Public Datasets for Cloud Computing: A Comprehensive Survey

GUOZHI LIU, School of Computer Science and Technology, South China University of Technology, Guangzhou, China
WEIWEI LIN, South China University of Technology, Guangzhou, China and Pengcheng Laboratory, Shenzhen, China
HAOTONG ZHANG, South China University of Technology, Guangzhou, China
JIANPENG LIN, South China University of Technology, Guangzhou, China
SHAOLIANG PENG, Hunan University, Changsha, China
KEQIN LI, Department of Computer Science, State University of New York, New Paltz, United States

Publicly available datasets are vital to researchers because they permit the testing of new algorithms under a variety of conditions and ensure the verifiability and reproducibility of scientific experiments. In cloud computing research, there is a particular dependence on obtaining load traces and network traces from real cloud computing clusters, which are used for designing energy efficiency prediction, workload analysis, and anomaly detection solutions. To address the current lack of a comprehensive overview and thorough analysis of cloud computing datasets and to gain insight into their current status and future trends, in this article, we provide a comprehensive survey of existing publicly cloud computing datasets. First, we utilize a systematic mapping approach to analyze 968 scientific papers from 6 scientific databases, resulting in the retrieval of 42 datasets related to cloud computing. Second, we categorize these datasets based on 11 characteristics to assist researchers in quickly finding datasets suitable for their specific needs. Third, we provide detailed descriptions of each dataset to assist researchers in gaining a clearer understanding of their characteristics. Fourth, we select 12 mainstream datasets and conduct a comprehensive analysis and comparison of their characteristics. Finally, we discuss the weaknesses of existing datasets, identify challenges, provide recommendations for long-term dataset maintenance and updates, and outline directions for the future creation of new cloud computing datasets. Related resources are available at https://github.com/ACAT-SCUT/Awesome-CloudComputing-Datasets.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Computer systems organization → Cloud computing;**

This work is supported by Guangdong Major Project of Basic and Applied Basic Research (Grant No. 2019B030302002), NSFC-FDCT under Grant No. 62361166662, National Key R&D Program of China under Grant No. 2023YFC3503400, and the Major Key Project of PCL, China under Grant No. PCL2023A09.

Authors' Contact Information: Guozhi Liu, School of Computer Science and Technology, South China University of Technology, Guangzhou, China; e-mail: csliuguozhi@mail.scut.edu.cn; Weiwei Lin (Corresponding author), South China University of Technology, Guangzhou, China and Pengcheng Laboratory, Shenzhen, Guangdong, China; e-mail: linww@scut.edu.cn; Haotong Zhang, South China University of Technology, Guangzhou, Guangdong, China; e-mail: hoyt.zhang77@gmail.com; Jianpeng Lin, South China University of Technology, Guangzhou, Guangdong, China; e-mail: csjianpenglin@mail.scut.edu.cn; Shaoliang Peng, Hunan University, Changsha, Hunan, China; e-mail: slpeng@hnu.edu.cn; Keqin Li, Department of Computer Science, State University of New York, New Paltz, New York, United States; e-mail: lik@newpaltz.edu.

ACM 0360-0300/2025/03-ART198
https://doi.org/10.1145/3719003

Additional Key Words and Phrases: Cloud computing, datasets, workload, security, energy efficiency, resource allocation, task scheduling, anomaly detection

ACM Reference Format:
Guozhi Liu, Weiwei Lin, Haotong Zhang, Jianpeng Lin, Shaoliang Peng, and Keqin Li. 2025. Public Datasets for Cloud Computing: A Comprehensive Survey. *ACM Comput. Surv.* 57, 8, Article 198 (March 2025), 38 pages.
https://doi.org/10.1145/3719003

1 Introduction

Cloud computing is a technology that provides computing and communication services such as computational resources, storage, applications, and networking through the internet (i.e., the cloud). [56, 62]. Its aim is to achieve flexible resource utilization to meet users' dynamic computing resource demands. Through cloud computing, customers can obtain highly available services, while service providers can leverage the elasticity of infrastructure to reduce management costs. Pay-as-you-go pricing is also one of the significant advantages driving the rapid development of cloud computing [38, 128]. Because of these advantages, major cloud service providers like Amazon, Tencent, and Alibaba have built large-scale cloud clusters to offer resource services, and an increasing number of organizations are transferring their internal operations and applications to the cloud [29, 128].

Although the enticing advantages of cloud computing, unpredictable circumstances may lead to performance degradation or even cluster downtime [128]. For instance, mismatches between workloads and resources may result in resource wastage, decreased performance, and energy inefficiency, while network attacks could potentially cause entire cluster downtime. Therefore, to enhance resource utilization, it is imperative to accurately predict workloads [30]. By effectively estimating future workloads, service providers can better plan and allocate resources, pre-allocating or withdrawing resources as needed. However, due to the inherent complexities of cloud workloads, efficiently and accurately predicting them is not a trivial task [128]. Currently, mainstream methods utilize machine learning or deep learning techniques to learn relevant knowledge from historical data [131].

However, these methods need massive data support for training. In many cases, without sufficient empirical data, it is impossible to determine the quality of an approach. For machine learning-based approaches, training and validation must be conducted on sufficiently large, and representative datasets [37, 122].

In cloud computing, there are typically three methods for obtaining relevant datasets. First, the data collection is performed in real cloud computing clusters, where researchers collect corresponding workload traces by running various applications in the clusters. This is a difficult task as it requires significant financial investment (for hardware procurement) and time (for continuous data capture). Second, researchers can simulate the operation of cloud computing clusters on specific platforms to generate datasets. Compared to the former method, this approach requires fewer resources. However, its challenge lies in achieving realistic behaviors of different components [37]. Finally, empirical data can be accessed by logging the actual usage of cloud computing workloads by real users [37]. However, the collection of such datasets poses various challenges, such as privacy issues and information security concerns.

When researchers are unable to gather their own datasets, they can utilize publicly available datasets. However, finding datasets suitable for specific experiments can be a difficult job. This article aims to assist cloud computing researchers by providing a comprehensive survey of publicly available datasets and categorizing them based on a variety of characteristics. To achieve this, we analyze 968 papers retrieved from six comprehensive scientific databases, namely, ACM Digital

Library, IEEE Xplore Digital Library, SpringerLink, Wiley Online Library, Elsevier ScienceDirect, and Google Scholar. Among these, 165 papers use publicly available datasets, with 86 papers (52%) published after 2020. This analysis results in the identification of 42 publicly available datasets, 34 (81%) of which are used within the 86 papers published after 2020.

In addition, this article systematically reviews the existing challenges and future directions of datasets in the field of cloud computing, which are primarily reflected in the following core dimensions: First, there is a significant gap in the availability and diversity of datasets, particularly in the context of network security, where publicly available datasets are scarce and lack comprehensive coverage of attack types, while the energy efficiency domain suffers from a lack of timely and well-structured benchmark data. Second, the scale and complexity of datasets are insufficient, as existing datasets often suffer from limited node sizes and simplistic monitoring metrics, making it difficult to accurately represent the heterogeneous characteristics of real-world cloud clusters. Third, the timeliness of data urgently needs improvement. Furthermore, current mainstream research paradigms focus on cutting-edge areas such as cold start optimization in serverless architectures, multi-tenant dynamic scheduling mechanisms, energy efficiency optimization models for green data centers, and multi-cloud collaborative governance frameworks, highlighting the need for specialized datasets tailored to these research areas. Finally, to ensure the sustainable development of the data ecosystem, this study summarizes relevant measures, specifically recommending the establishment of a dynamic data update mechanism through automated collection, open-source collaboration, and privacy compliance frameworks, thereby providing a sustainable experimental foundation for innovation in the field of cloud computing. Moreover, we have imposed limitations on the scope of this article, as described below.

1.1 Scope

This article aims to collect datasets that can be used to develop and verify various cloud computing schemes. These datasets cover not only host cluster architectures but also virtual machine cluster architectures and datasets based on web traces. Additionally, we concentrate on six application scenarios: workload analysis, energy efficiency, resource allocation, task scheduling, anomaly detection, and security.

As this article focuses on these application scenarios, data collection includes not only datasets containing various workloads but also network traffic data. The former primarily includes features like timestamp, CPU, GPU, memory, and Bandwidth usage. The latter mainly includes information such as IP addresses, HTTP request responses, URLs, whether the requests are malicious, and response status codes.

Finally, we have not found a way to directly access some of the mentioned datasets. Nevertheless, these datasets are still included for two reasons. First, they can be obtained by directly contacting the authors or through purchase. Second, we believe these datasets contain relevant data or demonstrate interesting measurement methods, which researchers can replicate to study similar use cases.

1.2 Contribution

The main contributions of this article are summarized as follows.

- We survey and summarize the publicly available datasets in cloud computing, which is the first systematic survey of datasets in this field.
- We utilize a systematic mapping approach to analyze 968 scientific papers from 6 comprehensive scientific databases in the field of computer science, resulting in the retrieval of 42 publicly available datasets related to cloud computing.

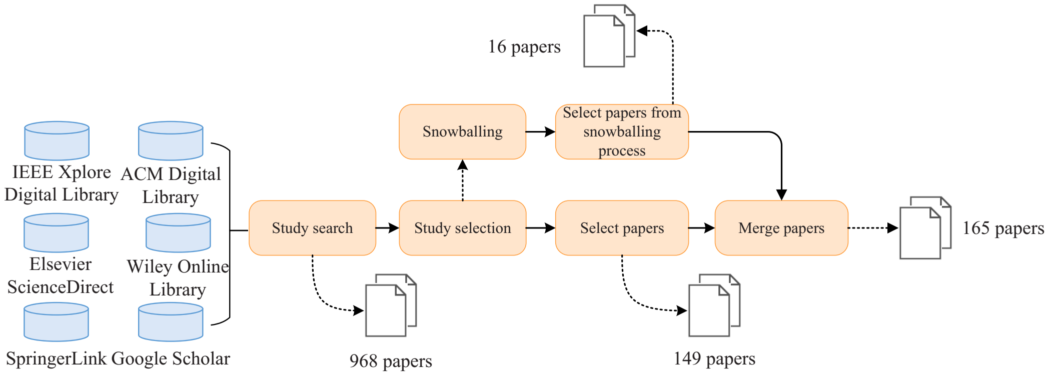


Fig. 1. Methodology for datasets gathering.

- We categorize these datasets based on several characteristics to assist researchers in quickly finding datasets suitable for their specific needs.
- We provide detailed descriptions of each dataset to assist researchers in gaining a clearer understanding of their characteristics, including descriptions, structure, fields, purposes, and so on.
- We conduct a comprehensive analysis and comparison of the characteristics of 12 mainstream datasets.
- We discuss the weaknesses of existing datasets, identify challenges, and outline directions for the future creation of new cloud computing datasets.

1.3 Organization

The rest of this article is organized as follows. In Section 2, we explain the methods we follow for investigating and selecting relevant datasets. Section 3 provides a detailed description of the characteristics we define for classifying datasets, while Section 4 presents the results of the classification itself. Subsequently, Section 5 provides a more detailed overview of these datasets. Section 6 presents a comprehensive analysis and comparison of 12 mainstream datasets. Section 7 discusses the current challenges and the future directions in generating datasets for cloud computing. Finally, Section 8 summarizes the content of this article.

2 Methodology

2.1 Process of Datasets Gathering

This section describes the dataset collection methodology for this article. As shown in Figure 1, the method includes four main steps: study search, study selection, snowballing process, and merge papers.

(1) Study search: Leveraging studies [99] and [37], we identified the most comprehensive scientific databases in the field of computer science: ACM Digital Library (ACM), IEEE Xplore Digital Library (IEEE), SpringerLink, Wiley Online Library (Wiley), Elsevier ScienceDirect (Elsevier), and Google Scholar. It is worth noting that Google Scholar only indexes literature that is not included in the other five major databases. To explore the above databases, we have established the below retrieval rules: (“Document Title”: “cloud computing” AND “Document Title”: “dataset” AND “Full Text Only”: “dataset”) OR (“Document Title”: “MapReduce” AND “Document Title”: “workload” AND “Full Text Only”: “dataset”) OR (“Document Title”: “Virtual machine management” AND “Full Text Only”: “dataset”) OR (“Document Title”: “Distribute” AND “Document Title”:

Table 1. Number of Papers Per Database

Database	Total	%	Filter	%Filtered
IEEE	445	45.97%	81	49.09%
Elsevier	224	23.14%	26	15.76%
ACM	181	18.70%	29	17.56%
Wiley	70	7.23%	8	4.86%
Google Scholar	32	3.31%	13	7.88%
SpringerLink	16	1.65%	8	4.85%
Total	968	100%	165	100%

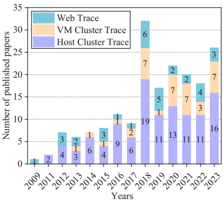


Fig. 2. Annual count of published papers.

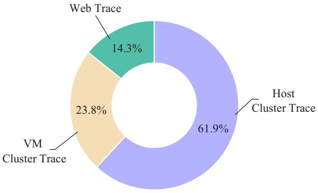


Fig. 3. Proportion of literature related to the datasets.

"Hardware resource" AND "Full Text Only": "dataset") OR ("Document Title": "cluster" AND "Document Title": "workload" AND "Full Text Only": "dataset") OR ("Document Title": "cluster" AND "Document Title": "trace" AND "Full Text Only": "dataset") OR ("Document Title": "cloud" AND "Document Title": "resource usage" AND "Full Text Only": "dataset") OR ("Document Title": "analysis" AND "Document Title": "cloud" AND "Full Text Only": "dataset") OR ("Document Title': "cloud" AND "Document Title": "workload" AND "Full Text Only": "dataset") OR ("Document Title": "decentralized hosting" AND "Document Title": "workload" AND "Full Text Only": "dataset").

(2) Study selection: We first define the following inclusion and exclusion criteria aimed at gathering preliminary research. Then, we employ the inclusion and exclusion standards defined in Step 2 to filter out articles from the papers retrieved in Step 1 that meet the inclusion criteria.

- `Inclusion criteria`: (i) This research is relevant to the field of cloud computing. (ii) this study should utilize publicly available datasets relevant to cloud computing; (iii) this study should be written in English.
- `Exclusion criteria`: (i) Short studies (less than three pages); (ii) studies do not have full text available; (iii) studies structured as tutorials, editorials, and others; (iv) studies without utilizing publicly available datasets relevant to cloud computing.

