Vrije Universiteit Amsterdam

Universiteit van Amsterdam

Master Thesis

Operational Analysis of OpenAI Services Using Self-Reported Outages and Incidents

Author: Qingxian Lu (2779768)

1st supervisor:	Alexandru Iosup
daily supervisor:	Xiaoyu Chu
2nd reader:	supervisor name

A thesis submitted in fulfillment of the requirements for the joint UvA-VU Master of Science degree in Computer Science

August 23, 2024

Abstract

Large Language Model (LLM) based services such as those provided by OpenAI have become popular in recent years. However, there is a lack of operational data analysis on the reliability of such services. This research aims to analyze the operational characteristics and patterns of OpenAI's services using their publicly reported incident and outage data. We developed a data pipeline to collect and process relevant data from OpenAI's status page. We then performed a set of failure analyses. Additionally, we applied the Latent Dirichlet Allocation (LDA) technique to extract underlying themes from the textual incident reports. The failure analysis revealed that OpenAI has an efficient failure response mechanism, with most issues resolved quickly. Clear temporal failure patterns were identified, which suggests a correlation with user activity. The LDA model extracted three distinct topics, describing the trending issues OpenAI's system faced. This study provides insights into the failure characteristics of a LLM-based AI service. The findings highlight opportunities for OpenAI to enhance its system reliability through improved resource allocation, and failure mitigation strategies tailored to specific services. The methodologies presented can be extended to analyze other LLM services.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Research Context

The last decades have seen machine learning evolve from statistical learning to deep learning, and to transformer-based Large Language Models (LLMs) [**?**]. LLMs have demonstrated remarkable capabilities in text-based Natural Language Processing (NLP) tasks [1, 2]. Moreover, LLMs have shown impressive results in text-to-media tasks, including image, audio, and video generation [3, 4, 5]. For commercial reasons, LLMs and their parameters are often not open-sourced. Instead, they are released as services, so-called Language-Model-as-a-Service (LMaaS) [6]. Users can access LLM-based services to solve various tasks via text prompts, paying per usage [7, 8].

Due to their exceptional capabilities, LLM-based services have gained immense popularity. OpenAI, a leading LLM-based service provider, is known for its chatbot ChatGPT and image generator DALL·E. The launch of ChatGPT marked a major breakthrough in artificial intelligence [9]. It sets a user adoption record with 100 million monthly active users within two months after launch[10]. These services now play a significant role in daily life, with many users relying on them for coding, writing, and problem-solving. Consequently, the reliability and sustainability of such services are important, as disruptions can severely impact user experience and cause financial losses.

This research focuses on OpenAI's services as a representative case of LLM-driven services. However, there is a lack of operational data analysis on such services. This gap exists for several reasons: (1) LLM is a state-of-art technique and only became popular in 2022 [11], so little attention has been paid to this problem; (2) There is a lack of datasets describing the operational status and failures of LLM service; (3) Previous studies focus

more on the internal ML or AI workload data, rather the high-level operational data that directly reveals the quality of service (QoS) and user experiences.

To bridge this gap, our research aims to develop methods for collecting operational data on LLM services and conducting various analyses. This method will provide insights into the operational characteristics of LLM-driven service. We begin by scraping service status data and incident reports from OpenAI's status page. We then propose a failure analysis approach to examine failure response efficiency and severity, and identify time patterns in OpenAI's system. To better understand system reliability, we investigate correlations in failure patterns across OpenAI's different services. Finally, we present a textual analysis using Latent Dirichlet Allocation (LDA) to extract trending keywords from incident reports, offering further insights into system issues.

1.2 Research Questions

Our research attempts to answer the main research question: **How to collect and analyze the operational data provided by OpenAI self-reported outages and incidents?** We break down this main research question into three sub-questions (RQ), which are as follows:

RQ1. How to design and implement an effective data pipeline for collecting and processing incident and outage data from the OpenAI status page?

The dynamic nature of the operational status webpage brings challenges to extracting data from it. It requires appropriate data collection techniques to capture dynamic web elements, ensuring data accuracy and completeness. Moreover, the website contains both real-time and historical data. A well-designed data pipeline is necessary to ensure scalability and re-productivity. By addressing these challenges and obtaining high-quality failure data, we can create a robust toolbox for collecting OpenAI's operational data and analyzing OpenAI's system reliability.

RQ2. What are the key characteristics and patterns of failures (including both incidents and outages) of OpenAI's services?

Identifying failure patterns is beneficial for understanding failures happened in AI services and improving failure mitigation strategies. However, it is challenging because the features of the incident and outage data are unknown. Appropriate analysis

and visualization methods are needed to characterize the data we collected. By overcoming these challenges, we can gain valuable insights into the nature of failures in LLM-based AI services.

RQ3. What are the main themes and trends in OpenAI's incident reports?

Understanding the trending themes in incident reports provides deeper insights into the problems faced by LLM services. The challenge lies in extracting distinct and meaningful themes from unstructured text data and interpreting these themes in the context of LLM service failures. By addressing these challenges through natural language processing techniques of the incident reports, we can uncover major failure types and inform strategies for improving LLM service's reliability.

1.3 Research Contributions

The research contributions (RC) of this study are as follows:

RC1. We designed and developed a robust data pipeline for collecting and processing historical incident and outage data from OpenAI's public status page. We implemented web scraping techniques to capture dynamic web content effectively. The raw data underwent transformation processes and resulted in two structured datasets: the Incident Data Product and the Outage Data Product. The datasets and software will be open-sourced on GitHub, which will ensure the reproducibility of this work and enable further research.

RC2. We proposed a methodology for failure analysis to uncover the characteristics and patterns of failures in OpenAI's systems. We investigated the distributions of failure durations and impact levels, revealing insights into the failure response mechanisms. Time series analysis techniques were applied to identify temporal patterns in failure occurrences. Additionally, we examined the correlations between service failures, uncovering potential shared dependencies between services. This comprehensive failure analysis provides insights for understanding the reliability of large-scale LLM systems.

RC3. We presented a clustering method for textual analysis to better understand the underlying themes and trends in OpenAI's incident reports. This Latent Dirichlet

Allocation (LDA) topic modeling technique allowed us to extract meaningful topics from the textual data in the incident reports, suggesting prevalent issues in the OpenAI's system.

1.4 Research Structure

This thesis is structured as follows: Chapter 2 provides background information on OpenAI services and key operational terms. Chapter 3 describes the data pipeline and the methods used for failure and report analysis. Chapter 4 presents and discusses the results of the failure analysis, including statistical distribution, temporal patterns, and inter-service correlations. Chapter 5 conducts and discusses the textual analysis of incident reports using LDA topic modeling. Chapter 6 addresses threats to the validity and limitations of the study. Chapter 7 gives a review of related work, and Chapter 8 provides a conclusion of this research.

2

Background

2.1 OpenAI Services

OpenAI provides four kinds of *services*:

(1) API. The OpenAI API service allows developers to access and use advanced LLM models provided by OpenAI through API keys. These models, such as GPT-3.5, can be used for multiple tasks. The API provides a simple interface for developers to use state-of-the-art LLMs without building or training their own models.

(2) ChatGPT. ChatGPT is a sibling model to InstructGPT, which is trained to follow instructions in a prompt and provide a detailed response. ChatGPT interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer follow-up questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.

(3) Labs. OpenAI labs provide DALL·E, which is an AI system developed by OpenAI that can create original, realistic images and art from a short text description. It can make realistic and context-aware edits, including inserting, removing, or retouching specific sections of an image from a natural language description. It can also take an image and make novel and creative variations of it inspired by the original.

(4) Playground. OpenAI playground is a place to explore resources, tutorials, API docs, and dynamic examples to get the most out of OpenAI's developer platform.

2.2 Operational Terms and Definitions

The failures that lie in OpenAI's service consist of incidents and outages. An Incident is an operational issue that possibly leads to a service outage. Once an incident happens, a

textual report of its failure-recovery process is disclosed to the public. An outage refers to a service status when the service is unavailable to users.

2.2.1 Outages and Outage-related Metrics

(1) Service status. There are five service statuses: (a) Operational, (b) Degraded Performance, (c) Under Maintenance, and two kinds of outages – (d) major outage and (e) partial outage. The status page primarily records and displays outages; degraded performance and maintenance are not reflected in the status timeline. According to the status page provider[1], the status page focuses on displaying unplanned downtime. While degraded performance (i.e., slow service) affects user experience, and maintenance is planned, neither is considered technically downtime and thus not included in the service status display. The service statuses are recorded separately in each service timeline on a daily basis. Notably, a service can experience different statuses within the same day, for example, undergoing both types of outages in a single day.

(2) Partial Outage and Major Outage. For a given day, partial outage minutes and total outage minutes represent the occurrence and duration of such outages within that day. In this research, a day with any recorded outage minutes is called an outage day. According to the status page provider's documentation, major outages affect 100% of the people that use a given service, while partial outages only affect a subset of those users. In terms of impact, partial outages are considered less severe than major outages. Specifically, partial outages are discounted to be only 30% as impactful as major outages. This discount factor is applied uniformly and cannot be manually configured.

(3) Scaled Total Outage Minutes. The outage minutes reflect the outage impact and are used to display service status. OpenAI uses a set of colors from green (operational) to red (long-period outage) to display the daily service status to the users. The status color is completely determined by the scaled total outage minutes, which represents the sum of partial and major outage minutes applied with the 30% discount factor. As scaled total outage minutes increase, the color shifts towards red, indicating a worse service status.
The rule and formula to calculate the total outage minutes is:

$$T = M + P$$

The rule and formula to calculate the scaled total outage minutes is:

$$Ts = M + (P \times 0.3)$$

[1]Atlassian Support: `https://support.atlassian.com/statuspage/docs/display-historical-uptime-of-components/`

T is the total outage minutes, indicating the sum of outage minutes per day. Ts is the scaled outage minutes. Compared to total outage minutes, scaled outage minutes provide a better reflection of the daily outage impact. M means major outage minutes, and P means partial outage minutes in a day.

2.2.2 Incidents and Incident Related Metrics

(1) Incident status. Throughout the incident recovery process, from the observation of an issue to the recovery from it, there are 5 incident statuses. (a)Investigating, (b)Identified, (c)Monitoring, (d)Resolved, and (e)Postmortem. "Investigating" means symptoms of an issue are observed but the root cause is not found. Once the root cause is found and a fix is being worked on, the status changes to "Identified". If the fix successfully resolves the issue, the incident moves to "Monitoring". After a period of monitoring, if everything functions properly and symptoms subside, the root cause is considered eliminated and systems are considered back to 100% performance. In this case, the incident is marked as 'Resolved'. "Postmortem" is a special status, existing only for significant incidents. For some incidents, a while after the resolution, OpenAI may publish a postmortem update to explain in detail what went wrong and what measures will be taken to prevent similar issues in the future.

(2) Affected services. Not like outages, which reflect individual service status and are recorded separately, incidents are events that can affect multiple services simultaneously. If an incident occurs in the system on a given day and is known to affect a specific service, and if that service experiences any outage on the same day, the incident will be linked to this service in the status timeline as a related incident. The link means this incident contributed to the (outage) status for this service on that specific day.

(3) Incident duration. Since an incident doesn't necessarily go through all 5 statuses, with "Resolved" being the only mandatory ending mark, the incident duration in this resarch is measured from the discovery or initial update status of an incident to its resolution. This duration represents how long it takes OpenAI to recover from an issue.

(4) Incident impact. Incident impact illustrates the severity of an incident. Incidents are categorized into 5 types based on their impact levels, displayed in different colors to reflect severity: None (black), Minor (yellow), Major (orange), Critical (red), and Maintenance (black). Incident impact is decided by service status. If an incident is linked to a service timeline, the impact-level increases as the service status worsens. Specifically, if the service is "Operational", the incident is impact-none, meaning issues occurred without causing downtime; if the service has a "Partial Outage", the incident is impact-minor, indicating

breakage affecting some users; if the service has a "Major Outage", the incident is impact-major, signifying significant breakage affecting all users. For incidents affecting multiple services, the impact level is escalated. If all affected services have "Partial Outage", the impact level escalates from minor to major. Similarly, if all affected services have "Major Outage", the impact level becomes critical, which is the worst case. Same as the logic for displaying service status, degraded performance and maintenance are not considered planned downtime. In these cases, the incident impact level would be impact-minor and impact-maintenance, respectively.

3

Methods for Failure and Textual Analysis of OpenAI services

3.1 Data Processing Pipeline

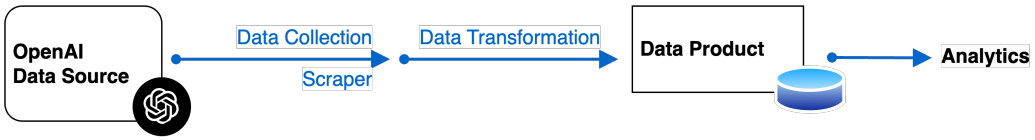


Figure 3.1: Data Pipeline Overview

The data processing pipeline consists of 2 main stages: data collection and data transformation, as illustrated in Figure 3.1. It begins with automated data collection from OpenAI's official website and then proceeds to transform the raw data. Finally, the pipeline produces structured datasets for operational analysis of OpenAI's service performance.

