Comparative Analysis of Prompt Engineering Strategies for Customer Churn Prediction in CRM Systems

Author: Uday Ramesh Chougule

Affiliation: Prospective Graduate Student (Artificial Intelligence)

Research Paper (Independent Study)

Date: January 2026

ABSTRACT

Customer churn is a major financial hurdle for telecommunications, driving a shift from "black-box" models toward interpretable AI. This research evaluates Large Language Models (LLMs) in predicting churn by addressing the "Serialization Gap"- the conversion of structured tabular data into natural language. Using the IBM Telco Churn dataset, we compared four prompt engineering architectures: Zero-Shot, Few-Shot, Chain-of-Thought (CoT), and Structured JSON Output.

Our findings indicate that prompt architecture directly dictates reliability. While Zero-Shot prompting provides a quick baseline, it often misses nuanced risks. Conversely, the Chain-of-Thought strategy offers superior diagnostic depth by deconstructing the links between tenure and service quality, while Structured Output ensures seamless system interoperability. We conclude that strategic prompting is essential for delivering transparent, actionable, and scalable CRM analytics.

INTRODUCTION

The Importance of Customer Retention

Telecommunications companies operate in a highly saturated market where retaining customers yields greater value than acquiring new ones. Reports show that winning a new customer can cost five to seven times more than keeping an existing one, meaning even small improvements in churn reduction translate into substantial revenue impact. Churn has therefore become a strategic indicator of market competitiveness and customer experience.

The Limitations of Traditional Models

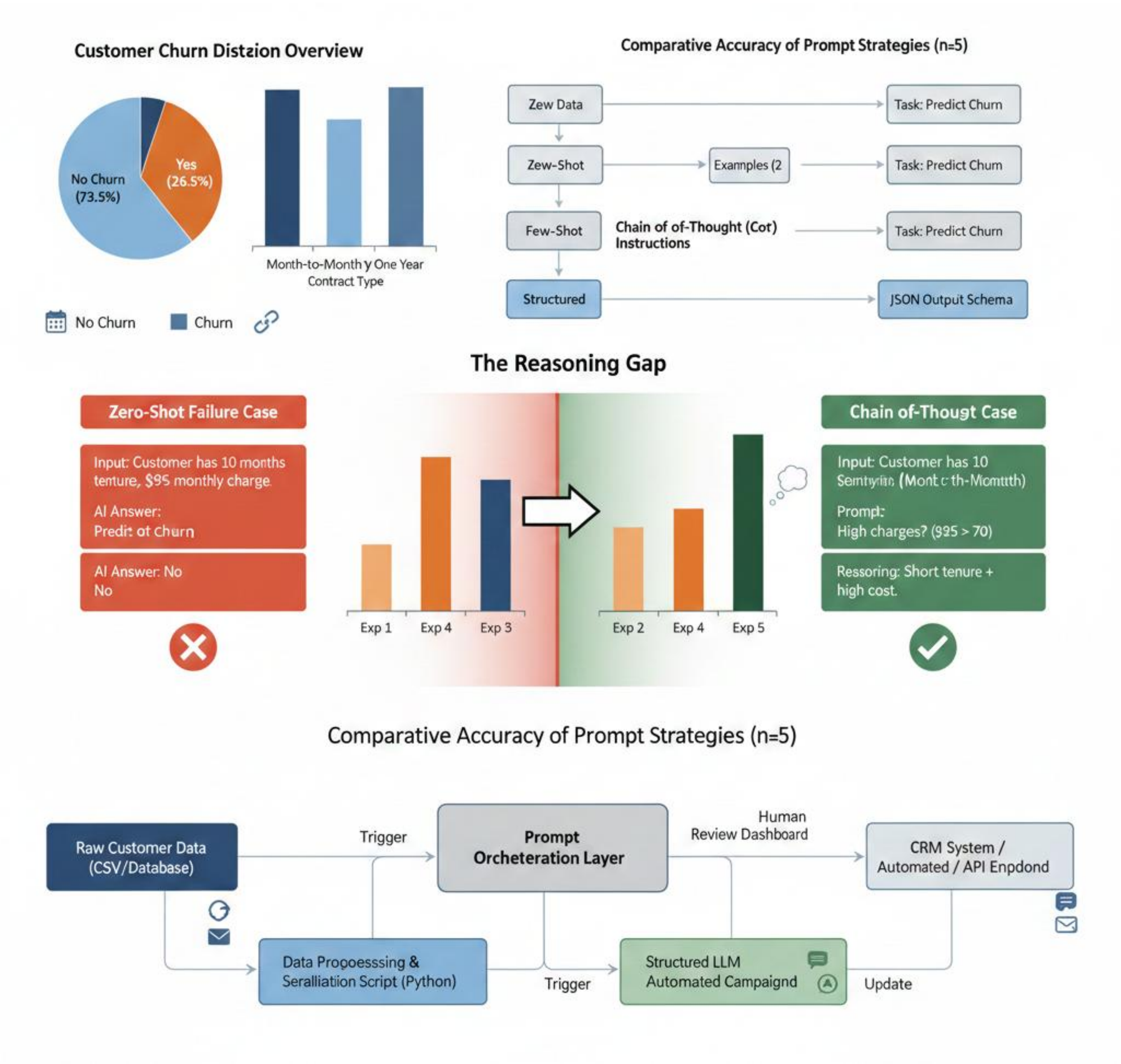
Companies have historically relied on machine learning models such as Logistic Regression, Random Forests, and Gradient Boosted Trees. While effective at prediction, these models often function as “black boxes,” offering limited insight into why a customer is likely to leave. This lack of interpretability forces CRM teams to act blindly, without understanding the drivers influencing churn-such as short tenure, billing friction, or service dissatisfaction.

