

Optimizing Deepseek Chat API Costs Based on Log Pattern Analysis using Business Intelligence and CRISP-DM

Della Anissa Putri Widodo¹, Nono Heryana², Azhari Ali Ridha³

^{1,2,3}Universitas Singaperbangsa Karawang, Indonesia

Email: dellaapw@gmail.com

Abstract

This study aims to analyze token usage and cost patterns of the DeepSeek Chat API and formulate cost optimization strategies based on historical log data. The study employs a Business Intelligence approach with the CRISP-DM framework through descriptive and diagnostic analysis of usage data from July 9 to November 30, 2025. The analysis covers token usage, cost, cost per token, usage patterns over time, correlation, and context caching. The results show a total usage of 81,612,706 tokens with a total cost of USD 11.212044 and an average daily usage of 562,846.25 tokens. The aggregate cost per million tokens was USD 0.137381, while the Pearson correlation between token usage and cost was 0.791494. Context caching analysis showed a cache hit rate of 83.97% and a cache miss rate of 16.03%. Based on the analysis, usage patterns with relatively higher costs can be identified to support token volume monitoring and the utilization of context caching as part of cost optimization strategies. The results are presented through a Business Intelligence dashboard as a tool for monitoring and evaluating DeepSeek Chat API usage.

Keywords: DeepSeek Chat, API, Token, Context Caching, Business Intelligence, CRISP-DM.



A. INTRODUCTION

The advancement of Artificial Intelligence (AI) has driven changes in the development of technology-based services, including chatbot systems that enable users to interact with computers using natural language. Progress in Natural Language Processing (NLP) and Machine Learning (ML) has facilitated the creation of chatbots capable of providing automated assistance and services across various sectors (Caldarini et al., 2022). With the evolution of Large Language Models (LLMs), the ability of generative chatbots to comprehend context and generate responses is increasingly being utilized and studied in fields such as education, healthcare, and business services (Aldhafeeri et al., 2025; Habibie, 2026). The increasing use of this technology makes technical and operational aspects, including resource requirements and processing costs, important considerations in managing generative AI services.

One operational aspect requiring attention in the use of generative AI is the cost of processing via Application Programming Interfaces (APIs). APIs allow applications to access AI model capabilities through interfaces provided by service providers. For LLM services that employ token-based billing, usage costs are linked to the volume of input and output tokens processed. Tokens are units of text used by models to represent inputs and generate outputs; consequently, a single token does not

necessarily equate to a single word or character. The volume of processed tokens can be tracked through usage data returned by the API service, allowing for an understanding of the usage components that drive costs.

Research on tokenization efficiency indicates that the number of tokens generated from text varies depending on the language and model employed; however, findings specifically focused on Ukrainian cannot be directly generalized to Indonesian (Maksymenko & Turuta, 2025). Consequently, token counts and their processing characteristics are relevant factors when analyzing usage patterns and the costs associated with LLM services.

The DeepSeek Chat API is one service available for accessing generative AI models. DeepSeek has garnered attention as its models are discussed in the context of balancing development costs, computational efficiency, and AI model performance (Cusumano, 2025). Studies on the implications of DeepSeek for data science and management also highlight the potential for utilizing AI at a lower cost compared to approaches requiring extensive computational resources (Xu et al., 2025). Furthermore, comparisons between DeepSeek-R1 and ChatGPT examine differences in technological approaches, capabilities, and implementation challenges (Jin et al., 2025). Research on DeepSeek-R1 also demonstrates how reinforcement learning is utilized to develop reasoning capabilities for verifiable tasks, such as mathematics and programming (Guo et al., 2025). While these studies provide context regarding DeepSeek's technological evolution, reasoning capabilities and model development efficiency cannot be directly equated to API usage cost-efficiency across all request patterns. Therefore, the cost of using the DeepSeek Chat API must be evaluated based on applicable rates and actual usage data.

Comparing API rates provides an initial basis for understanding cost differences between DeepSeek services and those offered by other providers. In the pricing announcement for DeepSeek-V3 published in December 2024, DeepSeek set rates effective February 8, 2025, at US\$0.27 per million input tokens for cache misses, US\$0.07 per million input tokens for cache hits, and US\$1.10 per million output tokens. Meanwhile, OpenAI documentation lists GPT-4o rates at US\$2.50 per million input tokens and US\$10.00 per million output tokens.

Based on a comparison of these unit rates, DeepSeek's announced "cache miss" input price for 2025 is approximately 89.2% lower than GPT-4o's input rate, while its output price is about 89% lower than GPT-4o's output rate. This comparison indicates that DeepSeek offers lower unit rates for the specific components compared. However, a difference in unit rates does not automatically prove that actual costs across all usage patterns will always be lower, as costs are also influenced by token volume, the ratio of cache hits to cache misses, the model type, and the rates applicable at the time of use.

Beyond unit rates, input processing mechanisms are also relevant to API cost management. DeepSeek offers a context caching mechanism that allows recurring input segments to be reused when cache matching criteria are met. A "cache hit" refers to the volume of input tokens that successfully utilize stored context, whereas a "cache

miss" refers to input tokens that do not utilize stored context. These two categories carry different rates within DeepSeek's pricing structure; therefore, their composition must be considered when analyzing API usage costs. Consequently, a cost evaluation cannot rely solely on total token counts but must distinguish between cache-hit input tokens, cache-miss input tokens, and output tokens. This distinction enables a more detailed analysis of the relationship between usage characteristics and incurred costs.

In practice, API usage generates log data containing information on usage timing, token counts, costs, and request volumes. This data can be utilized to examine usage patterns over time. However, viewing historical data in isolation does not necessarily provide a comprehensive picture of changes in token volume, cost fluctuations, or the relationships between usage components and costs. Systematic analysis is required to summarize and compare historical data across consistent periods. Through such analysis, researchers can identify variations in usage levels, compare costs between periods, and examine the relationships between token counts and costs, as well as between cache categories and costs. The analysis results must be interpreted in light of the data's characteristics, as a statistical relationship between two variables does not inherently prove a cause-and-effect relationship.