3.1.1 Data Sources

The data source for this research is OpenAI's official status page [1]. This platform serves as a monitoring system for the operational status of OpenAI services, providing both real-time and historical data on service performance, outages, and incidents. OpenAI publishes two primary categories of historical data: incidents and outages. These data offer detailed insights into past service disruptions, their recovery processes, and their impact.

[1] OpenAI Status: `https://status.openai.com`

Historical Incidents [1] provides detailed records of past issues, organized chronologically by month. Each incident report includes a title, a timeline of incident status updates with detailed descriptions, and the services affected.

Historical Outages [2] are presented in a calendar format, with separate calendars for each OpenAI service (API, ChatGPT, Labs, and Playground). Each history page displays a 3-month calendar view. By hovering over the calendar, one can reveal detailed information about outages, including the occurrence and duration of partial and major outages, as well as any related incidents.

3.1.2 Data Collection

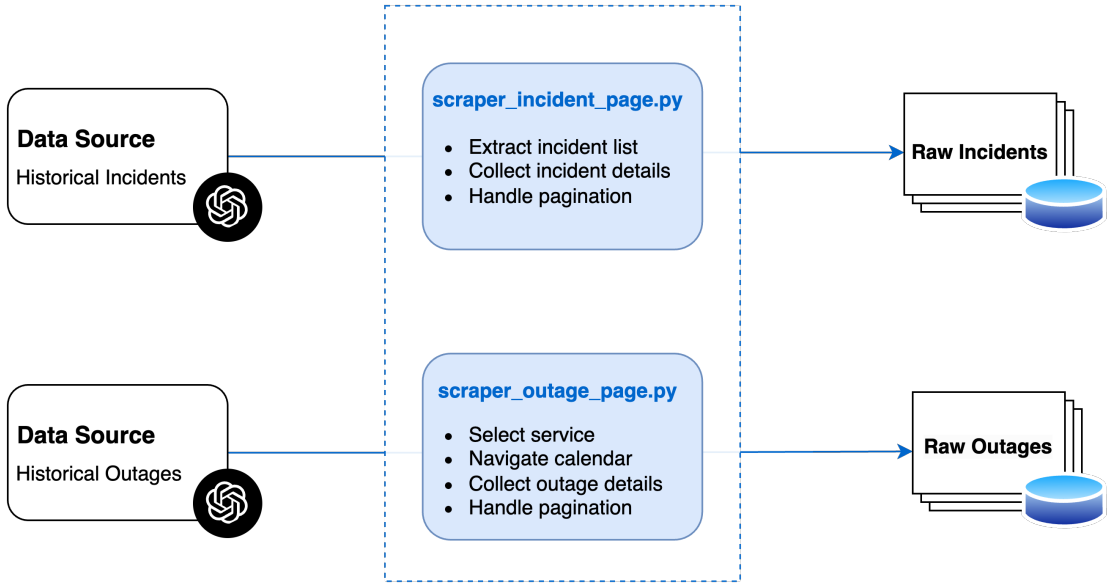


Figure 3.2: Data Collection

Given the dynamic nature of the OpenAI status page, we implemented an automated data collection method based on Selenium WebDriver [3], a robust tool for browser automation. Selenium was chosen over faster HTTP request libraries due to its capabilities in handling dynamic web content. Selenium WebDriver allows native browser automation by simulating real-user interactions. These features ensure the proper load and extraction of page elements, ensuring a comprehensive and accurate data capture.

[1] Incident History: `https://status.openai.com/history`
[2] Uptime History: `https://status.openai.com/uptime`
[3] Selenium WebDriver: `https://www.selenium.dev/documentation/webdriver/`

As shown in Figure 3.2, two Python scripts were developed to automate the collection of historical incident and outage data.

The historical incident data collection process involves three key steps. First, it extracts a list of incidents from the incident history page, organized by month. Second, for each incident, it simulates clicking on the incident title to open a new tab with detailed information. It then collects specific details including the incident title, unique identifier, impact level (represented by color), detailed updates (status, content, and timestamp), and affected services. Finally, the script handles pagination, navigating through multiple pages backward until reaching the earliest recorded incident. Throughout this process, it stores the collected data in a pandas DataFrame and exports it to a CSV file, with the filename reflecting the date range of the data collected from each page.

The historical outage data collection allows users to select different OpenAI services (API, ChatGPT, Labs, and Playground) via command-line arguments. For each day on the service calendar, the script simulates mouse hover actions to extract detailed tooltip information, including the date, outage details (type and duration), service status color, and related incidents. Similar to the incident collection, the outage collection script handles pagination to navigate through all available data. Upon completion, the collected outage data is exported to a CSV file, organized by service and execution date.

To ensure the reliability of the data collection process, both scripts implement exception handling mechanisms. This approach addresses potential issues such as network problems, stale elements, or unexpected page layouts. Additionally, the scripts include retry mechanisms and detailed logging functionality to track collection progress and record any encountered errors.

3.1.3 Data Transformation

The data transformation pipeline, illustrated in Figure 3.3, standardizes data format and extracts meaningful features. This process was implemented using Python, primarily leveraging the pandas library for efficient data processing.

Data transformation begins with loading historical data from CSV files and concatenating it into a single DataFrame for unified processing. The next step is initial preprocessing, including removing duplicate entries that have been introduced during the data collection phase. As part of this process, nested columns are parsed for easier manipulation. For example, the "Updates" column in the incident dataset and the "Outages" column in the outage dataset are converted to JSON format. Feature engineering is then tailored to the specific characteristics of incident and outage datasets.

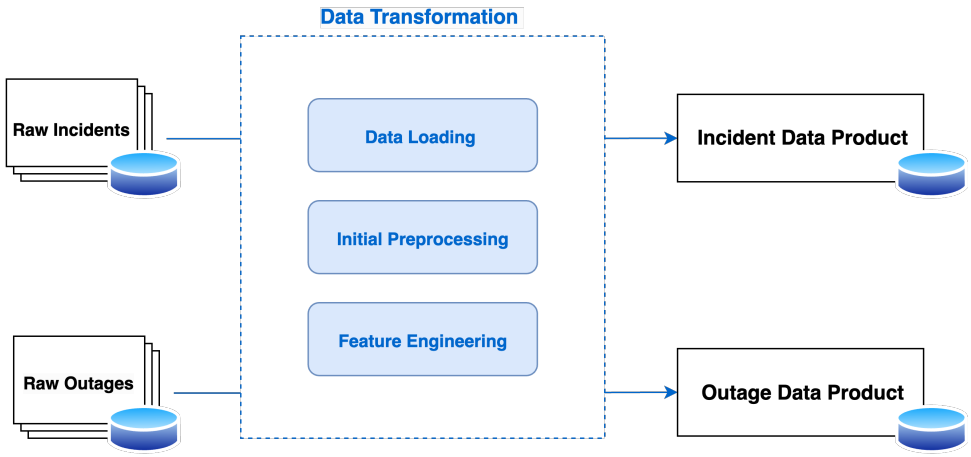


Figure 3.3: Data Transformation Pipeline

Several key features are extracted from the raw incident data including "incident_id", "incident_impact_level", and service flags. The textual impact levels ('impact-none', 'impact-minor', 'impact-major', 'impact-critical', 'impact-maintenance') are encoded to numerical values (0-4) for quantitative analysis. The service flags (0 or 1) identify which specific OpenAI services were affected by each incident. To describe the incident life cycle, incident status-related features are extracted from the "Updates" list. For each status ("Investigating", "Identified", "Monitoring", "Resolved", and "Postmortem"), we extract three types of features: "flag", "description", and "timestamp". The "flag" is a binary indicator showing whether this status occurred in the incident report, and "timestamp" indicates when the status was updated. Additionally, "incident_duration" is calculated by subtracting the timestamp of the first status from the "resolved" status timestamp, representing the total incident recovery time.

The feature engineering for the outage dataset aims to create outage descriptors for each service on a daily basis. Standardized features for partial and major outages are developed: a binary "flag" indicating the occurrence of that specific type of outage, and its corresponding duration in minutes. "Total_outage_minutes" and "scaled_total_outage_minutes" are calculated as the sum and weighted sum of outage minutes. A binary "outage_flag" is generated to indicate whether a day is an ouatge day with any outage minutes.

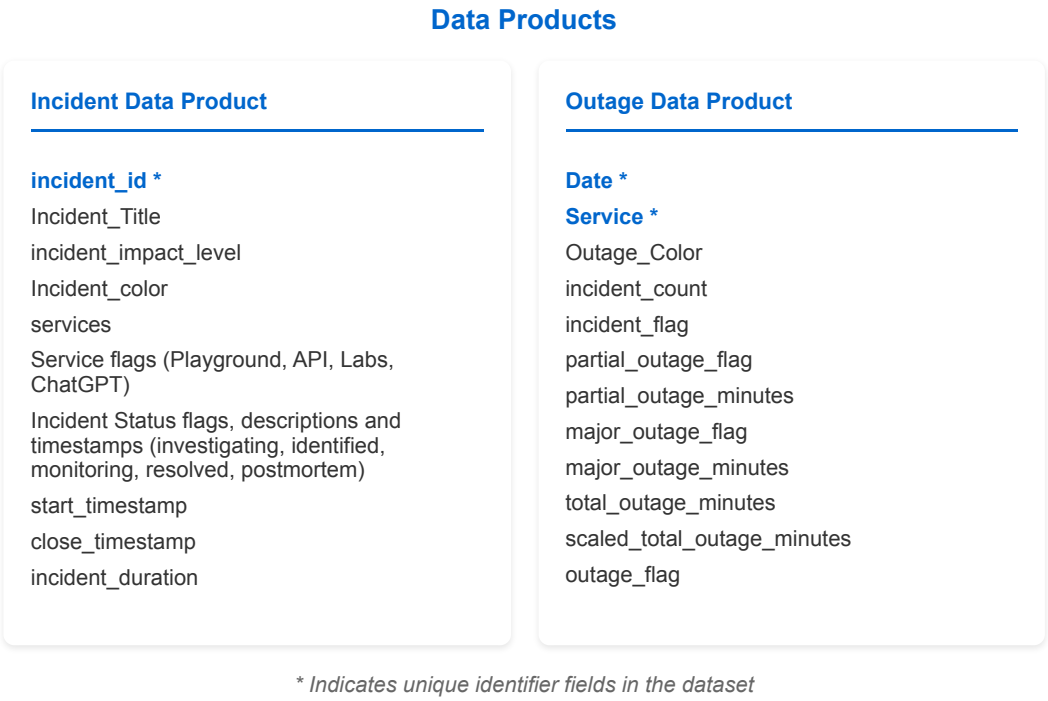


Figure 3.4: Data Products

3.1.4 Data Product

The data pipeline processes raw data collected from OpenAI's status page and produces two structured datasets: the Incident Data Product and the Outage Data Product, as illustrated in Figure 3.4. These datasets are archived following the structure shown in Figure 3.5. The initial data collection outputs are stored in the /raw folder, with filenames indicating the date range. The data transformation outputs, which also serve as the final data products of the pipeline, are stored in the staging folder /stg.

Data overview

The final data produced from the data pipeline covers the period from February 2021 to May 2024. The datasets provide comprehensive information on service disruptions and incident reports across multiple OpenAI services, including API, ChatGPT, Labs, and Playground.

The incident dataset comprises 322 unique incidents over approximately 40 months. Table 3.1 provides an overview of this dataset. The first row in the table describes general statistics for all incidents. While each incident progresses through various stages, not all

Data Archive Structure

Raw and Staging Data

```
data
├── raw
│   ├── incident
│   │   └── incident_history_202403_202405.csv
│   └── uptime
│       ├── api
│       │   └── 2024-05-29/uptime_history.csv
│       ├── chatgpt
│       ├── labs
│       └── playground
└── stg
    ├── incident
    │   └── 2024-07-06/incident_stages.csv
    └── outage
        └── 2024-07-12/outage_history.csv
```

Figure 3.5: Data Archive Structure

Service	Ealiest	Latest	Months	# of Failure-Recovery Status					# of Impact Levels					# of Incident
				Investigating	Identified	Monitoring	Resolved	Postmortem	Critical	Major	Minor	None	Maintenance	
ALL	2021-02-09	2024-05-24	40	229	132	199	322	25	42	107	130	42	1	322
API	2021-03-01	2024-05-23	39	155	90	135	207	17	21	70	94	22	0	207
ChatGPT	2023-02-14	2024-05-24	16	94	52	86	116	10	24	37	45	9	1	116
Labs	2023-02-21	2024-05-07	15	29	12	24	31	4	6	16	6	3	0	31
Playground	2021-04-01	2024-05-07	38	24	14	22	30	2	7	12	7	4	0	30

Table 3.1: Incident Dataset Grouped by Services Overview.

incidents necessarily go through every stage. Notably, 25 incidents (7.8%) have associated postmortem reports, indicating in-depth analysis for significant failures.