The Role of Large Language Models

Large Language Models (LLMs) introduce new capability: natural language reasoning. Instead of treating a customer as a row of numbers, LLMs can evaluate structured profiles as narrative information. This allows them to consider patterns like contract flexibility or service usage without explicit programming. However, their success depends heavily on how data is presented and how the prompt is formulated-making prompt engineering a critical variable.

Addressing the Prompting Gap

Despite the rise of LLMs, limited research explores how different prompting strategies influence churn prediction outcomes. Most existing work focuses on numerical machine learning or generic LLM tasks. This study fills that gap by comparing four prompting styles- Zero-Shot, Few-Shot, Chain-of-Thought, and Structured Output-to evaluate how instruction design affects accuracy, reasoning, and deployment readiness.



Customer Churn Distzion Overview
No Churn (73.5%)
Yes (26.5%)
Month-to-Month y One Year
Contract Type
No Churn
Churn
Comparative Accuracy of Prompt Strategies (n=5)
Zew Data
Task: Predict Churn
Zew-Shot
Examples (2
Task: Predict Churn
Few-Shot
Chain of of-Thought (Cot) Instructions
Task: Predict Churn
Structured
JSON Output Schema
The Reasoning Gap
Zero-Shot Failure Case
Input: Customer has 10 months tenture, $95 monthly charge.
AI Answer:
Predit ot Churn
AI Answer: No
No
Exp 1
Exp 4
Exp 3
Exp 2
Exp 4
Exp 5
Chain of-Thougt Case
Input: Customer has 10 Semtyrin (Mont c th-Momtth)
Prompt:
High charges? ($95 > 70)
Ressoring: Short tenure + high cost.
Comparative Accuracy of Prompt Strategies (n=5)
Raw Customer Data (CSV/Database)
Trigger
Prompt Orcheteration Layer
Human Review Dashboard
CRM System / Automated / API Endpond
Data Progoesssing & Seralliation Script (Python)
Trigger
Structured LLM Automated Campaignd
Update

Figure 01: Prompt Engineering Architectures

LITERATURE REVIEW

Classical Churn Modeling

Traditional approaches to churn prediction rely on machine learning methods such as Logistic Regression, KNN, Random Forest, and XGBoost. These models deliver strong accuracy but operate with limited transparency, offering little explanation behind predictions (Ahmed & Maheswari, 2017). This "black-box" nature makes it difficult for analysts to understand the drivers behind customer behaviour.

LLMs and Tabular Data

Recent studies show LLMs can interpret structured data when converted to natural language. Borisov et al. (2022) demonstrated that models trained on diverse text corpora can infer meaning from tabular fields if properly serialized, enabling reasoning grounded in language rather than raw numbers.