Transforming historical data into information suitable for decision-making requires a structured analytical approach. One relevant approach is Business Intelligence (BI), which involves the use of data, analytical processes, and information presentation to support data-driven decision-making. BI systems play a role in collecting, analyzing, and presenting relevant information to support organizational decision-making processes (Nugroho & Utama, 2025). In business analytics, an understanding of the data, transformation processes, and the analytical context is also essential to ensure that the results are interpreted accurately and do not lead to erroneous decisions (Staudinger et al., 2025).

The objective of this study is to characterize token usage patterns and costs associated with the DeepSeek Chat API service from July to November 2025; identify factors contributing to cost efficiency or waste in API usage within AI chatbot operations; formulate optimization strategies for token usage patterns to reduce operational costs; and develop recommendations for managing DeepSeek Chat API usage based on the analysis of historical log data.

B. LITERATURE REVIEW

1. Artificial Intelligence and Chatbot

Artificial Intelligence (AI) has evolved into a field dedicated to creating systems capable of mimicking human intelligence. Within this context, chatbots have emerged as computer programs specifically designed to engage in interactive conversations with users via text or voice (Caldarini et al., 2022). This development is driven by technological advancements such as Natural Language Processing (NLP) and large-scale machine learning models, which enable chatbots to comprehend user inquiries and automatically generate relevant responses (Atmauswan & Abdullahi, 2022; Florensa et al., 2024).

With their increasingly intelligent capabilities, chatbots have now been widely adopted in various sectors such as e-commerce, banking, education, and public services. Their goal is to improve operational efficiency and service quality (Laymouna et al., 2024). Chatbots enable organizations to provide round-the-clock customer service without relying on human staff availability; this not only accelerates response times but also alleviates manual workloads (Atmauswan & Abdullahi, 2022). From a business perspective, integrating chatbots into sales workflows has been shown to improve process consistency and speed, ultimately driving operational acceleration (Budiman et al., 2021; Caldarini et al., 2022). Furthermore, chatbots contribute to lowering operational costs by automating routine tasks that were previously handled manually, thereby reducing reliance on human resources (Laymouna et al., 2024).

From a user experience perspective, chatbots offer ease of interaction by enabling communication in more natural language alongside rapid responses (Laymouna et al., 2024). This aligns with findings by Nugraha et al. (2022), who emphasize that chatbot responsiveness is a key determinant of user satisfaction, as customers generally value speed when obtaining information (Atmauswan & Abdullahi, 2022).

Furthermore, attributes such as ease of use, efficiency, and reliability drive user interest in adopting chatbots as digital marketing tools (Florensia et al., 2024). Recent developments in the field of AI have even given birth to a generation of chatbots based on large language models, such as GPT-4, which are able to understand conversational context more deeply and produce more complex responses (Atmauswan & Abdullahi, 2022).

However, despite these benefits, implementing chatbots that utilize advanced AI models entails high costs. Consequently, analyzing cost optimization for AI API usage is critical to ensuring that the benefits gained justify the investment. This context underscores the relevance of further research into cost optimization for the DeepSeek Chat API.

2. APIs, Tokens, and Fee Structures

An Application Programming Interface (API) is an interface that enables an application to communicate with external services or systems. In the context of artificial intelligence, APIs allow developers to integrate AI model capabilities into applications without having to independently manage the entire computing infrastructure and model inference processes (Mishra et al., 2023). Generally, API usage involves sending a request containing instructions or input data to a service provider's server, processing the request via the model, and returning a response to the calling application. This mechanism enables the capabilities of Large Language Models (LLMs) to be utilized through software integration.

The cost of using an LLM API is typically calculated based on the number of tokens processed by the model. A token is a unit of text representation used by the model to process input and generate output. Tokens do not always correspond to

single words, as the tokenization process can split text into whole words, word fragments, numbers, or symbols.

The token count of a given text depends on the text's characteristics and the specific tokenization method employed by the model. Therefore, the actual number of tokens processed can be verified via usage information returned by the API.

The number of tokens in a text depends on the text's characteristics and the tokenization method employed by the model. Consequently, the actual number of tokens processed can be verified via usage information returned by the API.

API billing schemes may distinguish between input and output tokens and apply different rates to specific usage categories. For the DeepSeek API, input tokens are categorized into cache hits and cache misses, while output tokens are subject to a separate rate. Thus, usage costs depend not only on the total token count but also on the token composition and the rates applicable to each category.

API rates are subject to change in accordance with service provider policies. On September 29, 2025, DeepSeek announced an API price reduction of over 50% upon introducing the experimental DeepSeek-V3.2-Exp model. This change highlights that historical cost analysis must account for the periods during which specific rates were in effect, rather than assuming a single, constant rate over time. In this study, cost data recorded in usage logs serve as the basis for analysis, while official rate information provides the context for understanding cost differences across usage categories.

API usage efficiency is also linked to prompt design and model usage patterns. Excessively long instructions or the inclusion of unnecessary context can increase the volume of input tokens processed. Research on prompt compression indicates that prompt length can drive up computational demands; consequently, compression strategies must balance processing efficiency with the quality of the generated response (Gao et al., 2024). Studies on cost-based prompt optimization further highlight the importance of simultaneously considering prompt length, the number of model evaluations, and task performance when devising optimization strategies (Zehle et al., 2025). Accordingly, crafting concise prompts, limiting context to relevant information, and managing output length can serve as strategies to control token usage.

One mechanism that supports cost efficiency is context caching. This mechanism allows input segments that meet cache-matching criteria to be reused in subsequent requests. In the DeepSeek API, input prefixes matching cached data are classified as "cache hits," whereas non-matching input tokens are classified as "cache misses." Official DeepSeek documentation states that the caching mechanism operates automatically but does not guarantee a cache hit for every request, as the outcome depends on the suitability of the reused context. Therefore, the ratio of cache hits to cache misses can be used to analyze token usage composition and its relationship to costs.