(3) Snowballing process: We apply the "snowballing" process [126] to ensure that we do miss relevant studies. This technique is an iterative process that reviews the references of each selected study.

(4) Merge papers: In the end, we combine the studies gathered in Step 2 with those selected in Step 3.

2.2 Results

As illustrated in Table 1, after completing the first step, we retrieve a total of 968 papers (i.e., IEEE, Elsevier, ACM, Wiley, Springerlink, and Google Scholar are represented by 445, 224, 181, 70, 16, and 32, respectively). Subsequently, following the second step's screening, we identify a total of 165 papers that utilized publicly available datasets, with IEEE having the highest count of 81 papers, constituting 49.09% of the total papers.

To more clearly illustrate the usage trends of various datasets in cloud computing, we categorize the datasets into three types: Web Trace, VM Cluster Trace, and Host Cluster Trace based on the Data Content Characteristics in Section 3.1. Specifically, Figure 2 illustrates the annual publication count of these 165 papers from 2009 to 2023. It is evident that the number of papers utilizing three types of public datasets has shown a rising trend over time, highlighting the significance of public datasets in scholarly research. Notably, the research in cloud computing peaked in 2018, particularly with regard to studies based on host cluster trace. This is the reason for the significant advancements in large-scale models, notably the release of **Generative Pre-trained Transformer**

(GPT) by OpenAI in 2018 [103]. The training of large models requires substantial computational resources, thus leading to an increased demand for and attention to cloud computing clusters. The subsequent content will revolve around the analysis and exploration of the 165 papers identified in this context.

Further, Figure 3 demonstrates that the usage of host cluster trace datasets is notably prominent, constituting 61.90% of the total publication count. Hence, we observe a growing inclination among scholars in recent years within the cloud computing research domain to favor the utilization of real datasets, with host cluster trace datasets being particularly favored.

In addition, in Supplemental Material A, we provide the corresponding literature for each dataset along with the frequency of their usage. The subsequent content will revolve around the analysis and exploration of the 42 public datasets identified in this context.

3 Datasets Characteristic

This section introduces characteristics that are defined and used for categorizing survey datasets across various dimensions. Subsequent sections will assess these characteristics for each dataset to underline their advantages and drawbacks and determine their possible applications. Figure 4 illuminates these characteristics in the form of a feature map aligned with **Feature-Oriented Domain Analysis (FODA)** [37, 57].

3.1 Data Content

The data content refers to the nature of data contained within the datasets. In this article, we classify the dataset into two categories based on the type of data in the dataset: Web Trace and Cluster Trace.

Web Trace: The web trace dataset typically comprises information from websites, network services, or applications. Its primary records encompass user access patterns, network traffic, user interaction data, and other network activities. For instance, this includes user requests, response times, page visit counts, user IP addresses, HTTP requests, and other web-related information.

Cluster Trace: The cluster trace dataset typically involves the operation and management of clusters, servers, or distributed systems in cloud computing. It mainly includes data on inter-server communication, resource utilization, task scheduling, and container or virtual machine management, among others. For instance, it covers information related to CPU usage, GPU usage, memory usage, I/O, task scheduling, and other cluster-related data. Further, due to the clustered architecture in cloud computing commonly divided into cloud host cluster architecture and cloud virtual machine cluster architecture, in this article, we classify these datasets into ***host cluster trace*** and ***virtual machine cluster trace*** based on the cluster's architecture.

3.2 Year

Due to the rapid advancements in cloud computing, datasets recorded ten years ago may contain tracking records that are no longer relevant today. Simultaneously, significant changes have occurred in resource utilization, workloads, and other aspects due to the rapid progress in hardware and software technologies. Therefore, the age of a dataset is an important indicator for assessing its representativeness concerning real-world cloud computing tracking.

3.3 Available

The availability indicates whether the dataset is accessible to other researchers. In this article, we categorize it into three types: publicly available for free (✔), unavailable (✘), requiring payment (**$**), and needing to contact author to get (?).

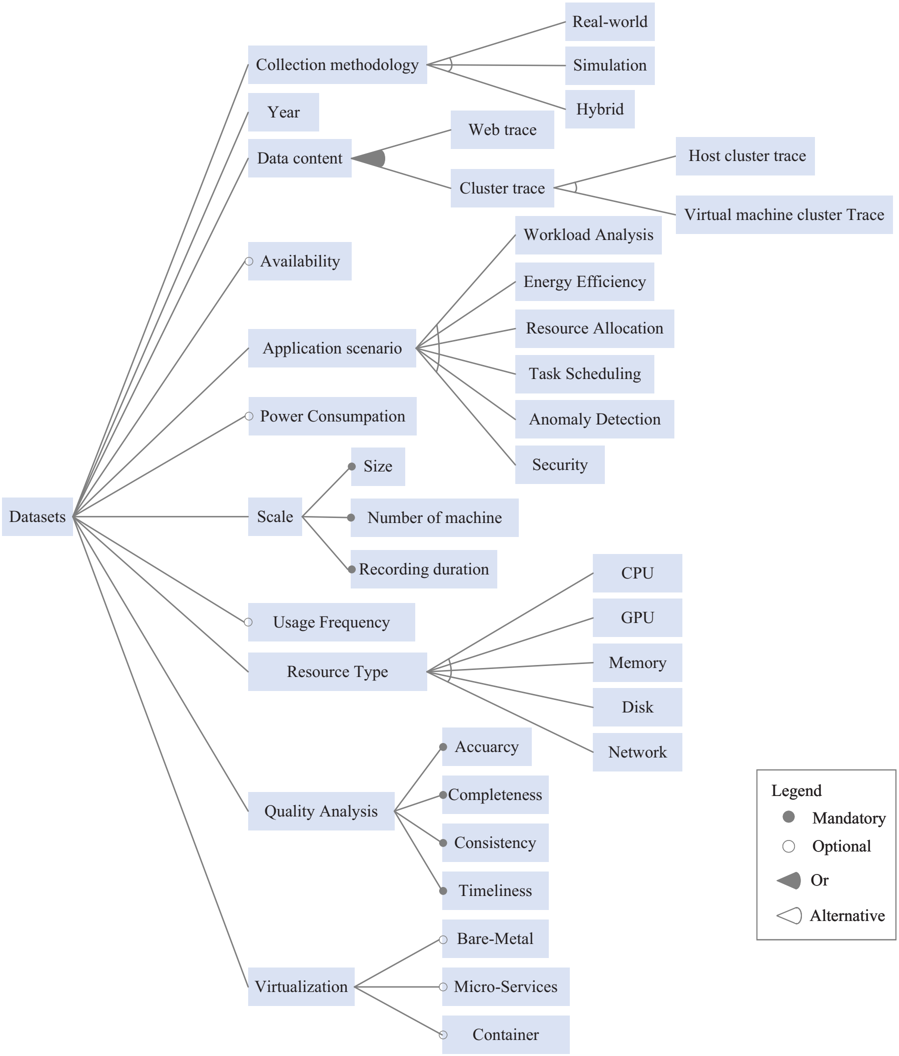


Fig. 4. Dataset characteristics as feature diagram.

3.4 Application Scenario

This feature indicates the scenarios in which researchers can apply the dataset. Through an extensive literature review, we classify cloud computing dataset applications into six classes: workload analysis, energy efficiency, resource allocation, task scheduling, anomaly detection, and security.

Workload Analysis: Workload analysis refers to the assessment and study of performance characteristics of tasks and applications running on the cloud platform. This analysis encompasses the execution patterns, resource demands, resource utilization, and the impact of different task types on system performance. By analyzing workloads, researchers can optimize resource allocation, enhance system efficiency, and predict future resource requirements.

Energy Efficiency: Research on energy efficiency in cloud computing environments aims to reduce energy consumption, enhance energy utilization, and decrease the environmental impact of clusters. Through monitoring and optimizing the energy usage of servers, clusters, and other devices, as well as improving energy utilization patterns, it is possible to lower energy consumption and reduce operational costs. This holds significant importance for sustainability and environmental friendliness.

Resource Allocation: Resource allocation involves dynamically assigning resources (e.g., CPU, GPU, network, etc.) in the cloud to satisfy the diverse demands of users. This encompasses the design and optimization of resource allocation algorithms to ensure efficient resource utilization and high-performance task execution.

Task Scheduling: Task scheduling refers to effectively managing and organizing the sequence and allocation of tasks within a cloud environment. Through appropriate task scheduling strategies, system efficiency and performance can be enhanced, reducing task execution time, preventing resource wastage, and balancing system loads.

Anomaly Detection: Anomaly detection focuses on identifying and addressing anomalies within a cloud environment, such as hardware failures, security vulnerabilities, or malicious activities. By monitoring abnormal behavior within the system, potential issues can be promptly identified and resolved, ensuring the stability and security of the system.

Security: Security research aims to protect cloud computing systems from unauthorized access, data breaches, or other security threats. This includes the use of encryption techniques, identity authentication, access control, and the formulation of security policies to ensure the security and integrity of data and operations within the cloud environment, thereby preventing potential security vulnerabilities and attacks.

3.5 Scale

The scale of the dataset can be more precisely described by the ***size*** of the dataset, ***the number of machines*** (e.g., VMs, nodes, etc.) in the cloud computing cluster, and the ***duration*** of data collection.

3.6 Collection Methodology

This characteristic showcases the methods researchers use to gather datasets. Various methods can be employed to represent real-world cloud computing tracking with different levels of accuracy:

Real-world: In the real-world collection, it is the most accurate but also the most challenging method. This approach involves directly collecting measurement results from existing real cloud computing clusters. Therefore, the prerequisite is having access to an actual cloud computing cluster.

Simulation: Researchers use cloud computing simulators and collect trace records from simulations. The advantage of this method is that it is easier to set up and replicate, and it also costs less. However, the traced records from simulations cannot accurately depict real-world trace records.

Hybrid: To leverage the strengths of different methods, researchers can also establish a hybrid testing platform that combines real and simulated components.

Notably, in this article, all 42 datasets are utilized in a real-world manner, meaning that all datasets are directly collected by measuring results from existing real cloud computing clusters.

3.7 Resource Type

This characteristic represents the types of resources included in the dataset. Through an extensive literature review, we classify cloud computing resource types into five categories, including *CPU, GPU, memory, disk, and network*. These metrics guide readers in identifying datasets that meet their specific requirements.

3.8 Quality Analysis

Quality analysis encompasses key metrics such as *accuracy, completeness, consistency, and timeliness*, which are essential for evaluating dataset reliability. Accuracy ensures the precision of the data, while completeness examines the extent to which the dataset covers all necessary aspects. Consistency checks for internal coherence, and timeliness evaluates the relevance of the data in relation to current conditions. These factors are critical for assessing the dataset's suitability for various analytical applications in cloud computing research.

3.9 Usage Frequency

This characteristic reflects the frequency with which the dataset is used in research and practical applications. By analyzing usage frequency, we can assess the relevance and popularity of the dataset within the academic and industry communities. In this study, we measure usage frequency based on the number of papers related to the dataset and the citation counts of those papers.

3.10 Power Consumption

This characteristic refers to whether the dataset includes information related to power consumption. Due to the significant energy consumption of clusters, energy efficiency is a crucial aspect of cloud computing. Therefore, in this study, we treat power consumption as a distinct metric to clearly inform users about which datasets are relevant for energy efficiency analysis.

3.11 Virtualization

This characteristic refers to the type of virtualization technology included in the dataset. Due to the widespread use of virtualization technologies in cloud computing, different virtualization architectures exhibit significant differences in resource management and performance optimization. Therefore, in this study, we treat virtualization technology as a distinct metric and classify it into three subcategories: *Bare-Metal*, *Micro-Services*, and *Container*. This classification helps readers conduct more targeted analysis and comparisons.

4 Datasets Overview

Before delving into detailed descriptions of each dataset, we have provided a comprehensive summary overview of them in table format, based on the dataset characteristics outlined in Section 3. This idea is that readers seeking datasets with specific characteristics can initially refer to the following content to identify datasets aligned with their interests. Subsequently, they can delve into the detailed descriptions of cloud computing-related datasets in Section 5. This design aims to provide readers with targeted information to meet their research needs.

Tables 2–4 provide a comprehensive listing of all publicly available datasets related to cloud computing that have been collected in this article. The organizational structure of the tables follows the features described in Section 3. Furthermore, these three tables categorize the data into three types based on *data content*: Table 2 presents datasets with a computer cluster architecture focused on *hosts*, while Table 3 showcases datasets with a computer cluster architecture centered around *VM*. Table 4 displays datasets related to *web trace* content.

In addition, we also present the relationship between the datasets and specific application scenarios in the Supplementary Materials *B* to help researchers more quickly identify datasets of interest.

5 Details of Datasets

In this section, we will provide detailed introductions to each type of dataset. We classify them based on the data content into three types: **datasets with host cluster trace**, **datasets with VM cluster trace**, and **datasets with web trace**.

