An Empirical Characterization of Outages and Incidents in Public Services for Large Language Models

Xiaoyu Chu, Sacheendra Talluri, Qingxian Lu, Alexandru Iosup
{x.chu,s.talluri,a.iosup}@vu.nl,{q.lu}@student.vu.nl
Vrije Universiteit Amsterdam, The Netherlands

ABSTRACT

People and businesses increasingly rely on public LLM services, such as ChatGPT, DALL·E, and Claude. Understanding their outages, and particularly measuring their failure-recovery processes, is becoming a stringent problem. However, only limited studies exist in this emerging area. Addressing this problem, in this work we conduct an empirical characterization of outages and failure-recovery in public LLM services. We collect and prepare datasets for 8 commonly used LLM services across 3 major LLM providers, including market-leads OpenAI and Anthropic. We conduct a detailed analysis of failure recovery statistical properties, temporal patterns, co-occurrence, and the impact range of outage-causing incidents. We make over 10 observations, among which: (1) Failures in OpenAI's ChatGPT take longer to resolve but occur less frequently than those in Anthropic's Claude; (2) OpenAI and Anthropic service failures exhibit strong weekly and monthly periodicity; and (3) OpenAI services offer better failure-isolation than Anthropic services. Our research explains LLM failure characteristics and thus enables optimization in building and using LLM systems. FAIR data and code are publicly available on https://zenodo.org/records/14018219 and https://github.com/atlarge-research/llm-service-analysis.

KEYWORDS

Failure characterization, LLM, performance modeling, reliability, OpenAI, Anthropic, Character.AI, operational data analytics, outage, incident, failure-recovery, time-series analysis

1 INTRODUCTION

In the past 5 years, increased availability of data and computation enabled Large Language Models (LLMs) to support scientists, businesses, and general users in a wide range of applications, such as coding [41, 64], image generation [36, 42], and general problem-solving [66, 71]. Hundreds of millions of users rely increasingly on public LLM services such as ChatGPT [53], DALL·E [29], and Claude [52]. Understanding service outages, and how incidents leading to them are addressed, is essential to enhancing the fault tolerance and quality of service (QoS) of LLM systems. However, relatively little data and no peer-reviewed studies exist in this rapidly emerging area. Addressing this problem, and complementing studies that focus on LLM resource utilization [28, 67] and user satisfaction [66], in this work we conduct the first data-driven, empirical characterization of outages and incidents in public LLM services. We conduct three classes of analysis on long-term datasets we collect from 8 public LLM services from OpenAI, Anthropic, and Character.AI.

The reliability of public LLM services is becoming increasingly important, as service failures can severely erode user experience and cause substantial financial losses under the competitive market. Driven by demand and market strategy, public LLM providers compete intensely, investing over $ 40 billion in 2024 [30]. Failures quickly affect many users and become highly visible, as the user cohort has already reached a global scale. For example, the launch of ChatGPT marked a major breakthrough in LLM applications [55] and set a user-adoption record, with 100 million monthly active users within 2 months after its 2022 launch [51].

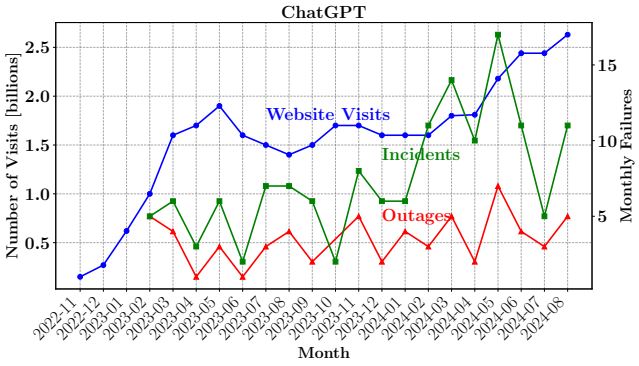


Figure 1: Monthly website visits, outages, and incidents for ChatGPT. Vertical axis: (left) number of website visits in billions; (right) monthly outage and incident counts. Data: website visits [49], outages (Table 3), and incidents (Table 4).

Although service reliability is important, users still frequently encounter issues with LLM services. For example, users report to DownDetector many login failures, request errors, and high response latency when using ChatGPT [19]. Figure 1 shows the monthly website visits [49], and outages and incidents for ChatGPT as reported by OpenAI. As ChatGPT's monthly web visits grow dramatically, the number of its outages and especially incidents also exhibit an upward trend. Thus, significant LLM failures continuously occur, decreasing user satisfaction and potentially causing financial loss, making reliable LLM services a challenge.

Understanding dependability aspects can help improve systems especially when the workload characteristics are also understood. Previous work already provides system-level workload characteristics for the workloads of machine learning [14, 15, 65], deep learning [38], big data [58], and more general clouds [50]. Recently, LLM workloads have received attention as well [28, 67]. What remains unaddressed in characterizing the failures of LLM.

We identify and address in this work two main challenges in understanding how public LLM services currently fail. First, **no longitudinal service failure data currently exists**. Ideally, the community would have access to a large number of similarly curated datasets that capture LLM-service failures, under the same failure model, over long periods of time. There are some efforts to provide available LLM workloads, such as BurstGPT [67] and AcmeTrace [28], but they each focus on one LLM service, and none provides service failure data for it.

Second, **no comprehensive analysis of failures in public LLM services currently exists**. At this stage in the scientific area, such an analysis would ideally be data-driven, and include for example general characteristics of failures, such as Mean Time Between Failures (MTBF) and To Recovery (MTTR) from classical dependability analysis [8], and also of the time spent in various stages of the recovery-process specific to LLM operations; a temporal analysis of failures; and an analysis of failure cascades (co-occurrences). Such kinds of analysis would enable future research into models, and future theoretical and practical studies of LLM systems.

Addressing both main challenges, this research aims to provide a thorough empirical characterization of LLM service failures, using data from official outages and incident reports, which are the two types of information self-disclosed by LLM service providers when significant failures occur. Our contribution is manyfold:

(i) We summarize the de facto industry standard for modeling LLM-service outages and exemplify the anatomy of an outage (Section 2).
(ii) We collect outage and incident data for 8 LLM services, and prepare the corresponding LLM-failure datasets (Section 3). This study covers representative, commonly used LLM services, across 3 LLM-service providers: OpenAI's API [5], ChatGPT [12], DALL·E [18], Playground [46]; Anthropic's API [4], Claude [16], Console [17]; and Character.AI [11].
(iii) We analyze the failure characteristics of 8 LLM services (Section 4). We analyze the MTTR and MTBF by provider and by service, the time spent in various stages of the recovery process, and quantify empirically the model parameters introduced in Section 2.
(iv) We analyze LLM service failures over time (Section 5). We explore service availability over hourly and daily intervals, identify various diurnal and weekly patterns, and investigate auto-correlations.
(v) We analyze the co-occurrence of failures (Section 6). Specifically, we analyze the co-occurrence of failures per provider, and of pairs of services across providers.
(vi) We follow the principles of open science and release as open, FAIR artifacts for public use the datasets[1] and software[2] used in this work. We expect them to provide opportunities to reproduce and expand on our research.

2 ANATOMY OF AN LLM-SERVICE INCIDENT: MODEL AND EXAMPLE

We present in this section a model, coupled with an example, of how an LLM-service incident occurs, affects actual users, and is managed by the LLM-service provider.

2.1 Model and Real-World Example

A failure-recovery process not only leads to addressing a system failure, but also shows a complete story of how the system experienced the (cascading) failure and provides insights to track and improve the system and services affected by it [20].

Industry leads, such as OpenAI and Anthropic, report availability data built around a de facto standard model of their failure-recovery process. Acting as a tutorial, this section summarizes this industry model and exemplifies it through an actual incident. The example considers both data self-reported by the LLM service provider, OpenAI, and data reported by users experiencing the incident; analyzing user-reported data across all incidents studied here is useful but outside the scope of this article.

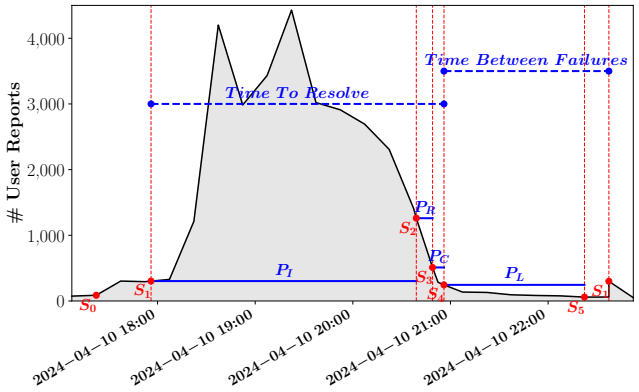


Figure 2: Visualization of the failure-recovery model with user reports of a selected ChatGPT incident, UDT time.

Selection of the incident. We selected the real-world incident in which a major outage happened with the ChatGPT LLM service on April 10, 2024; this is the first major incident since, on April 1, OpenAI enabled free-to-use access to ChatGPT without signup, effectively opening up ChatGPT trials for everyone [45]. OpenAI reported the April 10 incident with complete details about all the stages of its failure-recovery process [44], which is only done for significant incidents that require many local resources to address. In parallel with OpenAI team's efforts to identify and resolve the incident, we recorded two other data sources. First, users reported problems using the ChatGPT service on DownDetector [6]; user-reported failures of public services are increasingly used to check the truthfulness and completeness of self-reported failure reports [59], but so far they have not been used in peer-reviewed studies of LLM services. We also recorded reports about this outage from news media across the technical and political spectrum, such as Fox [72].