The incident reports data indicates which services were affected by each incident. With only 26 incidents (8.0%) lacking information on affected services, the incident data can be effectively grouped by service. It's important to note that a single incident may affect multiple services simultaneously, which is why the sum of incidents across different services exceeds the total number of incidents. The service-specific statistics suggest that OpenAI began reporting incidents for ChatGPT and Labs only from 2023, while incident reporting for API and Playground started earlier in 2021. The API presents as an important service for OpenAI, experiencing the highest number of incidents (207) over 39 months. ChatGPT, despite its later launch, recorded 116 incidents over 16 months and notably has the highest number of impact-critical incidents (24) in a relatively short time. Labs and Playground services experienced fewer incidents, with 31 and 30 respectively.

Table 3.2 provides an overview of the outages. The outage dataset records the daily status of each OpenAI service, showing a total of 119 outage days over 40 months. Notably, the

Service	Start	End	Months	Outage Counts		Outage Time [min]		# of Outage Days	# of Related Incidents
				Major	Partial	Total	Scaled		
ALL	2021-02-11	2024-05-29	40	-	-	-	-	119	296
API	2021-02-11	2024-05-29	40	24	67	7,113	3,049	82	217
ChatGPT	2023-02-14	2024-05-29	16	24	34	4,174	2,145	51	127
Labs	2023-02-21	2024-05-29	16	12	14	2,786	1713	22	33
Playground	2021-03-31	2024-05-29	39	11	11	1,489	949	22	33

Table 3.2: Outage Dataset Overview.

outage dataset contains fewer records than the incident dataset.

The API service experienced the highest number of outages (82) over 40 months, including 24 major and 67 partial outages. ChatGPT, despite reporting status for only 16 months, recorded 51 outages and has the same number of major outages as API. Both Labs and Playground services experienced 22 outages each, which could suggest either lower usage or more stable systems for these services.

3.2 Methods for Failure Analysis

The failure analysis is conducted from several perspectives using the processed incident and outage datasets. Statistical distribution analysis is performed to characterize failure recovery efficiency and severity. Box plots and violin plots are used to visualize the distributions of failure duration and impact levels. Box plots show the mean, median, quartiles, and potential outliers, while violin plots display the full distribution, including any multimodal patterns that might not be apparent in box plots. Temporal analysis aims to identify time patterns in OpenAI's system failures. A failure frequency trend analysis and a temporal distribution analysis are conducted. The frequency trend is visualized using histogram plots with a kernel density estimate line, which allows for the observation of both discrete counts and overall trends. The temporal distributions are examined on hourly, weekly, and monthly granularities using stacked bar charts. Inter-service correlation analysis investigates potential shared failure dependencies between OpenAI services. Co-occurrence matrices and correlation coefficient heatmaps are generated to reveal relationships of failures between service pairs. All the data for the failure analysis is prepared using pandas visualized using matplotlib.

3.3 Methods for Report Analysis

Latent Dirichlet Allocation (LDA) topic modeling is applied to the textual incident reports to extract meaningful failure themes. The reports undergo preprocessing steps including cleaning, tokenization, lemmatization, and stop-word removal. Hyperparameter tuning is conducted using a grid search method to optimize the LDA model. We utilize pandas and nltk libraries for textual data preprocessing and gensim for LDA model training and evaluation. The modeling results are visualized using the pyLDAvis library for topics overview and interpretation. The topic distributions are visualized using matplotlib and seaborn, with a stacked bar plot for the topic distribution per incident report and heatmaps for the topic distribution across services.

4

Failure Analysis of OpenAI Outages and Incidents

4.1 Overview and Approaches for Failure Analysis

To investigate the characteristics and patterns of OpenAI failures, we performed a set of failure analysis on the outage and incident data: statistical distribution analysis, temporal analysis, inter-service correlation analysis.

By analyzing the statistical distribution of failure duration and failure impact, We characterize how OpenAI reacts to its service failures and how bad the failures usually are. Failure duration stands for how long OpenAI typically takes to resolve incidents or recover from outages. The distribution of duration values explains the efficiency of failure response processes. Meanwhile failure impact represents the severity of OpenAI services failures.

Temporal data represents sequential observations over time, allowing for the examination of data trends, cyclical patterns. In the context of OpenAI's failure data analysis, temporal analysis aims to identify time patterns in OpenAI's system, suggesting when the system is more prone to failures. Sorting all data chronologically, we first analyze the failure frequency trends to showcase how the number of incidents and outages change over time. We then examine the temporal distribution of failures to investigate the peak times for incidents and outages. The temporal distribution analysis is conducted hourly, weekly, and monthly. Taking the weekly analysis as an example, we identify the corresponding day of the week for each data point. The total number of incidents or outages for each day of the week is then calculated.

The inter-service correlation analysis aims to identify shared failure dependencies between OpenAI services. The core of this analysis revolves around two key matrices, a

co-occurrence matrix and a correlation Matrix. The co-occurrence matrix illustrates how often a service pair fails concurrently, meanwhile, the correlation matrix displays the statistical correlations, Pearson Correlation Coefficients (PCC), between the failure patterns of service pairs. PCC values range between -1 and 1. The absolute PCC value closer to 1 indicates a stronger correlation. A strong positive correlation suggests that services tend to fail together or operate together. A strong negative correlation indicates that when one service fails, the other is less likely to fail. Values close to 0 indicate that the failure patterns of the service pair likely do not have a clear linear relationship. Those two matrices reveal the relationships between service failures and are visualized using heatmaps.

4.2 Statistical Distribution Analysis of Failures

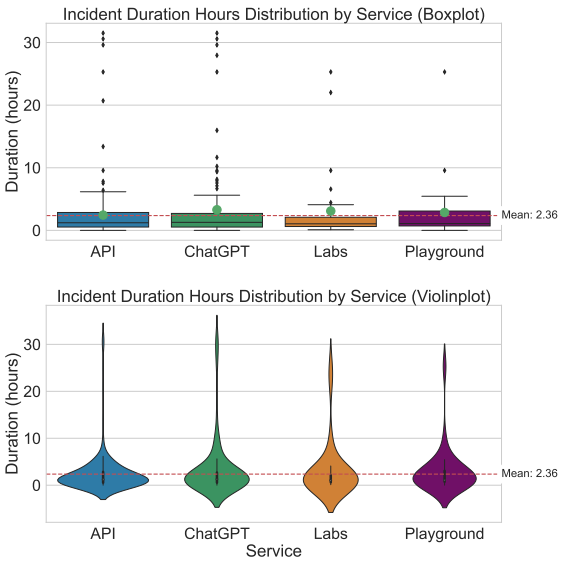
4.2.1 Failure Duration Distribution

O-1: *Failure durations across all services show a right-skewed distribution, with a small number of complex cases significantly inflating average recovery times.*
O-2: *Outage durations are generally shorter than incident resolution times.*

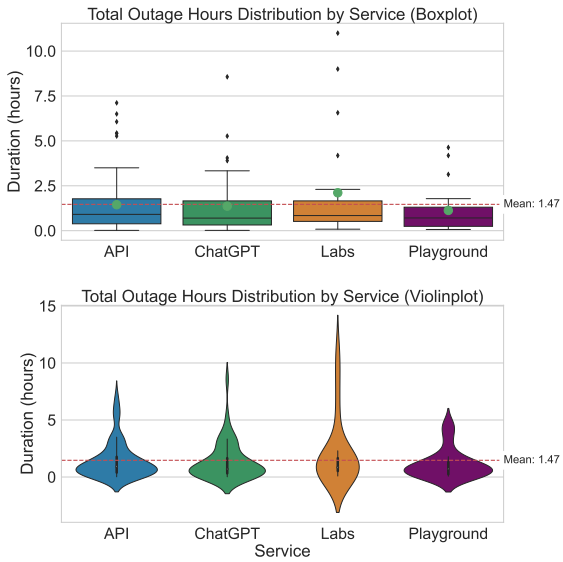
This section analyzes the distribution characteristics of failure duration, including incident duration and outage duration. For incidents, we analyze the "incident_duration" feature, defined as the elapsed time from the discovery or initial report of an incident to its resolution, representing the recovery time for each incident. For outages, we examine three features: "partial_outage_minutes", "major_outage_minutes", and "total_outage_minutes". "partial_outage_minutes" represents the duration during which a subset of users experiences service unavailability. "major_outage_minutes" represents the duration of service disruption affecting almost all users. "total_outage_minutes" is the sum of partial and major outage duration. For standardization, all duration values were converted to hours. The duration analysis provides insights into the efficiency of failure response processes and implies the complexity of issues faced.

The boxplots and violin plots in Figure 4.1 illustrate the distribution of failure duration for both incidents and outages. The x-axis represents the categories, which are different service types, while the y-axis shows the failure duration in hours. A red dashed line across both plots indicates the overall mean duration hours, providing a reference point for comparison across services.

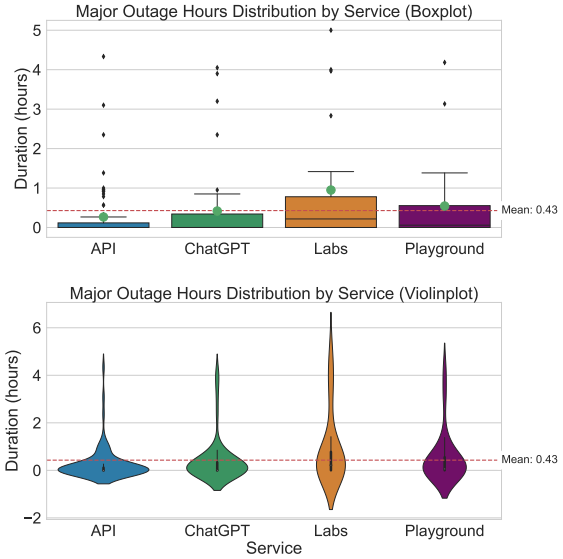
A consistent pattern observed across all services is the right-skewed distribution of failure durations. This skewness is further confirmed by the mean duration being larger than the median across all services, especially for incidents. This suggests that while most issues



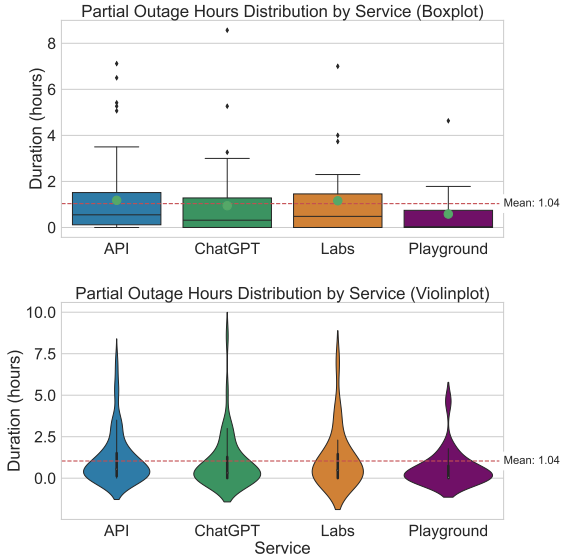
(a) Incident Duration Distribution



(b) Total Outage Hour Distribution



(c) Major Outage Hour Distribution



(d) Partial Outage Hour Distribution

Figure 4.1: Incident and Outage Duration Distribution by Service

are resolved quickly, a small number of outlier cases significantly inflate the average failure duration, indicating the occurrence of complex, time-consuming issues (**O-1**). As shown by the interquartile boxes, the incident duration is predominantly concentrated in the 0-3 hour range for all services. The total outage hours are concentrated in 0-2 hours. The minimum outage duration across services is low (0.3 to 1.5 minutes), suggesting that all services can

experience minor, quickly-resolved issues. This observation indicates that OpenAI has a sensitive failure response mechanism for quickly detecting and addressing minor issues.

Comparing the distributions of incident and outage duration, it's worth noting that outage durations are generally shorter than the incident resolution times recorded in incident reports. (**O-2**) This discrepancy might arise because service recovery often precedes the closure of an incident report. Except for resolving the issue itself, additional time is typically spent on monitoring the fix and learning from the failure. Consequently, the time to close an incident report is often significantly longer than the actual outage duration.

ChatGPT exhibits the widest distribution and the highest mean incident duration of 3.30 hours, significantly above the overall average. This service stands out with a broad range of outliers, suggesting that it encounters the most varied and potentially complex incidents. Interestingly, ChatGPT's outage performance is more moderate, with a mean total outage duration of 1.36 hours, balancing between partial (0.95 hours) and major (0.42 hours) outages. This discrepancy might indicate that ChatGPT faces complex issues requiring extensive investigation and monitoring, however, OpenAI has developed effective strategies for quick service restoration.

The API service demonstrates a wide distribution but with fewer outliers in incident duration distribution. Its mean duration of 2.45 hours is relatively low considering its widespread, implying that while complex incidents do occur, they are prone to be resolved more quickly than those affecting ChatGPT. For outages, API's mean total outage duration (1.45 hours) is close to the overall average (1.47 hours), with the highest mean for partial outages (1.18 hours) but the lowest for major outages (0.25 hours). This pattern suggests that the API service experiences longer minor disruptions but suffers less from severe, widespread issues. The low major outage hours indicate either less complexity or effective mitigation strategies for severe issues.