Table 2. Classification for Datasets with Host Cluster Trace

Dataset	Source	Year	Data Content	Application Scenario						Scale			Resource Type					Quality Analysis				Usage Frequency	Power Consumption	Availability
				W	R	T	A	S	E	Size	Machines	Duration	C	G	M	D	N	Acc	Com	Con	Tim			
Google Power Data	Google	2024	Host	✘	✘	✘	✘	✘	✔	45.4 KB	57	1mo	✘	✘	✘	✘	✘	✔	✔	✔	8 mo/0	53/1	✔	✔
Azure LLM Inference Dataset	Microsoft	2023	Host	✔	✘	✔	✔	✘	✘	313 KB	U	1d	✘	✘	✘	✘	✘	✔	✔	✔	5 mo/2	68/2	✘	✔
		2024								702 KB	U	9d												
Acme	Shanghai AI Lab	2023	Host	✔	✔	✔	✔	✘	✔	80 GB	588	6mo	✔	✔	✘	✘	✘	✔	✔	✔	1 y/0	21/1	✔	✔
Alibaba GPU Traces	Alibaba	2020	Host	✔	✔	✔	✔	✘	✘	1.36 GB	1,800	2mo	✔	✔	✔	✘	✘	✔	✔	✔	1 y/1	241/1	✘	✔
		2023								12.2 MB	1,523	7d												
Helios	S-Lab	2020	Host	✔	✔	✔	✔	✘	✘	343 MB	802	6mo	✔	✔	✘	✘	✘	✔	✔	✔	4 y/0	143/2	✘	✔
Marconi100	Marconi	2020	Host	✔	✔	✔	✔	✘	✔	33.4 GB	960	9mo	✔	✔	✔	✔	✘	✔	✔	✔	4 y/3	2/2	✔	✔
Stadia Cloud Gaming Dataset	Universitat Pompeu Fabra	2020	Host	✔	✘	✔	✔	✘	✘	153 MB	U	5mo	✘	✘	✘	✘	✘	✔	✔	✔	4 y/0	115/1	✘	✔
SURFsara	SURFsara	2019	Host	✔	✔	✔	✔	✘	✔	41.3 GB	5	~8mo	✔	✔	✔	✔	✔	✔	✔	✔	5 y/0	2/1	✔	✔
Google Cluster Data	Google	2009	Host	✔	✔	✔	✔	✘	✘	29.8 MB	U	7h	✔	✘	✔	✔	✘	✔	✔	✔	5 y/2	5,685/71	✘	✔
		2011								186 GB	12,583	29d												
		2019								2.4 TB	96,000	1mo												
SPEC Cloud Iaa s	SPEC	2018	Host	✔	✔	✔	✔	✘	✘	?	?	~1y	✔	✘	✔	✔	✔	✔	✔	✔	6 y/6	106/1	✘	✔
Alibaba Cluster Trace	Alibaba	2017	Host	✔	✔	✔	✔	✘	✘	232 MB	1,300	12h	✔	✘	✔	✔	✘	✔	m	✔	6 y/1	986/13	✘	✔
		2018								98 GB	4,000	8d												
Dionatrafk	Dionatr a F. Kirchoff	2018	Host	✔	✘	✘	✘	✘	✘	321 KB	U	1mo	✘	✘	✘	✘	✘	✔	✔	✔	6 y/0	38/1	✘	✔
ATLAS Cluster Traces	Carnegie Mellon University	2011	Host	✘	✘	✔	✔	✘	✘	16.7 MB	1,600	5y	✘	✘	✘	✘	✘	✔	✔	✔	6 y/3	16/1	✘	✔
		2018	Host	✘	✘	✔	✔	✘	✘	1.189 MB	9,408	3mo	✘	✘	✘	✘	✘	✔	m	✔				✔
		2016	Host	✘	✘	✔	✔	✘	✘	?	872	9mo	✘	✘	✘	✘	✘	✔	?	✔				?
		2016	Host	✘	✘	✔	✔	✘	✘	?	441	9mo	✘	✘	✘	✘	✘	✔	?	✔				?
Philly	Microsoft	2017	Host	✔	✔	✔	✔	✘	✘	6.6 GB	552	3mo	✔	✔	✔	✘	✘	✔	✔	✔	7 y/0	422/2	✘	✔
SPEC CPU	SPEC	2006	Host	✔	✔	✔	✔	✘	✘	?	?	~1y	✔	✘	✔	✔	✔	✔	✔	✔	7 y/7	6/1	✘	✔
		2017																						
Autoscale Analyser	H.S. Bhathiya	2015	Host	✘	✔	✘	✘	✘	✘	1.086 MB	2	47h	✔	✘	✔	✘	✘	✔	✔	✔	9 y/0	23/1	✘	✔
Intel Netbatch logs	Ohad Shai	2012	Host	✔	✔	✔	✔	✘	✘	2.53 GB	U	1mo	✔	✘	✔	✘	✘	✔	✔	✔	12 y/0	57/1	✘	✔
OpenCloud Hadoop Workload	Carnegie Mellon University	2010	Host	✔	✘	✔	✔	✘	✘	U	64	20mo	✘	✘	✘	✘	✘	?	?	?	13 y/0	90/2	✘	?
Yahoo Webscope Dataset	Yahoo	2010	Host	✔	✔	✔	✔	✘	✘	33 KB	U	U	✔	✘	✔	✔	✔	✔	✔	✔	14 y/0	26/2	✘	✔
SWIM Workload	Facebook	2009	Host	✔	✘	✘	✔	✘	✘	0.42 MB	600	1d	✘	✘	✘	✘	✘	✔	✔	✔	14 y/1	1317/3	✘	✔
		2010	Host	✔	✘	✘	✔	✘	✘	2.14 MB	3,000	1d	✘	✘	✘	✘	✘	✔	✔	✔				✔
SPEC power_ssj2008	SPEC	2008	Host	✘	✘	✘	✘	✘	✔	?	?	~1y	✘	✘	✘	✘	✘	✔	✔	✔	16 y/16	40/1	✔	✔
GWA-T-4 AuverGrid	Delft University of Technology	2006	Host	✔	✔	✔	✔	✘	✘	13 MB	5	U	✔	✘	✔	✘	✔	✔	✔	✔	18 y/0	4/1	✘	✔
GWA-T-1 DAS2	Delft University of Technology	2005	Host	✔	✔	✔	✔	✘	✘	35 MB	5	U	✔	✘	✔	✘	✔	✔	✔	✔	19 y/0	4/1	✘	✔
GWA-T-10 SHARCNet	Delft University of Technology	2005	Host	✔	✔	✔	✔	✘	✘	33 MB	10	U	✔	✘	✔	✘	✔	✔	✔	✔	19 y/0	4/1	✘	✔
GWA-T-3 NorduGrid	Delft University of Technology	2003	Host	✔	✔	✔	✔	✘	✘	20 MB	75	U	✔	✘	✔	✘	✔	✔	✔	✔	21 y/0	4/1	✘	✔
Parallel Workloads Archive	DG Feitelson	1996-2018	Host	✔	✔	✔	✔	✘	✘	U	U	U	✔	✘	✔	✘	✘	✔	✔	✔	28 y–6 y /40	578/4	✘	✔

U: Unspecified; **$**: Paid service; **?**: Need to contact author to get.
Application Scenario W: Workload Analysis; R: Resource Allocation; T: Task Scheduling; A: Anomaly Detection; E: Energy Efficiency; S: Security.
Duration y: years; mo: months; w: weeks; d: days; h: hours.
Resource Type C: CPU; G: GPU; M: Memory; D: Disk; N: Network.
Quality Analysis Acc: Accuracy; Com: Completeness (m: missing values); Con: Consistency; Tim: Timeliness.
Data Content Host: Host cluster trace; VM: virtual machine cluster trace; Web: Web trace.

5.1 Datasets with Host Cluster Trace

In this section, a detailed description of each dataset categorized under the host cluster trace type is provided, including information such as time, purpose, and fields.

Google Power Data (2024) [53]: The Google Power Data dataset provides power utilization information for 57 power domains in Google data centers. It primarily records PDU, total power utilization averaged over 5-min intervals, and the power utilization attributed to production workloads. Notably, the energy data in this dataset corresponds to the trace data in the Google Cluster Data V3 [52], making it possible to combine these two datasets for more comprehensive analysis.

Table 3. Classification for Datasets with VM Cluster Trace

Dataset	Source	Year	Data Content	Application Scenario						Scale			Resource Type					Quality Analysis				Usage Frequency	Power Consumption	Virtualization	Availability
				W	R	T	A	S	E	Size	Machines	Duration	C	G	M	D	N	Acc	Com	Con	Tim				
Alibaba MicroServices Traces	Alibaba	2021 2022	VM	✓	✓	✓	✓	✗	✗	61.1 GB 2 TB	~10,000 ~40,000	12 h 13 d	✓	✗	✓	✗	✗	✓	✓	✓	2 y/1	211/2	✗	MS	✓
CERIT-SC Workloads	Czech CERIT Scientific Cloud	2022	VM	✓	✓	✓	✓	✗	✗	11 MB	U	1 y	✓	✓	✓	✗	✗	✓	✓	✓	2 y/5	15/1	✗	Docker	✓
OpenNebula Virtual Machine Profiling Dataset	Prasad Purnaye	2021	VM	✓	✓	✓	✓	✓	✗	2.6 MB	6	63 h	✓	✗	✗	✓	✗	✓	✓	✓	3 y/0	23/1	✗	BM (KVM)	✓
Fruktus	Piotr Nawrocki	2020	VM	✓	✓	✗	✓	✗	✗	176 KB	U	11 mo	✓	✗	✗	✗	✗	✓	✓	✓	4 y/0	7/1	✗	U	✓
Chameleon Cloud traces	Science Cloud Blog	2019	VM	✓	✓	✓	✓	✗	✗	72.9 MB	U	~5 y	✓	✗	✓	✗	✗	✓	✓	✓	5 y/3	0/1	✗	BM (KVM)	✓
Azure Public Dataset	Microsoft	2017 2019	VM	✓	✓	✓	✓	✗	✗	117 GB 235 GB	~2 mi ~2.6 mi	30 d 30 d	✓	✗	✓	✗	✗	✓	✓	✓	5 y/1	792/7	✗	BM (Hyper-V)	✓
Scout	Hsu Chin-Jung	2018	VM	✓	✓	✓	✓	✗	✗	587 MB	U	1 d	✓	✗	✓	✓	✓	✓	✓	✓	6 y/0	23/1	✗	BM (Xen)	✓
IBM Docker Registry traces	IBM	2018	VM	✓	✗	✗	✗	✗	✗	1.5 GB	U	75 d	✗	✗	✗	✗	✗	✓	✓	✓	6 y/0	148/2	✗	Docker	✓
Business Critical Workloads	Bitbrain	2015	VM	✓	✓	✓	✓	✗	✗	284 MB	1,250	1 mo	✓	✗	✓	✓	✓	✓	✓	✓	9 y/0	877/13	✗	BM (KVM)	✓
Planetlab VM traces	Beloglazov Anton	2011	VM	✓	✓	✓	✓	✗	✗	5.2 MB	U	10 d	✓	✗	✗	✗	✗	✓	✓	✓	13 y/0	607/7	✗	BM (Xen)	✓

U: Unspecified; **$**: Paid service; **?**: Need to contact author to get.
Application Scenario W: Workload Analysis; R: Resource Allocation; T: Task Scheduling; A: Anomaly Detection; E: Energy Efficiency; S: Security.
Methodology r: Real-world; s: Simulation; h: Hybrid.
Duration y: years; mo: months; w: weeks; d: days; h: hours.
Resource Type C: CPU; G: GPU; M: Memory; D: Disk; N: Network.
Quality Analysis Acc: Accuracy; Com: Completeness (m: missing values); Con: Consistency; Tim: Timeliness.
Virtualization BM: Bare-Metal; MS: Micro-Services.
Data Content Host: Host cluster trace; VM: virtual machine cluster trace; Web: Web trace.

Table 4. Classification for Datasets with Web Trace

Dataset	Source	Year	Data Content	Application Scenario						Scale			Resource Type					Quality Analysis				Usage Frequency	Power Consumption	Availability
				W	R	T	A	S	E	Size	Machines	Duration	C	G	M	D	N	Acc	Com	Con	Tim			
ISCX	Canadian Institute for Cybersecurity	2016	Web	✗	✗	✗	✗	✓	✗	~26.3 MB	U	1y	✗	✗	✗	✗	✓	✓	✓	✓	8 y/5	18/1	✗	✓
Amazon Resource Cost	Queen's University	2012	Web	✓	✗	✗	✗	✗	✗	87.1 KB	U	4d	✗	✗	✗	✓	✓	✓	m	✓	12 y/1	17/1	✗	✓
Cloud Intrusion Detection Dataset	Pisa University	2012	Web	✗	✗	✗	✗	✓	✗	~70 MB	45	7w	✗	✗	✗	✗	✓	✓	✓	✓	12 y/0	85/1	✗	✓
Wikipedia Web Traces from WikiBench	Wikipedia	2007	Web	✓	✗	✗	✗	✗	✗	~5 TB	U	9y	✗	✗	✗	✗	✓	?	?	?	17 y/0	551/9	✗	✗
KDD Cup 1999's Dataset	MIT Lincoln Labs	1998	Web	✗	✗	✗	✗	✓	✗	1.31 MB	U	U	✗	✗	✗	✗	✓	✓	✓	✓	26 y/0	315/3	✗	✓
NASA HTTP Traces	Kennedy Space Center	1995	Web	✓	✗	✗	✗	✗	✗	20.7 MB	U	1mo	✗	✗	✗	✗	✓	✓	✓	✓	29 y/0	73/8	✗	✓

U: Unspecified; **$**: Paid service; **?**: Need to contact author to get.
Application Scenario W: Workload Analysis; R: Resource Allocation; T: Task Scheduling; A: Anomaly Detection; E: Energy Efficiency; S: Security.
Methodology r: Real-world; s: Simulation; h: Hybrid.
Duration y: years; mo: months; w: weeks; d: days; h: hours.
Resource Type C: CPU; G: GPU; M: Memory; D: Disk; N: Network.
Quality Analysis Acc: Accuracy; Com: Completeness (m: missing values); Con: Consistency; Tim: Timeliness.
Machines mi: million.
Data Content Host: Host cluster trace; VM: virtual machine cluster trace; Web: Web trace.

Based on this dataset, Sakalkar et al. [106] conduct research on the energy efficiency of cloud platforms, proposing a new medium-voltage power layer and workload scheduling techniques.

Azure LLM Inference Dataset (2023, 2024) [16]: The Azure LLM Inference Dataset, released by Microsoft in 2024, provides tracking examples for multiple LLM inference services on the Azure cluster. The dataset includes two scenarios: code generation and conversation. Two versions of the dataset record traces from November 11, 2023, and from May 10 to May 19, 2024. Each log primarily documents the number of context tokens and the number of generated tokens for inference tasks. Based on this dataset, Shah et al. [98] analyze the latency, throughput, memory, and power consumption characteristics at different stages of LLM inference, focusing on model deployment and scheduling techniques. Stojkovic et al. [115], by analyzing features such as the number of instances, model parallelism, and GPU frequency in the LLM inference system, focus on the development of an energy management framework for LLM inference environments.

Acme (2023) [54]: Acme, released by the Shanghai Artificial Intelligence Laboratory in 2023, provides traces of training large language models on the Acme cluster. This dataset encompasses job submissions between March 2023 and August 2023, covering two independent GPU clusters. The dataset size is 80 GB, documenting a total of 880,740 jobs, including 470,497 GPU jobs. Detailed records include each job's submission, start, and end times, the number of GPUs and CPUs utilized, and the total GPU resources consumed per job. Based on this dataset, Hu et al. [59] conducted an in-depth feature study of the workload traces, focusing on analyzing the differences between LLMs and previous **deep learning (DL)** workloads for specific tasks. Their research explores resource utilization patterns and identifies the impact of various job failures.