Incident visualization and model parameters: Focusing on the major ChatGPT outage on April 10, 2024, Figure 2 visualizes the failure-recovery process as reported by ChatGPT, overlapping it with the number of user reports as reported by DownDetector. Around 2024-04-10 17:30, with the service believed to operate normally, some faults started to happen and the number of user reports increased to an abnormal level. This triggered an alert to the ChatGPT operational team, who started *investigating* at 17:56 (status S_1). At 20:39, they *identified* the issue (S_2). They quickly implemented a fix, which they released and started *monitoring* at 20:49 (S_3). They confirmed that the issue had been *resolved* at 20:56 (S_4). During this period, the user reports increased to a peak, around which it fluctuated until the fix was released to increasingly more users, at which point a sharp drop of user reports, toward a normal level, can be seen in the figure. Finally, after a period of *postmortem analysis*, an incident summary was released by the ChatGPT team, to explain the cause of this incident to its users (S_5).

[1] https://zenodo.org/records/14018219
[2] https://github.com/atlarge-research/llm-service-analysis

Table 1: Parameters and, below the double line, output metrics of the failure-recovery model proposed in this work.

ID	Name	Definition
S_1	Investigating Status	The operational team has started investigating an incident.
S_2	Identified Status	The issues have been identified.
S_3	Monitoring Status	A fix has been implemented and the operational team started monitoring the results.
S_4	Resolved Status	The incident has been resolved.
S_5	Postmortem Status	A summary of the incident after it has been resolved.
P_I	Investigating Period	From S_1 to S_2, showing the time from observing to identifying the issues.
P_R	Repairing Period	From S_2 to S_3, showing the time to repair the issues.
P_C	Checking Period	From S_3 to S_4, showing the time to confirm the fix is stable and effective.
P_L	Learning Period	From S_4 to S_5, showing the time to provide the incident's root cause.
$MTTR$	Mean Time To Resolve	From S_1 to S_4, covering P_I, P_R, P_C, showing the full time of resolving issues.
$MTBF$	Mean Time Between Failures	From the S_1 of the current incident to the next, showing how frequently failures happen.
T, T_S, A	Outage time, scaled, availability	Definitions discussed in Section 2.2

Table 2: Values of parameters for the selected incident [44], UDT time. Status-markers S_1 through S_4 occur on April 10, 2024.

Incident ID	S_1	S_2	S_3	S_4	S_5	P_I [h]	P_R [h]	P_C [h]	P_L [h]	Time To Resolve [h]
w20mcckg1748	17:56	20:39	20:49	20:56	2024-04-18 00:01	2.72	0.17	0.12	171.08	3.00

This incident exemplifies a failure-recovery process with five key status-markers, S_1 through S_5. Table 1 summarizes the industry-wide model, including these status-markers, the periods they delimit, and the operational metrics. Among the periods, whereas P_I, P_R, and P_C capture how the operational team resolved the outage and are thus synchronous with users experiencing the incident, P_L captures the period, sometimes ending long after the incident has been resolved, during which the operational team analyses the incident and devises new measures to prevent it and related incidents from happening again.

The model outputs include the industry-standard Time To Resolve and, with knowledge about prior and following incidents, Time Between Failures. Table 2 summarizes the status-markers, the periods spanning the failure-recovery process, and the Time To Resolve for this incident. It shows how the operational team took much longer than usual to find the cause of this incident, 2.72 h vs. the 0.65 h found in our analysis in Section 4 (see Table 6), but then it was able to resolve the failure and restore service much faster than normal. However, by then it was already late, as the media has picked up the incident.

As Section 2.2 details, the model also includes LLM-specific metrics to capture the outage duration, scale it according to the severity of the outage, and estimate the (daily) service availability.

2.2 LLM-Specific Terms and Metrics

Incident: An operational issue that may cause a service outage, e.g., "getting an error of having reached a limit of GPT-4 usage" [6]. Once an incident happens, a textual report of its failure-recovery process is produced and is (expectedly) disclosed to the public. An incident can have different *impact levels* on single or multiple services, which include critical, major, minor, minimal, and maintenance, and are similarly defined by the LLM operators.

Outage: Time when the service is unavailable. An outage can have multiple impact ranges: *major outage*, where most of the service's users experience it, and *partial outage*, where a relatively small fraction of the users experience the outage. Operators such as OpenAI *scale* (discount) partial outages as being about 30% (0.3×) as bad as major outages [7].

Outage duration: For an operator, per day, let T_M be the duration of major outage minutes and T_P be the partial outage minutes. The formula to calculate the *daily total outage minutes (T)* is:

$$T = T_M + T_P \tag{1}$$

Similarly, the *daily scaled outage minutes (T_S)* has the formula:

$$T_S = T_M + (T_P \times 0.3) \tag{2}$$

Organizations such as OpenAI use primarily the scaled outage minutes to assess and report outage impact [7].

Availability: Derived from the daily scaled outage minutes, we define the *daily availability*, A, as the percentage of time a service or a group of services are available, given by the formula:

$$A = (1 - \frac{T_S}{24 \times 60}) \times 100\% \tag{3}$$

3 DATASET COLLECTION AND PREPARATION

We collect for this research long-term datasets from 8 LLM services across 3 LLM service providers. We then process these datasets to prepare data useful to characterize LLM service outages and incidents. Tables 3 and 4 summarize the processed outage and incident datasets, respectively.

3.1 LLM Services

Selection process: Addressing the main challenge of lacking longitudinal failure data about LLM services, particularly under the same failure model, we carefully investigate the current LLM services,

Table 3: Summary of LLM outages, per service. Legend: Incident Count = the number of related incidents.

ID	Service	Provider	Start	End	Months	Outage Count			Outage Minutes		Incident
						Total	Major	Partial	Total	Scaled	Count
O_1	API	OpenAI	2021-02-11	2024-08-31	43	104	26	78	7,891	3,340	242
O_2	ChatGPT		2023-02-14	2024-08-31	19	70	28	42	5,185	2,744	157
O_3	DALL·E		2023-02-21	2024-08-31	19	27	13	14	2,821	1,748	34
O_4	Playground		2021-03-31	2024-08-31	42	24	12	12	1,636	1,018	36
A_1	API	Anthropic	2023-07-11	2024-08-31	14	25	0	25	1,675	502	80
A_2	Claude		2023-07-11	2024-08-31	14	30	2	28	3,017	983	90
A_3	Console		2023-07-11	2024-08-31	14	27	1	26	2,032	662	72
C_1	Character.AI	Character.AI	2023-10-19	2024-08-31	11	32	17	15	3,351	1,878	41

Table 4: Summary of LLM incident reports, per service. Legend: Maint. = Maintenance; Inv. = Investigating; PM = Postmortem.

ID	Provider	First Date	Last Date	# of	# of Impact Levels					# of Failure-Recovery Status				
				Reports	Critical	Major	Minor	None	Maint.	Inv.	Identified	Monitoring	Resolved	PM
P_1	OpenAI	2021-02-09	2024-08-28	365	46	125	141	52	1	259	144	225	365	29
P_2	Anthropic	2023-03-25	2024-08-30	141	5	43	48	44	1	96	45	51	141	2
P_3	Character.AI	2023-10-24	2024-08-07	36	19	11	4	2	0	26	16	15	36	2

and select 8 LLM services from 3 service providers based on the following reasons: (1) *Data availability:* Selected services should have public status pages running for long durations, so our data collection can provide rich datasets for the community to further analyze. (2) *Popularity:* Selected services should be popular, with many users and applications with daily use, so the impact of outages is significant, and there is high likelihood users and media will also report on such outages if left unattended. This pressures operators to respond quickly, so the data we collect represents the best performance LLM operators can currently deliver. (3) *Diversity:* Selected services should cover most types of LLM services provided by different companies. This will ensure the generality of our results.

Selected LLM services. (1) *OpenAI API:* The OpenAI API allows developers to access and use advanced LLM models provided by OpenAI through API keys without building or training from scratch. (2) *ChatGPT:* ChatGPT is a chatbot that interacts with users conversationally. ChatGPT can answer follow-up questions with prompts and provide a detailed response. (3) *Labs (DALL·E):* DALL·E is a text-to-image model that can create original, realistic images from a short text description. (4) *Playground:* Playground is a web-based interface for users to interact with and experiment with OpenAI's language models. (5) *Anthropic API:* Similar to OpenAI API, Anthropic API allows developers to integrate language models such as Claude, into their applications and services. (6) *Claude* Similar to OpenAI's ChatGPT, Claude is an AI chatbot and is trained to have natural, text-based conversations with users. (7) *Console:* Similar to the OpenAI's playground, the Anthropic Console is a web-based interface that allows users to interact with Anthropic's AI models directly. (8) *Character.AI:* is an innovative chatbot platform that leverages LLMs to facilitate a series of chatbots that emulate the personas of various figures, such as historical icons, fictional heroes, modern celebrities, etc.

3.2 Data Collection and Dataset Preparation

We collected all available outage and incident data reported publicly by of OpenAI, Anthropic, and Character.AI, up to 2024-08-31, on their public status pages [61–63] and incident pages [31–33]. The starting dates differ: OpenAI has started reporting on February 11, Anthropic on July 11, and Character.AI on October 19, all dates in 2023. (Our study misses none of the published reports.)