3. Business Intelligence (BI)

Business Intelligence (BI) is an approach that leverages technology and data analysis processes to transform operational data into information that supports organizational decision-making. The BI process encompasses data collection, integration, processing, analysis, and presentation, enabling the resulting information to be used for understanding conditions and evaluating organizational performance. In practice, BI facilitates data-driven decision-making by helping organizations identify patterns, trends, and relevant insights within available datasets (Michael et al., 2024; Mekimah et al., 2024).

Technically, BI implementation can include reporting, data analysis, Online Analytical Processing (OLAP), and information presentation via visualizations and dashboards. These components assist users in examining data from various perspectives and understanding performance indicators more systematically. BI dashboards typically feature indicators, charts, tables, and interactive elements that allow users to explore information according to their analytical needs. The effectiveness of this information presentation depends on the quality and completeness of the data, as well as how the visualizations are designed to support decision-making tasks (Burnay et al., 2024; Hjelle et al., 2024).

In business analytics practice, four common approaches are distinguished based on the objective of the questions they aim to answer: descriptive analytics, diagnostic analytics, predictive analytics, and prescriptive analytics. These four approaches serve distinct functions, ranging from summarizing historical events to supporting the selection of actions based on analytical findings (Balasingam, 2022).

4. CRISP-DM

CRISP-DM (Cross-Industry Standard Process for Data Mining) is a process framework used to systematically organize data analysis projects through six phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. These six phases are interconnected and can be executed iteratively based on project goals and requirements. Although CRISP-DM originated in data mining practices, the evolution of data analytics demonstrates that this framework can be adapted for various types of data science projects and diverse analytical needs (Martínez-Plumed et al., 2021).

CRISP-DM does not necessarily have to be applied to build predictive models. In the context of descriptive analytics, activities within the Modeling, Evaluation, and Deployment phases can be adjusted to meet the need for generating descriptive information.

Modeling activities can be replaced by the formulation and execution of queries or analyses that address information requirements, while evaluation focuses on the accuracy and interpretation of the analysis results. Thus, CRISP-DM can be used to organize historical data analysis without the need to train predictive models (Staudinger et al., 2025).

The application of CRISP-DM can also be tailored to support business intelligence (BI) needs, particularly when projects focus on data integration, data quality, validation, and information presentation via dashboards. One development of the CRISP-DM framework for BI establishes Dashboard Development as a distinct phase while emphasizing validation and sustainable data management. This development represents an adaptation to a specific context rather than a universal replacement of the original six CRISP-DM phases (Sousa et al., 2026). Furthermore, the development of analytical processes aligned with specific domain requirements demonstrates that data analysis frameworks can be customized based on problem characteristics, governance, and the needs of information users (Plotnikova et al., 2023).

C. METHOD

This study employs a quantitative approach utilizing the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework, adapted to the context of analyzing and optimizing costs associated with the DeepSeek Chat API. The CRISP-DM framework structures the research process systematically, encompassing problem understanding, data understanding, data preparation, analysis, and evaluation, through to the presentation of findings and recommendations. The study focuses on the usage of the DeepSeek Chat API service, specifically the 'deepseek-chat' model. The data utilized consists of historical service usage logs spanning the period from July 9 to November 30, 2025. The analysis centers on token counts, usage costs, and context caching components, specifically cache hits and cache misses. This data is processed to identify usage patterns, relationships between variables, usage levels, and characteristics relevant to cost efficiency.

D. RESULT AND DISCUSSION

1. Data Preparation (CRISP-DM)

The data preparation stage is a phase within the CRISP-DM methodology aimed at preparing data for consistent use during the analysis process. In this study, data preparation follows the data understanding stage to ensure that the DeepSeek Chat API usage log data aligns with the research requirements. This process involves merging data from multiple files, cleaning and adjusting the data structure, establishing a complete observation date range, and creating derived variables necessary for analysis.

The data used in this study originates from several monthly log files categorized by information type: token usage data and cost data. The token usage data includes the various token types used to calculate daily token totals, while the cost data contains information on service usage costs by date. Consequently, an integration process is required to consolidate token usage and cost information into a single daily dataset.

The data preparation stage in this study comprises four main processes: data integration, data cleaning, handling missing values, and feature engineering. These

processes are executed using the Python programming language within the Google Colab environment. The resulting output serves as the final dataset for descriptive and diagnostic analyses.

2. Basic Calculations

Following the data preparation process, basic calculations were performed to obtain a quantitative overview of token usage and DeepSeek Chat API costs during the observation period. These calculations covered total token usage, total usage costs, average daily token usage, average daily costs, and cost per token.

The data spanned 145 days of observation: 23 days in July 2025, 31 days in August 2025, 30 days in September 2025, 31 days in October 2025, and 30 days in November 2025. Token usage was calculated based on three categories: output tokens, input cache miss tokens, and input cache hit tokens. The request count variable was excluded from the total token count because it represents the number of requests rather than the number of tokens. The results of the overall basic calculations can be summarized as follows:

Table 1. Basic Calculation of Token Usage and Costs

Month	Day	Total Token	Average Tokens/Day	Total Cost (USD)	Average Cost/Day (USD)	Cost per 1 Million Tokens (USD)
July	23	7,644,234	332,358.00	0.922949715	0.040128248	0.120738
August	31	17,756,495	572,790.16	2.304493950	0.074338515	0.129783
September	30	20,686,582	689,552.73	4.474594084	0.149153136	0.216304
October	31	21,128,456	681,563.10	1.994836872	0.064349577	0.094415
November	30	14,396,939	479,897.97	1.515169684	0.050505656	0.105242
Total/Average	145	81,612,706	562,846.25	11.212044305	0.077324443	0.137381

Based on Table 1, the cost per million tokens varies across the months of observation. This figure is calculated from the ratio of total cost to total token usage for each month. The aggregate value for the entire period of US\$0.137381 per million tokens is calculated based on a total cost of US\$11,212044305 and a total usage of 81,612,706 tokens, and is therefore not obtained through an average of five monthly cost values per million tokens.