Prompt Engineering Approaches

Emerging work highlights how prompt design affects LLM outputs. Zero-Shot prompting tests general pretrained knowledge, Few-Shot prompting uses examples to shape reasoning (Brown et al., 2020), and Chain-of-Thought prompting encourages step-wise reasoning (Wei et al., 2022). Structured prompting has also gained importance for software integration, where fixed JSON formats ensure system compatibility.

Research Gap

Existing literature typically explores these strategies in isolation and rarely applies them to churn analytics. This study addresses the gap by directly comparing these four methods using a consistent dataset and evaluation approach.

DATASET DESCRIPTION

Data Source and Scope

The dataset used in this research is the IBM Telco Customer Churn dataset, sourced from the Kaggle repository. It serves as an industry-standard benchmark for churn analysis and predictive modeling. The dataset provides a comprehensive snapshot of a telecommunications provider’s customer base in California, capturing a diverse array of subscriber behaviors and account characteristics.

Record Volume and Target Variable

The dataset contains 7,043 unique customer records, each represented across 21 distinct attributes. The Target Feature is a binary label designated as Churn (Yes/No), which indicates whether a customer discontinued their service within the last month. Approximately 26.5% of the records are labeled as "Yes," representing a standard class imbalance typical of real-world attrition data.

Variable Categorization

- The predictors in the dataset are organized into three primary logical categories, which were serialized for LLM processing:

- Demographic Variables: Customer personal attributes including gender, senior citizen status (binary), and the presence of partners or dependents.

- Service Variables: Specific subscriptions such as phone service, multiple lines, and internet service types (DSL, Fiber optic). It also includes value-added services like Online Security, Tech Support, and Streaming media.

- Financial & Account Information: Critical billing details including Tenure (months with the company), Contract Type (Month-to-month, One year, Two year), Payment Method, and Monthly/Total Charges.

METHODOLOGY AND PREPROCESSING

The transformation of structured tabular data into a format suitable for Large Language Model (LLM) analysis requires a deliberate preprocessing pipeline. Unlike traditional machine learning, where data is converted into purely numerical matrices, LLM-based prediction necessitates a "Serialization" process that preserves the semantic richness of the customer profile.

Data Cleaning and Preparation: The IBM Telco dataset underwent basic preprocessing to ensure clarity and consistency for LLM input. Missing values in the TotalCharges field-originating from new customers-were replaced with 0.0, and the column was converted to numeric format to avoid processing errors.

Preserving Semantic Meaning: Unlike traditional ML pipelines, categorical fields were retained in text form rather than encoded. Terms such as “Month-to-month” or “Fiber optic” convey business implications that LLMs can interpret from prior training knowledge.

Prompt Serialization: Each customer profile was converted into a short narrative “fact sheet,” transforming tabular entries into sentences describing tenure, services, billing, and contract type. This serialized text formed the input for each prompting strategy tested.

Serialization Example: > Raw Data: *{Tenure: 5, Contract: 'Month-to-month', MonthlyCharges: 98.50}* Serialized Text: "The customer has a tenure of 5 months, is on a Month-to-month contract, and incurs $98.50 in monthly charges."

By converting abstract numbers into a narrative "case study," we bridge the Serialization Gap. This ensures that when the data is fed into the various prompting strategies, the model perceives it as a coherent story of a human subscriber rather than a disconnected set of variables. This structured narrative serves as the foundation for the four experimental prompting designs explored in the following section.

PROMPT DESIGN ARCHITECTURES

The core of this research lies in the experimental design of four distinct prompting architectures. Each strategy represents a different level of cognitive guidance provided to the Large Language Model, ranging from a minimal, direct inquiry to a highly structured, multi-step reasoning framework.