The industry has standardized presenting outage data in a calendar format, with separate information for each service. Each outage 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 and any related incidents. Incident reports provide 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. Not all outages have corresponding incident reports. Conversely, some incidents, e.g., with minimal impact, do not report a service outage.

We developed an *automated data-collection method*, able to collect industry-standard outage and incident reports. Our tools leverage Python Selenium WebDriver [56], a robust tool allowing native browser automation by simulating real-user interactions. Our tools implement exception-handling mechanisms, addressing potential issues such as network problems, stale elements, and unexpected page layouts. They parse and extract information from the dynamic pages, and store them as raw datasets.

To *prepare the datasets*, we did typical data transformation, including filling in missing values, extracting data from text, processing JSON formats, splitting columns, and feature engineering to get the metrics used in this study. Last, before the analysis, we performed data cleanup, as discussed in the following, to obtain the cleaned outage and incident datasets whose properties are summarized in Table 3 and Table 4, respectively.

Table 5: Status counts of incident reports (see Table 4).

S_1	S_2	S_3	S_4	S_5	Count	Percent
✓		✓	✓		131	24.39%
✓			✓		110	20.48%
✓	✓	✓	✓		77	14.34%
			✓		62	11.55%
	✓	✓	✓		39	7.26%
✓	✓		✓		35	6.52%
	✓		✓		32	5.96%
		✓	✓		18	3.35%
✓	✓	✓	✓	✓	12	2.23%
✓		✓	✓	✓	7	1.30%
✓			✓	✓	5	0.93%
			✓	✓	4	0.74%
	✓	✓	✓	✓	3	0.56%
	✓		✓	✓	2	0.37%

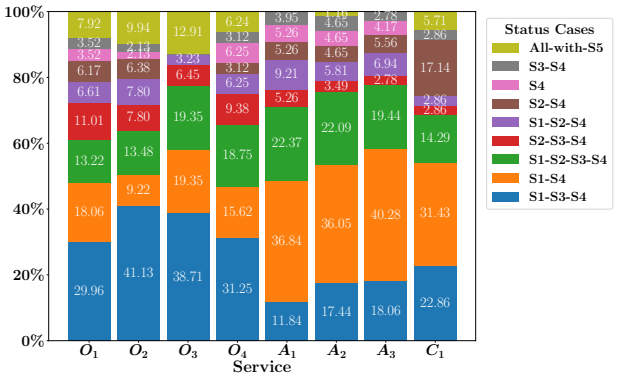


Figure 3: Presence of different status combinations, by service [%]. Due to small counts, status combinations with S_5 are merged into 'All-with-S5'. (Table 3 indexes the services.)

Data cleanup related to the failure-recovery model in Section 2.1: (1) The incident reports from, e.g., ChatGPT, include 6 statuses: *investigating, identified, monitoring, update, resolved*, and *postmortem*. The *update* status is not considered in the model, but in all the reports we have analyzed it seems to be used only as a keep-alive of the recovery process, to mark the operational team is still actively working on the incident, so we do not consider it in our analysis; (2) Out of the over 500 hundreds of incidents we analyzed in this work, only 5 cases do not follow the order of status markers S_1 through S_5. In 2 of these cases, the status-marker *identified* comes before *investigating*, and in 3 other cases, the status-marker *monitoring* comes before *identified*. None of these cases involves unusual durations or recovery times, so we safely exclude these 5 corner cases in our analysis.

4 FAILURE-RECOVERY ANALYSIS

We investigate the time spent on the key operational metrics (MTTR, MTBF) and compare the failure-recovery performance across the 8 LLM services and 3 service providers. We conduct several types of analysis to investigate the failure-recovery processes of LLM services: (1) Statues count and percent of different services; (2) Mean values for main model parameters; (3) Percent of different periods in the MTTR process; (4) Distribution of MTTR and MTTF duration by service; (5) Distribution of MTTR and MTTF duration

Table 6: Mean value for model parameters by service. Legend: h=hour(s), D=day(s).

ID	Service	P_I [h]	P_R [h]	P_C [h]	P_L [D]	$MTTR$ [h]	$MTBF$ [D]
O_1	API-OpenAI	0.72	1.63	1.46	4.10	2.56	5.64
O_2	ChatGPT	0.65	1.64	1.73	4.79	3.64	4.01
O_3	DALL·E	1.01	0.96	1.81	1.86	3.03	18.24
O_4	Playground	0.37	1.56	2.22	4.30	2.95	39.93
A_1	API-Anthropic	1.04	1.11	1.37	-	2.81	5.22
A_2	Claude	1.35	1.72	2.05	0.21	3.16	4.79
A_3	Console	0.94	0.34	0.58	-	2.05	5.73
C_1	Character.AI	0.40	0.50	1.73	3.61	3.95	8.74
	Arith. Mean	0.84	1.40	1.58	4.01	2.94	7.41
	Geom. Mean	0.53	1.15	0.87	3.45	3.99	3.26

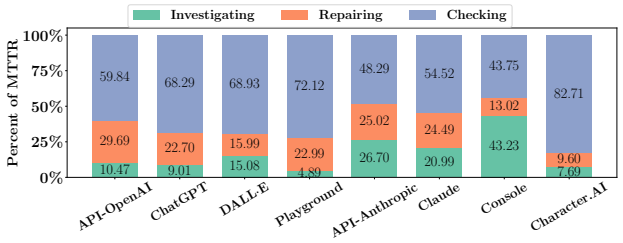


Figure 4: Percent of time spent in the Investigating, Repairing, and Checking periods, from the overall duration for failure resolution [%].

by providers. For each analysis, we begin with key observations, followed by detailed descriptions and discussions.

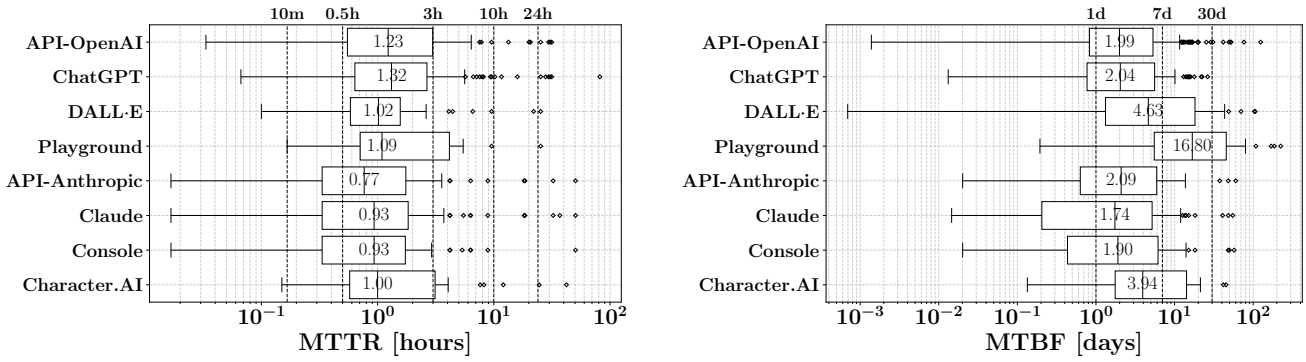
Observation #1: *Most incident reports lack information for all statuses. Although 100% of the incidents have been resolved, only 6.15% of reports disclose a postmortem.*

Updated status information is important for users waiting for a service to recover so that they can plan their work, recovery, and communication with customers. Table 5 gives the numbers and percent of different combinations of statuses for all reports. In most cases, incident reports do not include every status. Cases with $S_5(postmortem)$ account for the fewest percent, with only 6.15% of incident reports having a postmortem. The most prevailing case combination is S_1-S_3-S_4 (24.39%), in which the important status $S_2(identified)$ is missing. This means the operators do not communicate to the users that they have identified the issue. 20.48% of cases only provide information about $S_1(investigating)$ and $S_4(resolved)$ statuses. 14.34% of reports update every status (S_1-S_2-S_3-S_4) throughout the duration of incidents, while 11.55% only update once the incidents have been resolved (S_4).

These results indicate that operators fail to communicate the state of a failure to the user and update the user about potential fix times. Users must create their own failure models [21] and fault-tolerance systems [48] and expect little input from the operator.

Observation #2: *ChatGPT includes postmortems in 12.91% of its reports. Anthropic's reports contain the fewest postmortems, with none provided for its API and Console services.*

The status combinations also vary depending on different services, as Figure 3 shows. The primary status combination for OpenAI services is S_1-S_3-S_4, representing 29.96% for API, 41.13% for ChatGPT, 38.71% for DALL·E, and 31.25% for Playground. In contrast, Anthropic and Character.AI primarily use the S_1-S_4 combination, accounting for over 30% of each service. OpenAI publishes



(a) Mean Time To Resolve (Shorter is better). **(b) Mean Time Between Failures (Longer is better).**

Figure 5: Distribution of MTTR and MTBF by service, with median values indicated.

more status information in incident reports compared to Anthropic and Character.AI.