Based on preliminary calculations, usage totaling 81,612,706 tokens was recorded between July and November 2025, with an aggregate cost of US\$11.212044305. Average usage stood at 562,846.25 tokens per day, while the average daily cost was US\$0.077324443. A monthly breakdown reveals that September 2025 recorded the highest average token usage at 689,552.73 tokens per day. October followed with an average of 681,563.10 tokens per day, whereas July recorded the lowest average usage at 332,358 tokens per day.

In terms of cost, September also registered the highest daily average at US\$0.149153136. This aligns with the high volume of token usage during that month; however, the relationship between token volume and cost extends beyond mere averages.

Meanwhile, an analysis of the cost per million tokens reveals monthly variations. September recorded the highest rate at US\$0.216304 per million tokens, while October recorded the lowest at US\$0.094415 per million tokens. These figures indicate variations in cost relative to token volume across the different periods. At this stage, the cost-per-token figure serves as an indicator of relative cost efficiency rather than as the sole basis for determining an optimal pattern.

3. Usage Pattern Analysis

The usage pattern analysis in this study aims to identify trends in token consumption and DeepSeek Chat API service costs during the observation period. The analysis employs a descriptive analytics approach applied to time-series data that has undergone a data preparation phase. The analysis in this section covers trends in token usage and daily costs, usage patterns by day of the week and by month, and the distribution of daily token usage.



Figure 1. Trends in Daily Token Usage and Costs

Figure 1 illustrates the daily changes in token usage and DeepSeek Chat API service costs throughout the observation period. The data used consists of daily figures aggregated by date. Based on the visualization, token usage shows a fluctuating pattern throughout the observation period. There are days with high usage and days with low usage, including days with no usage activity. Daily usage ranged from 0 to 1,615,491 tokens. Over the 145-day observation period, there were 137 active days and 8 inactive days. Overall, changes in token usage appeared to align with changes in costs. Total daily token usage is derived by summing three token components: output tokens, input cache miss tokens, and input cache hit tokens.

An analysis based on the day of the week was conducted to determine differences in average token usage across the days. The metric used was average daily token usage, ensuring that comparisons were not influenced by variations in the frequency of occurrence for each day during the observation period.

Table 2. Calculation of Average Daily Token Usage

Day	Number of days	Total Token	Average Tokens/Day
Monday	20	14,033,283	701,664.15
Tuesday	20	10,961,864	548,093.20

Wednesday	21	10,604,476	504,975.05
Thursday	21	8,742,911	416,329.10
Friday	21	9,456,194	450,294.95
Saturday	21	11,236,542	535,073.43
Sunday	21	16,577,436	789,401.71

Based on the results of the table above, Sunday has the highest average token usage, namely 789,401.71 tokens per day, while Thursday has the lowest average, namely 416,329.10 tokens per day. This difference indicates a variation in usage levels based on the day of the week. These results serve to illustrate temporal usage patterns but are not used to infer the causes of user behavior, as the study did not directly measure behavior factors.

A month-by-month analysis was conducted to determine changes in token usage volume across each observation period. Total token usage for a given month was calculated by summing the token usage for every day within that month.

Table 3. Total Token Usage by Month

Month (Year 2025)	Number of days	Total Tokens
July	23	7,644,234
August	31	17,756,495
September	30	20,686,582
October	31	21,128,456
November	30	14,396,939

Based on the calculations in Table 3, October 2025 recorded the highest total token usage at 21,128,456 tokens. This figure is slightly higher than that of September 2025, which reached 20,686,582 tokens. Conversely, July 2025 recorded the lowest total token usage at 7,644,234 tokens. This value needs to be considered based on the number of observation days because the research period in July 2025 started on the 9th, so it only covers 23 days, while other months have observation periods between 30 and 31 days. Thus, the analysis reveals variations in token usage volume across the months of the observation period. These results illustrate usage fluctuations for each period and serve as a basis for examining cost characteristics and relative efficiency in the subsequent analysis.

Distribution analysis was employed to examine how token usage values were distributed across the 145-day observation period. A histogram was used for visualization, with daily token usage serving as the variable grouped into several value ranges. Based on the daily data, token usage ranged from a minimum of 0 tokens to a maximum of 1,615,491 tokens. This range indicates variations in usage intensity across different days during the observation period. To facilitate the creation of a usage-level variable, the token usage range was divided into three groups of equal interval width.

With an interval width of 538,497 tokens, the range of token usage is divided into three groups as follows:

- a) Low : 0 to 538,497 tokens;

- b) Medium: 538,497 to 1,076,994 tokens;
- c) High: 1,076,994 to 1,615,491 tokens.

Subsequently, each daily observation is assigned to a group based on the token usage for that day. The grouping results are presented in Table 4.

Table 4. Distribution of Token Usage Levels

Usage Rate	Token Interval	Total Days	Total Token
Low	0-538,497	78	21,067,378
Medium	538,497–1,076,994	53	42,381,200
High	1,076,994–1,615,491	14	18,164,128
Total		145	81,612,706

For example, the number of days in the "Low" group was determined by counting all daily observations with token usage in the range of 0 to 538,497 tokens. The grouping results show that there are 78 days in this group. The "Medium" group comprises 53 days, while the "High" group comprises 14 days. Based on this distribution, the "Low" group accounts for the highest number of days (78 days), followed by the "Medium" group (53 days) and the "High" group (14 days). Thus, high-intensity usage occurs on fewer days compared to usage at the "Low" and "Medium" levels. In this study, the "High" level is not automatically regarded as a data error or wasteful expenditure; rather, this classification serves to distinguish the intensity of token usage across different days.

Usage patterns for the DeepSeek Chat API service from July to November 2025 can be summarized as follows:

- a. Token usage fluctuated over time.
Daily usage ranged from 0 to 1,615,491 tokens. Out of the 145 days observed, there were 137 active days and 8 inactive days.
- b. Usage varied by day of the week.
The highest average usage occurred on Sundays (789,401.71 tokens/day), while the lowest average usage occurred on Thursdays (416,329.10 tokens/day).
- c. Usage varied across months.
Total token usage peaked in October 2025 at 21,128,456 tokens, while the lowest total was recorded in July 2025 at 7,644,234 tokens.
- d. Most days fell into the Low and Medium usage categories.
Of the 145 days observed, 78 fell into the Low category, 53 into the Medium category, and 14 into the High category.
- e. Changes in token usage and costs require further quantitative analysis.
Trend visualizations indicate that changes in token usage generally moved in tandem with changes in costs; however, the strength of the relationship between these two variables cannot be determined based on visualization alone.