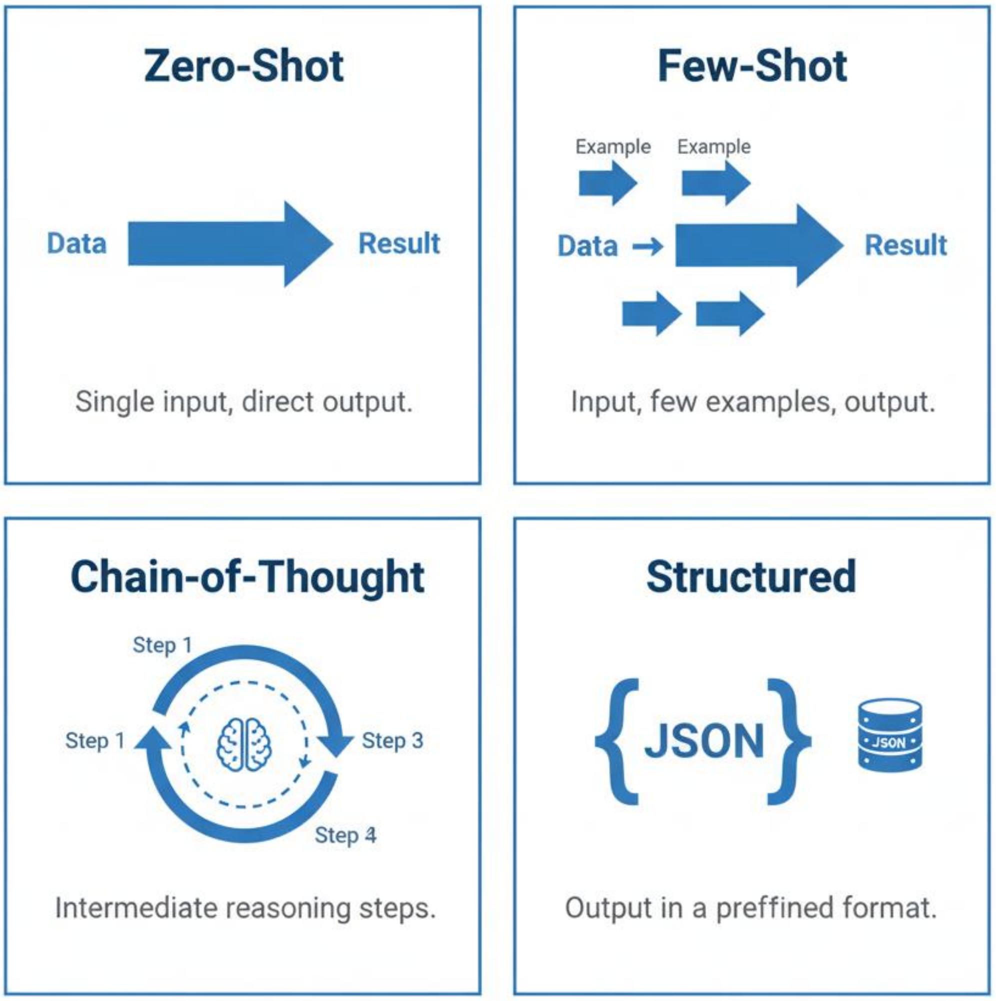

Zero-Shot
Data
Result
Single input, direct output.
Few-Shot
Example
Example
Data
Result
Input, few examples, output.
Chain-of-Thought
Step 1
Step 1
Step 3
Step 4
Intermediate reasoning steps.
Structured
{JSON}
JSON
Output in a preffined format.

Figure 02: End-to-End Experimental Framework for LLM-Based Churn Prediction.

Zero-Shot Prompting (Baseline)

The Zero-Shot strategy serves as the control group for our experiment. In this architecture, the serialized customer data is provided to the model alongside a direct instruction to predict churn, without any prior examples or context.

- Mechanism: It relies entirely on the LLM's internal pre-trained knowledge and "raw intuition" about business patterns.

- Purpose: To evaluate the baseline capability of the model to identify risk based solely on the provided variables.

Few-Shot Prompting (Contextual Learning)

The Few-Shot architecture introduces a "In-Context Learning" layer. Before the target customer data is presented, the model is provided with two labeled examples-one representing a "Loyal" customer and one representing a "Churned" customer.

- Mechanism: By seeing successful and unsuccessful profiles, the model develops a temporary "mental schema" for what constitutes a risk factor within the specific context of the Telco dataset.

- Purpose: To determine if providing minimal comparative context reduces prediction errors and helps the model distinguish between standard and high-risk profiles.

Chain-of-Thought (CoT) Prompting (Logical Decomposition)

The Chain-of-Thought strategy is designed to bypass "shallow" decision-making by forcing the model to generate intermediate reasoning steps. Instead of requesting a binary "Yes/No," the prompt instructs the model to decompose the problem into a logical sequence:

1. Analyze Contractual Risk: Evaluate the impact of contract type (e.g., Month-to-month).
2. Assessment of Tenure and Loyalty: Weigh the duration of the relationship against recent charges.