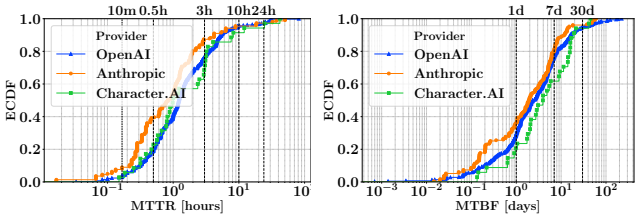
> **Observation #3:** *Claude spent the longest time on the periods of investigating (1.35 hours), repairing (1.72 hours), and checking (2.05 hours).*

The time a service takes to resolve incidents affects the fault-tolerance strategies a user can use. For example, users can maintain a local cache to tolerate very short failures. Table 6 shows the mean value of different model parameters. For MTTR and MTBF, because some records don't have *investigating* timestamps, we use the minimum ones before the resolved status here (This also explains why Claude has the longest P_I, P_R, and P_C respectively, but it does not have the longest *MTTR*). The learning period (P_L) takes the longest time (2.94 days) in all services.

> **Observation #4:** *Significant differences are observed across the 8 services in the percentage of periods within failure resolutions. For Character.AI, 82.71% of the time is spent on monitoring and checking if the fix is stable and effective. Anthropic services spent more percent of the time for investigating and resolving issues than OpenAI services.*

Figure 4 shows the percent of the 3 periods (Investigating, Repairing, Checking) in *MTTR*. The majority of the resolution time is used for checking, ranging from the highest 82.71% for Character.AI to the lowest 43.75% for Console. Most services spent more percent of the time on repairing than investigating, except for API and Console from Anthropic. Anthropic API spent 26.70% of the time investigating issues, while Console spent 43.23% in the same period.

The large fraction of the time used for checking indicates that deploying a fix to production takes a long time. Operators should employ faster testing and continuous deployment techniques to deploy fixes faster [73]. However, this is challenging as LLMs are a new technology, and there isn't much work on improving testing and deployment time.



(a) MTTR. **(b) MTBF.**

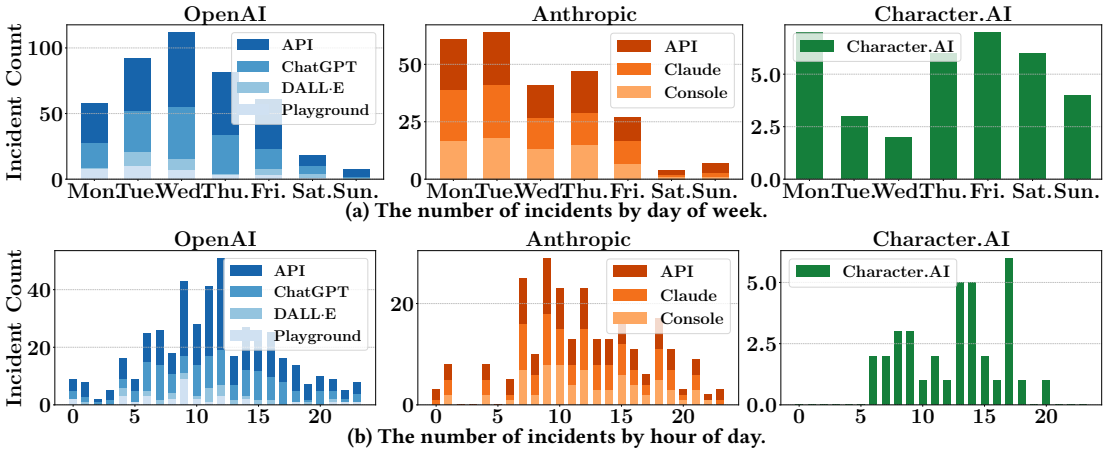
Figure 6: MTTR and MTBF by provider, ECDF plot. The closer the line is to the upper left, the shorter the time it takes.

> **Observation #5:** *OpenAI's API and ChatGPT recover slower from failures than Anthropic's API and Claude, with 1.6x and 1.4 longer MTTR, respectively.*

Figure 5a depicts *MTTR* of the 8 LLM service incidents, showing how quickly failures are resolved. Most failures are resolved between 0.5 and 3 hours, with the median values around 1 hour. APIs and chatbots are the most popular LLM services. The median *MTTR* of OpenAI API (1.23 hours) is 1.6x longer than Anthropic API (0.77 hours). Similarly, the median *MTTR* of ChatGPT (1.32 hours) is 1.4x longer than Claude (0.93 hours).

> **Observation #6:** *Playground is the most reliable service (16.80 median MTBF), followed by DALL·E (4.53 median MTBF) and Character.AI (3.94 median MTBF). OpenAI's ChatGPT is more reliable than Anthropic's Claude, though its API is less reliable in comparison.*

Awareness of how frequently a service fails is important for users to assess the reliability they can offer when they depend on the service. It's also important to assess which fault-tolerance mechanisms they should use as each has a different overhead. For example, active replication, frequent checkpointing, or infrequent checkpointing. Figure 5b depicts *MTBF* of the 8 LLM incidents, showing how frequently failures occur. The *MTBF* of failures varies significantly across services. The most reliable service is Playground, with a median *MTBF* of 16.80 days, which is nearly 9.66 times higher than the lowest median *MTBF* of 1.74 days from Claude. The median *MTBF* for OpenAI's API is 1.99 days, which is lower than Anthropic's API at 2.09 days; however, ChatGPT at 2.04 days is higher than Claude's 1.74 days. DALL·E and Character.AI are relatively reliable services, with median *MTBF* values of 4.63 days and 3.94 days, respectively.



(a) The number of incidents by day of week.

(b) The number of incidents by hour of day.

Figure 7: Temporal distributions for incidents, PDT time.

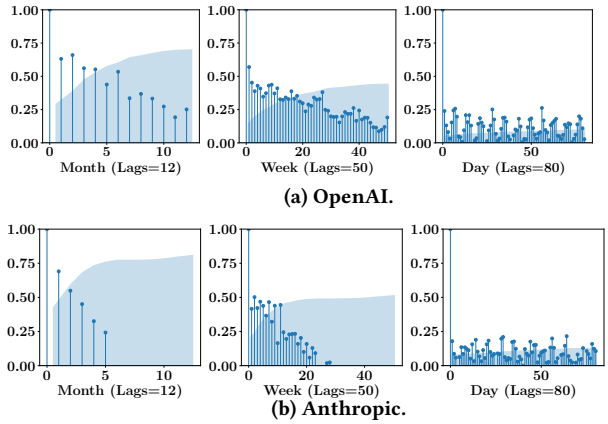
The MTBF of LLM services (4-40 days) is much higher than the MTBF of single-node failure (0.25-1 day) and system-wide failure (6.6 days) in other large-scale systems [24, 43]. This indicates that LLM operators use effective fault-tolerance mechanisms. It also indicates that users can use low-overhead fault-tolerance techniques like infrequent checkpointing to provide a reliable service that depends on LLMs.

Observation #7: *Over 90% of the incidents end within 10 hours for all measured providers. Specifically, Anthropic's services resolved failures more quickly but also experienced the highest frequency of incidents, based on its MTTR (2.70 hours) and MTBF (5.22 days) on average.*

To understand how the MTTR and MTBF values are distributed and compare the distributions for different LLM providers, Figure 6 displays the Empirical Cumulative Distribution Function (ECDF) plot of *MTTR* in hours and *MTBF* in days, grouped by provider. It also marks vertically different time points for better observation and comparison.

A small percentage of incidents can be resolved within 10 minutes, such as 8.55% for Anthropic. Anthropic also solved the highest percent of incidents (37.18%) in 0.5 hours, significantly more than OpenAI (19.72%) and Character.AI (22.86%). Most failures are addressed within 3 hours, with 74.25% for OpenAI, 82.91% for Anthropic, and 68.57% for Character.AI. After 10 hours, 92.34% of OpenAI, 90.60% of Anthropic, and 91.43% of Character.AI's failures are solved. However, a small proposition of failures for all providers lasted over 1 day, with 6.03%, 7.69%, and 5.71%, respectively. Overall, Anthropic resolved failures more quickly, despite a higher percentage of extreme cases lasting over 1 day.

Although Anthropic resolves failures the fastest, it also encounters them most frequently, with every 5.22 days on average. In contrast, OpenAI and Character.AI are more reliable, with failure occurring every 8.48 and 8.74 days, respectively. A notable percentage of incidents occur within a day: 35.47% for Anthropic, 28.77% for OpenAI, and 20.00% for Character.AI. Within 1 week interval, nearly three-quarters of failures occur for OpenAI (75.64%) and for Anthorpic (78.63%), with a slightly lower rate for Character.AI (60.00%). Over 90% of incidents for all providers happen within a month of each other.



(a) OpenAI.

(b) Anthropic.

Figure 8: Auto-correlations with the numbers of incidents aggregated at different time granularities.

Our findings indicate that users of LLMs should expect failures regularly (at least once a month). Therefore, failure should not be an exceptional event but should be baked into the users' normal operating procedure.

5 FAILURE PATTERNS OVER TIME

This section conducts time series analyses to examine the failure patterns over time, including: (1) Weekly and daily incident distributions, (2) Auto-correlations in different time intervals; and (3) Daily service available time.