4. Analysis of Relationships and Factors

The analysis of relationships and factors in this study aims to identify the connection between token usage and costs, as well as to pinpoint variables that can be observed and analyzed based on the dataset. A diagnostic analytics approach is employed to examine inter-variable relationships once usage patterns have been identified.

The main variables analyzed include the number of tokens used, usage costs, usage levels, and context caching components in the form of cache hits and cache misses. These variables were selected because they are available in the log data and can be quantitatively measured. In addition to the main variables, the research dataset also has derived variables in the form of cost per token, usage level, high usage flag, and active, which are used to support the analysis and comparison of usage characteristics.

The analysis in this section is not intended to experimentally establish causal relationships. In this study, the term "factor" refers to a variable or usage characteristic that demonstrates an empirical relationship or association with costs based on historical data. Consequently, the analysis results serve to explain the patterns of relationships found in the data and should not be interpreted as evidence that a specific variable directly causes changes in costs.



Figure 2. Scatter plot of the relationship between token usage and cost

Figure 2 is used to examine the relationship between daily token usage and daily usage costs. The horizontal axis represents the number of tokens, while the vertical axis represents the cost in USD. Each point on the scatter plot represents a single day of observation; thus, there are 145 data points representing the entire period from July to November 2025. The data points show a trend moving from the bottom-left to the top-right. This pattern indicates a positive relationship between token usage and cost. In other words, days with higher token usage tend to incur higher usage costs.

To quantitatively measure the strength and direction of the relationship between token usage and cost, the Pearson correlation coefficient is used. A correlation coefficient of 0.791494 indicates a strong positive relationship between

token usage and costs during the observation period. This means that an increase in token usage tends to be accompanied by an increase in costs. To determine whether this relationship remains consistent across months, the correlation was also calculated on a monthly basis.

Table 5. Correlation between Token Usage and Cost by Month

Month	Pearson Correlation Coefficient
July	0.976456
August	0.967652
September	0.869020
October	0.860466
November	0.882776
Total	0.791494

The calculations indicate a positive relationship between tokens and costs across all months of observation. The highest correlation was observed in July (0.976456), while the lowest was in October (0.860466). Nevertheless, all monthly correlation values consistently demonstrate a strong positive relationship. These results reinforce the finding that the number of tokens is strongly linked to costs. However, the fact that the correlation values do not reach 1 indicates that cost fluctuations are not fully explained by changes in total token counts. Consequently, token composition and context caching mechanisms were further analyzed as observable factors within the dataset.

Based on the variables available in the dataset and the results of prior analysis, the variables that can be empirically analyzed in this study include token usage volume, the composition of cache hits and cache misses, and daily usage intensity. These three aspects can be calculated and compared using the available usage data, allowing the analysis to be conducted based on historical conditions recorded in the log data.

a. Token Usage Volume

Token usage volume is a key factor influencing usage costs. Correlation analysis reveals a Pearson correlation coefficient of 0.791494 between token usage and costs across the entire observation period. This value indicates a strong positive relationship between token usage and costs; consequently, days with higher token usage tend to incur higher costs.

b. Cache Hit and Cache Miss Composition

The second aspect analyzed is the composition of input tokens recorded as cache hits and cache misses. The DeepSeek Chat API provides a context caching mechanism, allowing specific input segments that meet caching criteria to be reused in subsequent requests. Of the total 76,909,934 tokens recorded as cache components, 83.97% were cache hits and 16.03% were cache misses.

The calculation results indicate a positive relationship between cache components and costs. However, this positive relationship does not imply that cache hits or cache misses directly cause changes in cost; rather, both

components are part of the overall token usage. Thus, the cache analysis results demonstrate that the composition of tokens recorded as cache hits and cache misses is a relevant factor in understanding cost variations.

c. Daily Usage Intensity

The third aspect analyzed is daily usage intensity. Usage intensity is represented by the "usage level" variable, which categorizes daily token usage into three levels: Low, Medium, and High. The calculation results show that the Low group has the largest number of days, namely 78 days, followed by the Medium group with 53 days, and the High group with 14 days.

An analysis based on usage levels was conducted to determine whether differences in daily usage intensity were accompanied by differences in average costs. The data were divided into three groups based on token usage intervals: Low, Medium, and High. Average daily costs show a pattern of higher values in groups with higher levels of token usage. The "High" group recorded the highest average daily cost at US\$0.192990, while the "Low" group recorded the lowest at US\$0.036655. The "Medium" group fell between the two, with an average daily cost of US\$0.106624.

These results indicate that days with higher token usage intensity tend to be associated with higher daily costs. This pattern is consistent with a positive relationship between token usage and cost, as numerically measured by the Pearson correlation. However, the differences in average costs across usage levels are not used to infer a causal relationship. This analysis merely demonstrates variations in cost characteristics based on usage intensity levels within the observed historical data.

Based on the various analyses, there are several key findings. First, token usage shows a strong positive correlation with costs. Pearson correlation analysis based on 145 days of observation yielded an r-value of 0.791494. Monthly correlations also revealed a strong positive relationship throughout the observation period, with values ranging from 0.860466 to 0.976456. These results indicate a strong link between changes in token volume and changes in costs.

Second, daily usage intensity reveals distinct cost differences. Average costs rose from US\$0.036655 per day in the "Low" usage group to US\$0.106624 per day in the "Medium" group, and to US\$0.192990 per day in the "High" group. Thus, usage levels can serve as an indicator to distinguish between days with relatively low costs and those with relatively high costs.

Third, the composition of context caching indicates that the majority of recorded input tokens resulted in cache hits. Of the total 76,909,934 tokens consisting of cache hits and cache misses, 64,584,256 tokens, or 83.97%, are cache hits. However, the cache hit rate declined from 94.30% in July to 76.64% in November, indicating a shift in cache composition over the study period.