Alibaba GPU Traces (2020, 2023) [7]: This dataset includes log information for hybrid training and inference jobs executing popular machine learning methods. The traces are gathering from a substantial production cluster within Alibaba **Platform for Artificial Intelligence (PAI)**, which consists of more than 6,500 GPUs across approximately 1,800 machines, covering the period of July and August 2020. This trace provides detailed records of each machine's configuration (e.g., gpu_type, cpu_capacity), operational information (e.g., machine_load, machine_cpu_iowait), and task execution details (e.g., plan_gpu, gpu_type, cpu_usage, gpu_usage). Based on this dataset, Weng et al. [124] conduct a characterization study of workload traces and highlight the challenges faced in cluster scheduling. These challenges include low GPU utilization, long queuing delays, difficult-to-schedule tasks requiring high-end GPUs with stringent scheduling requirements, load imbalance across heterogeneous machines, and potential CPU bottlenecks. Weng et al. [125] investigate the issue of GPU fragmentation and propose corresponding scheduling solutions.

Helios (2020) [108]: Helios, released by SenseTime in 2020, is a trace dataset from SenseTime's private data center designed for developing and producing deep learning models. The data center comprises 8 independent GPU clusters with a total of over 12,000 GPUs. The dataset, sized at 343 MB, includes traces from 4 independent GPU clusters, encompassing 3,362,981 total jobs, of which 1,580,464 are GPU jobs. It provides a comprehensive analysis of deep learning workloads in Helios from April 2020 to September 2020. The trace details each job's GPU count, CPU count, and duration, although specific resource utilization rates are not included. Based on this dataset, Hu et al. [58] conducted an in-depth analysis, focusing on service scheduling and cluster energy efficiency. Additionally, Hu et al. [59] compared this dataset comprehensively with other existing GPU datasets.

Marconi100 (2020) [22]: The Marconi100 dataset is provided by the Italian National Supercomputing Center (CINECA) for studying and analyzing large-scale scientific computing and data processing workloads. This dataset covers the entire system, including internal information on more than 980 compute nodes such as *weather forecasts*, *system status alerts*, *power units*, *core load*, *frequency*, *job-related information*, *memory read/wems*, *system status alerts*, *workload manager*

Table 5. Datasets and Fields for the Stadia Cloud Gaming Dataset

Datasets	Name	Fields	Characteristics
Dataset 1	Time trends	F1, F2, F3	TO, TP, SL; RE: 1080p; VC: VP9; DU: 30 s; DT & UT; RTP, DTLS, STUN
Dataset 2	Traffic features	F1, F2, F3	TO, TP, SL; RE: 1080p, VC: VP9, DU: 600 s; DT & UT, RTP, DTLS, STUN
Dataset 3	Game status	F1, F2, F3	TO, SL; RE: 1080p, VC: VP9, DU: 540 s, DT & UT
Dataset 4	Codecs	F1, F2, F3	TO, SL; RE: 1080p, VC: VP9 & H.264, DU: 600 s, DT
Dataset 5	Resolutions	F1, F2, F3	TO, SL; RE: 720p, 1080p, 4K, VC: VP9, DU: 600 s, DT
Dataset 6	Varying bandwidths	F1, F2, F3	TO, SL; RE: 720p, 1080p, VC: VP9, DU: 60 s, DT
Dataset 7	Abrupt bandwidth fluctuations	F4, F5, F6, F7	TO, SL; RE: 720p, 1080p, VC: VP9, DU: 500 s, DT
Dataset 8	delay	F7, F8	TO, SL; RE: 720p, 1080p, VC: VP9, DU: 60-600 s, DT
Dataset 9	delayed changes	F5, F6, F9, F10	TO: 1080p, VC: VP9, DU: 180 s, DT & UT

RE: resolution, **VC**: video codec, **DU**: duration, **DT**: downlink traffic, **UT**: uplink traffic.
Three games: Rise of the Tomb Raider: 20th anniversary edition (**TO**), Thumper (**TP**) and Spitlings (**SL**).

statistics, *job-related information*, and *temperature*. The dataset contains measurement data for hundreds of metrics from each compute node, as well as hundreds of other metrics collected from monitoring sensors on all system components. The recorded timeframe spans from April 1, 2020, to September 1, 2022, with varying collection frequencies for different metrics, but most are collected every 20s. Based on this dataset, Molan et al. [86] conduct research on anomaly detection in HPC systems, while Lin et al. [78] study the optimization of cooling control in data centers to reduce energy consumption.

Stadia Cloud Gaming Dataset (2020) [21]: This Dataset is a traffic tracking of Stadia, Google's cloud gaming solution, which is publicly available by work [21] in 2020. The Stadia Cloud Gaming Dataset comprises a total of nine datasets. Each dataset may contain multiple traffic traces, with each traffic trace being a text file containing the following two or more variable fields:

- F1: Epoch-formatted packet arrival timestamp.
- F2: Second-based relative timing of packet arrivals.
- F3: UDP data length in bytes.
- F4: Height of video frame in pixels.
- F5: FPS (frames per second) of video.
- F6: Seconds per round trip.
- F7: Rate of packet loss in seconds.
- F8: Jitter Buffer delay, measured per second.
- F9: Uplink RTCP data rate in bits/second.
- F10: Downlink RTP data rate in bits/second.

Table 5 specifies which fields are used in each dataset, along with information such as the game, measurement duration, video codec used, resolution, and so on for each dataset. Based on this dataset, Carrascosa et al. [21] first analyze the traffic characteristics of Stadia, then examine how different Stadia games and configurations affect the generated traffic patterns. Finally, they study the network performance in the presence of Stadia traffic.

SURFsara (2019) [116]: The SURFsara dataset, provided by the SURFsara Computing and Data Research Center in the Netherlands, is utilized for researching and analyzing large-scale scientific computing and data processing workloads. This dataset encompasses authentic workloads from various domains such as bioinformatics, physics, meteorology, and more. Specifically, the data spans from December 29, 2019, to August 7, 2020, with a sampling frequency of every 15 s, covering over 300 servers. Each node's maximum sample size per metric is 1,258,646, resulting in a total measurement count of 66,541,895,243. The dataset's total size is 41.3 GB. This dataset includes 100+ metrics, such as

- `NVIDIA NVML`: Each GPU data includes metrics such as power, chip temperature, fan speed, and memory usage.
- `IPMI`: The data for each server includes metrics such as power consumption and internal temperature.
- `OS-level`: Low-level operating system metrics from procfs, sockstat, or netstat data include the status of each server, including CPU load, disk, memory, network utilization, context switches, and interrupts.

Based on this dataset, Versluis et al. [121] conduct a comprehensive analysis of data center operations, providing statistical characteristics related to nodes, energy consumption, and workloads.

Google Cluster Data (2009, 2011, 2019): Google Cluster Data is a comprehensive trace log released by Google Data Centers regarding its production clusters. There are three versions available: Version 1 (2009), Version 2 (2011), and Version 3 (2019).

Version 1 (2009) [50]: This version documents detailed trace records of production workloads executed on Google clusters over a 7-h period. These workloads consist of a series of tasks, where each task is part of a job, and a job contains multiple tasks. Each row in the dataset represents the execution details of an individual task within a 5-min interval, primarily recording fields such as *average CPU usage* and *average memory usage*. The trace includes 75 5-min reporting intervals, totaling 3,535,029 observations, 9,218 unique jobs, and 176,580 unique tasks. Garraghan et al. [49] conducted a large-scale analysis of server resource utilization using this dataset and characterized the features of production cloud data centers. Patel et al. [97] propose a cloud workload analysis method based on the Gaussian Mixture Model using this dataset.

Version 2 (2011) [51]: The second version released in November 2011 constitutes a solid working foundation, documenting the trace logs of Google Cloud. It encompasses the activities of 12,583 heterogeneous servers over a 29-day period, including approximately 672,024 jobs and over 25 million tasks. Compared to the first version, this version provides more comprehensive information, such as *task resource requirements (CPU, memory, and disk), priority levels, and the maximum and average resource consumption of tasks*. Agrawal et al. [3] propose an adaptive anomaly detection mechanism based on this dataset to detect and monitor anomalies in cloud computing clusters. Cheng et al. [32] use this dataset as a training set to develop a cloud cluster resource prediction algorithm. Zhang et al. [133] propose a dynamic workload management approach in heterogeneous cloud environments.

Version 3 (2019) [52]: The third version logs the overall resource usage for each task on eight various clusters spread across the world. This tracking starts on Wednesday, May 1, 2019, and captures the resource utilization for various tasks over 1 month. The trace mainly contains 5-min averaged normalized CPU and memory usage. The version is composed of approximately 2.4 TiB of compressed dataset. Compared to Version 2, this version captures more detailed information, such as *cpu_usage_distribution, tail_cpu_usage_distribution, sample_rate, and page_cache_memory*. Based on this dataset, Rossi et al. [105] conduct workload distribution prediction, Park et al. [96] explore resource reuse and real-time migration to enhance data center resource utilization, and Christofidi et al. [34] investigate cloud resource usage prediction.

SPEC Cloud Iaas (2018) [114]: SPEC Cloud IaaS, released in 2018 by **Standard Performance Evaluation Corporation (SPEC)**, is a set of benchmarks designed to evaluate the performance of cloud infrastructure services. This benchmark aims to simulate real-world cloud computing environments and provides a series of test workloads to measure cloud computing performance. These tests cover various types of cloud computing workloads, including virtual machine deployment, storage performance, network performance, and so on. Notably, there is a $2,000 fee to use the SPEC Cloud IaaS 2018 benchmark. Based on this dataset, Papadopoulos et al. [1] analyze and establish standards for cloud performance measurement.

Alibaba Cluster Trace (2017, 2018): Alibaba C luster Trace is a comprehensive trace log published by Alibaba Group about its production clusters. There are two versions available: Version 1 (2017) and Version 2 (2018).

Version 1 (2017) [5]: The dataset begins in September 2017 and captures the cluster activities of a production environment over 12-h intervals. It encompasses data from around 1300 machines, which are utilized for running both online services and batch processing tasks. The version is composed of approximately 232 MB of compressed dataset, including six main components. Specifically, two files are utilized to detail the resource usage of the compute nodes. Two files are used to describe the resource utilization for batch jobs and two others are used to describe the resource usage for online services. Based on this dataset, Chen et al. [26] and Liu et al. [81] analyze workloads in production clouds to enhance resource management efficiency and quality of service. Additionally, Mahesh et al. [55] analyze the dataset and propose a prediction method for resource usage in production clouds.

It is worth noting that the dataset exhibits the following issues: first, some task_id and job_id are missing in the batch_instance.csv file; second, certain instance_id values in the container_usage.csv file do not appear in the container_event.csv file; furthermore, there is missing usage information in the batch_instance.csv file.

Version 2 (2018) [6]: Similar to the first release, the trace provides detailed information about batch jobs, service jobs, and servers. In particular, the dataset begins in September 2018 and encompasses cluster information over an 8-day period, and involves approximately 4,000 machines that operate both online services and batch jobs. The version is composed of approximately 98 GB of compressed dataset, including six main components. Compared to Version 1, this version includes additional machine configuration details (e.g., cpu_num and mem_size), incorporates network-related metrics in machine usage (e.g., net_in and net_out), and records more detailed information at the instance level (e.g., cpu_max, cpu_avg, mem_max, and mem_avg). Based on this dataset, Everman et al. [40] conduct a comprehensive analysis of Alibaba cluster traces, identifying issues of over-subscription (leading to resource waste and low utilization) and under-subscription (resulting in performance degradation). They also develop a simulator to evaluate potential solutions to these problems. Zhu et al. [136] propose a production cloud benchmarking method through the analysis of this dataset.

Dionatrafk (2018) [2]: This dataset is a real-time workload trace conducted by the study [2] on a genuine experimental platform, publicly released in 2018, spanning over a month. The platform utilizes machines that are equipped with an Intel Core i7-6500U processor, which operates at a clock speed of 2.50 GHz and has 16 GB of memory. Meanwhile, the authors employ Python 2.7 and use an interactive environment, Google Colaboratory, to implement and execute all techniques. The dataset consists of 13 CSV files, each storing the *workload* conditions generated by the platform at different time intervals. Based on this dataset, Kirchoff et al. [70] conduct research on cloud workload prediction.

ATLAS Cluster Traces (2011, 2016, 2018) [15]: The trace repository, established by Carnegie Mellon University and **Los Alamos National Laboratory (LANL)**, is designed to gather real workload traces from diverse platforms. Currently, it contains four sets of job scheduling logs that have been made available. These logs originate from LANL's general-purpose Mustang cluster, the advanced LANL Trinity supercomputer, and two data centers operated by Two Sigma. The configurations of these logs are detailed as follows:

- `LANL Mustang cluster`: Mustang is composed of 1,600 equal compute nodes with 102 TB of RAM and 38,400 AMD Opteron 6176 2.3 GHz cores.
- `LANL Trinity supercomputer`: Trinity is composed of 9,408 equal compute nodes with 1.2 PB of memory and 301,056 Intel Xeon E5-2698v3 2.3 GHz cores.

- `Sigma's datacenters`: The two data centers have a total of 1,313 equal compute nodes with 328 TB of memory and 31,512 CPU cores.

Additionally, the author only provides detailed information on public datasets *LANL Mustang cluster* and *LANL Trinity supercomputer*, each comprising a CSV file. Based on this dataset, Amvrosiadis et al. [11] analyze heterogeneous workloads in cloud computing clusters. It is worth noting that the datasets related to the Sigma data center are available upon request via email by contacting the authors.

Notably, a small amount of start_time and dispatch_time data is missing in the dataset. Users can remove these entries before use, as this does not have a significant impact on the overall usability of the dataset.

Philly (2017) [104]: Philly, released by Microsoft Research in 2017, is a dataset capturing **deep neural network (DNN)** training workloads on Microsoft's internal Philly cluster. The dataset spans from August 7, 2017, to December 22, 2017, totaling 6.6 GB in size and documenting 117,325 jobs. It provides detailed records of each job's submission time, completion time, the number of GPUs used, and specific GPU IDs. However, it does not include information on resource utilization rates. Based on this dataset, Jeon et al. [65] conducted a comprehensive analysis focusing on factors affecting cluster utilization for DNN training workloads in multi-tenant clusters. Their research addressed three key aspects: (1) the impact of group scheduling and locality constraints on queuing delays, (2) the influence of locality on GPU utilization, and (3) failures occurring during training. Additionally, Hu et al. [59] performed a detailed comparison of this dataset with other existing GPU workload datasets.