5.1 Temporal Distributions

Observation #8: *OpenAI and Anthropic's services failures exhibit periodic patterns that more frequent on weekdays than on weekends. However, Character.AI has fewer failures on Tuesdays and Wednesdays. All services show a diurnal pattern of failures, typically peaking from 8:00 to 16:00.*

To investigate the temporal distributions of LLM incidents, we aggregate service incidents by day of week in Figure 7a, and hour of day in Figure 7b. Incident times are given in local time (PDT)

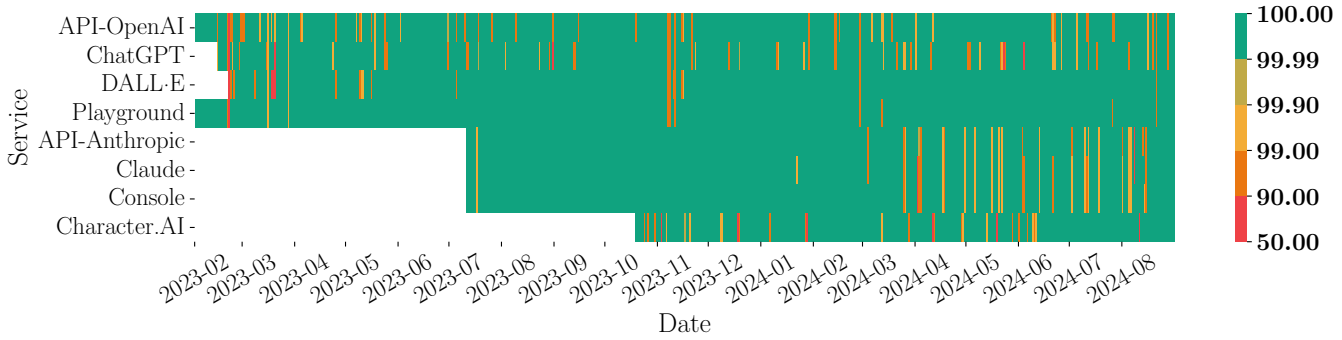


Figure 9: Service daily availability by scaled outage minutes [%]. Some services started reporting later (see Section 3.2).

Table 7: Service availability by scaled outage minutes (from all periods).

Service	Min	Max	Mean	Median	=100%	>=99.999%	>=99.99%	>=99.9%	>=99%	>=90%	<90%
API-OpenAI	80.57%	100.0%	99.82%	100.0%	92.68%	92.68%	92.68%	92.84%	95.30%	99.77%	0.23%
ChatGPT	77.15%	100.0%	99.66%	100.0%	88.85%	88.85%	88.85%	89.20%	93.10%	99.12%	0.88%
DALL·E	74.17%	100.0%	99.78%	100.0%	95.88%	95.88%	95.88%	95.88%	96.77%	99.28%	0.72%
Playground	82.57%	100.0%	99.94%	100.0%	98.08%	98.08%	98.08%	98.16%	98.88%	99.84%	0.16%
API-Anthropic	85.44%	100.0%	99.92%	100.0%	94.02%	94.02%	94.02%	94.26%	97.61%	99.76%	0.24%
Claude	82.02%	100.0%	99.84%	100.0%	93.06%	93.06%	93.06%	93.06%	97.13%	99.52%	0.48%
Console	82.56%	100.0%	99.89%	100.0%	93.54%	93.54%	93.54%	93.54%	97.61%	99.76%	0.24%
Character.AI	85.33%	100.0%	99.59%	100.0%	90.88%	90.88%	90.88%	91.19%	94.03%	98.11%	1.89%

as they were originally reported in PDT. OpenAI and Anthropic's services display a clear weekday pattern in incidents, with significantly more failures on weekdays than on weekends. In contrast, Character.AI follows a different pattern, with fewer failures occurring on Tuesdays and Wednesdays. This may be due to the differing purposes of using LLM services: Character.AI is primarily used for leisure [69], while API and conversational services are more often used for work-related tasks, such as writing and coding [66]. All services exhibit a diurnal pattern, with incident peaks occurring during typical work hours, such as 8:00 to 16:00, and lower at night hours. Similar periodic failure patterns are also found in machine learning jobs [14, 15, 65], deep learning jobs [38], and general user request in BurstGPT workloads [67].

5.2 Auto-correlations

Observation #9: *LLM service failures have strong monthly auto-correlations, with OpenAI incidents showing longer-lasting correlations than Anthropic. Both services display distinct weekly periodicity.*

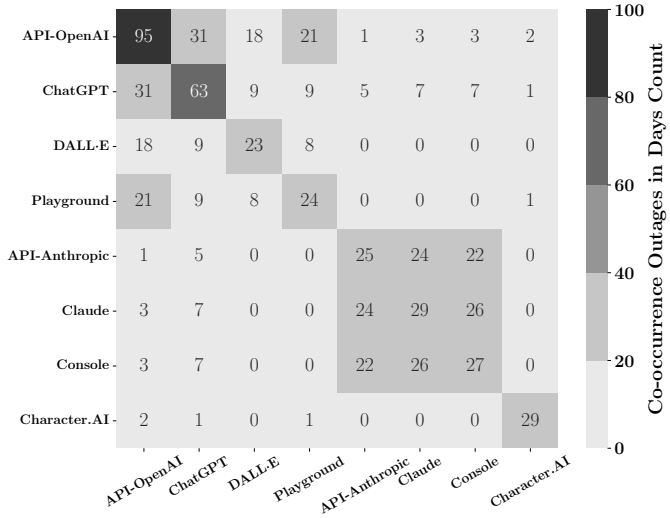
We investigate if a failure is immediately followed by another failure and how often it happens. Figure 8 depicts the auto-correlation for the number of incidents at month, week, and day granularities. Confidence intervals are drawn as the blue area. By default, this is set to a 95% confidence interval, suggesting that correlation values outside of this area are significant, which are real patterns rather than random noise. Lags represent the time intervals at which a time series is compared to itself, and autocorrelation measures how similar a time series is to itself at different lags.

For OpenAI, the auto-correlation plots display significant positive correlations up to lag 3 on a monthly scale and up to lag 12 on a weekly scale, indicating that both monthly and weekly incidents are strongly related to their previous values. Anthropic shows similar correlations with shorter lags, with up to lag 1 for monthly data and lag 7 for weekly data, likely affected by Anthropic's shorter operational history. The consistent but gradual decay in auto-correlations at every 7-day interval for both OpenAI and Anthropic suggests strong weekly periodic behavior, supporting our previous findings in Figure 7. Compared to the auto-correlations observed in ML failures from the previous study [15], the auto-correlation in LLM service failures shows stronger periodic trends. The periodic characteristics can be utilized to predict future incidents, similar to workload failure predictions [40].

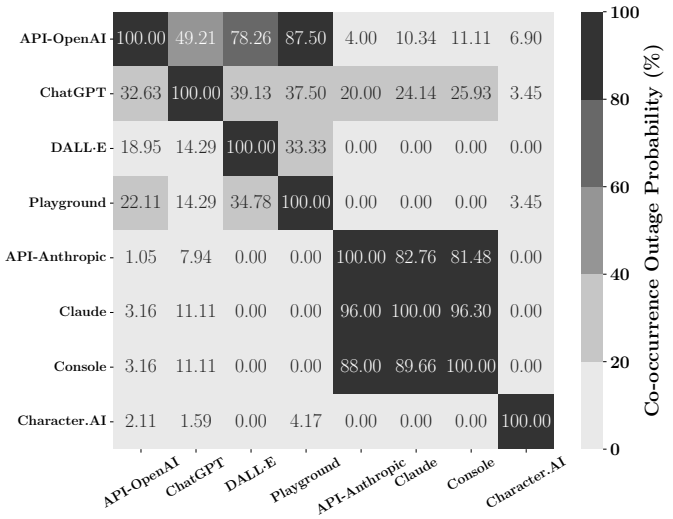
5.3 Service Availability Over Time

Observation #10: *ChatGPT is the least consistently available service, with only 88.85% of days fully available, followed by Character.AI at 90.88%. Availability of Anthropic's services declined after April 2024, possibly due to product release and the sharp increase in user demands.*

We provide a high-level view of what level of service reliability a user can expect in this section. Figure 9 shows the service daily availability by scaled outage minutes, from February 2023 to August 2024. We categorized availability into five levels based on their value ranges. Days without outages, which mean full service availability, are colored green, while days with longer outage durations are represented by colors closer to red. Table 7 gives the specific statistics of service availability. DALL·E and Playground have the highest availability, with 95.88% and 98.08% of days fully accessible, respectively. In contrast, ChatGPT is the least available



(a) Co-occurrence outages in days count.



(b) Conditional probabilities of co-occurrence outages. Notes: y-axis = service A, x-axis = service B, cells = P(A|B).

Figure 10: Co-occurrence of outages between service pairs.

service, with only 88.85% of days fully accessible. Availability of Anthropic's services declined after April 2024, possibly due to product release and the sharp increase in user demands [2, 3]. Character.AI also shows noticeable instability, with only 90.88% of days fully available and over 1.89% of days with availability falling below 90%.

6 CO-OCCURRENCE OF FAILURES

This section examines the co-occurrence of failures across services. When an outage occurs in one service, do other services also experience outages? How about the impacted range of incidents for different services and providers? To address these questions, we analyze (1) the co-occurrence of outages, and (2) the impact range of incidents.