Labs and Playground, while having fewer incidents and fewer recorded outage days, present interesting contrasts. Labs shows the second-highest mean incident duration of 3.09 hours, influenced by a few significant outliers, including two incidents lasting over 20 hours. It consistently exhibits the highest mean durations across all outage metrics (total: 2.12 hours, major: 0.95 hours, partial: 1.16 hours), with the widest distributions and most extreme outliers. This pattern indicates that the Labs service exhibits the highest volatility in terms of recovery time among OpenAI's services. Certain issues within Labs pose exceptional challenges in diagnosis and resolution. Playground demonstrates an incident duration distribution with fewer outliers and a relatively lower mean incident duration of 2.86 hours. It also shows the lowest mean outage durations for total (1.13 hours) and partial

(0.58 hours) outages. This suggests that Playground might have less complex features or small usage workload.

4.2.2 Failure Impact Distribution

O-3: *Minor impacts, which accounts for 40.37%, are the most common type of incident across all services.*
O-4: *API and ChatGPT share a pattern where the number of incidents decreases as the impact level increases.*

This section conducts a comparative distribution analysis of the incident impact-levels across all OpenAI services. The categorical "incident_impact_level" feature is assesed. This analysis provides insights into the frequency of different types of incidents in terms of servity. It also helps in understanding the overall user experience and the robustness of OpenAI's systems. By the definition of "impact-maintenance", the single "incident" case in this category represents a planned system maintenance rather than an unexpected failure. We exclude it from the comparative analysis.

Incident Impact-level	Count	Percentage (%)
impact-none	42	13.04
impact-minor	130	40.37
impact-major	107	33.23
impact-critical	42	13.04
impact-maintenance	1	0.31

Table 4.1: Incident Impact-level Overall Description

The overall statistics in Table 4.1 reveal that across all services, minor impacts constitute the largest category, accounting for 40.37% of all incidents (**O-3**), followed by major impacts at 33.23%. The predominance of non-critical impact incidents suggests that OpenAI's systems are generally robust. Although they frequently encounter issues, not all users of all services are affected simultaneously. There are still parts of the system functioning during these incidents.

Critical impact incidents are less frequent (13.04%). It still represent a significant concern as it represents the most severe situations where all users experience outages for multiple services. The presence of these critical incidents might indicate interconnected dependencies between service failures that, when affected, can lead to widespread disruptions. There are also similar proportion of none impact incident. These could represent

issues that were successfully mitigated before causing any outages.

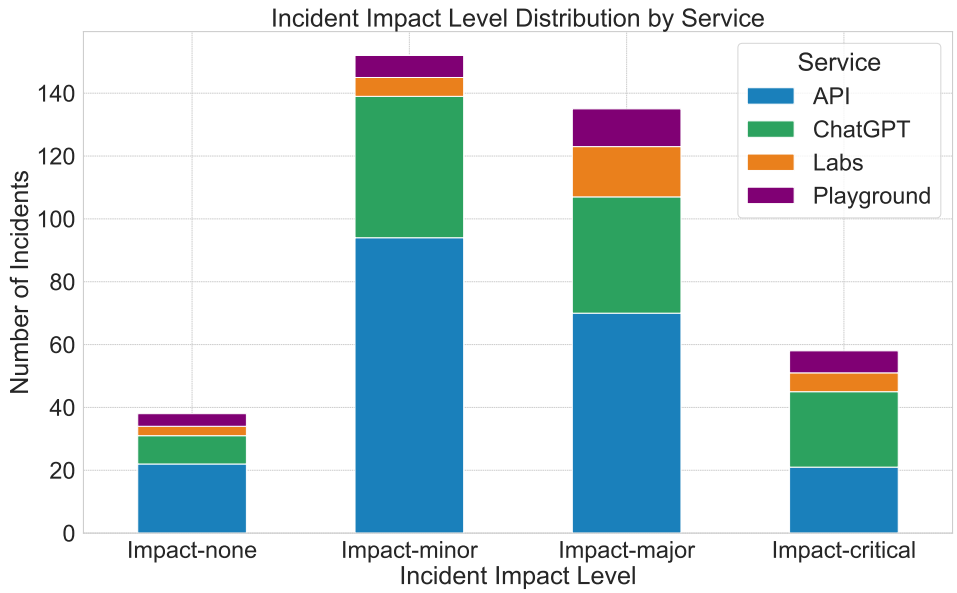


Figure 4.2: Incident Impact-level Distribution

As shown in Figure 4.2, the pattern found in API ChatGPT differs from that in Labs Playground.

For API ChatGPT, they have the least number of impact-none incidents. Excluding incidents with no impact, as the impact level increases, the number of incidents decreases (**O-4**). More specifically, the API service demonstrates the highest number of incidents across all impact levels, as we already observed in the dataset overview. It shows a clear predominance of minor impacts (94 incidents), followed by major impacts (70 incidents) and critical impacts (21 incidents). This distribution suggests that while the API frequently encounters issues, a significant portion of these are of lower severity. ChatGPT exhibits a similar pattern to the API but with a more balanced distribution between minor (45) and major (37) impacts. Notably, ChatGPT has the highest number of critical impacts (24) among all services. By the definition of critical impacts, this suggests that when a major outage happens in ChatGPT, it is likely there is another service suffering from a major outage as well.

Unlike API and ChatGPT, we observed more impact-major incidents than impact-minor incidents for both Labs and Playground. Labs has a higher proportion of major impacts (16) compared to its minor impacts (6), and Playground has more major impacts (12) compared to minor impacts (7), indicating that when issues do occur, they tend to be more

severe with more users affected. The different pattern observed in Labs and Playground could be due to less usage, resulting more unpredictability in incident impact.

4.2.3 Failure Duration and Failure Impact

O-5: *Incident duration increases with impact severity, from no impact to major impact.*
O-6: *Incidents with critical impact have an average resolution time of 1.9 hours, which is quicker than incidents with less severe impacts. Both impact-minor and impact-major incidents take an average of over 2.5 hours to resolve.*

This section aims to determine if incidents with higher severity require longer time to resolve. We group the incident dataset by impact level and then visualize the duration distribution for each level. This analysis helps in investigating the potential correlation between incident impact levels and its resolution times, and providing suggestions to improve incident management processes.

Figure 4.3 offers a nuanced view of incident duration distribution across various impact categories. A gradient color scheme, ranging from green to red, is employed to represent the severity of failures.

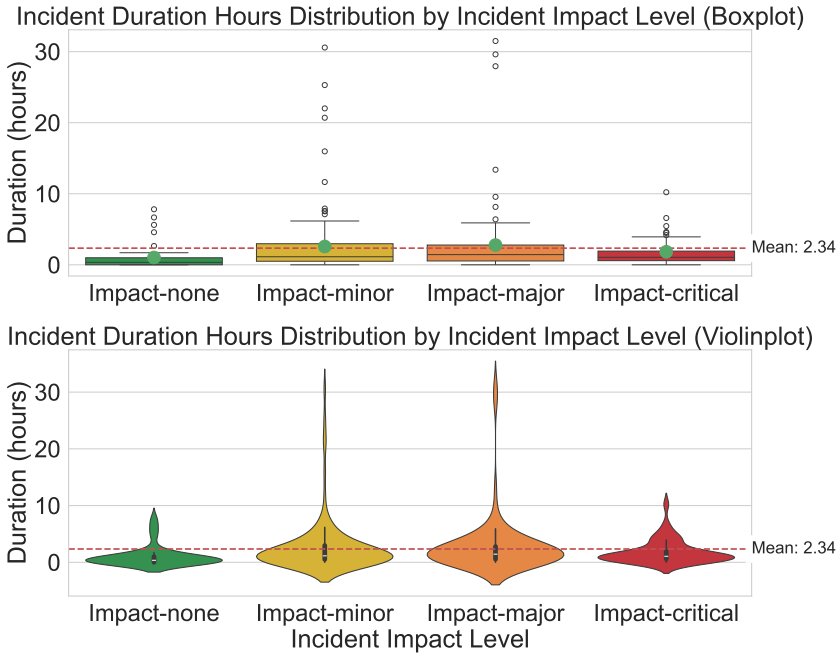


Figure 4.3: Incident Duration Distribution by Impact-level

Incidents categorized as Impact-none show the shortest mean duration of 1.03 hours, with a compact distribution and few outliers. This suggests that incidents causing no

outages are typically straightforward and quickly resolved. As we move to higher impact levels, we observe a general trend of increasing mean duration: Impact-minor incidents 2.57 hours, while Impact-major incidents extend to 2.78 hours(**O-5**). This pattern aligns with the expectation that higher impact incidents, which cause more significant service disruptions, are typically more complex and time-consuming to resolve.

Interestingly, Impact-critical incidents break this pattern, presenting a lower mean duration of 1.90 hours compared to lower impact categories (**O-6**). This unexpected pattern suggests that OpenAI might have implemented specialized rapid response mechanism for critical failures.

4.2.4 Implications

Excludes a small number of outlier instances, OpenAI appears to have effective mechanisms for quickly resolving most of the issues across all services. The shortest outage duration is less than 2 minutes. This indicate that OpenAI has a robust and sensitive failure response system. The complexity of ChatGPT failures indicate a need for continued investment in understanding its unique challenges. The API service shows a pattern of shorter major ones but longer minor outages, suggesting a focus on improving minor issue resolution.

The prevalence of non-critical impacts (minor and major) over critical ones indicates that OpenAI's systems are generally robustIt can often maintain partial functionality during incidents. This robustness is particularly evident in API service. ChatGPT, however, is the service most prone to impact-critical incidents, suggesting that additional resources and attention are needed when ChatGPT fails to prevent concurrent failures in other services.

The increasing incident resolution time from Impact-none to Impact-major suggests that the complexity of issues grow with impact severity. This highlights the need for scalable response mechanisms and resource allocation. The unique behavior of Impact-critical incidents underscores the effectiveness of rapid response mechanisms for high-priority issues. This could serve as a model for improving response times across other impact levels.

4.3 Temporal Analysis of Failures

This section examines the temporal patterns observed in both the incident and the outage dataset. For the incident dataset, we transform the data into a time series based on the "start_timestamp" feature, which indicates when each incident occurred. The incidents are then sorted chronologically. The outage dataset is broken down into service-specific subsets. Each subset is inherently time-series data, ordered by "date".

4.3.1 Overview of Failure Frequency Trends

O-7: *The incident frequency shows a general increase over time, while the overall outage frequency fluctuates significantly without a clear upward trend.*
O-8: *The failure frequency of ChatGPT shows a steady increase over time since its introduction.*

The analysis of the frequency trend of incidents and outages aims to illustrate how failure rates evolve. This analysis provides a comprehensive overview of OpenAI's significant failures over time.

Incident Frequency Trend

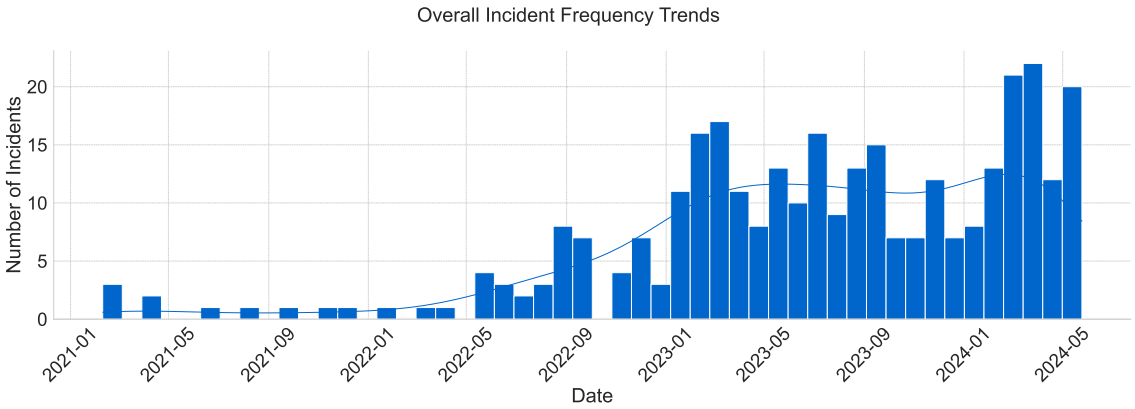


Figure 4.4: Overall Incident Trends

As illustrated in Figure 4.4, there is a general increase in the number of incidents over time (**O-7**), with notable fluctuations. A significant spike in incidents is observed around early 2023, followed by a slight decrease and then an increase towards another peak in 2024. The early 2023 spike coincides with the widespread adoption of ChatGPT, which was released in late 2022 (ref). The general upward trend could be attributed to be attributed to growth in user base or system complexity. As OpenAI's services gain popularity, the increased load on the system and the introduction of new features may lead to more incidents.