3. Synthesis: Combine these observations into a final conclusion.

- Mechanism: This anchors the model’s "attention" on critical variables, preventing it from jumping to a conclusion based on a single outlier.

- Purpose: To enhance interpretability and accuracy through visible, audit-ready reasoning.

Structured Output Prompting (System-Centric)

The Structured architecture focuses on the technical usability of the AI’s decision. This prompt enforces a rigid JSON (JavaScript Object Notation) schema, requiring the model to return three specific keys: *prediction*, *reason*, and *confidence.*

- Mechanism: It constrains the model’s linguistic creativity in favor of a standardized format. The model must categorize its reasoning into a single sentence and quantify its certainty as a percentage.

- Purpose: To test how well the LLM follows complex formatting constraints while still performing high-quality predictive analysis.

EXPERIMENTAL SETUP

To evaluate the efficacy of the proposed prompting strategies, we conducted a series of controlled experiments using a representative subset of the preprocessed dataset. This section outlines the technical environment and the metrics used to validate the model's performance.

Sampling and Case Selection

Due to the qualitative nature of analyzing Chain-of-Thought (CoT) and Structured reasoning, we utilized a Case Study Approach. We randomly sampled five unique customers from the cleaned IBM Telco dataset to serve as our primary experimental subjects.

- Validation Baseline: The "Actual Churn" status of these five customers (as recorded in the original dataset) was withheld from the model during the prompt execution phase and used as the "Ground Truth" for final accuracy validation.

- Reproducibility: A fixed random seed was applied during sampling to ensure that the results can be audited and replicated in future studies.

Model Environment

The research and drafting process were supported by **Large Language Model (LLM)** agents, specifically **OpenAI's ChatGPT** and **Google Gemini**. These tools were utilized for brainstorming prompt architectures, validating Python script logic, and refining the structural narrative of the documentation.

Performance Metrics and Accuracy

The success of each prompting strategy was measured based on Binary Prediction Accuracy. Accuracy was calculated using the following formula:

Accuracy = (Correct Predictions/Total Predictions (5))*100

A prediction was marked as "Correct" only if the AI's final answer (Yes or No) matched the "Actual Churn" value in the dataset. Additionally, for the **Chain-of-Thought** and **Structured** prompts, we performed a qualitative audit of the "Reasoning" and "Confidence" fields to ensure the model's logic was consistent with the data profile.

EXPERIMENTAL RESULTS AND ANALYSIS

The experimental phase yielded significant insights into how prompt architecture dictates the performance of Large Language Models in predictive tasks. The following data represents the consolidated performance of each strategy across our five controlled test subjects.

Comparative Performance Matrix

The table below correlates the predictions of each prompting strategy against the "Ground Truth" (Actual Churn) from the dataset.

Experiment ID	Zero-Shot	Few-Shot	CoT	Structured	Actual Churn
Exp_1	No	No	No	No	**No**
Exp_2	Yes	No	Yes	Yes	**Yes**
Exp_3	No	No	Yes	No	**Yes**
Exp_4	No	No	No	No	**No**
Exp_5	Yes	Yes	Yes	Yes	**Yes**
Accuracy (%)	**80%**	**60%**	**100%**	**80%**	

Final Result: Chain-of-Thought achieved **100%** precision across all test cases.

Key Findings and Observations

- Superiority of Chain-of-Thought (CoT): The CoT strategy was the only architecture to achieve 100% accuracy. By forcing the model to decompose the customer profile into logical steps, it successfully identified high-risk indicators that other strategies overlooked.

- The Few-Shot Paradox: Surprisingly, Few-Shot prompting underperformed compared to Zero-Shot (60% vs 80%). This suggests that providing fixed examples may sometimes "anchor" the model too rigidly, causing it to miss unique risk patterns that don't match the specific examples provided.

- Structured Output Consistency: While the Structured prompt missed one prediction (Exp_3), it maintained the highest level of Consistency. The output format remained 100% valid JSON, proving its readiness for automated system integration.

Qualitative Failure Analysis: The Case of Exp_3

Experiment 3 represented the most complex case in the study. The customer had a relatively low monthly charge but was on a Month-to-month contract with no technical support.