6.1 Co-occurrence of Outages

Observation #11: *Co-occurrence is particularly high among services from the same provider, suggesting a strong interdependence between those services. For Anthropic's services, the likelihood of any two services experiencing outages on the same day is over 80%, indicating a severe lack of isolation across different services.*

The Figure 10a shows the number of co-occurring outages across different services on the same day. The counts of outages may be affected by the maximum number of outages. For example, the number of co-occurrences among Anthropic services is lower than for OpenAI services, however, the probability of co-occurrence among Anthropic services is higher. To avoid this impact of the number of outages, we also give the conditional probabilities of co-occurring outages in Figure 10b. The conditional probability indicates the likelihood that if service B experiences an outage, service A will also experience an outage. For instance, the 49.21% in row 1, column 2 means that if ChatGPT is down, there is a 49.21% chance that OpenAI's API will also experience an outage on the same day. The probability that service A is also outage while service

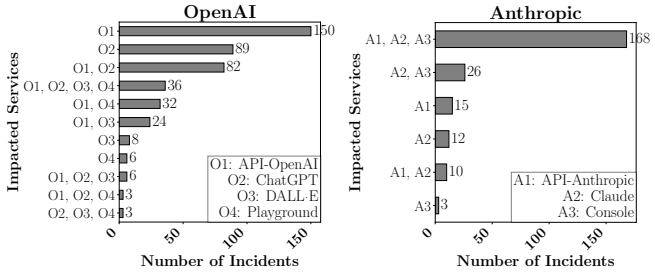


Figure 11: Impact Services of OpenAI and Anthropic incidents, respectively.

B is outage can be formulated in:

$$P(A|B) = \frac{P(A \cap B)}{P(A) \times P(B)} = \frac{O_{AB}}{O_B} \quad (4)$$

O_{AB} represents the number of days when both services A and B experience outages simultaneously, while O_B indicates the number of days that service B has an outage.

The heatmaps show that co-occurrence is notably high among services from the same provider. For OpenAI services, the API is more likely to have an outage with DALL·E (78.26%) and Playground (87.50%) than ChatGPT (49.21%). For Anthropic's services, the likelihood of any two services experiencing outages on the same day is extremely high over 80%, this may be caused by a lack of isolation across different services. There is no correlation observed between services from different providers. The lack of correlation suggests that user can use one service as the other's backup to increase their reliability. The difference in co-occurrence between OpenAI and Anthropic suggests that outages could be reduced through better service isolation.

Table 8: Percentage of the number of impacted services.

	1	2	3	4
OpenAI	57.63%	31.44%	2.73%	8.20%
Anthropic	12.82%	15.38%	71.79%	-

6.2 Impact Range of Incidents

Observation #12: *71.79% of Anthropic incidents affect all its services, compared to only 8.20% for OpenAI.*

The incident reports indicate that one incident can impact several services, which is the impact range of incidents. Figure 11 gives the impacted services combinations of OpenAI and Anthropic incidents based on their reports, respectively. For Anthropic, the majority of incidents (71.79%) impact its 3 services jointly. However, for OpenAI, only 8.20% of services are impacted together, and over half (57.63%) of incidents affect a single service.

7 LIMITATIONS AND VALIDITY

The *generality* of our work is limited to the services we analyze. We analyze LLM-related services from three popular operators, including the currently most popular (OpenAI). However, other, very different services could exist, e.g., allowing users to self-host LLMs (e.g., Anyscale), and LLM services from big cloud providers and/or used primarily internally (e.g., Google Gemini).

The *accuracy* of our failure dataset is limited to what the LLM operators themselves report. That makes our data subject to the operators' bias. Prior work [26, 59] suggests the operator's reports already capture the most user-visible failures as those generate widespread social media coverage, making it difficult for the operators to hide failure; to confirm this for LLM services, we need to collect data from other sources such as the user devices [10] or user failure reports [26, 59].

The *depth* of our analysis regarding the root cause of failures is limited. We glean limited information from the operators' failure reports regarding the hardware and software infrastructure. To confirm our findings, we would ideally use detailed infrastructure and application-level data. However, this requires active help from the service operators, e.g., releasing their system traces as Google did with its cluster workloads [23].

The *scope* of our work is limited to LLM services. We ignore other deep learning services such as Image Generation (e.g., Stable Diffusion, Midjourney), Translation (e.g., DeepL), etc. However, we believe the operational characteristics of LLM services are valuable in and of themselves. LLMs have gained broad general public adoption and mindshare, as described in Section 1. LLM services now also support multi-modal use cases such as image generation and image-based question answering, making them some of the most general deep learning tools currently available.

8 RELATED WORK

Overall, this work complements the existing body of work on failure characterization, modeling, and more generally failure-recovery, with a focus on the emerging area of LLM services. Ours is the first comprehensive, longitudinal data collection and empirical characterization of public LLM services.

Operational failure characterization of workstations [34], HPC sites [24], clouds [22], big data jobs [54], networks [47], storage devices [1], CPUs [25], and GPUs [60] has led to improved application designs and fault-tolerance mechanisms. Leading from these, we have better failure detection [10, 26], checkpointing [21], retry [48], and replication mechanisms [57]. However, these do not cover deep learning and particularly LLM services.

There is existing work on operational characteristics of GPUs for deep learning [39], ML jobs on HPC clusters [14], and deep-learning clusters [35]. However, no work has described the *operational failure characteristics* of user-facing deep learning services. Our study addresses this gap, focusing on LLMs.

Deep learning workloads have been characterized including their GPU utilization [27, 38], network characteristics [9], and storage characteristics [13]. User-facing machine learning workloads have also been characterized [68]. The studies complement our work as they explore different hardware/software stack layers. We complement the studies by enhancing the community's understanding of LLM failures at the user-facing application layer.

LLM workloads have been characterized at the preliminary-level for training [28], fine-tuning [67, 70], and inference [37]. Failures have been assessed briefly; e.g., found to occur frequently (~9 hour MTBF) in LLM training [28], compared to around 4 days MTBF for the user-facing services in this work. Fine-tuning and inference workloads have not been characterized, especially concerning failures. Ours is the first study to focus on failures occurring in public LLM services, with unique contributions in longitudinal analysis and in collecting comprehensive data from multiple services.

9 CONCLUSION

Understanding the characteristics of failures in the operation of public LLM services has become a stringent problem, driven by the rapid increase in the popularity of such services, market competitiveness, and increasingly self-reported presence of such failures by LLM service providers. Addressing this problem, in this work we have conducted a comprehensive empirical characterization of outages and incidents in public LLM services.

We have collected long-term failure data from 8 relevant services, and produced corresponding outage and incident datasets under the same failure model. We analyzed the failure characteristics of these services, per service and overall, and specifically analyzed MTTR and MTBF, time spent in various stages in the recovery process, service availability over hourly and daily intervals, diurnal and weekly availability distributions, auto-correlations; and failure co-occurrence for pairs and groups of services per provider, and for pairs of services across providers.

In our analysis, we emphasized over 10 observations, which scientists, engineers, and users could include directly in their knowledge base, and from which improvements to LLM systems could occur in time.

For the future, we aim to lead a community effort where datasets, collected long-term and processed to provide similar information, can be shared. Future analysis could include the popular LLM services powered by Google DeepMind's Gemini and Mistral AI's Mixtral, and the promising emerging LLMs from US-, China-, and Japan-based big tech companies, e.g., NVIDIA's Nemotron.

ACKNOWLEDGMENTS

We thank the China Scholarship Council (CSC) for supporting Xiaoyu Chu. We thank the support of Netherlands-funded projects NWO OffSense and GFP 6G FNS, and EU-funded projects MCSA-RISE Cloudstars and Horizon Graph-Massivizer.

REFERENCES

[1] Jacob Alter, Ji Xue, Alma Dimnaku, and Evgenia Smirni. 2019. SSD failures in the field: symptoms, causes, and prediction models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2019, Denver, Colorado, USA, November 17-19, 2019*, Michela Taufer, Pavan Balaji, and Antonio J. Peña (Eds.). ACM, 75:1–75:14. https://doi.org/10.1145/3295500.3356172

[2] Anthropic. 2024. Release notes of Anthropic API. https://docs.anthropic.com/en/release-notes/api.

[3] Anthropic. 2024. Release notes of Claude. https://docs.anthropic.com/en/release-notes/claude-apps.

[4] Anthropic API. 2024. https://www.anthropic.com/api.

[5] OpenAI API. 2024. https://openai.com/index/openai-api/.

[6] Internet Archive. 2024. Archived page of OpenAI status at 2024/04/10. https://web.archive.org/web/20240410235249/https://downdetector.com/status/openai/, Accessed: 2024-09-30.

[7] Atlassian. 2024. Atlassian Support. https://support.atlassian.com/statuspage/docs/display-historical-uptime-of-components/.

[8] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl E. Landwehr. 2004. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Trans. Dependable Secur. Comput.* 1, 1 (2004), 11–33. https://doi.org/10.1109/TDSC.2004.2

[9] Ammar Ahmad Awan, Arpan Jain, Ching-Hsiang Chu, Hari Subramoni, and Dhabaleswar K. Panda. 2020. Communication Profiling and Characterization of Deep-Learning Workloads on Clusters With High-Performance Interconnects. *IEEE Micro* 40, 1 (2020), 35–43. https://doi.org/10.1109/MM.2019.2949986

[10] Sam Burnett, Lily Chen, Douglas A. Creager, Misha Efimov, Ilya Grigorik, Ben Jones, Harsha V. Madhyastha, Pavlos Papageorge, Brian Rogan, Charles Stahl, and Julia Tuttle. 2020. Network Error Logging: Client-side measurement of end-to-end web service reliability. In *17th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2020, Santa Clara, CA, USA, February 25-27, 2020*, Ranjita Bhagwan and George Porter (Eds.). USENIX Association, 985–998. https://www.usenix.org/conference/nsdi20/presentation/burnett

[11] Character.AI. 2024. https://character.ai/.

[12] OpenAI ChatGPT. 2024. https://chatgpt.com/.