Figure 4.5 reveals distinct patterns for different services. API demonstrates a similar trend to the overall incident frequency. ChatGPT shows a steady increase in incidents from its introduction to a peak in May 2024 (**O-8**). This trend could be related to the continuous updates of the ChatGPT service, its growing user base, and its increased complexity of

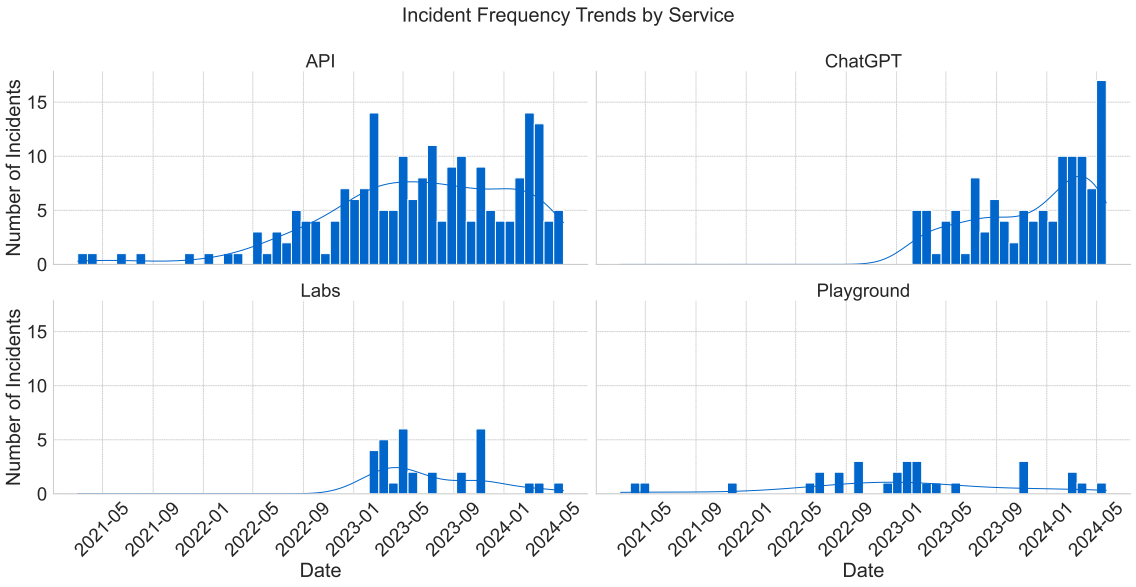


Figure 4.5: Incident Trends by Service

user interactions over time. Labs and Playground exhibit intermittent incidents with lower frequencies than API and ChatGPT. They show no clear upward or downward trend.

Outage Frequency Trend

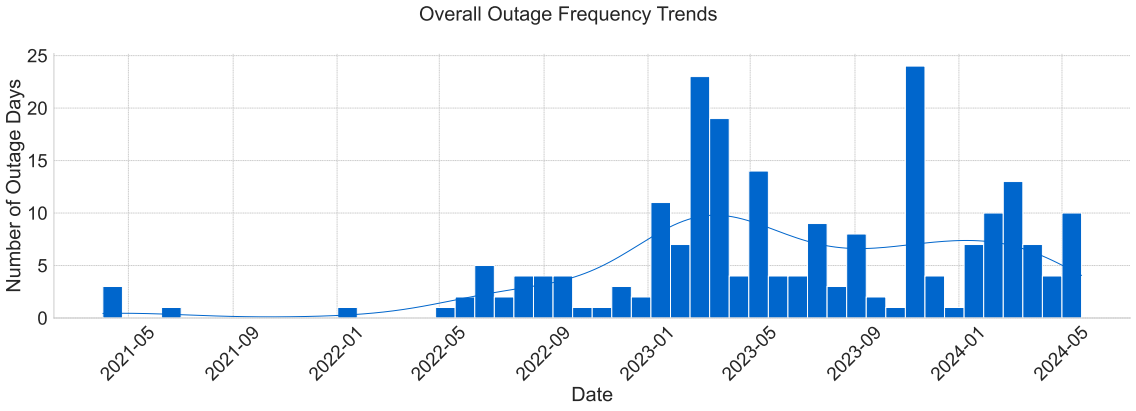


Figure 4.6: Overall Outage Trends

As shown in Figure 4.6, unlike the incident trend, outage days do not display a general increase over time. Instead, they fluctuate dramatically with two notable spikes (**O-7**).

One spike is in early 2023. This spike coincides with the surge in incident frequency. The other one is in the end of 2023. This peak is not mirrored in the incident trends.

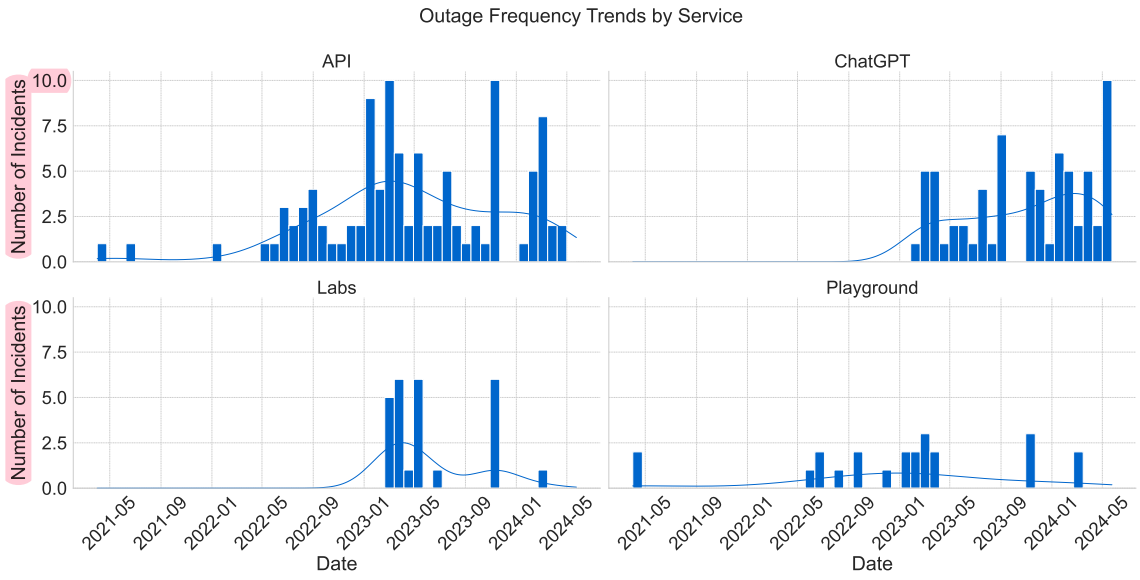


Figure 4.7: Outage Trends by Service

The outage trends by service in Figure 4.7 provide further insights. API shows a fluctuating trend with occasional spikes. The early 2023 spike aligns with peaks in the incident frequency trend, indicating periods where API issues frequently led to service downtime. Other services (ChatGPT, Labs, Playground) generally mirror their respective incident trends but with lower frequencies.

Across the services, the outage frequency trends broadly correlate with the incident frequency trends, but outage days are less frequent than incidents. The discrepancy between incident and outage trends could be explained by the different incident impact levels. The incidents with less severe impacts typically do not result in widespread service outages. The different incident and outage frequency trends suggest that many incidents are resolved without causing significant service outages.

4.3.2 Temporal Distribution of Failures

O-9: *Daily Pattern: Incident reaches highest levels during the morning, with a notable peak around noon.*
O-10: *Weekly Pattern: Failures show a clear mid-week peak, typically on Tuesday and Wednesday.*
O-11: *Seasonal Pattern: OpenAI experiences the highest failure rates in spring and the lowest in summer.*

The objective of this analysis is to identify the cyclic patterns of incidents and outages across different OpenAI services. By understanding when these failures occur, we can develop strategies to improve system reliability and performance. This temporal distribution analysis is conducted on several time granularities: hourly (for incidents only; 00:00 to 23:00), weekly (Monday to Sunday), and monthly (January to December). This multi-granular approach allows for a comprehensive understanding of when failures are more likely to occur.

Incident Temporal Distribution

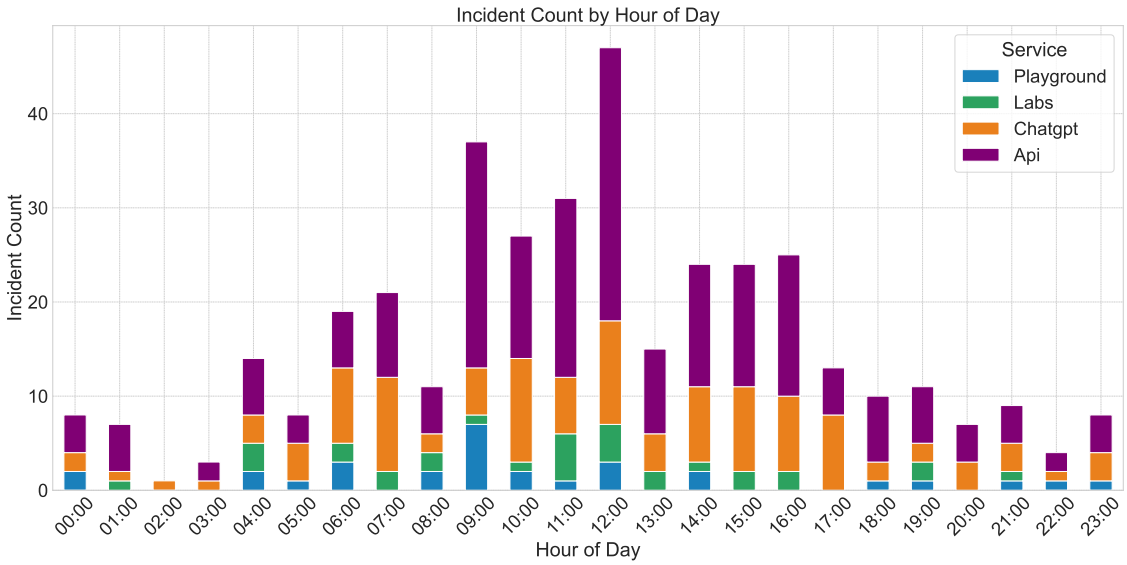
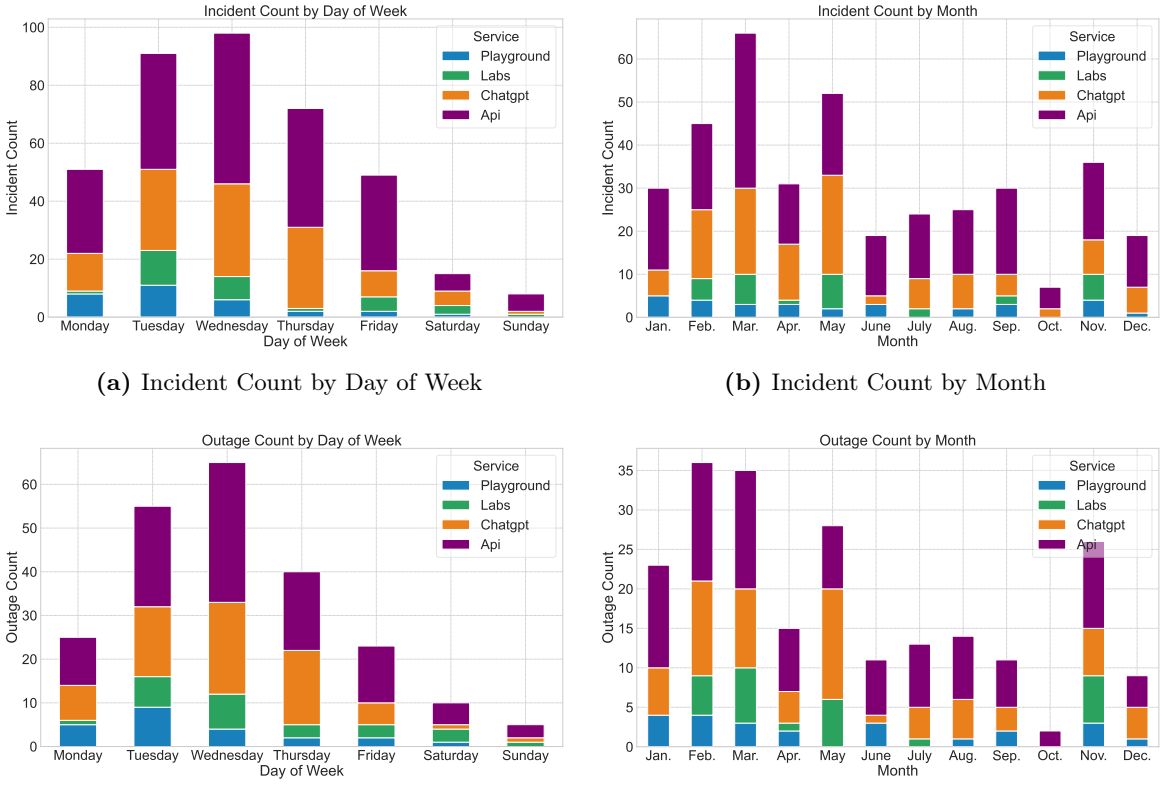


Figure 4.8: Incident Count by Hour of Day

Figure 4.8 illustrates the distribution of incident counts by hour, segmented by specific services: Playground, Labs, ChatGPT, and API. The times are presented in Pacific Standard Time (PST) or Pacific Daylight Time (PDT), corresponding to North American Time

Zones.