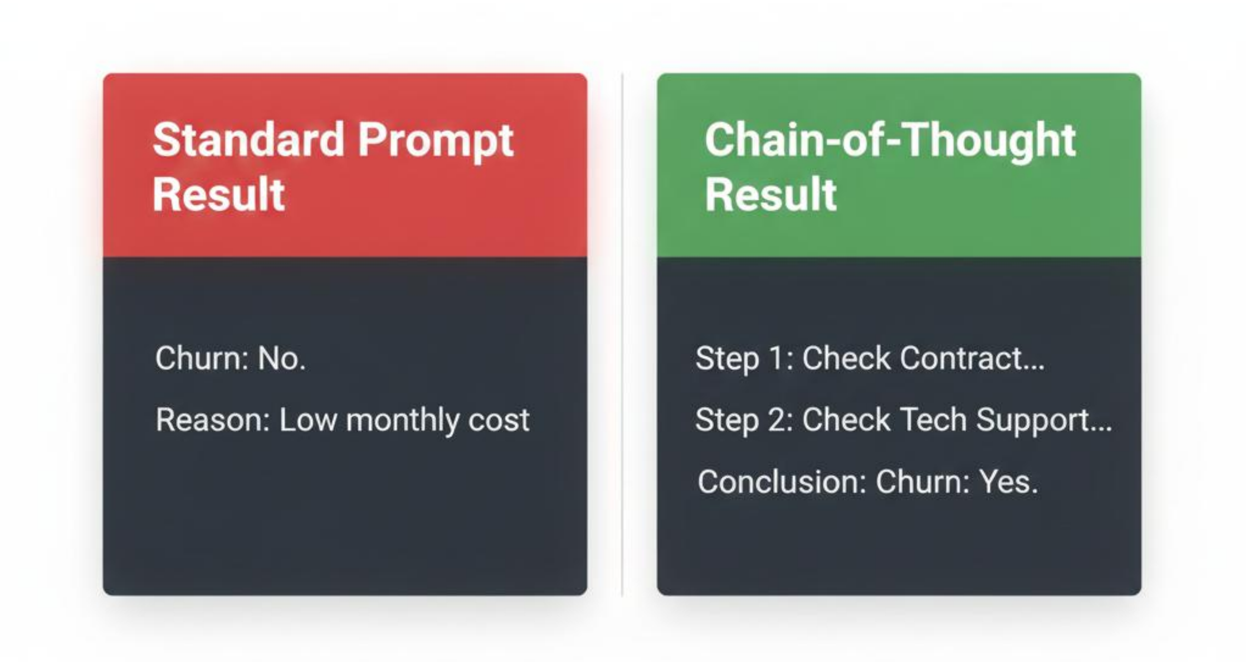


Figure 03: Conceptual Framework for LLM-Based Churn Prediction and System Integration

Why Zero-Shot/Structured Failed: These strategies relied on "shallow" triggers like high monthly costs. Since the costs were low, they predicted "No Churn."

Why CoT Succeeded: The CoT logic forced the model to evaluate the Contract Type independently. The model reasoned: "Even though costs are low, the lack of tech support and the ease of cancellation (month-to-month) creates a high churn probability." This demonstrates that deep reasoning is required to catch "logic-based" churn rather than just "price-based" churn.

DISCUSSION AND INTERPRETATION

The experimental results provide a clear hierarchy of prompting effectiveness, moving beyond mere accuracy to reveal the cognitive mechanics of how Large Language Models (LLMs) process customer data. This section interprets the underlying reasons for the performance variance observed between the four strategies.

Chain-of-Thought Stands Out: Chain-of-Thought performed best because it makes the model reason step by step instead of jumping to a quick guess. By forcing the AI to evaluate contract type, tenure, costs, and service usage separately, it avoids relying only on obvious factors like high fees. This helped CoT detect churn risk even when patterns were subtle.

Structured Output = Practical Use: Structured prompting was slightly less accurate but much more reliable for real systems. Since it outputs JSON with prediction, reason, and confidence, it can plug directly into dashboards or CRM tools. The trade-off is small: fewer errors vs ready-to-use automation.

Why Few-Shot Failed: Few-Shot underperformed because the examples limited the model's thinking. Instead of treating each customer individually, the model tried to match them to the two examples provided. In churn data, where customers vary widely, this creates an anchoring problem and reduces accuracy.