[13] Steven Wei Der Chien, Stefano Markidis, Chaitanya Prasad Sishtla, Luís Santos, Pawel Andrzej Herman, Sai Narasimhamurthy, and Erwin Laure. 2018. Characterizing Deep-Learning I/O Workloads in TensorFlow. In *3rd IEEE/ACM International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems, PDSW-DISCS@SC 2018, Dallas, TX, USA, November 12, 2018*. IEEE, 54–63. https://doi.org/10.1109/PDSW-DISCS.2018.00011

[14] Xiaoyu Chu, Daniel Hofstätter, Shashikant Ilager, Sacheendra Talluri, Duncan Kampert, Damian Podareanu, Dmitry Duplyakin, Ivona Brandic, and Alexandru Iosup. 2024. Generic and ML Workloads in an HPC Datacenter: Node Energy, Job Failures, and Node-Job Analysis. arXiv:2409.08949 [cs.DC] https://arxiv.org/abs/2409.08949

[15] Xiaoyu Chu, Sacheendra Talluri, Laurens Versluis, and Alexandru Iosup. 2023. How Do ML Jobs Fail in Datacenters? Analysis of a Long-Term Dataset from an HPC Cluster. In *Proceedings of the International Conference on Performance Engineering, Coimbra, Portugal, April, 2023*.

[16] Anthropic Claude. 2024. https://claude.ai/.

[17] Anthropic Console. 2024. https://console.anthropic.com/.

[18] OpenAI DALL·E. 2024. https://openai.com/index/dall-e-3/.

[19] DownDetector. 2024. OpenAI user reports. https://downdetector.com/status/openai/.

[20] Marc Gamell, Daniel S. Katz, Hemanth Kolla, Jacqueline Chen, Scott Klasky, and Manish Parashar. 2014. Exploring Automatic, Online Failure Recovery for Scientific Applications at Extreme Scales. In *International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2014, New Orleans, LA, USA, November 16-21, 2014*, Trish Damkroger and Jack J. Dongarra (Eds.). IEEE Computer Society, 895–906. https://doi.org/10.1109/SC.2014.78

[21] Rohan Garg, Tirthak Patel, Gene Cooperman, and Devesh Tiwari. 2018. Shiraz: Exploiting System Reliability and Application Resilience Characteristics to Improve Large Scale System Throughput. In *48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2018, Luxembourg City, Luxembourg, June 25-28, 2018*. IEEE Computer Society, 83–94. https://doi.org/10.1109/DSN.2018.00021

[22] Peter Garraghan, Paul Townend, and Jie Xu. 2014. An Empirical Failure-Analysis of a Large-Scale Cloud Computing Environment. In *15th International IEEE Symposium on High-Assurance Systems Engineering, HASE 2014, Miami Beach, FL, USA, January 9-11, 2014*. IEEE Computer Society, 113–120. https://doi.org/10.1109/HASE.2014.24

[23] Google. 2019. Google Cluster Traces 2019. https://github.com/google/cluster-data?tab=readme-ov-file.

[24] Saurabh Gupta, Tirthak Patel, Christian Engelmann, and Devesh Tiwari. 2017. Failures in large scale systems: long-term measurement, analysis, and implications. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2017, Denver, CO, USA, November 12 - 17, 2017*, Bernd Mohr and Padma Raghavan (Eds.). ACM, 44. https://doi.org/10.1145/3126908.3126937

[25] Peter H. Hochschild, Paul Turner, Jeffrey C. Mogul, Rama Govindaraju, Parthasarathy Ranganathan, David E. Culler, and Amin Vahdat. 2021. Cores that don't count. In *HotOS '21: Workshop on Hot Topics in Operating Systems, Ann Arbor, Michigan, USA, June, 1-3, 2021*, Sebastian Angel, Baris Kasikci, and Eddie Kohler (Eds.). ACM, 9–16. https://doi.org/10.1145/3458336.3465297

[26] Jiyao Hu, Zhenyu Zhou, Xiaowei Yang, Jacob Malone, and Jonathan W. Williams. 2020. CableMon: Improving the Reliability of Cable Broadband Networks via Proactive Network Maintenance. In *17th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2020, Santa Clara, CA, USA, February 25-27, 2020*, Ranjita Bhagwan and George Porter (Eds.). USENIX Association, 619–632. https://www.usenix.org/conference/nsdi20/presentation/hu-jiyao

[27] Qinghao Hu, Peng Sun, Shengen Yan, Yonggang Wen, and Tianwei Zhang. 2021. Characterization and prediction of deep learning workloads in large-scale GPU datacenters. In *International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2021, St. Louis, Missouri, USA, November 14-19, 2021*, Bronis R. de Supinski, Mary W. Hall, and Todd Gamblin (Eds.). ACM, 104. https://doi.org/10.1145/3458817.3476223

[28] Qinghao Hu, Zhisheng Ye, Zerui Wang, Guoteng Wang, Meng Zhang, Qiaoling Chen, Peng Sun, Dahua Lin, Xiaolin Wang, Yingwei Luo, Yonggang Wen, and Tianwei Zhang. 2024. Characterization of Large Language Model Development in the Datacenter. In *21st USENIX Symposium on Networked Systems Design and Implementation, NSDI 2024, Santa Clara, CA, April 15-17, 2024*, Laurent Vanbever and Irene Zhang (Eds.). USENIX Association, 709–729. https://www.usenix.org/conference/nsdi24/presentation/hu

[29] Singularity Hub. 2022. OpenAI Says DALL-E Is Generating Over 2 Million Images a Day. https://singularityhub.com/2022/10/03/openai-says-dall-e-is-generating-over-2-million-images-a-day-and-thats-just-table-stakes/.

[30] IDC. 2024. A Deep Dive Into Global AI and Generative AI Spending. https://blogs.idc.com/2024/08/16/a-deep-dive-into-idcs-global-ai-and-generative-ai-spending/.

[31] Anthropic Incident. 2024. https://status.anthropic.com/history.

[32] Character.AI Incident. 2024. https://status.character.ai/history.

[33] OpenAI Incident. 2024. https://status.openai.com/history.

[34] Bahman Javadi, Derrick Kondo, Alexandru Iosup, and Dick H. J. Epema. 2013. The Failure Trace Archive: Enabling the comparison of failure measurements and models of distributed systems. *J. Parallel Distributed Comput.* 73, 8 (2013), 1208–1223. https://doi.org/10.1016/J.JPDC.2013.04.002

[35] Myeongjae Jeon, Shivaram Venkataraman, Amar Phanishayee, Junjie Qian, Wencong Xiao, and Fan Yang. 2019. Analysis of Large-Scale Multi-Tenant GPU Clusters for DNN Training Workloads. In *Proceedings of the 2019 USENIX Annual Technical Conference, USENIX ATC 2019, Renton, WA, USA, July 10-12, 2019*, Dahlia Malkhi and Dan Tsafrir (Eds.). USENIX Association, 947–960. https://www.usenix.org/conference/atc19/presentation/jeon

[36] Jing Yu Koh, Daniel Fried, and Russ Salakhutdinov. 2023. Generating Images with Multimodal Language Models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.). http://papers.nips.cc/paper_files/paper/2023/hash/43a69d143273bd8215578bde887bb552-Abstract-Conference.html

[37] Małgorzata Łazuka, Andreea Anghel, and Thomas Parnell. 2024. LLM-Pilot: Characterize and Optimize Performance of your LLM Inference Services. *arXiv preprint arXiv:2410.02425* (2024).

[38] Baolin Li, Rohin Arora, Siddharth Samsi, Tirthak Patel, William Arcand, David Bestor, Chansup Byun, Rohan Basu Roy, Bill Bergeron, John T. Holodnak, Michael Houle, Matthew Hubbell, Michael Jones, Jeremy Kepner, Anna Klein, Peter Michaleas, Joseph McDonald, Lauren Milechin, Julie Mullen, Andrew Prout, Benjamin Price, Albert Reuther, Antonio Rosa, Matthew L. Weiss, Charles Yee, Daniel Edelman, Allan Vanterpool, Anson Cheng, Vijay Gadepally, and Devesh Tiwari. 2022. AI-Enabling Workloads on Large-Scale GPU-Accelerated System: Characterization, Opportunities, and Implications. In *IEEE International Symposium on High-Performance Computer Architecture, HPCA 2022, Seoul, South Korea, April 2-6, 2022*. IEEE, 1224–1237. https://doi.org/10.1109/HPCA53966.2022.00093

[39] Guanpeng Li, Siva Kumar Sastry Hari, Michael B. Sullivan, Timothy Tsai, Karthik Pattabiraman, Joel S. Emer, and Stephen W. Keckler. 2017. Understanding error propagation in deep learning neural network (DNN) accelerators and applications. In *Proceedings of the International Conference for High Performance Computing,*

Networking, Storage and Analysis, SC 2017, Denver, CO, USA, November 12 - 17, 2017, Bernd Mohr and Padma Raghavan (Eds.). ACM, 8. https://doi.org/10.1145/3126908.3126964

[40] Jie Li, Rui Wang, Ghazanfar Ali, Tommy Dang, Alan Sill, and Yong Chen. 2023. Workload Failure Prediction for Data Centers. In *16th IEEE International Conference on Cloud Computing, CLOUD 2023, Chicago, IL, USA, July 2-8, 2023*. IEEE, 479–485. https://doi.org/10.1109/CLOUD60044.2023.00064

[41] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023. Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.). http://papers.nips.cc/paper_files/paper/2023/hash/43e9d647ccd3e4b7b5baab53f0368686-Abstract-Conference.html

[42] Yujie Lu, Xianjun Yang, Xiujun Li, Xin Eric Wang, and William Yang Wang. 2023. LLMScore: Unveiling the Power of Large Language Models in Text-to-Image Synthesis Evaluation. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.). http://papers.nips.cc/paper_files/paper/2023/hash/47f30d67bce3e9824928267e9355420f-Abstract-Conference.html

[43] Catello Di Martino, Zbigniew T. Kalbarczyk, Ravishankar K. Iyer, Fabio Baccanico, Joseph Fullop, and William Kramer. 2014. Lessons Learned from the Analysis of System Failures at Petascale: The Case of Blue Waters. In *44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2014, Atlanta, GA, USA, June 23-26, 2014*. IEEE Computer Society, 610–621. https://doi.org/10.1109/DSN.2014.62

[44] OpenAI. 2024. Elevated errors in ChatGPT. https://status.openai.com/incidents/w20mcckg1748, Accessed: 2024-09-30.