A noticeable pattern is the lower incident rates during off-work hours, particularly late at night and early in the morning. From 4:00 AM, a gradual increase in incidents begins, escalating significantly from 9:00 AM. This time marks the beginning of typical work hours and is followed by a slight decrease before reaching another peak at noon. The peak around noon represents the highest incident counts of the day (**O-9**). The morning hours consistently show more incidents compared to the afternoons. This pattern holds true across all services. The correlation between typical business hours and increased incident rates suggests a link between user activity and system failures.



(a) Incident Count by Day of Week

(b) Incident Count by Month

(c) Outage Count by Day of Week

(d) Outage Count by Month

Figure 4.9: Incident and Outage Temporal Distribution by Service

Figure 4.9a presents the distribution of incidents across different days of the week. A steady increase in incidents starts from Monday, with a peak on midweek, particularly on Wednesday (**O-10**). This midweek surge is particularly notable in the API and ChatGPT services, which maintain high incident rates even into Thursday. After the midweek peak, there is a gradual decline towards the weekend, with Sunday showing minimal incidents.

There are barely any incidents on Sunday except for API. This pattern indicates a surge of user activity in midweek, correlated with increased incident rates.

Figure 4.9b shows the monthly distribution of incidents throughout the year. A year starts with a moderate level of incidents in January, which gradually increases and peaks in March. This peak is the highest point for incidents throughout the whole year. Post-March, there is a decline in incident counts. April, in particular, shows a substantial drop in incidents. This drop is mainly related to the drop in API service. Throughout the summer, from June to September, there is a consistently low level of incidents. There is a slow and gradual increase starting from the low level in summer. October appears to be an exceptional month with a remarkably low number of incidents. Notably, only ChatGPT and API reported incidents in October. The incident counts across the months exhibit a clear seasonal pattern. The highest incident rates are observed in the spring, particularly in March. Besides, there is a consistently low incident rate during the summer months (June to August) (**O-11**). This pattern could be correlated with reduced user engagement during the summer vacation period. Some of the seasonal variations could also be influenced by the evolving maturity of the services and specific events, such as launching a new product or releasing a new version.

Outage Temporal Distribution

While discrepancy are observed between incident and outage frequency trends, the temporal patterns of outages largely align with those of incidents.

Specifically, the data presented in Figure Figure 4.9c reveals a similar mid-week peak for outages. Examining the outage distribution by month, as shown in Figure Figure 4.9d, outage rates are observed high during spring from January to March, and notably decrease during the summer months from June to September. Interestingly, compared to the patterns observed in incident reports, the outage count declines starting in September and hits its lowest point in October. Another distinct difference is observed in the final quarter of the year. While the incident count stays at a low level, the outage count in November is relatively high throughout the year.

4.3.3 Implications

The fluctuations in incident and outage frequencies overall trends might be a result of rapid innovation and frequent updates. OpenAI may need to find a balance between introducing new features and maintaining system stability. The varying patterns across

services suggest that OpenAI may need to tailor its failure management approaches to each service. Especially the increasing failure frequency for ChatGPT, suggests the need for continuous improvement in scalability as the user base grows.

The clear daily, weekly, and seasonal patterns in failure rates indicate that the performance of OpenAI's services is highly influenced by user activities. This observation highlights the need for dynamic resource allocation strategies. During peak periods - particularly around noon of the day, mid-week, and spring months throughout the year - OpenAI should increase both computational resources and customer support availability. Additionally, OpenAI could benefit from implementing enhanced monitoring and response systems during these peak periods. The correlation between failure rates and user activities also suggests the significance of user study. A deep analysis of user activity patterns could help OpenAI to minimize the disruptions that users experience and improve its system reliability.

4.4 Inter-service Correlation Analysis of Failures

O-12: *The correlations across services are generally weak.*
O-13: *API and ChatGPT demonstrate a negative linear relationship, as indicated by their negative (-0.35 in incidents and -0.48 in outages).*
O-14: *ChatGPT shows more independence from other services. This is evidenced by its negative PCC with API and its near-zero PCC with both Labs and Playground.*

This section analyzes the relationships between OpenAI's services. The goal of this analysis is to investigate whether there are underlying shared triggers of service failures.

Figures 4.10 and 4.11 visualize these correlations with two main types of heatmaps: Co-occurrence Counts and Correlation Coefficients. Each cell on Co-occurrence Count heatmaps, except for the diagonal cells, represents the number of times two services failed concurrently. The diagonal cells are instances where only a single service failed. They use a grayscale color scheme to represent the frequency of joint failures, with darker colors indicating higher co-occurrence. The Correlation Coefficient heatmaps display PCC between the service failures. These coefficients assess the linear relationship of a service pair. The correlation heatmaps use a diverging color scheme, with red representing positive correlations, blue representing negative correlations, and white or light colors representing no or weak correlations.

The patterns revealed for incidents in 4.10 and outages in 4.11 are very similar.

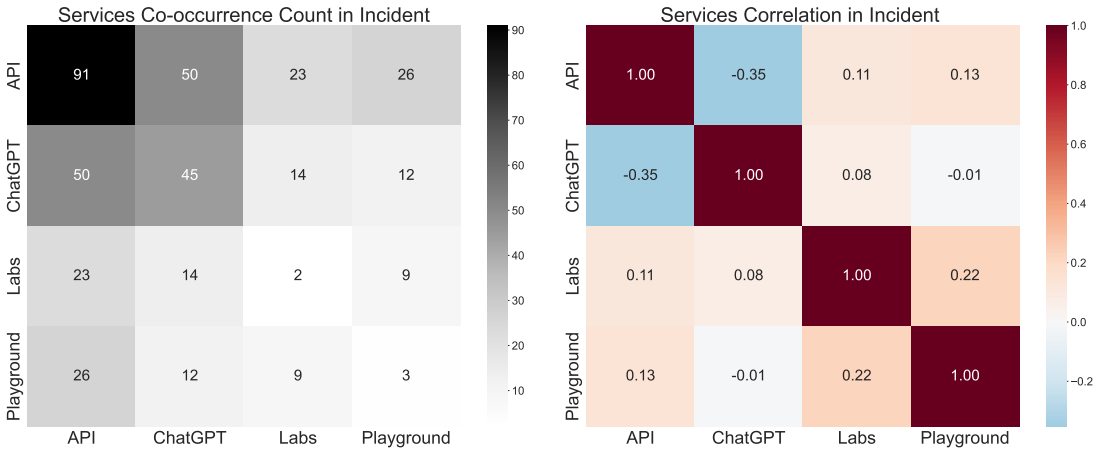


Figure 4.10: Inter-service Correlation Analysis for Incidents

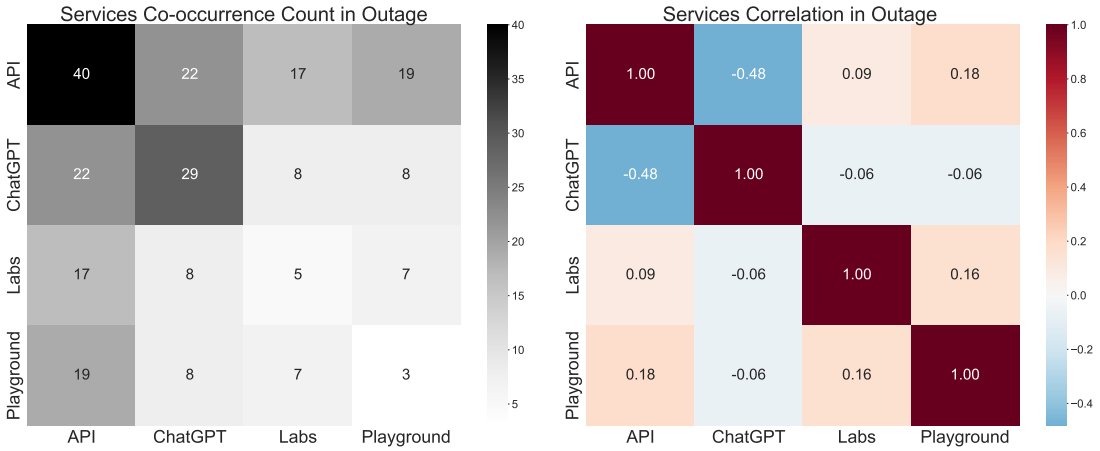


Figure 4.11: Inter-service Correlation Analysis for Outages

API and ChatGPT share high co-occurrence but exhibit a negative correlation. Despite often failing simultaneously (with co-occurrence counts of 50 in incidents and 22 in outages), they show negative correlation coefficients (-0.35 in incidents and -0.48 in outages). This suggests that while they frequently experience failures on the same days, their failure modes are not positively linearly dependent (**O-13**). This might be because API and ChatGPT are highly utilized services, which increases the chance of coincidental simultaneous issues. However, there are also many instances where only one of the services fails. Especially the API, which alone experienced incidents 90 times, nearly twice the co-occurrence count, and outages 40 times, nearly double the co-occurrence count of 22. This indicates

that often when the API experiences problems, ChatGPT may operate normally, and vice versa.

Labs and Playground show weak positive correlations with each other and with the API (Labs-Playground, Labs-API, API-Playground). Labs and Playground present low single-occurrence failure frequency (Labs alone had 2 incidents and 5 outages; Playground had 3 incidents and 3 outages), suggesting they coincide with other services more often. This indicates they may share some common dependencies with other services, but the relationships are not strong.

ChatGPT exhibits more independence from other services (**O-14**). Its negative correlation with the API suggests that when API fails, the ChatGPT is unlikely to fail. The PCC between ChatGPT and both Labs and Playground tends towards zero, indicating no significant linear relationship between their failures.

Overall, inter-service correlations are generally weak (**O-12**). Across the correlation matrices, the coefficients are close to zero, both positive and negative. This implies that there is no strong linear relationship in the failure patterns across services, indicating a degree of operational independence.

Implications

The high co-occurrence but negative correlation between API and ChatGPT failures indicates that these services might benefit from service isolation. This could help maintain the availability of one service even when the other experiences issues.

The weak positive correlations between Labs, Playground, and API suggest that these services might share some potential dependencies. The failure response mechanism of OpenAI could be optimized to focus on these shared components to improve overall system reliability.

4.5 Summary of Failure Analysis

In our failure analysis, we present approaches to comprehensively characterize OpenAI's system failure patterns. The characterization is conducted from several perspectives: failure recovery efficiency, severity, time patterns, and potential shared failure dependencies between services.

The results suggest that OpenAI has an efficient and robust failure response mechanism. OpenAI is capable of detecting minor disruptions and resolving most issues quickly. The robustness of OpenAI's system is also proved by the fact that most of its problems do not

cause critical impact. It keeps partial functionality while addressing the issues. OpenAI appears to have an effective priority protocol for incident recovery. It resolves severe incidents much quicker than those with less impact. API and ChatGPT show higher failure frequency, with ChatGPT presenting more challenges in failure mitigation. This is evidenced by its longer incident resolution times and more impact-critical incidents.

From the perspective of time patterns, a general upward trend in failure frequency is found, indicating increasing system complexity and workload. Interestingly, the temporal distribution of failures suggests that OpenAI's service failures are highly correlated with user behaviors. It is showing clear failure surges during morning hours and midweek, while experiencing a significant reduction during summer months, likely due to vacation periods.

Investigating correlations between OpenAI's services, the results indicate that OpenAI's services are well-isolated. Failures of each service generally do not present strong positive linear dependencies with others. Notably, API and ChatGPT display a negative correlation, suggesting they tend to fail or operate independently of each other.

5

Textual Analysis of OpenAI Incident Reports

5.1 Overview and Apoproaches for Report Analysis

The goal of the report analysis is to extract meaningful patterns and characteristics of system failures from these textual incident reports. We apply LDA topic modeling to the incident dataset and present the visualization of its result for better interpretation. The performance of the LDA model is evaluated and optimized using a hyperparameter Engineering method.

The LDA model considers a document as a mixture of various subjects or themes. LDA assigns a probability distribution over a set of topics for each document, indicating the likelihood of each topic being present in that particular document [12]. By analyzing the word frequencies within the document, LDA can identify the main themes discussed in the text. A key advantage of LDA is that it is a generative model that doesn't require predefined topics, making it valuable for exploring the underlying thematic structures of OpenAI's incidents.

The LDA modeling process consists of three main stages: preprocessing, training, and evaluation. The preprocessing stage involves several steps to prepare the text data for LDA modeling. First, the text undergoes cleaning and standardization for consistency. Next, the text is tokenized into individual words and lemmatized to reduce words to their base forms Bigrams are also created to capture common two-word phrases. Finally, stop words and short words are removed to focus on the most meaningful terms.

The LDA model takes the preprocessed text data as training input and learns a set of latent topics from it. The number of topics is determined by hyperparameter Engineering.

The evaluation of the LDA model is performed utilizing the coherence score. A higher coherence score indicates that the words within a topic are more semantically related, suggesting better topic interpretability. The results of the LDA model are visualized using pyLDAvis, providing an interactive view of the topics generated. Each topic can be interpreted by its most relevant terms. Complementary to pyLDAvis, we examine the topic distribution per Document, and the proportional distribution of the dominated topics for each service, providing insights into which topics are more prevalent in distinct services.