SPEC CPU (2006, 2017) [113]: SPEC CPU benchmark is a suite of CPU benchmarking tools released by SPEC (Standard Performance Evaluation Corporation), which aim to measure the computing capability of computer systems. It is primarily used to evaluate and benchmark the performance of computationally intensive tasks, putting stress on aspects such as processors, memory subsystems, and compilers to obtain realistic workloads. Currently, there are two main versions of the SPEC CPU benchmark: SPEC CPU2006 and SPEC CPU2017, where SPEC CPU2006 has been discontinued. Notably, there is a $1,000 fee to use the SPEC CPU2017 benchmark. Based on this dataset, Iglesias et al. [60] conduct a comprehensive analysis of the relationship between the cost of running computational assets and their resource capacity (i.e., CPU and RAM).

Autoscale Analyser (2015): [19] This dataset comprises real-time workload traces obtained in 2015 from two servers, spanning over 47 h. The original dataset consists of four log files, recording the CPU and memory utilization of the two computing nodes. The detailed fields for each file are as follows:

- `server-cpu-usage.log`: This file provides detailed records of the server's CPU usage, primarily encompassing eight fields: time, CPU, %user, %nice, %system, %iowait, %steal, and %idle.
- `server-mem-usage.log`: This file provides detailed records of the server's memory usage, primarily encompassing eight fields: kbbuffers, %commit time, kbcommit, kbmemused, %memused, kbcached, and kbmemfree.

Based on this dataset, Shariffdeen et al. [110] conduct research on cloud workload prediction methods and automated cloud resource scaling techniques.

Intel Netbatch logs (2012) [109]: This dataset includes one month of accounting records (e.g., CPU and memory utilization) from the Intel Netbatch grid, as disclosed by Ohad Shai of Intel in November 2012. These data originate from four clusters, with three situated on the U.S. West Coast and one in Israel. The trace is composed of approximately 2.53 GB of compressed dataset, including four main CSV files labeled as *Intel-NetbatchA-2012-0*, *Intel-NetbatchB-2012-0*, *Intel-NetbatchC-2012-0*, and *Intel-NetbatchD-2012-0*.

These four CSV files contain records with identical fields, as listed below:*end time*, *machine ID*, *iteration number*, *group*, *submit time*, *user*, *command*, *start time*, *exit status*, *suspend time (seconds)*, *memory (GB total for all threads)*, *wall time (seconds)*, *user CPU time (seconds total for all threads)*, *iteration submit time (local timestamp)*, *system CPU time (seconds total for all threads)*, *max RSS (4 KB pages)*, *job ID*, *max VM (4 KB pages)*, and *cores*. Based on this dataset, Jeddi et al. [64] conduct research on workload prediction for network function virtualization.

OpenCloud Hadoop Workload (2010) [95]: This trace records real-time logs of the OpenCloud Hadoop cluster administered at the Parallel Data Lab. The trace spans from May 2010 to December 2011, totaling 20 months. The logs comprise 51,975 successfully completed jobs, 4,614 failed jobs, and 1,762 aborted jobs, with a total of 78 users submitting jobs during this period. The cluster consists of 64 nodes, each equipped with a 2.8 GHz dual quad-core CPU (8 cores), 16 GB RAM, 10 Gbps Ethernet NIC, and 4 Seagate 7200 rpm SATA hard disk drives. Throughout the entire data collection period, the cluster runs Hadoop version 0.20.1. It is important to note that the authors do not provide the original data but offer anonymized data fields along with the relevant hardware and software configurations. Therefore, researchers can obtain the data as needed or contact the authors via email to access the relevant data. Based on this dataset, El-Sayed et al. [39] analyze the behavioral patterns of unsuccessful jobs across different clusters and develop an anomaly detection algorithm for large-scale computing platforms. Additionally, Kaur et al. [67] conduct research on job scheduling methods in cloud computing clusters.

Yahoo Webscope Dataset (2010) [130]: The Yahoo Webscope project offers datasets in six domains for researchers' use, including advertising and market datasets, competition datasets, computer system datasets, chart and social datasets, language datasets, and rating and classification datasets. In the computer system dataset, there exists one called the Database Platform System Tracing dataset, which includes a number of trace data on the resource utilization of the compute nodes during the execution of the PNUTS/Sherpa database. These data measure metrics such as *network traffic*, *CPU usage*, *memory usage*, *disk usage*, and so on. It is worth noting that access to this dataset requires authors to apply for it through the official website [130].

Based on this dataset, Mian et al. [84] conduct research on task scheduling and resource allocation, while Agrawal et al. [3] investigate detection techniques for cloud anomalous activities.

SWIM Workload (2009, 2010) [41]: This is a dataset collected by Facebook about MapReduce cluster workloads for Hadoop. The workloads span one day, encompassing 24 historical trace samples, with each sample representing 1 h of data. There are two versions: Version 1 (2009), and Version 2 (2010).

Version 1 (2009) [41]: The first version comes from a historical trace of Hadoop on Facebook's 600-machine cluster. The original trace spanned a six-month period from May 2009 through October 2009 and included about 1 million jobs. The trace consists of a dataset of approximately 0.42 MB, including three main tsv files labeled *FB-2009_samples_24_times_1hr_0_first50jobs*, *FB-2009_samples_24_times_1hr_0*, and *FB-2009_samples_24_times_1hr_1*, where *FB-2009_samples_24_times_1hr_0_first50jobs* is the test dataset. The corresponding workload for each job is stored in each file and the fields for each job are as follows: inter_job_submit_gap_seconds, shuffle_bytes, submit_time_seconds, new_unique_job_id, map_input_bytes, and reduce_output_bytes.

Version 2 (2010) [41]: The second version is derived from a historical Hadoop trace on the same Facebook cluster, now expanded to 3,000 machines. This original trace covers a period of 1.5 months, from October 2010 to November 2010, and includes approximately 1 million jobs. The trace consists of a dataset of approximately 0.42 MB, including two main tsv files labeled

FB-2010_samples_24_times_1hr_withInputPaths_0 and *FB-2010_samples_24_times_1hr_0*. The fields for job are the same as in version one.

Based on this dataset, Chen et al. [28] study the workload of MapReduce tasks and develop a corresponding scheduling framework based on the characteristics of these workloads. Ling et al. [80] investigate the issue of resource utilization imbalance in the cloud, while Chen et al. [27] propose new features for MapReduce workloads

SPEC power_ssj2008 (2008) [112]: SPEC power_ssj2008, released in 2008 by SPEC, is the first industry-standard benchmark designed to evaluate the energy efficiency and performance characteristics of server systems, including single-server and multi-node configurations. The goal of the benchmark is to evaluate the scalability of the CPU, cache, memory hierarchy, and Shared Memory Processor, while measuring the performance of specific elements of the Just-in-Time compiler, Java Virtual Machine, threading, garbage collection, and operating system implementation. The score is calculated by dividing the total performance obtained at each target workload level (ssj_ops) by the total average power consumption at each target workload level (including active and idle), enabling comparison of power consumption and performance across different servers. Based on this dataset, Lin et al. [79] analyze the power consumption of cloud server components (i.e., CPU, memory, and storage) and develop power consumption models.

GWA-T-4 AuverGrid (2006) [93]: AuverGrid team in 2006, meticulously documents the workload of a cluster. This cluster comprises 475 CPUs and has completed 404,176 jobs during the recording period, catering to 405 users. The dataset provides detailed records of the resources utilized by each job, including CPU, memory, and network, as well as the resources required by each job. The dataset contains 29 fields, primarily including *SubmitTime*, *WaitTime*, *RunTime*, *AverageCPUTimeUsed*, *Used Memory*, *ReqMemory*, *UsedNetwork*, *UsedResources*, *ReqNetwork*, among others. Based on this dataset, Ali et al. [4] propose an unsupervised feature selection method for cloud workload tracing, while Cetinski et al. [25] conduct research on workload prediction techniques.

GWA-T-1 DAS2 (2005) [91]: The dataset records the workload of the DAS-2 system in February 2005, capturing metrics such as CPU utilization, memory consumption, and network activity. During the recording period, the system consisted of 400 CPUs and completed 1,124,772 jobs, serving 333 users. The dataset contains 29 fields, primarily including *SubmitTime*, *WaitTime*, *RunTime*, *AverageCPUTimeUsed*, *Used Memory*, *ReqMemory*, *UsedNetwork*, *UsedResources*, *ReqNetwork*, among others. Based on this dataset, Ali et al. [4] propose an unsupervised feature selection method for cloud workload tracing.

GWA-T-10 SHARCNet (2005) [92]: The dataset released by John Morton and Clayton Chrusch in 2005, meticulously documents the workload of a cluster. This cluster comprises 6,828 CPUs and has completed 1,195,242 jobs during the recording period, catering to 412 users. The dataset provides detailed records of the resources utilized by each job, including CPU, memory, and network, as well as the resources required by each job. The dataset contains 29 fields, primarily including *SubmitTime*, *WaitTime*, *RunTime*, *AverageCPUTimeUsed*, *Used Memory*, *ReqMemory*, *UsedNetwork*, *UsedResources*, *ReqNetwork*, and so on. Based on this dataset, Ali et al. [4] propose an unsupervised feature selection method for cloud workload tracing.

Parallel Workloads Archive (2005) [42]: This repository, initiated in 2005, collects real parallel workload logs from production systems spanning from 1993 to 2016. It encompasses 40 sources, including NASA iPSC (1993), LANL CM5 (1994), SDSC Par95 (1994), KIT FH2 (2016), and so on. Detailed information for each dataset can be found on the website [42]. Based on this dataset, Baldan et al. [17] conduct research on cloud workload prediction and Bossche et al. [120] propose a deadline-aware hybrid cloud workload scheduling strategy. Additionally, they investigate selection strategies for outsourcing cloud service providers.

GWA-T-3 NorduGrid (2003) [90]: The dataset released by the NorduGrid team in 2003, meticulously documents the workload of a cluster. This cluster comprises 2,000 CPUs and has completed 781,370 jobs during the recording period, catering to 387 users. The dataset provides detailed records of the resources utilized by each job, including CPU, memory, and network, as well as the resources required by each job. The dataset contains 29 fields, primarily including *SubmitTime*, *WaitTime*, *RunTime*, *AverageCPUTimeUsed*, *Used Memory*, *ReqMemory*, *UsedNetwork*, *UsedResources*, *ReqNetwork*, among others. Based on this dataset, Ali et al. [4] propose an unsupervised feature selection method for cloud workload tracing.

5.2 Datasets with VM Cluster Trace

In this section, a detailed description of each dataset is categorized under the VM cluster trace type, including information such as time, purpose, and fields.

Alibaba MicroServices Traces (2021, 2022): Alibaba MicroServices Traces is a comprehensive trace log published by Alibaba Group about its production clusters. There are two versions: Version 1, and Version 2.

Version 1 (2021) [8]: The first version of the trace includes detailed runtime metrics for around 20,000 **microservices (MS)**. The data is collected over a 12-h period in 2021 from Alibaba's production cluster of more than 10,000 **bare metal (BM)** nodes. This trace records **response time (RT)** information, BM node runtime information, **Microservice call rate (MCR)**, MS runtime information, and MS Call Graphs information. The trace is composed of approximately 61.1 GB of compressed dataset, including four main components labeled as *node*, *MSCallGraph*, *MSResource*, and *MSRTQps*. Specifically,

- `node`: Bare metal node runtime info.
- `MSCallGraph`: The information on MCR and RT details the MCR and RT for calls made through various communication paradigms among over 1,300 microservices, encompassing more than 90,000 containers within the same production cluster.
- `MSResource`: The MS runtime information captures CPU and memory utilization data for over 90,000 containers associated with more than 1,300 microservices within the same production cluster.
- `MSRTQps`: The MS Call Graphs information involves sampling the call graph at a 0.5% rate due to the extensive scale of the data. This results in approximately over 20 million call graphs across more than 20,000 microservices within more than ten clusters.

Version 2 (2022) [9]: This second version provides a longer duration (13 days) for this updated trace compared to the previous trace version (v2021) and includes additional information such as the service ID in the call graph. In addition, the number of BMs and MSs has increased to 40,000+ and 28,000+ from the previous 1,300+ and 1,300+, respectively. The trace is composed of approximately 61.1 GB of compressed dataset, including four main components labeled as *node*, *MSCallGraph*, *MSResource*, and *MSRTMCR*. Specifically,

- `node`: BM Node runtime information.
- `MSCallGraph`: The MS Call Graphs information comprises over twenty million call graphs involving more than 17,000 microservices distributed across more than ten clusters.
- `MSResource`: The MS runtime information tracks the CPU and memory utilization of over 470,000 containers for more than 28,000 microservices within the same production cluster.
- `MSRTMCR`: The information on the MCR and RT captures the MCR and RT metrics for calls conducted through various communication paradigms among more than 28,000 microservices, using over 470,000 containers within the same production cluster.

Based on this dataset, Luo et al. [99] conduct an in-depth analysis of microservice call graphs to quantify their differences from traditional DAGs used in data-parallel jobs. Khodabandeh et al. [68] analyze the dependencies and collaborative relationships among microservices, identifying similarities between service graphs.

CERIT-SC Workloads (2022) [71]: This dataset, published by Dalibor Klusacek in 2016, aims to track the real workload of the CERIT-SC system. The CERIT-SC system represents the largest portion of the Czech national grid and cloud infrastructure, MetaCentrum, featuring a total of 5,224 CPU cores. Among these, 3,912 CPU cores (constituting 75% of the total) are completely virtualized utilizing the OpenNebula framework, making them available for a range of applications, while the remaining 1,312 CPU cores are non-virtualized and dedicated to "bare-metal" grid computing. The dataset primarily presents job-related metrics, including *job submission time*, *wait time*, *runtime*, *allocated CPU count*, *average CPU utilization time*, and *memory usage*. Additionally, Dalibor Klusacek has also released other versions of the dataset on the website [71]. Based on this dataset, Klusáček et al. [72] conduct research on workload scheduling problems.