Fourth, there is month-to-month variation in relative costs. The cost per million tokens ranged from US\$0.094415 to US\$0.216304. This variation demonstrates that actual costs cannot be adequately analyzed based solely on total token volume; one must also consider usage composition and the applicable rate structures for the period in question.

5. Analysis and Recommendations for Cost Optimization

The assessment in this section employs several indicators derived from the preceding analysis: cost per token, cost per million tokens, daily usage rate, and the composition of cache hits and cache misses. These indicators are necessary because a low cost figure does not necessarily reflect superior cost efficiency if the volume of tokens used is also low. Consequently, the evaluation is conducted by considering the relationship between cost and usage volume, alongside the characteristics of context caching. The cost-per-token value and the cost per million tokens serve as indicators of relative cost efficiency. These indicators are used to compare cost characteristics against usage volume across different periods or groups, rather than as absolute measures of the success of service usage.

Based on the previous calculation results, October 2025 has the lowest relative cost, namely USD 0.094415 per one million tokens, while September has the highest relative cost, namely USD 0.216304 per one million tokens. Compared to the overall average of USD 0.137381 per million tokens, October's relative cost was approximately 31.28% lower, whereas September's was approximately 57.45% higher.

Thus, October can be identified as the period with the best relative cost efficiency within the observed data, while September showed the lowest relative cost efficiency. This identification is based on the observed cost per million tokens for each month. The differences in relative costs between months must be interpreted with the understanding that DeepSeek's API service pricing structure changed during the observation period. However, this study could not empirically identify off-peak or on-peak usage because the log data did not provide hourly timestamps or more granular usage timing. Therefore, the differences in cost-per-token values between months in this study are not used to conclude that one period was more efficient due to off-peak or on-peak usage. Consequently, the term "relative cost efficiency" in this study refers to a comparison of observed costs per million tokens based on monthly data, rather than the determination of an absolute optimal pattern or one based on specific pricing periods.

A key characteristic of the DeepSeek Chat API is its context caching mechanism. This mechanism allows specific segments of the input context that meet caching criteria to be reused in subsequent requests. In the DeepSeek Chat API, a "cache hit" signifies input tokens successfully served from the cache, whereas a "cache miss" indicates input tokens not served from the cache that require processing as new input. The caching mechanism operates automatically within the API service. Consequently, a cache hit does not merely imply the reuse of the entire input; rather, it relates to the prefix portion that satisfies the caching conditions. Thus, the occurrence of cache hits and misses in the log data enables an analysis of how input tokens are distributed based on the utilization of the caching mechanism. In this study, cache hit and cache miss components were calculated based on data volumes collected between July and November 2025. Of the 76,909,934 tokens recorded as cache components, 64,584,256 tokens (83.97%) were cache hits, while 12,325,678 tokens (16.03%) were cache misses.

To examine the distribution of these two components over the study period, the numbers of cache hits and cache misses were broken down by month. The values in the visualization were obtained by summing the respective cache components across all days of each month. The calculation results are presented in Table 6.

Table 6. Composition of Cache Hits and Cache Misses Relative to Total Cache

Month	Cache Hit	Cache Miss	Total Cache Token	Cache Hit (%)
July	6,713,216	405,456	7,118,672	94.30%
August	15,046,720	1,608,341	16,655,061	90.34%
September	16,171,648	3,434,653	19,606,301	82.48%
October	16,227,264	3,700,194	19,927,458	81.43%
November	10,425,408	3,177,034	13,602,442	76.64%
Total	64,584,256	12,325,678	76,909,934	83.97%

Based on Table 6, the number of cache hits exceeded the number of cache misses throughout the observation period. However, the number of cache misses increased during certain periods, particularly in September and October. Next, to examine changes in context caching usage over the study period, an analysis was conducted based on the monthly cache hit percentage. The cache hit percentage represents the proportion of cached tokens successfully served via the cache relative to the total number of tokens recorded as either cache hits or cache misses during the same month. Based on Table 6, the cache hit percentage gradually declined over the observation period. In July, the cache hit percentage was 94.30%, then dropped to 90.34% in August, 82.48% in September, 81.43% in October, and reached 76.64% in November.

Consequently, the cache hit rate declined by 17.66 percentage points between July and November. Conversely, the cache miss rate rose from 5.70% in July to 23.36% in November. These changes indicate that the proportion of tokens successfully served via the cache did not remain constant throughout the study period. At the beginning of the observation period, the cache hit rate was relatively high, whereas by the end, it was lower. From a cost optimization perspective, this situation is significant because context caching enables the reuse of specific contexts; thus, tokens meeting cache hit criteria are subject to different input rate structures compared to cache misses, in accordance with the service pricing applicable during the relevant period. However, the fluctuation in the cache hit rate cannot be cited as the sole cause of changes in total costs or cost per token. The study period encompassed changes in pricing structures and DeepSeek Chat service versions, while the data analyzed was aggregated on a daily basis. Therefore, within this study, context caching is positioned as a mechanism related to token usage efficiency rather than as the sole determinant of costs.

Based on the results of cost per token analysis, usage level, and context caching, usage patterns in this research can be differentiated based on the relative cost efficiency observed. A pattern indicating relatively greater efficiency is usage that results in a lower cost-per-token value compared to the value for the overall period. Based on these figures, the months of July, August, October, and November had a cost

per token below the overall average, whereas September was above it. Notably, October demonstrated the best relative cost efficiency, with a rate of 0.094415 USD per million tokens. October also recorded a total usage of 21,128,456 tokens, indicating that high usage volume does not automatically result in a high cost per token. September recorded the highest cost per token at 0.216304 USD per million tokens. This figure is significantly higher than the overall average of 0.137381 USD per million tokens. Consequently, September was the period with the highest relative cost based on the volume of tokens used.

Based on the analysis, a cost optimization strategy was formulated by linking empirical findings, implications, and actionable measures.

a. Monitor cost per token as a cost-efficiency indicator

Research results indicate monthly variations in relative costs, with the cost per token ranging from US\$0.094415 to US\$0.216304 per million tokens.