5.2 Hyperparameter Engineering

The model takes several parameters, including the number of topics, alpha (document-topic density), eta (topic-word density), number of passes(the number of times the model goes through the entire corpus), and chunk size. These parameters can be adjusted to optimize the model's performance. A grid search method is implemented to systematically explore different combinations of these hyperparameters, allowing for model tuning.

Parameter	**Values**
# of topics	3, 4
alpha	symmetric, asymmetric
eta	symmetric, auto
passes	10, 20, 30
chunksize	100, 200

Table 5.1: Grid Search Parameters for LDA Model

A set of potential values is defined in Table 5.1 for each hyperparameter. All possible combinations of these hyperparameter values are generated, creating a comprehensive grid of potential model configurations. For each combination of hyperparameters, an LDA model is trained using the specified parameters, and the model's performance is evaluated using a coherence score. The coherence score and corresponding parameters are recorded. After evaluating all combinations, the set of hyperparameters that produced the highest coherence score is identified as the optimal hyperparameters (Table 5.2). Using these hyperparameters, a final LDA model is trained on the incident reports, and the coherence score for the best model is 0.3284.

Parameter	# of topics	Alpha	Eta	Passes	Chunksize	Best Coherence Score
Optimal configuration	3	Asymmetric	Symmetric	30	200	0.33

Table 5.2: Optimal hyperparameters for the LDA model

5.3 pyLDAvis Visulisation Results

O-15: *The LDA model generated three distinct topics: Elevated Error Rates, Service Outages, and Customer Experience Issues.*

The pyLDAvis visualization provides an insightful representation of the LDA results. The left side of the visualization displays an Intertopic Distance Map, while the right side shows bar charts of the Top-30 Most Relevant Terms for each topic. In the Intertopic Distance Map, three distinct circles represent the three topics. The size of each circle corresponds to the prevalence of that topic in the corpus, while the distance between circles indicates the degree of distinction between topics. The clear separation observed between these circles suggests that the LDA model has successfully extracted 3 relatively independent topics from the incident reports: Elevated Error Rates, Service Outages, and Customer Experience Issues (**O-15**). The bar charts on the right display the most relevant terms for each topic, along with the specific token percentages, which represent the proportion of each topic in the entire document collection. The bars with red color indicate the estimated term frequency within a selected topic, while blue ones show the overall term frequency across all documents.

5.3.1 Topic 1: Elevated Error Rates

Topic 1, represented in Figure 5.1, accounts for the largest share of tokens (45%), indicating that this is the most frequently encountered type of issue in OpenAI's system. Key terms associated with this topic include "error," "elevate," "rate," "ChatGPT," "implement," and "API." This topic can be interpreted as incidents related to elevated error rates. Such incidents typically represent situations where system performance degrades or fails to function normally, but usually do not result in service unavailability. These issues often affect multiple services, especially ChatGPT and API services.

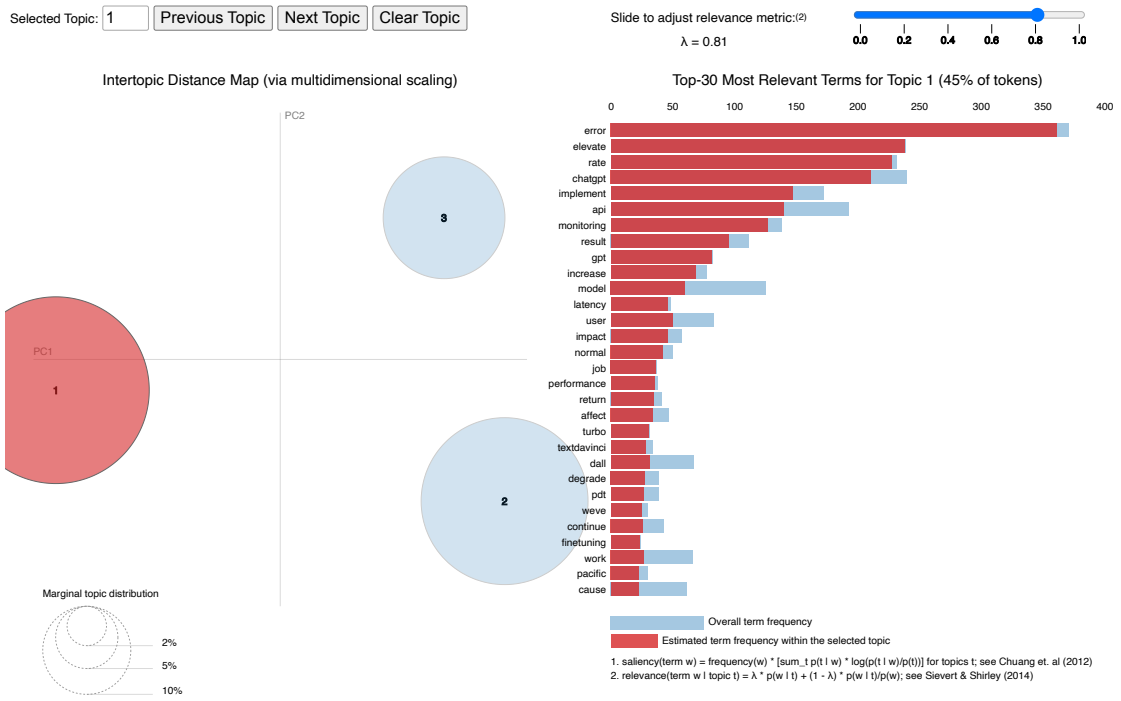


Figure 5.1: pyLDAvis Results: topic=1

5.3.2 Topic 2: Service Outages

Topic 2, shown in Figure 5.2, represents 35.9% of the tokens, indicating that while not as prevalent as elevated-error-rates related incidents, it still constitutes a significant topic of failures. Key terms for this topic include "outage," "model," "request," "DALL·E" "cluster," "work," and "finetune." This topic likely corresponds to incidents with one or more services experiencing outages, resulting in service unavailability for users. These outages, indicated by the key terms, can affect various components of OpenAI's infrastructure, from specific models like DALL-E to the underlying compute clusters that run the models. The term 'request' might indicate that these outages often manifest as failures to process user requests.

5.3.3 Topic 3: Customer Experience Issues

Topic 3, illustrated in Figure 5.3, has the lowest share of tokens (19.1%), suggesting that while these issues occur, they are relatively less common than the previous two topics. Key terms for this topic include "failure," "traffic," "customer," and "change." This topic appears to emphasize incidents related to user interactions, covering various issues that may

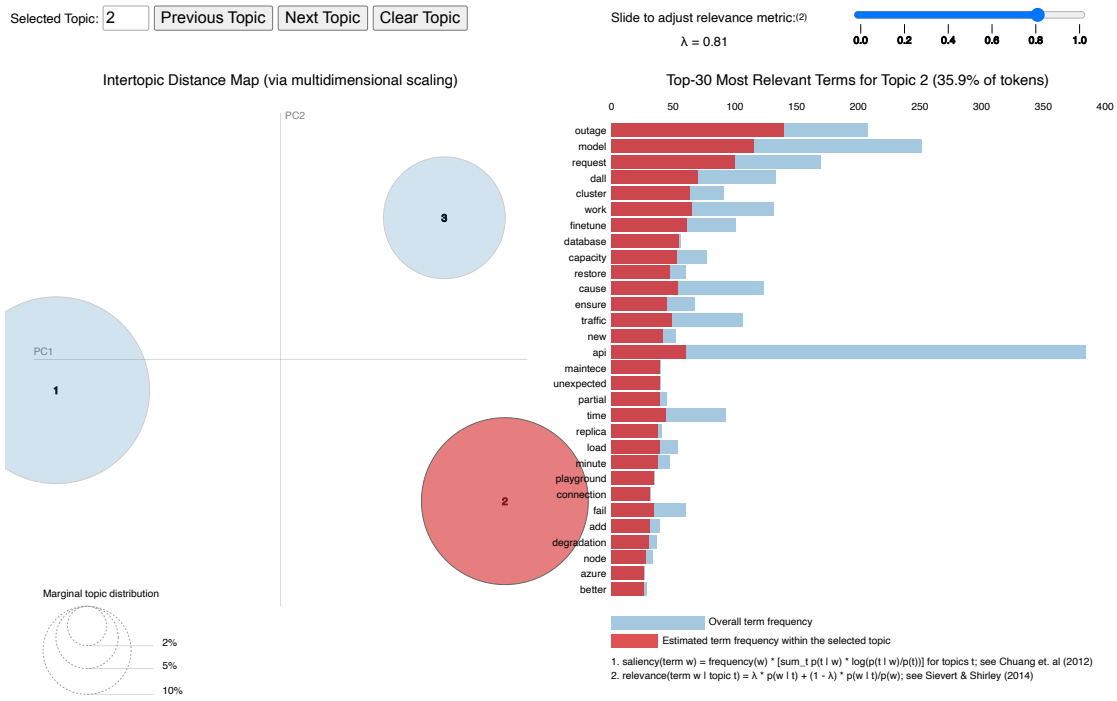


Figure 5.2: pyLDAvis Results: topic=2

not necessarily lead to system downtime but impact user experience. These could include malfunction or performance degradation caused by system updates or traffic fluctuations.

5.4 Topic Distribution Analysis

This section digs the results of LDA deeper by investigating the topic distributions.

5.4.1 Topic Distribution per Document

O-16: *While the topic "elevated error rates" is dominant across documents, there is an increase in the topic "service outage" in later documents.*

Figure 5.4 illustrates the topic distribution across individual incident reports (documents). The x-axis represents each document, ordered chronologically from left to right. The y-axis shows the probability of different topics in each document, with the total probability summing up to 1. Different colors distinguish the topics: blue for Topic 1 (elevated error rates), orange for Topic 2 (service outages), and green for Topic 3 (customer experience issues).

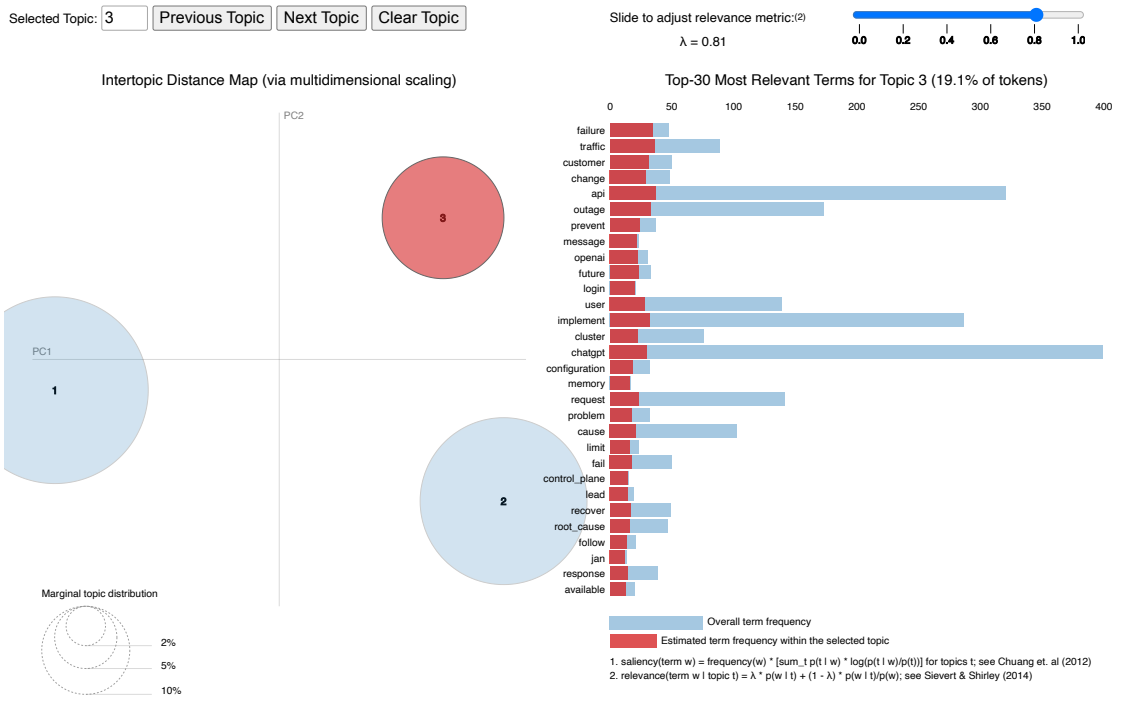


Figure 5.3: pyLDAvis Results: topic=3

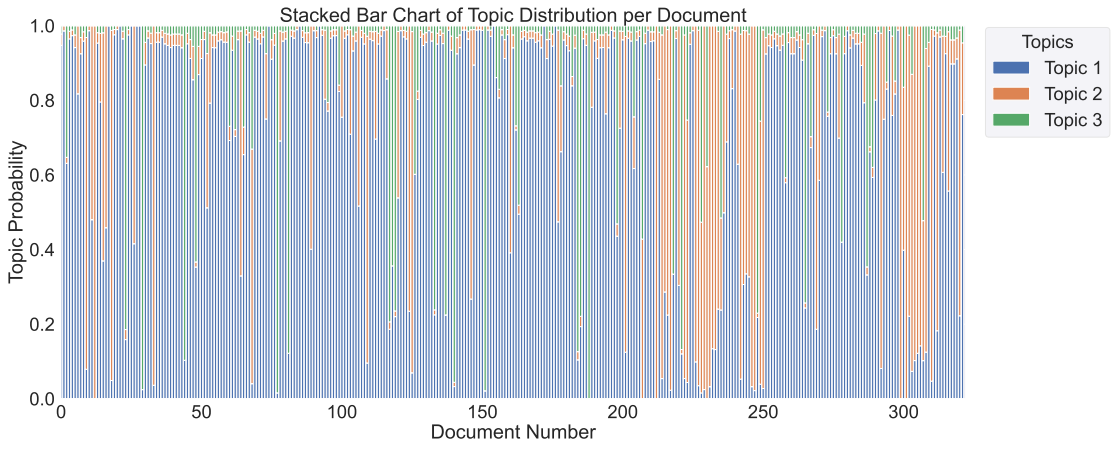


Figure 5.4: Topic Distribution per Document

This visualization reveals a changing trend in topic distribution over time. Generally, Topic 1 (blue) appears to dominate across documents, but there is a noticeable increase in Topic 2 (orange) in later documents (**O-16**). Topic 3 (green) consistently shows a smaller

presence throughout the timeline.

