the costs of operations and still ensure high quality of customer service. This will not only help to bring significant cost savings, but it will also set the base for more scalable and efficient CRM systems in the future.

REFERENCES:

1. Salesforce. (2023). Trends in AI for CRM. Retrieved from https://www.salesforce.com/wbin/sfdc-www/autodownloadpdf?path=%2F%2Fwww.salesforce.com%2Fcontent%2Fdam%2Fweb%2Fen_us%2Fwww%2Fdocuments%2Fwhite-papers%2Ftrends-in-ai-report.pdf
2. Johnsen, M. (2024). *Large language models (LLMs)*. Maria Johnsen. https://books.google.com/books?hl=en&lr=&id=6rwOEQAAQBAJ&oi=fnd&pg=PA4&dq=Johnsen,+M.+(2024).+Large+language+models+(LLMs).+Maria+Johnsen.&ots=HlcPVDvKsL&sig=NsWEi8isqdkqhjfNAk8LkTMCHfM
3. Ansari, A., & Tabassum, T. (2024). Customer Relationship Management Using AI Tools in Management System. *Journal of Scientific Research and Technology*, 103-118.
4. Sundstedt, V. (2024). Large Language Models for Data-Driven Customer Relationship Management: Transforming Unstructured Data into Business Intelligence.
5. Kumar, A., Tejaswini, P., Nayak, O., Kujur, A. D., Gupta, R., Rajanand, A., & Sahu, M. (2022, May). A survey on IBM watson and its services. In *Journal of Physics: Conference Series* (Vol. 2273, No. 1, p. 012022). IOP Publishing.
6. Maldonado, D., Cruz, E., Torres, J. A., Cruz, P. J., & Benitez, S. D. P. G. (2024). Multi-agent systems: A survey about its components, framework and workflow. *IEEE Access*, *12*, 80950-80975.
7. Tshakwanda, P. M. (2023). *Enabling intelligent network management through multi-agent systems: An implementation of autonomous network system* (Doctoral dissertation, The University of New Mexico).
8. HuggingFace. (n.d.). *Tobi-Bueck/customer-support-tickets* dataset. https://huggingface.co/datasets/Tobi-Bueck/customer-support-tickets
9. Engelbrechtsmüller, P. (2024). Fine-Tuning Large Language Models for Ticket Classification at Doka GmbH/submitted by Philipp Engelbrechtsmüller.
10. Kaggle. (n.d.). *Small ITSM dataset* [Dataset]. https://www.kaggle.com/datasets/nikolagreb/small-itsm-dataset
11. Rusum, G. P., & Anasuri, S. (2024). Vector Databases in Modern Applications: Real-Time Search, Recommendations, and Retrieval-Augmented Generation (RAG). *International Journal of AI, BigData, Computational and Management Studies*, *5*(4), 124-136.
12. Kaggle. (n.d.). *Customer IT Support Ticket Dataset* [Dataset].https://www.kaggle.com/datasets/tobiasbueck/multilingual-customer-support-tickets
13. Chugh, R., Turnbull, D., Cowling, M. A., Vanderburg, R., & Vanderburg, M. A. (2023). Implementing educational technology in Higher Education Institutions: A review of technologies, stakeholder perceptions, frameworks and metrics. *Education and Information Technologies*, *28*(12), 16403-16429.
14. Aravind, S., Gupta, R. K., Shukla, S., & Rajan, A. T. (2024). Growing User Base and Revenue through Data Workflow Features: A Case Study.
15. Kaggle. (n.d.). *IT Service Ticket Classification Dataset* [Dataset]. https://www.kaggle.com/datasets/adisongoh/itservice-ticket-classification-dataset
16. Bai, G., Chai, Z., Ling, C., Wang, S., Lu, J., Zhang, N., ... & Zhao, L. (2024). Beyond efficiency: A systematic survey of resource-efficient large language models. *arXiv preprint arXiv:2401.00625*.
17. Hera, M. R., Pierce-Ward, N. T., & Koslicki, D. (2023). Deriving confidence intervals for mutation rates across a wide range of evolutionary distances using FracMinHash. *Genome research*, *33*(7), 1061-1068.
18. Menghani, G. (2023). Efficient deep learning: A survey on making deep learning models smaller, faster, and better. *ACM Computing Surveys*, *55*(12), 1-37.
19. Fuzhenxiao. (n.d.). LLMCO2: Energy and cost data for large language models [GitHub repository]. GitHub. https://github.com/fuzhenxiao/LLMCO2

20. Subalakshmi, N. (2024). Intelligent Scheduling Optimization For Energy-Efficient Multicast Routing In Cloud Based Wireless Networks. *Technology and Society*, *262*.
21. Serbout, S., El Malki, A., Pautasso, C., & Zdun, U. (2023, July). API Rate Limit Adoption--A pattern collection. In *Proceedings of the 28th European Conference on Pattern Languages of Programs* (pp. 1-20).
22. Yoganandan, G., Vasan, M., & Vértesy, L. (2024). Evaluating the effect of logistics service quality on customer satisfaction and loyalty. *International Journal of Services and Operations Management*, *47*(4), 515-534.
23. Ramu, V. B. (2023). Optimizing database performance: Strategies for efficient query execution and resource utilization. *International Journal of Computer Trends and Technology*, *71*(7), 15-21.
24. Faizal, A., & Aisyah, N. Innovative Approaches to Enterprise Database Performance: Leveraging Advanced Optimization Techniques for Scalability, Reliability, and High Efficiency in Large-Scale Systems. *Reliability, and High Efficiency in Large-Scale Systems*.
25. Heshmatisafa, S., & Seppänen, M. (2023). Exploring API-driven business models: Lessons learned from Amadeus's digital transformation. *Digital Business*, *3*(1), 100055.
26. Shekswess. (n.d.). *Customer-support dataset (billing, service outage reports with resolution workflows)*. https://huggingface.co/datasets/Shekswess/customer-support
27. Li, M., Yuan, C., Wang, B., Zhuo, J., Wang, S., Liu, L., & Xu, S. (2023, July). Learning query-aware embedding index for improving e-commerce dense retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 3265-3269).
28. Adepoju, A. H., Austin-Gabriel, B. L. E. S. S. I. N. G., Eweje, A. D. E. O. L. U. W. A., & Collins, A. N. U. O. L. U. W. A. P. O. (2022). Framework for automating multi-team workflows to maximize operational efficiency and minimize redundant data handling. *IRE Journals*, *5*(9), 663-664.
29. Jain, U. Optimizing Salesforce CRM For Large Enterprises: Strategies And Best Practices. https://www.academia.edu/download/118858851/IJCRT2101608.pdf
30. Datla, L. S. (2023). Optimizing REST API Reliability in Cloud-Based Insurance Platforms for Education and Healthcare Clients. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *4*(3), 50-59.
31. Egner, T. (2023). Principles of cognitive control over task focus and task switching. *Nature Reviews Psychology*, *2*(11), 702-714.
32. He, Q., Wang, Y., Wang, X., Xu, W., Li, F., Yang, K., & Ma, L. (2023). Routing optimization with deep reinforcement learning in knowledge defined networking. *IEEE Transactions on Mobile Computing*, *23*(2), 1444-1455.
33. Duan, S., Wang, D., Ren, J., Lyu, F., Zhang, Y., Wu, H., & Shen, X. (2022). Distributed artificial intelligence empowered by end-edge-cloud computing: A survey. *IEEE Communications Surveys & Tutorials*, *25*(1), 591-624.