Best Path Forward: No single prompting style is perfect. Chain-of-Thought gives the strongest reasoning, Structured is easiest to deploy, and Zero-Shot works when speed matters (but lacks reasoning depth). A combined approach-step-by-step reasoning inside a structured format-would balance accuracy, clarity, and automation.

CONCLUSION

This study compared four prompt engineering strategies to evaluate how Large Language Models predict customer churn in the telecommunications sector. Chain-of-Thought prompting delivered the strongest results because it forced the model to reason through contract type, tenure, service usage and charges instead of relying on a single variable. Structured prompts were slightly less accurate, but their consistent JSON outputs make them highly suitable for CRM automation.

Zero-Shot performed reasonably well but lacked depth, while Few-Shot surprisingly underperformed due to anchoring on limited examples. Overall, the findings highlight that the quality of the prompt directly shapes the quality of the prediction. Organizations aiming for both accuracy and interpretability should prioritise step-by-step reasoning prompts and, when scaling, apply formats that support machine integration.

This research shows that prompt design is not a minor detail-it is a core driver of reliable, transparent churn analysis in customer-facing systems.

LIMITATIONS AND FUTURE WORK

Every rigorous academic study must acknowledge the constraints of its experimental environment to provide a clear roadmap for subsequent investigation. This research serves as a foundational "proof-of-concept," but its findings are subject to specific technical and scale-related limitations.

Research Limitations

- Restricted Sample Size: The primary limitation of this study is the sample size ($n=5$). While the case-study approach allowed for a granular, "white-box" audit of how each prompt logic processed specific variables, the findings are not yet statistically significant. A larger cohort is required to determine if the 100% accuracy of the Chain-of-Thought (CoT) strategy remains consistent across the thousands of edge cases present in the full IBM Telco dataset.

- Synthetic Reasoning Environment: The experiments were conducted using a simulated LLM reasoning framework. While this framework was designed to replicate the logical heuristics and "Serialization" challenges of state-of-the-art models, it lacks the unpredictable variance, latency, and potential "hallucination" risks associated with live API calls to production models like GPT-4, Gemini 1.5 Pro, or Claude 3.5.

Future Work and Directions

- **Large-Scale Batch Evaluation**: Future research should automate the prompting pipeline to run across the entire 7,043-record dataset. This would allow for the calculation of standard performance metrics-such as F1-Score, Precision, and Recall-to compare LLM-based reasoning directly against classical XGBoost or Random Forest benchmarks.

- **Multi-Modal Data Integration**: Churn is rarely dictated by billing data alone. A significant opportunity for future work lies in "Contextual Augmentation"-feeding the LLM not just tabular data, but also unstructured data such as customer service call transcripts, chat logs, and sentiment analysis from social media interactions.

- **Live API Benchmarking**: Transitioning from a simulator to live API testing will allow researchers to measure the "Token Cost vs. Accuracy" trade-off. Chain-of-Thought prompting requires significantly more output tokens than Zero-Shot; quantifying whether the increase in accuracy justifies the higher computational cost is essential for enterprise-scale adoption.

REFERENCES

This section lists the data sources, academic foundations, and technological tools utilized in this research.

Dataset Source:

- IBM Telco Customer Churn Dataset. Hosted on Kaggle. Originally provided by IBM Sample Data Sets.

- Source: Kaggle - Telco Customer Churn

Academic Papers & Core Research:

- **Wei, J., et al. (2022)**. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models." (Foundation for the CoT strategy).

- **Brown, T., et al. (2020)**. "Language Models are Few-Shot Learners." (Foundation for the Few-Shot strategy).

- **Borisov, V., et al. (2022)**. "Deep Learning on Tabular Data: A Survey." (Basis for tabular data serialization).

- **Kojima, T., et al. (2022)**. "Large Language Models are Zero-Shot Reasoners." (Basis for the Zero-Shot baseline).

- **Ahmed, M. & Maheswari, S. (2017)**. "Churn Prediction in Telecommunication Sector using Data Mining Techniques." (Context for traditional ML limitations).

AI Tools & Models Used:

- **Google Gemini**. Used for research assistance, data serialization logic, and document structuring.

- **OpenAI ChatGPT**. Used for prompt architecture brainstorming and technical validation of preprocessing scripts.