[45] OpenAI. 2024. Start using ChatGPT instantly (Apr 1, 2024). https://help.openai.com/en/articles/6825453-chatgpt-release-notes#h_126f2fa257.

[46] OpenAI Playground. 2024. https://platform.openai.com/playground/chat.

[47] Rahul Potharaju and Navendu Jain. 2013. When the network crumbles: an empirical study of cloud network failures and their impact on services. In *ACM Symposium on Cloud Computing, SOCC '13, Santa Clara, CA, USA, October 1-3, 2013*, Guy M. Lohman (Ed.). ACM, 15:1–15:17. https://doi.org/10.1145/2523616.2523638

[48] Mia Primorac, Katerina J. Argyraki, and Edouard Bugnion. 2021. When to Hedge in Interactive Services. In *18th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2021, April 12-14, 2021*, James Mickens and Renata Teixeira (Eds.). USENIX Association, 373–387. https://www.usenix.org/conference/nsdi21/presentation/primorac

[49] Similarweb Pro. 2024. https://pro.similarweb.com/.

[50] Charles Reiss, Alexey Tumanov, Gregory R. Ganger, Randy H. Katz, and Michael A. Kozuch. 2012. Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In *ACM Symposium on Cloud Computing, SOCC '12, San Jose, CA, USA, October 14-17, 2012*, Michael J. Carey and Steven Hand (Eds.). ACM, 7. https://doi.org/10.1145/2391229.2391236

[51] Reuters. 2023. ChatGPT sets record for fastest-growing user base - analyst note. https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/

[52] Reuters. 2024. Google-backed Anthropic releases Claude chatbot across Europe. https://www.reuters.com/technology/google-backed-anthropic-releases-claude-chatbot-across-europe-2024-05-13/.

[53] Reuters. 2024. OpenAI says ChatGPT's weekly users have grown to 200 million. https://www.reuters.com/technology/artificial-intelligence/openai-says-chatgpts-weekly-users-have-grown-200-million-2024-08-29/.

[54] Andrea Rosà, Lydia Y. Chen, and Walter Binder. 2015. Predicting and Mitigating Jobs Failures in Big Data Clusters. In *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2015, Shenzhen, China, May 4-7, 2015*. IEEE Computer Society, 221–230. https://doi.org/10.1109/CCGRID.2015.139

[55] Konstantinos I. Roumeliotis and Nikolaos D. Tselikas. 2023. ChatGPT and Open-AI Models: A Preliminary Review. *Future Internet* 15, 6 (2023). https://doi.org/10.3390/fi15060192

[56] Selenium. 2024. WebDriver Documentation. https://www.selenium.dev/documentation/webdriver/.

[57] Siqi Shen, Alexandru Iosup, Assaf Israel, Walfredo Cirne, Danny Raz, and Dick H. J. Epema. 2015. An Availability-on-Demand Mechanism for Datacenters. In *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2015, Shenzhen, China, May 4-7, 2015*. IEEE Computer Society, 495–504. https://doi.org/10.1109/CCGRID.2015.58

[58] Sacheendra Talluri, Alicja Luszczak, Cristina L. Abad, and Alexandru Iosup. 2019. Characterization of a Big Data Storage Workload in the Cloud. In *Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering, ICPE 2019, Mumbai, India, April 7-11, 2019*, Varsha Apte, Antinisca Di Marco, Marin Litoiu, and José Merseguer (Eds.). ACM, 33–44. https://doi.org/10.1145/3297663.3310302

[59] Sacheendra Talluri, Leon Overweel, Laurens Versluis, Animesh Trivedi, and Alexandru Iosup. 2021. Empirical Characterization of User Reports about Cloud Failures. In *IEEE International Conference on Autonomic Computing and Self-Organizing Systems, ACSOS 2021, Washington, DC, USA, September 27 - Oct. 1, 2021*, Esam El-Araby, Vana Kalogeraki, Danilo Pianini, Frédéric Lassabe, Barry Porter, Sona Ghahremani, Ingrid Nunes, Mohamed Bakhouya, and Sven Tomforde (Eds.). IEEE, 158–163. https://doi.org/10.1109/ACSOS52086.2021.00039

[60] Devesh Tiwari, Saurabh Gupta, James H. Rogers, Don Maxwell, Paolo Rech, Sudharshan S. Vazhkudai, Daniel Oliveira, Dave Londo, Nathan DeBardeleben, Philippe Olivier Alexandre Navaux, Luigi Carro, and Arthur S. Bland. 2015. Understanding GPU errors on large-scale HPC systems and the implications for system design and operation. In *21st IEEE International Symposium on High Performance Computer Architecture, HPCA 2015, Burlingame, CA, USA, February 7-11, 2015*. IEEE Computer Society, 331–342. https://doi.org/10.1109/HPCA.2015.7056044

[61] Anthropic Uptime. 2024. https://status.anthropic.com/uptime.

[62] Character.AI Uptime. 2024. https://status.character.ai/uptime.

[63] OpenAI Uptime. 2024. https://status.openai.com/uptime.

[64] Priyan Vaithilingam, Tianyi Zhang, and Elena L. Glassman. 2022. Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models. In *CHI '22: CHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA, 29 April 2022 - 5 May 2022, Extended Abstracts*, Simone D. J. Barbosa, Cliff Lampe, Caroline Appert, and David A. Shamma (Eds.). ACM, 332:1–332:7. https://doi.org/10.1145/3491101.3519665

[65] Laurens Versluis, Mehmet Çetin, Caspar Greeven, Kristian Laursen, Damian Podareanu, Valeriu Codreanu, Alexandru Uta, and Alexandru Iosup. 2023. Less is not more: We need rich datasets to explore. *Future Gener. Comput. Syst.* 142 (2023), 117–130. https://doi.org/10.1016/J.FUTURE.2022.12.022

[66] Jiayin Wang, Weizhi Ma, Peijie Sun, Min Zhang, and Jian-Yun Nie. 2024. Understanding User Experience in Large Language Model Interactions. *CoRR* abs/2401.08329 (2024). https://doi.org/10.48550/ARXIV.2401.08329 arXiv:2401.08329

[67] Yuxin Wang, Yuhan Chen, Zeyu Li, Xueze Kang, Zhenheng Tang, Xin He, Rui Guo, Xin Wang, Qiang Wang, Amelie Chi Zhou, and Xiaowen Chu. 2024. BurstGPT: A Real-world Workload Dataset to Optimize LLM Serving Systems. arXiv:2401.17644

[68] Qizhen Weng, Wencong Xiao, Yinghao Yu, Wei Wang, Cheng Wang, Jian He, Yong Li, Liping Zhang, Wei Lin, and Yu Ding. 2022. MLaaS in the Wild: Workload Analysis and Scheduling in Large-Scale Heterogeneous GPU Clusters. In *19th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2022, Renton, WA, USA, April 4-6, 2022*, Amar Phanishayee and Vyas Sekar (Eds.). USENIX Association, 945–960. https://www.usenix.org/conference/nsdi22/presentation/weng

[69] WIRED. 2024. The obsession with Character AI is becoming more common. https://wired.me/technology/character-ai-obsession/.

[70] Yuchen Xia, Jiho Kim, Yuhan Chen, Haojie Ye, Souvik Kundu, Cong Hao, and Nishil Talati. 2024. Understanding the Performance and Estimating the Cost of LLM Fine-Tuning. *CoRR* abs/2408.04693 (2024). https://doi.org/10.48550/ARXIV.2408.04693 arXiv:2408.04693

[71] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Ben Hu. 2024. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. *ACM Trans. Knowl. Discov. Data* 18, 6 (2024), 160:1–160:32. https://doi.org/10.1145/3649506

[72] FOX5 New York. 2024. ChatGPT recovers following outage affecting thousands of users. https://www.fox5ny.com/news/chatgpt-down-users-openai-internet, Accessed: 2024-09-30.

[73] Yang Zhang, Bogdan Vasilescu, Huaimin Wang, and Vladimir Filkov. 2018. One size does not fit all: an empirical study of containerized continuous deployment workflows. In *Proceedings of the 2018 ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/SIGSOFT FSE 2018, Lake Buena Vista, FL, USA, November 04-09, 2018*, Gary T. Leavens, Alessandro Garcia, and Corina S. Pasareanu (Eds.). ACM, 295–306. https://doi.org/10.1145/3236024.3236033