OpenNebula Virtual Machine Profiling Dataset (2021) [100]: This dataset, published by Prasad Purnaye on IEEEDataPort in 2021, records the operational information of virtual machines, including both normal and attack data. The dataset is collected by performing a series of probing programs supplied by OpenNebula, gathering monitoring information from 6 virtual machines over a period of 63 h, with simulations of attacks of varying durations conducted on a subset of the virtual machines. Based on known attack scenarios, the data has been labeled as normal or attack data. This dataset can be utilized for analyzing virtual machine behavior based on the monitoring capabilities of OpenNebula and is widely used for intrusion detection purposes. It is noteworthy that access to this dataset requires a subscription fee of $40. Based on this dataset, Saxena et al. [107] develop a virtual machine threat prediction model, aiming to safeguard computational data and minimize adversarial disruptions through proactive estimation of virtual machine threats.

Fruktus (2020) [87]: This dataset, released by Nawrocki in 2020, pertains to the CPU usage of virtual machines within a cluster. The cluster comprises more than 1700 VMs from the Nokia Solutions Krakow cluster and more than 100 VMs from the Polcom cluster. The dataset consists of 100 CSV files with a total size of 176 KB. Each file contains two fields: *timestamp* and *CPU usage (average)*. Based on this dataset, Nawrocki et al. [88] conduct research on anomaly detection techniques in the context of long-term cloud resource usage planning.

Chameleon Cloud traces (2019) [35]: This dataset is based on the OpenStack Nova/Blazar/Ironic services and has been released by Science Clouds in 2017. Since then, ten cloud traces spanning from 2017 to 2020 have been made available. The dataset includes information on the *CPU allocation*, *memory*, and detailed *workload*. It is suitable for purposes such as workload prediction. Detailed descriptions of the data table structure can be found at https://scienceclouds.org/cloud-traces/cloud-trace-format/. Based on this dataset, Kang et al. [66] investigate the "resource silo" issue in cloud computing, where resources are restricted to specific models, thereby reducing utilization potential and flexibility while increasing costs.

Azure Public Dataset (2017, 2019) [85]: This dataset, released by Microsoft, aims to record the workload status of VMs on the Microsoft Azure platform, covering key aspects such as CPU and memory usage. The dataset is divided into two versions, released in 2017 and 2019, respectively, corresponding to the workload status of Azure virtual machines for those two years. The first version of the dataset has a size of 117 GB and includes the following fields: *Max CPU utilization during the 5 min*, *VM category*, *Encrypted subscription id*, *Timestamp in seconds*, *VM virtual core count*, *Count VMs created*, *VM memory (GBs)*, *Avg CPU utilization during the 5 min*, *P95 of Max CPU utilization*, *Deployment size*, *Timestamp VM created*, *Timestamp in seconds*, *Timestamp VM deleted*, *Max CPU utilization*, *Avg CPU utilization*, *Min CPU utilization during the 5 min*, *Encrypted VM id*, and *Encrypted deployment id*. In comparison to the first version, the 2019 release has expanded to 235 GB. Based on this dataset, Jayakumar et al. [117], Cortez et al. [36], and Sathiya et al. [102] conduct research on cloud workload prediction. Zhang et al. [132] analyze Microsoft Azure's VM workloads in terms of memory, lifecycle, average utilization, and cost, and develop a pricing model.

Scout (2018) [33]: This dataset, released by Hsu Chin-Jung in 2018, contains extensive performance data of Hadoop and Spark applications running on AWS EC2. The dataset has a size of 587 MB and is stored in CSV format. It covers both single-node and multi-node scenarios. The single-node section comprises 18 types of virtual machine configurations, while the multi-node section encompasses 69 configurations, consisting of 9 virtual machine types with varying numbers of instances. Within each configuration, execution time and underlying performance metrics are collected using the sar tool, by combining input sizes and running programs. The primary fields include *CPU idle time ratio*, *pages read from disk per second*, *kernel-mode CPU utilization*, and *user-mode CPU utilization*, totaling 73 features. Based on this dataset, Bilal et al. [20] conduct research on black-box optimization techniques for implementing automated cloud configuration.

IBM Docker Registry traces (2018) [13]: This dataset, released by Ali Anwar et al. in 2018, involves tracing of Docker registries. It captures all GET, PUT, HEAD, PATCH, and POST requests. The tracing period spans from June 20, 2017, to February 9, 2017, covering 75 days and totaling over 38 million requests. The dataset is compressed to a size of 1.5 GB. It comprises 10 fields: *host*, *id*, *http.request.uri*, *timestamp*, *http.response.writing*, *http.request.method*, *http.request.useragent*, *http.request.remoteaddr*, *http.response.status*, and *http.request.duration*. Based on this dataset, Zhao et al. [135] conduct research on registry technologies in Docker, focusing on deduplication techniques. Similarly, Anwar et al. [14] analyze production workloads within the dataset, emphasizing improvements to Docker registry design.

Business Critical Workloads (2015) [94]: This dataset, released by Bitbrains in 2015, aims to document the workload conditions of 1750 VMs in Bitbrains' distributed cloud clusters in 2013. The dataset is divided into two parts. The first part, called fastStorage, includes the workload conditions of 1,250 virtual machines during August 2013. This section consists of 1,250 CSV files with a total size of 128 MB. The dataset contains the following fields: *Timestamp*, *Memory usage*, *CPU cores*, *Disk write throughput*, *Disk read throughput*, *CPU capacity provisioned (CPU requested)*, *CPU usage (MHz)*, *Network received throughput*, *CPU usage (%)*, *Network transmitted throughput*, *Memory provisioned*, and *Memory provisioned*. The second part, named Rnd, includes the workload conditions of 500 virtual machines during July, August, and September 2013. This portion of the dataset has a size of 156 MB and comprises 150 CSV files, with fields identical to those in the first part. Based on this dataset, Leka et al. [76], Matoussi et al. [83], and Bhagtya et al. [18] conduct research on cloud virtual machine workload prediction, while Shen et al. [111] perform an in-depth analysis of cloud virtual machine workload characteristics.

Planetlab VM traces (2011) [12]: This dataset, published by Beloglazov Anton in 2011, aims to record the CPU utilization of virtual machines on the Planetlab platform. The dataset covers a period from March to April 2011, during which random ten-day intervals are selected, and *CPU utilization* is measured every 5 min for over 11,000 virtual machines. Based on this dataset, Qiu et al. [101] and Zhang et al. [134] conduct research on cloud virtual machine workload prediction, while Yadav et al. [129] focus on predicting the minimum utilization of virtual machines.

5.3 Datasets with Web Trace

In this section, a detailed description of each dataset is categorized under the web trace type, including information such as time, purpose, and fields.

ISCX (2016) [45]: The Information Security Center of Excellence is a network traffic dataset released by the Canadian Institute for Cybersecurity in 2016, covering the period from 2009 to

2016. It is utilized for researching and developing technologies related to network security. The ISCX dataset comprises multiple distinct network traffic datasets, including ISCXVPN2016, ISCX-Tor2016, ISCX-URL2016, ISCX-Bot-2014, and ISCX IDS 2012.

- `ISCXVPN2016` [48]: This dataset covers both regular sessions and VPN sessions, encompassing 14 traffic classifications, including VOIP, VPN-VOIP, P2P, VPN-P2P, and others. Additionally, the dataset generates seven various categories of traffic: Browsing, Email, Chatting, Streaming, File Transfer, VoIP, and TraP2P.
- `ISCXTor2016` [47]: This dataset encompasses real-world traffic data, divided into non-Tor and Tor traffic categories. The non-Tor traffic utilizes benign traffic from the VPN project described in Reference [48] as its source; meanwhile, the Tor traffic includes seven distinct types of traffic: Email, VoIP, Chat, P2P, Video-Streaming, Browsing, FTP, and Audio-Streaming.
- `ISCX-URL2016` [46]: This dataset contains a collection of real-world URLs, which are classified based on their potential attack types into five categories: Benign URLs, Spam URLs, Phishing URLs, Malware URLs, and Defacement URLs.
- `ISCX-Bot-2014` [44]: This dataset is designed to evaluate botnet detection methods and is split into training and testing datasets. The training set encompasses seven types of botnets, while the testing dataset contains sixteen types. The size of the training set reaches 5.3 GB, of which 43.92% is marked as malicious, with the remainder representing normal network traffic. The size of the testing set is 8.5 GB, with 44.97% of the data flow identified as malicious.
- `ISCX IDS 2012` [43]: This dataset is dedicated to the evaluation of *network intrusion detection* and exhibits the following characteristics: it reflects real-world network environments and traffic patterns, the dataset is annotated, it encompasses comprehensive interaction capture, achieves data capture completeness, and includes diverse intrusion scenarios.

Based on this dataset, Alqahtani et al. [10] conduct research on intrusion detection, aiming to enhance security by analyzing network traffic data.

Amazon Resource Cost (2012) [118]: The dataset, released by Rizwan Mian and others from Queen's University in 2012, aims to investigate the workload characteristics of VMs on the Amazon Web Services cloud computing platform. Importantly, it provides various configurations of virtual machines along with their corresponding *costs* ($). This dataset encompasses three types of virtual machine specifications: small, large, and xlarge. Furthermore, the dataset includes ten fields for describing and analyzing the performance and cost aspects of virtual machines, such as *I/O*, *I/O($)*, *data in (GB)*, *data in ($)*, *data out 1 TB ($)*, and *data out 10 TB ($)*. Based on this dataset, Mian et al. [84] explore an experimental-driven performance model for executing data-intensive workloads in **Infrastructure-as-a-Service (IaaS)** public clouds and conduct cloud workload prediction.

It is worth noting that there are small amounts of missing data for certain metrics in the dataset, such as provisioned (GB-Ho), provisioned ($), I/O, data in (GB), and data out 1 TB (GB). Users can remove these missing values before use, as this does not significantly affect the overall usability of the dataset.

Cloud Intrusion Detection Dataset (2012) [89]: The dataset, released by the MIT Lincoln Laboratory DARPA intrusion detection evaluation team in 2012, supports research in cloud intrusion detection. It not only facilitates the identification of masquerade attacks but also enables the detection of over one hundred different categories of attacks, including ***User to Root (U2R)***, *Data Attacks (including Data Modification and Tampering)*, ***Remote to Local (R2L)***, *Unusual User Behaviors*, *surveillance and probing*, and ***Denial of Service (DoS)***.

The dataset is divided into two main parts: (1) Contains Unix Solaris auditing data and its associated TCP dump data; (2) Including Windows NT auditing data and its associated TCP dump data. These data aim to provide an empirical basis for training and testing **Intrusion Detection Systems (IDS)**.

Based on this dataset, Kholidy et al. [69] conduct research on cloud intrusion detection and release the dataset as part of this work.

Wikipedia Web Traces from WikiBench (2007) [61]: This dataset, released by Wikipedia, covers the HTTP request information of the Wikipedia website from 2007 to 2016, spanning 9 years. The dataset is massive, estimated to be around 5 TB in size. It includes the following fields:

- `domain_code`: Request domain name in shortened form.
- `page_title`: For page-level files, it contains the title of the part following "/wiki/" in the request URL, which has not been normalized (for example, Main_Page, Berlin).
- `count_views`: The count of views this page has received during the specified hour.
- `total_response_size`: The total size of responses generated by requests for this page within the given hour.

Based on this dataset, Wu et al. [127] conduct research on cloud workload prediction. Urdaneta et al. [119] classify client requests and analyze aspects such as the number of read and save operations, significant load variations, and requests for non-existent pages, while discussing approaches to handling Wikipedia workloads. Ling et al. [80] investigate the issue of virtual machine resource allocation, and Lei et al. [75] explore sharing techniques among MapReduce jobs.

It's worth noting that this dataset is discontinued on August 1, 2016, and users need to contact the responsible person to obtain it.

KDD Cup 1999's Dataset (1998) [74]: The dataset serves as a classic dataset for network intrusion detection. It is initially simulated by the MIT Lincoln Labs between 1998 and 1999 to mimic the environment of a United States Air Force **Local Area Network (LAN)**. Launch several attacks are launched against this LAN, and corresponding network traffic records are collected. The dataset aims to provide researchers with a realistic environment of network traffic to estimate the quality of intrusion detection systems. It includes a vast amount of network connection data recorded during real-time monitoring of simulated network traffic. These data cover diverse categories of network activities, including natural user behaviors and varying types of network attacks such as *R2L*, *DoS*, *U2R*, and *Probing*. Each record contains various features regarding the connection, for instance, the *source IP address*, *destination IP address*, *used protocol*, *connection duration*, *transmitted data volume*, and so on. Detailed descriptions of the data table structure can be found at https://kdd.ics.uci.edu/databases/kddcup99/kddcup.names. Based on this dataset, Zuhair et al. [137] and Chen et al. [31] investigate intrusion detection techniques for cloud platforms to protect cloud servers.

NASA HTTP Traces (1995) [23]: This dataset is released by the Kennedy Space Center in 1995. It records all HTTP requests received by the NASA Kennedy Space Center WWW server located in Florida for a period of two months in 1995. Each request is logged in a log file with the following fields:

- `Host initiating the request`: If the hostname cannot be determined, then the IP address is provided.
- `Timestamp`: The format is "DAY MON DD HH:MM:SS YYYY," with a time zone of −0400.
- `HTTP request`: For example, "GET /logs/USA-1995.log HTTP/1.0."
- `HTTP response status code`: The server returns the response status code, such as 404, 500, etc.
- `Number of bytes in the response`: The server provides the size of the response content in bytes.

Table 6. Comparison of Four General Resource Characteristics Datasets

	Google Cluster Data [52]	**Alibaba Cluster Trace [6]**	**Azure Public Dataset [85]**	**Alibaba MicroServices Traces [9]**
Datacenter	Google cluster	Alibaba production cluster	Microsoft Azure	Alibaba cloud
Year	2011/2019	2017/2018	2017/2019	2021/2022
Duration	7 h/29 days/1 month	12 h/8days	30 days	12 h/13 days
#VM	—	—	2M/2.6M (VMs)	470,000+(containers)
#Microservices	—	—	—	28,000+
Resource Type	CPU, Memory, Disk	CPU, Memory, Disk	CPU, Memory	CPU, Memory
Data content	Host	Host	VM (Bare-metal)	VM (Micro-Services)

Based on this dataset, Kirchoff et al. [70] and Kumar et al. [73] conduct research on cloud workload prediction. Li et al. [77] investigate load-balancing-based resource management in edge and cloud environments.