This distribution aligns with the pyLDAvis results, confirming that elevated error rates constitute the main topic for OpenAI's failure patterns. Moreover, the changing distribution over time suggests that OpenAI's failure patterns or incident management approaches have evolved, with an increased emphasis on outage-related descriptions in later incident reports.

5.4.2 Topic Distribution by Service

Figure 5.5 presents the distribution of dominant topics across different OpenAI services. For each document, the topic with the highest probability is considered the dominant topic. The left subfigure shows the raw incident count for each topic grouped by affected service. Due to varying total incident counts across services, the right subfigure presents a normalized version of the incident count, offering a more balanced comparison.

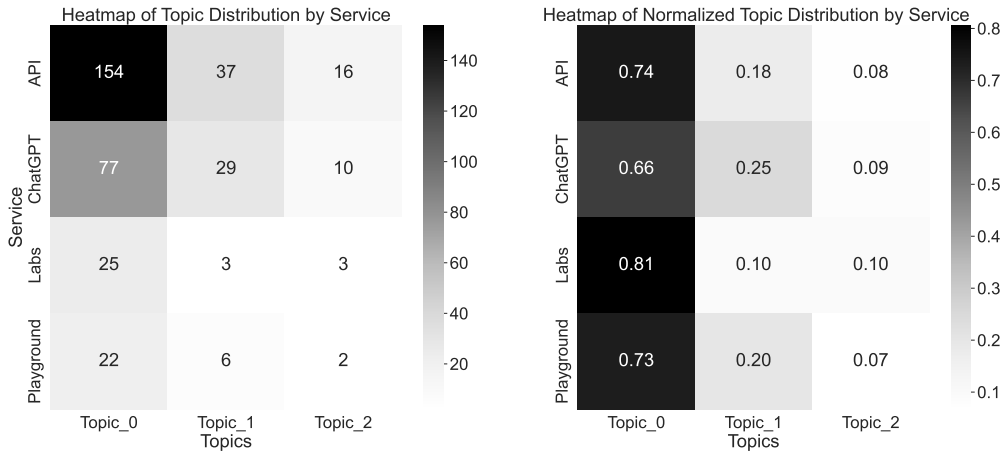


Figure 5.5: Topic Distribution by Service

For API Service, Topic 1 (elevated error rates) is most prevalent (74%), indicating that the API service frequently experiences performance fluctuations and increased error rates. While still showing a high proportion of Topic 1 (66%), ChatGPT exhibits a relatively higher proportion of Topic 2 (service outages, 25%) compared to other services. This suggests that ChatGPT may face more challenges in terms of availability. Labs and Playground display less interesting results with fewer incidents overall. Their most prevalent topic is Topic 1.

5.5 Summary of LDA Analysis

In the textual analysis, we conducted LDA modeling on the OpenAI incident reports. The model hyperparameters are fine-tuned using a grid research method to improve performance.

The modeling result of the optimized LDA model reveals 3 distinct topics of issues: elevated error rates, service outages, and customer experience problems. The prevalence of elevated error rate incidents holds across all services, particularly in the API. This pattern suggests the potential challenge of maintaining consistent performance in complex AI systems.

The evolution of topic distribution over time suggests that OpenAI's focus is shifting. They might pay increasing attention to outage-related issues in recent periods. This could indicate either an actual increase in outages or, more likely, evolved disruption detection and reporting mechanisms.

The service-specific analysis underscores the unique challenges faced by different components of OpenAI's ecosystem. While the API service struggles predominantly with increased error rate issues, ChatGPT shows a higher frequency for outage-related issues.

6

Threats To Validity and Limitation

(1) The data pipeline used in this study can only collect data up to the point of execution at once, potentially missing the most recent information. This limitation could lead to an incomplete representation of OpenAI operational data. To address this, implementing a cloud-based solution with scheduled data collection would ensure more up-to-date data. The scope of our data collection is limited, which may reduce the significance of the pattern observed from the data. To enhance the data, future research should expand data collection to include a wider range of LLM-based AI systems, and more diverse data sources such as user-reported data. This would provide a more comprehensive view of AI services beyond just OpenAI.

(2) In our failure analysis, the underlying causes and implications of observed patterns are largely based on subjective interpretations. The lack of concrete data, such as actual user activity metrics and computational resource usage, limits our ability to draw definitive conclusions. Future research should incorporate quantitative data to support and validate qualitative observations.

(3) The topic modeling approach used in this study is constrained by limited data and a lack of priori knowledge. As LDA, the model we use to generate topics, is an unsupervised learning method, its results are less accurate compared to supervised approaches. The absence of ground truth for topic categories further reduces the value of the LDA results. To improve this, manual categorization to construct ground truth data would be necessary, allowing for more robust topic modeling outcomes.

These limitations highlight areas for improvement in future research on AI failures. Addressing these threats to validity would enhance the significance and generalizability of findings in this area of study.

7

Related Work

In the domain of AI, many prior researches have investigated the workload characteristics of AI tasks, especially on High-Performance Computing (HPC) systems. Li et al.[13] analyzed Slurm scheduler logs from MIT Supercloud system, which is designed for AI workloads. They collected the logs for over 40 thousand GPU jobs and they analyzed the job characteristics. This study propose a job categorization method based job life cycle and it compares resource utilization between different type of jobs. Silimarly, Wang et al.[14] examined the performance of Deep Learning training jobs on Alibaba's Platform of Artificial Intelligen. They bring up a lightweight characterization framework to extract the DL-specific workload features. Furthermore, Ren et al.[15] conducted a comprehensive analysis. They compare the performance of different Deep Learning models on several cutting-edge systems, providing guidance for selecting the most suitable system for specific deep learning software.

Previous research has primarily focused on analyzing the workload performance of AI tasks, evaluating metrics such as resource utilization, communication bandwidth, and so on. There is a lack of failure analysis of AI tasks or services, evaluating the service reliability. However, extensive research has been conducted on cloud services, which share similar characteristics with AI services. Both cloud services and AI services rely on large-scale distributed infrastructure to handle large amount of data and complex tasks. Tola et al.[16] collected over 10 thousand failure events published by cloud service providers. They used graphical and statistical techniques to model the failure process and the failure intervals were found to be modeled by exponential distribution for most cloud services. Instead of self reports, Talluri et al.[17] collected user-reported failures for 12 popular cloud services. They analyzed failure counts, symptoms, and extracted time patterns from the reports with solid validation.

Despite the growing importance of AI services, there remains a significant gap in the analysis of their service failures. This studu aims to bridge this gap by analyzing the failure data of OpenAI, including its historical incidents and outages published on OpenAI's status page.

8

Conclusion

This thesis presents an operational analysis of the failure characteristics and patterns of OpenAI's services using their publicly reported incident and outage data. Through developing a robust data pipeline, we successfully collected the outage and incident data from dynamic web pages and obtained two structured datasets: Incident and Outage, providing cleaned datasets for further operational data analysis. We proposed a methodology to analyze failure and incident reports of LLM services from OpenAI, and give 16 key observations from the analysis results.

The failure analysis revealed key insights into OpenAI's system reliability. Despite the increasing complexity and user base, OpenAI demonstrates an efficient failure response mechanism, with most issues resolved quickly. ChatGPT experienced relatively higher failure frequencies. Its vulnerabilities are also present in its long incident resolution time with higher impact, indicating areas for improvement. The identified temporal failure patterns appear to be strongly correlated with user behaviors.

The textual analysis of incident reports using LDA topic modeling extracted three distinct themes: elevated error rates, service outages, and customer experience issues. The prevalence of error-related incidents across services underscores the challenge of maintaining consistent performance in complex AI systems. The evolving topic distribution over time suggests a shift in OpenAI's focus towards outage-related issues.

In conclusion, this study provides valuable insights for OpenAI and other AI service providers to enhance system reliability. The analytical methods presented in this thesis can be extended to analyze other AI services, contributing to a more general understanding of reliability in the rapidly evolving field of AI applications. Future research can focus on expanding the scope of data collection and validating observations with quantitative

metrics. Addressing the limitations in this study will further strengthen the findings in the domain of LLM-based AI service reliability.

References

[1] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. **Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond**. *ArXiv*, **abs/2304.13712**, 2023. 1

[2] Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. **Aligning Large Language Models with Human: A Survey**. *ArXiv*, **abs/2307.12966**, 2023. 1

[3] Chenyang Lyu, Minghao Wu, Longyue Wang, Xinting Huang, Bingshuai Liu, Zefeng Du, Shuming Shi, and Zhaopeng Tu. **Macaw-LLM: Multi-Modal Language Modeling with Image, Audio, Video, and Text Integration**. *ArXiv*, **abs/2306.09093**, 2023. 1

[4] Jing Yu Koh, Daniel Fried, and R. Salakhutdinov. **Generating Images with Multimodal Language Models**. *ArXiv*, **abs/2305.17216**, 2023. 1

[5] Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. **LLM-grounded Diffusion: Enhancing Prompt Understanding of Text-to-Image Diffusion Models with Large Language Models**. *ArXiv*, **abs/2305.13655**, 2023. 1

[6] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. **Black-Box Tuning for Language-Model-as-a-Service**. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, **162** of *Proceedings of Machine Learning Research*, pages 20841–20855. PMLR, 17–23 Jul 2022. 1

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam,

Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. **Language Models are Few-Shot Learners**, 2020. 1

[8] Emanuele La Malfa, Aleksandar Petrov, Simon Frieder, Christoph Weinhuber, Ryan Burnell, Raza Nazar, Anthony G. Cohn, Nigel Shadbolt, and Michael Wooldridge. **Language Models as a Service: Overview of a New Paradigm and its Challenges**, 2023. 1

[9] Konstantinos I. Roumeliotis and Nikolaos D. Tselikas. **ChatGPT and Open-AI Models: A Preliminary Review**. *Future Internet*, **15**(6), 2023. 1

[10] Reuters. **ChatGPT sets record for fastest-growing user base - analyst note**, 2023. 1

[11] Lizhou Fan, Lingyao Li, Zihui Ma, Sanggyu Lee, Huizi Yu, and Libby Hemphill. **A Bibliometric Review of Large Language Models Research from 2017 to 2023**, 2023. 1

[12] Bhagyashree Vyankatrao Barde and Anant Madhavrao Bainwad. **An overview of topic modeling methods and tools**. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 745–750, 2017. 35

[13] Baolin Li, Rohin Arora, Siddharth Samsi, Tirthak Patel, William Arcand, David Bestor, Chansup Byun, Rohan Basu Roy, Bill Bergeron, John Holodnak, et al. **AI-enabling workloads on large-scale GPU-accelerated system: Characterization, opportunities, and implications**. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 1224–1237. IEEE, 2022. 44

[14] Mengdi Wang, Chen Meng, Guoping Long, Chuan Wu, Jun Yang, Wei Lin, and Yangqing Jia. **Characterizing deep learning training workloads on alibaba-pai**. In *2019 IEEE international symposium on workload characterization (IISWC)*, pages 189–202. IEEE, 2019. 44

[15] Yihui Ren, Shinjae Yoo, and Adolfy Hoisie. **Performance analysis of deep learning workloads on leading-edge systems**. In *2019 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, pages 103–113. IEEE, 2019. 44

[16] Besmir Tola, Yuming Jiang, and Bjarne E Helvik. **Failure process characteristics of cloud-enabled services**. In *2017 9th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pages 1–7. IEEE, 2017. 44

[17] Sacheendra Talluri, Leon Overweel, Laurens Versluis, Animesh Trivedi, and Alexandru Iosup. **Empirical Characterization of User Reports about Cloud Failures**. In *2021 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*, pages 158–163. IEEE, 2021. 44