Therefore, using a dashboard to periodically monitor the cost per token can serve as a cost-control mechanism. Significant deviations above the benchmark value can signal a need to examine usage patterns and token composition.

b. Monitor cache hit and cache miss composition.

During the study period, cache hits accounted for 83.97% of total tokens with cache information. However, this percentage declined from 94.30% in July to 76.64% in November.

Consequently, monitoring the cache hit rate can serve as a key indicator for cost management. Input structures that allow for the reuse of the same prefix can be maintained when consistent with application requirements. DeepSeek explains that cache hits depend on matching prefixes already stored in the cache.

c. Control token usage volume

The Pearson correlation of 0.791494 between token usage and cost indicates a strong positive relationship. Furthermore, the average daily cost rose from US\$0.036655 at the "Low" usage level to US\$0.106624 at the "Medium" level and US\$0.192990 at the "High" level.

Therefore, controlling token volume can be a primary strategy for managing costs, particularly on days with high usage intensity.

d. Evaluate usage spikes

Research data reveals significant variation in daily usage, with a peak of 1,615,491 tokens per day. Therefore, days showing usage significantly higher than normal patterns can serve as evaluation points to investigate the causes of the increase. This strategy does not assume that every spike represents waste; rather, it treats such spikes as signals prompting further examination.

e. Accounting for rate changes when evaluating costs across periods

The study period spans a change in DeepSeek's pricing structure in September 2025. Official documentation states that the new pricing took effect on September 5, 2025, at 16:00 UTC, coinciding with the discontinuation of off-peak discounts.

Consequently, month-to-month cost comparisons should always take into account the applicable rate periods. However, this study could not empirically determine usage occurring during off-peak versus on-peak periods because the log data did not provide hourly timestamps or more granular usage timing details.

Thus, a usage strategy based on selecting off-peak or on-peak times was not established as an outcome of this study. If usage timing information becomes available in the system implementation, the applicable rate periods could be considered a factor in cost management.

f. Using analysis results as a basis for periodic evaluation

Cost management should be conducted periodically by monitoring at least three indicators: total tokens, total cost, and cost per million tokens, along with an additional indicator: the cache hit percentage.

This approach allows cost fluctuations to be observed not only in terms of absolute values but also in terms of efficiency relative to usage volume.

E. CONCLUSION

Based on an analysis of usage log data for the DeepSeek Chat API service from July to November 2025, this study reveals that service usage followed a fluctuating and uneven pattern throughout the observation period. Of the 145 days of observation, there were 137 active days and 8 inactive days, with total usage reaching 81,612,706 tokens and an average of 562,846.25 tokens per day, consisting of 78 days with low usage levels, 53 medium days, and 14 high days. Average daily token usage rose from 332,358.00 tokens in July to 572,790.16 tokens in August, peaking at 689,552.73 tokens per day in September, before declining to 681,563.10 tokens in October and 479,897.97 tokens in November. Regarding the type of day, usage was higher on weekends, averaging 662,237.57 tokens per day, compared to 522,317.75 tokens on weekdays. In terms of cost, total expenditure during the study period amounted to USD 11.212044, with a daily average of USD 0.077324; a Pearson correlation of 0.791494 indicates a strong positive relationship between token volume and cost in the daily data. Cost variations were also evident based on usage levels, with average costs of USD 0.036655 for low usage, USD 0.106624 for moderate usage, and USD 0.192990 for high usage. The context caching component also shows a relationship with cost variations, with 64,584,256 cache hit tokens and 12,325,678 cache miss tokens, or 83.97% and 16.03% of the total caching usage, respectively, and the correlation of cache hits and cache misses to costs is 0.715489 and 0.677420, respectively, although this relationship is not used to conclude a causal relationship. From a relative cost-efficiency perspective, the cost per million tokens ranges from USD 0.094415 to USD 0.216304; October recorded the lowest cost (USD 0.094415) and September the highest (USD 0.216304), while the overall average was USD 0.137381 per million tokens. The cache hit percentage also changed from 94.30% in July to 76.64% in November, or decreased by 17.66 percentage points, which indicates variations in the use of context caching throughout the research period, but cannot be

used as the sole basis for explaining differences in cost efficiency because costs are also related to volume, token composition, and the applicable tariff structure. Based on these findings, optimizing DeepSeek Chat API usage could focus on controlling token volume, monitoring usage intensity, evaluating costs per million tokens, and managing context caching for tasks involving recurring contexts. Monitoring of high usage days can be used as part of cost control, while consistent context structures can be considered to support the chance of cache hits without assuming that increased cache hits directly lead to decreased costs. Changes in the pricing structure also need to be considered during cost evaluation; however, since the available data lacks hourly timestamps, this study could not determine usage during off-peak versus peak periods, and thus, time-based usage selection was not established as an optimization strategy. Consequently, the cost management strategy resulting from this study focuses on monitoring token volume and usage rates, cost per million tokens, and context caching characteristics based on historical data, enabling the analysis results to serve as a basis for managing DeepSeek Chat API usage in systems with similar characteristics.