6 Analysis and Comparison of Mainstream Datasets

In this section, we select 12 currently mainstream datasets from 42 datasets. “Mainstream” datasets refer to high-quality datasets that are widely used in the field of cloud computing, are updated in a timely manner and cover current research hotspots. Specifically, the selection of these datasets is based on three key categories: *usage frequency*, *timeliness*, and *year*. Usage frequency is measured by the frequency with which these datasets are cited and used in cloud computing literature. Timeliness focuses on the update cycle of the datasets, while year considers whether the datasets were created in recent years, ensuring that they reflect the latest technological trends.

To better analyze and compare these datasets, we classify them into three categories based on the resource information recorded in the datasets: (1) datasets containing general resource logs, such as CPU, memory, and disk usage; (2) datasets recording GPU-related log information; and (3) datasets covering energy-related log information. Specifically, the first category of datasets is mainly suitable for studying the load, resource consumption, and performance optimization of small-scale tasks processed by CPUs. These datasets provide detailed records of general computing resource usage. The second category of datasets is suitable for more complex research on GPU-based computing tasks, helping to study GPU resource utilization, performance bottlenecks, and parallel computing efficiency. The third category of datasets is suitable for energy efficiency research, as they help researchers gain insights into the energy efficiency of different computing tasks, explore ways to optimize system energy usage and improve the efficiency of green computing. Additionally, considering **large language models (LLMs)** as the mainstream direction of current research, we introduce a fourth category of datasets, which includes logs related to LLM inference. Subsequently, we compare and analyze each category of datasets and present the results in the form of charts to help readers better select the datasets they need.

6.1 General Resource Characteristics Datasets

As shown in Table 6, we select four currently mainstream general resource characteristics datasets: Google Cluster Data, Alibaba Cluster Trace, Azure Public Dataset, and Alibaba MicroServices Traces. Table 6 provides a comparative analysis of the specifications and tracking information of these four datasets. Among them, Google Cluster Data and Alibaba Cluster Trace belong to the cloud host cluster architecture, while Azure Public Dataset and Alibaba MicroServices Traces are part of the cloud virtual machine cluster architecture. Additionally, all four datasets record tracking information related to CPU and memory, with Google Cluster Data and Alibaba Cluster Trace further including logs associated with disk usage.

Resource Utilization of Jobs. As shown in Figure 5, we use the **Cumulative Distribution Function (CDF)** as a metric to analyze the distribution of CPU utilization and memory utilization

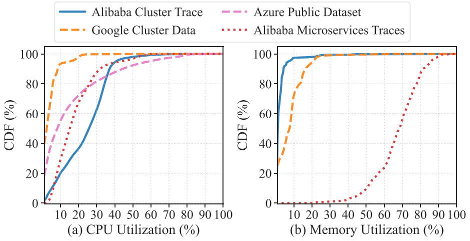


Fig. 5. Overview of different dataset characteristics. (a) CDF vs. CPU utilization. (b) CDF vs. Memory utilization, where Azure public dataset is not available.

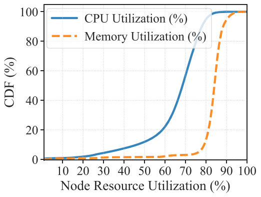


Fig. 6. CDF vs. node resource utilization of Alibaba MicroServices Traces.

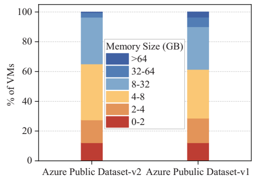


Fig. 7. VM memory configuration information in the Azure Public Dataset.

across various datasets. In this case, due to the large scale of the data (e.g., the total size of the Google Cluster Data is 2.4 TB), we select 100,000 traces from each dataset for analysis. The analysis shows that the CPU utilization in the Google Cluster Data is relatively low (i.e., 90% of the jobs have CPU utilization below 20%), and the memory utilization is also below 20%. This phenomenon indicates that the Google Cluster Data primarily consists of small-scale jobs. In contrast, the overall resource utilization in the Alibaba MicroServices Traces is higher (i.e., 60% of the jobs have CPU utilization above 20%, and 80% of the jobs have memory utilization exceeding 60%). This suggests that the jobs in the Alibaba MicroServices Traces are mainly large-scale jobs. The resource utilization in the Alibaba Cluster Trace and Azure Public Dataset appears relatively balanced.

Resource Utilization of nodes in Alibaba Microservices Traces. As shown in Figure 6, we further analyze the distribution of CPU and memory utilization for each node in the Alibaba Microservices Traces dataset. Similarly, we observe that the resource utilization at the node level is also high (i.e., 80% of the nodes have both CPU and memory utilization exceeding 60%). If users are seeking datasets with large-scale jobs and high resource utilization, then the Alibaba Microservices Traces dataset is a suitable choice.

VM memory configuration in Azure Public Dataset. As shown in Figure 7, we present the memory distribution of virtual machines in versions 1 and 2 of the Azure Public Dataset. The analysis indicates that this dataset contains a diverse range of virtual machine memory configurations, with the majority of virtual machines having memory sizes between 4 and 32 GB. Therefore, readers can select traces with specific memory configurations.

6.2 GPU-enhanced Resource Characteristics Datasets

As shown in Table 7, we select four mainstream datasets that include GPU-related log information: Alibaba GPU Traces, Acme (), Helios, and Philly. Among them, Acme primarily records workloads related to the development of LLMs, while Alibaba GPU Traces, Helios, and Philly cover general DL workloads from various domains. For example, Helios consists of four clusters dedicated to training tasks in computer vision and reinforcement learning, whereas Alibaba GPU Traces includes multiple server configurations used for training and serving tasks. Specifically, Acme uses A100 GPUs, Helios includes 1080Ti and V100 GPUs, Alibaba GPU Traces includes T4, P100, and V100 GPUs, and Philly only provides GPU memory size, without specifying the GPU model. Additionally, the average number of GPUs requested in Alibaba GPU Traces may be less than 1 (i.e., the average number of GPUs is 0.7). For the Acme, we use the traces from the *Seren* and *Kalos* clusters as examples. Specifically, Seren includes 368K CPU tasks and 664K GPU tasks, while the Kalos task trace contains 42K CPU tasks and 20K GPU tasks.

Table 7. Comparison of Four GPU-enhanced Resource Characteristics Datasets

	Alibaba GPU Traces [7]	**Acme [54]**	**Philly [104]**	**Helios [108]**
Datacenter	Alibaba PAI	Acme	Microsoft Philly	SenseTime Helios
Year	2020	2023	2017	2020
Duration	2 months	6 months	3 months	6 months
#Jobs	1.26M	1.09M	113K	3.36M
Avg. #GPUs	0.7	6.3	1.9	3.7
GPU Model	T4/P100/V100	A100	12 GB/24 GB	1080Ti/V100
Total #GPUs	6,742	4,704	2,490	6,416
Job Type	DL Models	LLMs	DL Models	DL Models

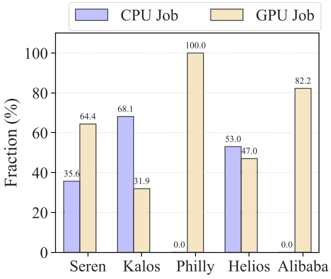


Fig. 8. Distribution of CPU jobs and GPU jobs.

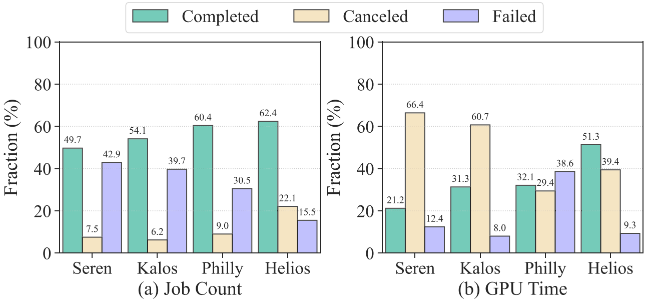
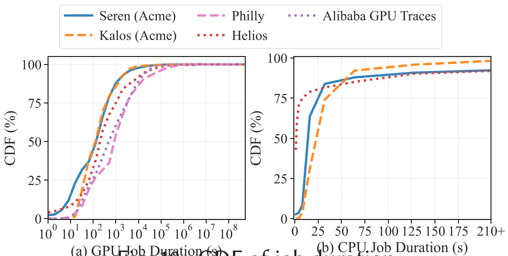


Fig. 9. Final status of the jobs: (a) number of jobs and (b) GPU resources usage, where Alibaba GPU Traces is not available.

Job distribution and its final state. As shown in Figure 8, we analyze the distribution of CPU tasks and GPU jobs across four datasets. The results show that Philly and Alibaba GPU Traces contain only GPU-related workloads, while Helios and Acme (i.e., Seren and Kalos) include both CPU and GPU jobs. Kalos primarily consists of GPU jobs, whereas Seren primarily consists of CPU jobs. Figure 9 summarizes the distribution of three key final states across the four datasets. The result shows that in the Acme, which primarily involves LLMs jobs, approximately 40% of the jobs end in failure, consuming 10% of the GPU resources. In contrast, the Philly and Helios datasets, which focus on traditional DL jobs, exhibit a relatively lower proportion of failed tasks. Additionally, we observe that a significant amount of GPU resources is wasted on debugging and failed tasks. For example, in the Seren and Kalos, the GPU resources consumed by debugging jobs account for 66.4% and 60.7%, respectively.

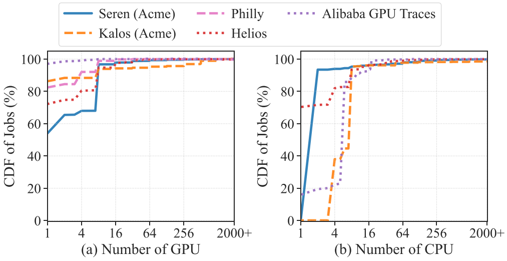
Job duration. Figures 10(a) and 10(b) show the duration distributions of GPU and CPU jobs, respectively. Intuitively, LLM-related tasks typically require longer runtimes. However, we observe that the workloads in the Acme (represented by the blue and orange lines) exhibit shorter GPU job durations. This may be due to the fact that a large number of tasks in Acme have a final status of failure. Additionally, we find that the workloads in the Helios exhibit shorter CPU job durations.

Workload distribution. Figures 11(a) and 11(b) show the proportions of tasks with different numbers of GPUs and CPUs, respectively. The analysis indicates that approximately 95% of the tasks in the Alibaba GPU Traces use only one GPU, and the number of CPUs required is the highest among all datasets. This means that the jobs in Alibaba GPU Traces are highly dependent on CPU resources. In contrast, in the Acme and Helios, the number of GPUs required for jobs is relatively higher, indicating a greater demand for GPU resources in these datasets.



(a) GPU Job Duration (s)
(b) CPU Job Duration (s)

Fig. 10. CDF of job duration.



(a) Number of GPU
(b) Number of CPU

Fig. 11. GPU and CPU demands vs. CDF of jobs.

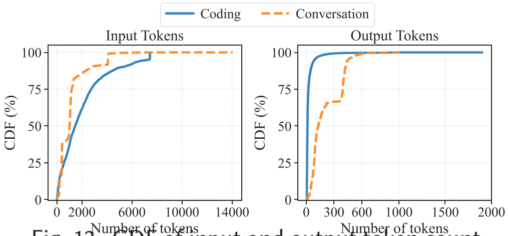


Fig. 12. CDF of input and output token count.

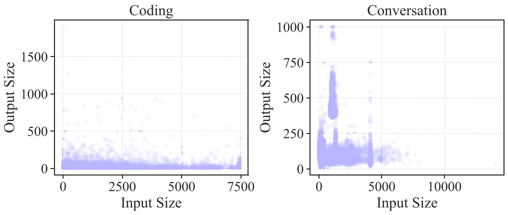


Fig. 13. Relationship between input and output size.

6.3 LLMs Inference Datasets

In this section, given that LLMs are one of the most popular research directions in recent years, we analyze the Azure LLM Inference Dataset [16], a cloud dataset currently collected for LLM inference. This dataset primarily contains traces of two types of jobs: Coding and Conversion jobs.

As shown in Figure 12, we plot the CDF of input token counts and output token counts. The analysis reveals that coding jobs typically require more input tokens but generate fewer output tokens. In contrast, conversion jobs generally require fewer input tokens but produce more output tokens. Further, Figure 13 shows the relationship between input size and output size for the two types of tasks. The analysis reveals that the output size of coding tasks is primarily concentrated around 100, with little variation as the input size increases. In contrast, the output size of conversion tasks ranges from 10 to 1,000, exhibiting a wider distribution.

6.4 Power-aware Resource Characteristics Datasets

In this section, we focus on analyzing three major datasets related to energy data: SURFsara, Marconi100, and Google Power Data. As shown in Table 8, both the SURFsara and Marconi100 datasets record relatively complete log information, including CPU, GPU, memory, disk, and power data. Additionally, the Marconi100 records comprehensive ambient data such as temperature, humidity, wind speed, visibility, clouds, and pressure. In contrast, the Google Power Data only records power-related log information, with other resource logs (such as CPU, memory, and disk) requiring integration with the Google Cluster Data [52].

Power utilization analysis. As shown in Figure 14, we plot the CDF of total power and GPU power for SURFsara and Marconi100. The analysis reveals that both the overall power consumption and GPU power of SURFsara are lower compared to Marconi100 (e.g., the GPU power of SURFsara is less than 250 W, while that of Marconi100 is concentrated between 500 W and 1,000 W). This likely indicates that Marconi100's workloads are more GPU-dependent and consist of larger jobs, whereas SURFsara primarily supports CPU-dependent smaller jobs.

Table 8. Comparison of Three Power-aware Resource Characteristics Datasets

	SURFsara [116]	**Marconi100 [22]**	**Google Power Data [53]**
Datacenter	SURFsara	Marconi	Google cloud
Year	2019	2020	2024
Duration	8 months	9 months	1 month
Resource Type	CPU, GPU, Memory, Disk, Network, Power	CPU, GPU, Memory, Disk, Power	Power
Ambient Factor	Temperature	Temperature, Humidity, Wind Speed, Visibility, Clouds, Pressure	—
cooling systems	—	Air-cooling, Water-cooling	—

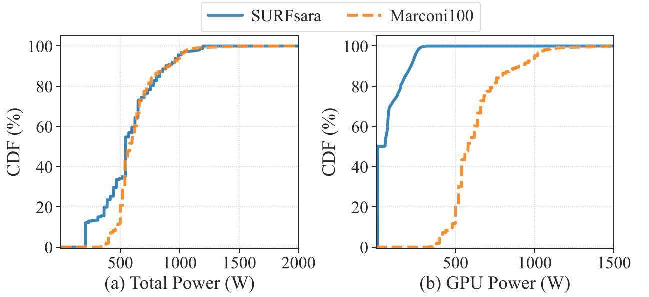


Fig. 14. CDF of total power and GPU power for SURFsara and Marconi100.