REFERENCES

1. Aldhafeeri, L., Aljumah, F., Thabyan, F., Alabbad, M., AlShahrani, S., Alanazi, F., & Al-Nafjan, A. (2025). Generative AI chatbots across domains: A systematic review. *Applied Sciences*, 15(20), 11220. <https://doi.org/10.3390/app152011220>
2. Atmauswan, P. S., & Abdullahi, A. M. (2022). Intelligent chatbot for university information system using natural language approach. *Albukhary Social Business Journal*, 59-64. <https://doi.org/10.55862/asbjv3i2a007>
3. Balasingam, I. (2022). A review of business analytics methods. *International Journal on Research and Development—A Management Review*, 11(1), 129–132. <https://doi.org/10.65521/ijrdmr.v11i1.1778>
4. Budiman, A. A., Yudha, A., & Utomo, C. S. (2021). Pembuatan Sistem Chatbot Menggunakan Metode Contextual. *Jurnal Sains & Teknologi Fakultas Teknik Universitas Darma Persada*, 11(2), 20-24. <https://doi.org/10.70746/jstunsada.v11i2.157>
5. Burnay, C., Lega, M., & Bouraga, S. (2024). Business intelligence and cognitive loads: Proposition of a dashboard adoption model. *Data & Knowledge Engineering*, 152, 102310. <https://doi.org/10.1016/j.datak.2024.102310>
6. Caldarini, G., Jaf, S., & McGarry, K. (2022). A literature survey of recent advances in chatbots. *Information*, 13(1), 41. <https://doi.org/10.3390/info13010041>
7. Cusumano, M. A. (2025). DeepSeek inside: origins, technology, and impact. *Communications of the ACM*, 68(7), 18-22. <https://doi.org/10.1145/3732796>
8. Florensia, N. P., Safa, Y. N., Patimah, Y., Priskila, R., & Pranatawijaya, V. H. (2024). Implementasi Open AI pada website restoran makanan India. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(3), 3766-3772. <https://doi.org/10.36040/jati.v8i3.9770>

9. Gao, J., Cao, Z., & Li, W. (2024). Selfcp: Compressing over-limit prompt via the frozen large language model itself. *Information Processing & Management*, 61(6), 103873. <https://doi.org/10.1016/j.ipm.2024.103873>
10. Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., ... & Tan, Y. (2025). DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081), 633-638. <https://doi.org/10.1038/s41586-025-09422-z>
11. Habieby, N. (2026). Workload Analysis Using The FT (Full Time Equivalent) Method To Determine The Effective Employee Requirements At Pt. Bank Rakyat Indonesia. *Techne: Journal of Engineering, Technology and Industrial Applications*, 2(2), 1-9.
12. Hjelle, S., Mikalef, P., Altwaijry, N., & Parida, V. (2024). Organizational decision making and analytics: An experimental study on dashboard visualizations. *Information & Management*, 61(6), 104011. <https://doi.org/10.1016/j.im.2024.104011>
13. Jin, I., Tangsrivimol, J. A., Darzi, E., Hassan Virk, H. U., Wang, Z., Egger, J., ... & Krittanawong, C. (2025). DeepSeek vs. ChatGPT: prospects and challenges. *Frontiers in Artificial Intelligence*, 8, 1576992. <https://doi.org/10.3389/frai.2025.1576992>
14. Laymouna, M., Ma, Y., Lessard, D., Schuster, T., Engler, K., & Lebouché, B. (2024). Roles, users, benefits, and limitations of chatbots in health care: rapid review. *Journal of medical Internet research*, 26, e56930. <https://doi.org/10.2196/56930>
15. Maksymenko, D., & Turuta, O. (2025). Tokenization efficiency of current foundational large language models for the Ukrainian language. *Frontiers in artificial intelligence*, 8, 1538165. <https://doi.org/10.3389/frai.2025.1538165>
16. Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernández-Orallo, J., Kull, M., Lachiche, N., ... & Flach, P. (2019). CRISP-DM twenty years later: From data mining processes to data science trajectories. *IEEE transactions on knowledge and data engineering*, 33(8), 3048-3061. <https://doi.org/10.1109/TKDE.2019.2962680>
17. Mekimah, S., Zighed, R., Mili, K., & Bengana, I. (2024). Business intelligence in organizational decision-making: a bibliometric analysis of research trends and gaps (2014–2024). *Discover Sustainability*, 5(1), 532. <https://doi.org/10.1007/s43621-024-00692-7>
18. Michael, C. I., Ipede, O. J., Adejumo, A. D., Adenekan, I. O., Adebayo, D., Ojo, A. S., & Ayodele, P. A. (2024). Data-driven decision making in IT: Leveraging AI and data science for business intelligence. *World Journal of Advanced Research and Reviews*, 23(01), 432-439. <https://doi.org/10.30574/wjarr.2024.23.1.2010>
19. Mishra, R., Kaur, I., Sahu, S., Saxena, S., Malsa, N., & Narwaria, M. (2023). Establishing three layer architecture to improve interoperability in Medicare using smart and strategic API led integration. *SoftwareX*, 22, 101376.
20. Nugraha, Y., Masnita, Y., & Kurniawati, K. (2022). Peran responsiveness chatbot artificial intelligence dalam membentuk customer satisfaction. *Jurnal Manajemen Dan Bisnis Sriwijaya*, 20(3), 143-158. <https://doi.org/10.29259/jmbs.v20i3.18528>

21. Nugroho, A. W., & Utama, A. A. G. S. (2025). Business intelligence systems and their impact on organizational decision-making and performance outcomes: Literature review. *Owner: Riset dan Jurnal Akuntansi*, 9(2), 1269-1284. <https://doi.org/10.33395/owner.v9i2.2646>
22. Plotnikova, V., Dumas, M., Nolte, A., & Milani, F. (2023). Designing a data mining process for the financial services domain. *Journal of Business Analytics*, 6(2), 140-166. 140–166. <https://doi.org/10.1080/2573234X.2022.2088412>
23. Sousa, B., Guerreiro, G. R., & Espadinha-Cruz, P. (2026). From fragmented data to business intelligence: a data-centric CRISP-DM framework for delegated insurance. *Information Systems and e-Business Management*, 1-28. <https://doi.org/10.1007/s10257-026-00729-6>
24. Staudinger, S., Schuetz, C. G., Schrefl, M., & Neuböck, T. (2025). Knowledge graph support for descriptive business analytics. *DECISION*, 52(3), 285-306. <https://doi.org/10.1007/s40622-025-00432-4>
25. Xu, Z., Liu, S., Huang, W., Shi, J., Xu, F., Tang, X., & Lu, H. (2025). Deepseek: implications for data science and management in the AI era. *Data Science and Management*. <https://doi.org/10.1016/j.dsm.2025.09.001>
26. Zehle, T., Schlager, M., Heiß, T., & Feurer, M. (2025). CAPO: Cost-aware prompt optimization. *arXiv preprint arXiv:2504.16005*. <https://proceedings.mlr.press/v293/zehle25a.html>