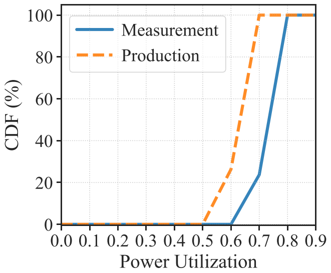


Fig. 15. CDF of power utilization for Google Power Data.

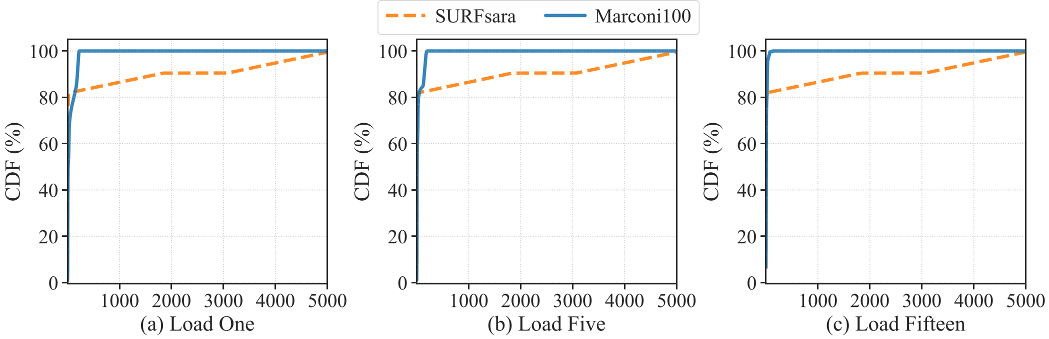


Fig. 16. CDF of load for SURFsara and Marconi100.

Moreover, as shown in Figure 15, we analyze the CDF of two metrics provided by Google Power Data: measured power and production power. Measured power represents the actual power utilization, while production power is estimated through interpolation based on CPU utilization and idle/busy machine power. The analysis reveals that the estimated production power utilization is significantly lower than the measured power utilization.

Workload analysis. As shown in Figure 16, we analyze the CDF of the load for SURFsara and Marconi100 using time windows of 1 min, 5 min, and 15 min. The analysis reveals that the load for SURFsara is higher, reaching up to around 5,000, while the load for Marconi100 is mostly concentrated around 100.

Temperature analysis. As shown in Figure 17, we first analyze the CDF of GPU temperature for SURFsara and Marconi100. The analysis reveals that the GPU temperature of SURFsara is

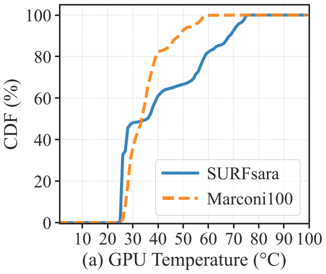


Fig. 17. CDF of GPU temperature for SURFsara and Marconi100.

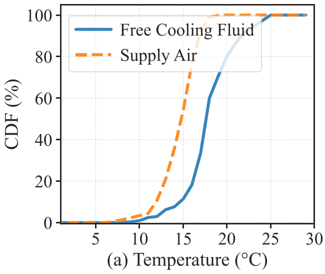


Fig. 18. CDF of temperature of free cooling fluid and supply air for Marconi100.

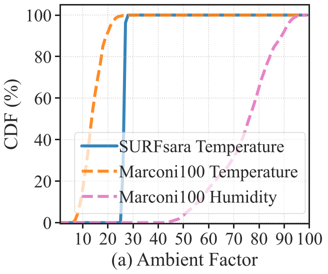


Fig. 19. CDF of ambient factor for SURFsara and Marconi100.

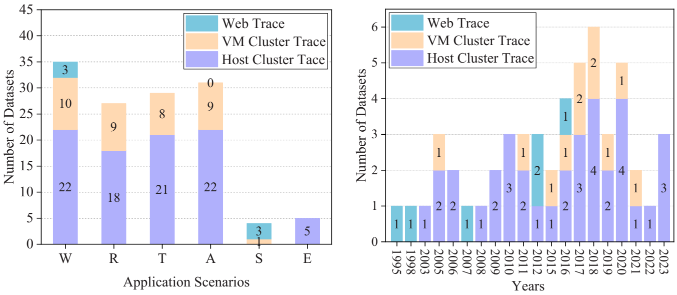
significantly higher than that of Marconi100 (e.g., the GPU temperature of SURFsara reaches up to 80°C, while the GPU temperature of Marconi100 peaks at 60°C). We hypothesize that this difference may be due to Marconi100's use of both air cooling and water cooling systems. Subsequently, in Figure 18, we present the CDF of temperature for the free cooling fluid and supply air in Marconi100. The results show that free cooling fluid has a higher overall temperature than supply air, which further validates the effectiveness of the water cooling system.

Ambient analysis. As shown in Figure 19, we plot the CDF of ambient factors, including ambient temperature and ambient humidity. The analysis reveals that the ambient temperature of Marconi100 is significantly lower than that of SURFsara (i.e., the average ambient temperature of Marconi100 is 15°C, while the average ambient temperature of SURFsara is approximately 28°C). Combined with Figure 17, this further indicates that ambient factors have a direct impact on reducing cluster temperature.

7 Challenges and Future Directions

In this section, we emphasize the challenges associated with the datasets we cover and present directions for further research. Specifically, we identify the shortage of datasets containing complete attack types, the scarcity of energy-related datasets, the poor timeliness of datasets, and the small scale of the provided datasets.

7.1 Security and Attach Traffic



(a) The number of datasets for application scenarios.

(b) The number of datasets per years.

Fig. 20. Number of datasets for application scenarios and for each year.

In Figure 20(a), the quantities of publicly available data in six application scenarios are depicted. It is observed that there are only four datasets for the security scenario (S). Despite the theoretical significance of the security scenario, only four datasets are currently applicable to it. This may indicate that the security scenario has not received widespread attention in practical applications or has not garnered sufficient focus from industry or academia. In summary, for this scenario, there is only a minimal amount of publicly available data for researchers to study new algorithms.

Additionally, there are limitations in some of the datasets that may reduce their utility for security researchers. On the one hand, the datasets may lack timeliness, as they are created years ago and may not accurately represent current conditions due to the rapid evolution of computer hardware and software. For instance, the KDD Cup 1999's Dataset [74] is created in 1998, and the attack methods at that time may no longer be relevant today. On the other hand, most datasets contain a limited range of attack traffic. For example, the Cloud Intrusion Detection Dataset [89] only includes *U2R*, *surveillance and probing*, *Data Attacks (including Data Modification and Tampering)*, *R2L*, *DoS*, and *Unusual User Behaviors*, but it does not cover *R2L*, *SQL Injection*, and *Man-in-the-Middle* attacks. To our knowledge, there are currently no publicly available datasets that include all popular types of network attacks.

Therefore, we expect future researchers in this domain to consider generating and publishing publicly available datasets that include all popular types of network attacks. The public release of such datasets would be extremely beneficial for research in network security for cloud computing.

In some cases, researchers may be unwilling to publicly release the data they have collected. This may be due to considerations of privacy and confidentiality, or it may be restricted by funding sources (such as industry collaborations). To partially address this issue, works [24] proposes a method that allows for both sharing datasets and protecting personal information [37]. Their approach suggests using small portions of the data, known as "annotation units," which represent subsets of real-world traffic traces [37, 123]. These units are easier to anonymise and then share with academia and industry.

7.2 Energy Efficiency and Green Computing

Recently, the importance of green computing in cloud computing has been increasingly emphasized [63]. With growing global concern for sustainable development, there is a rising demand from both businesses and individuals for environmentally friendly solutions. Cloud computing, as a powerful information technology tool, provides users with flexible, scalable, and efficient computing resources [82]. However, its energy consumption issue has also attracted widespread attention. To mitigate the environmental impact of cloud computing, green computing has become a significant solution [63]. Green computing focuses on reducing energy use and carbon emissions through enhancing the energy efficiency of data centers, utilizing renewable energy sources and energy-conserving technologies, and decreasing waste of hardware resources. Therefore, energy-related datasets have become extremely important, as they can assist researchers and practitioners in understanding and analyzing energy usage in data centers, thus enabling the development of effective green computing strategies and solutions.

Unfortunately, our survey results indicate that only a small number of datasets are related to energy efficiency. As shown in Figure 20(a), we found that there are only five datasets applicable to the energy efficiency scenario (E). This may suggest that energy efficiency has not received widespread attention in practical applications, or has not garnered enough focus from industry or academia. In summary, for this scenario, only a very limited amount of public data is available for researchers to understand and conduct replicable experiments.

Therefore, we hope that scholars engaged in research in this field in the future will consider generating and releasing comprehensive and up-to-date energy-efficiency public datasets. The public release of such datasets is crucial for studying the energy-efficiency and green computing of cloud computing.

7.3 Scale and Complexity

As shown in Tables 2–4, the majority of datasets are collected from a relatively small number of computing nodes. For example, GWA-T-1 DAS2 [91], Autoscale Analyse [19], and OpenCloud Hadoop Workload [95] have only 5, 2, and 64 computing nodes, respectively. However, the scale of computing clusters used in real-world applications is typically very large. Therefore, we advocate for the academic community to collect and release more datasets from such large-scale cloud computing clusters.

Another unresolved concern is the complexity of current datasets. Some datasets contain only specific metrics; for example, Autoscale Analyser [19] only records *CPU* and *memory usage*, while SPEC CPU [113] only records *CPU usage*. However, in cloud computing clusters, it is often necessary to monitor multiple metrics simultaneously, such as *CPU*, *I/O*, *memory*, *GPU*, *network*, and so on. Thus, we advocate that future datasets must reflect this complexity.

7.4 Obsolete

Due to the rapid development in cloud computing, datasets recorded a decade ago may contain tracking records that are no longer relevant today. Simultaneously, significant changes occur in resource utilization, workloads, and other aspects due to the rapid progress in software and hardware technologies. Therefore, the year of the dataset is a key indicator in assessing its representativeness in real-world tracking.

As shown in Figure 20(b), the creation times of all datasets (including all published versions) are displayed. We found that close to half of the datasets are created 10 years ago, and some datasets have been in existence for over 20 years (such as NASA HTTP Traces (1995)). Researching these datasets has lost practical significance for current computer clusters. Therefore, we urge the academic community to collect and publicly release more datasets collected in recent years, and to use the latest datasets for experimental support.

Furthermore, as shown in Figure 20(b), we found that the majority of datasets related to Web Trace are published 10 years ago. Hence, We expect that future researchers working in this field will generation and publication of public datasets related to web trace. The public release of such datasets is crucially beneficial for studying network security in cloud computing.

Additionally, the Supplemental Materials provide an in-depth exploration of **mainstream technologies and future research directions**, alongside a comprehensive discussion on **long-term dataset maintenance and updates**. See Supplemental Materials C and D for details.

8 Conclusion

Datasets play a vital role in numerous research fields. Having a sufficient number of high-quality datasets is essential for generating new scientific knowledge and ensuring the accuracy and dependability of experimental findings. Specifically, public datasets enable the benchmarking of solutions created by various groups and support the sharing of results that can be replicated and verified by others.

In this article, we conduct a comprehensive survey of available datasets suitable for cloud computing research. Through the analysis of 968 scientific papers, we collect and classify a total of 42 datasets. The majority of these datasets are applicable to three application scenarios: workload analysis, resource allocation, and task scheduling, while fewer datasets are suitable for energy efficiency, anomaly detection, and security scenarios. We classify the datasets based on 11 observed characteristics, such as data collection dates, duration, and number of machines, to assist researchers in finding datasets tailored to their specific needs. Third, We provide detailed descriptions of each dataset to assist researchers in gaining a clearer understanding of their characteristics, including descriptions, structure, fields, purposes, and so on. Fourth, we select 12 mainstream datasets and conduct a comprehensive analysis and comparison of their characteristics. Finally, we discuss the existing issues with the datasets and highlight future directions for creating datasets in cloud computing.

Our conclusion is that publicly available datasets covering energy efficiency and Web Trace are still scarce or even lacking. Moreover, most datasets contain few performance metrics, and there is currently a lack of large-scale datasets containing comprehensive metrics. Additionally, a significant number of current datasets lack attack traffic. While datasets containing only benign traffic are still valuable for security research, incorporating attack traffic to enhance them needs extra effort and careful planning. Therefore, we encourage future authors to publicly release more comprehensive datasets and to integrate a variety of attack scenarios into their experiments.

References

[1] A. V. Papadopoulos et al. 2021. Cloud Benchmarking Methodology TSE. Retrieved from https://drive.google.com/file/d/151guslA9SYV-8BJNMXa1udvMrF4jn2ae/view?usp=embed_facebook

[2] Dionatra F . Kirchoff. 2018. Dionatrafk. Retrieved 2018 from https://github.com/dionatrafk/workload_prediction

[3] Bikash Agrawal, Tomasz Wiktorski, and Chunming Rong. 2016. Adaptive anomaly detection in cloud using robust and scalable principal component analysis. In *Proceedings of the 15th International Symposium on Parallel and Distributed Computing (ISPDC'16)*. 100–106. https://doi.org/10.1109/ISPDC.2016.22

[4] Shallaw Mohammed Ali and Gabor Kecskemeti. 2023. SeQual: An unsupervised feature selection method for cloud workload traces. *J. Supercomput.* 79, 13 (Sep. 2023), 15079–15097. https://doi.org/10.1007/s11227-023-05163-w

[5] Alibaba. 2017. Alibab Cluster Trace Version 1. Retrieved from https://github.com/alibaba/clusterdata/tree/master/cluster-trace-v2017

[6] Alibaba. 2018. Alibab Cluster Trace Version 2. Retrieved from https://github.com/alibaba/clusterdata/tree/master/cluster-trace-v2018

[7] Alibaba. 2020. Alibab GPU Trace. Retrieved 2020 from https://github.com/alibaba/clusterdata/tree/master/cluster-trace-gpu-v2020

[8] Alibaba. 2021. Alibab MicroServices Traces V1. Retrieved from https://github.com/alibaba/clusterdata/tree/master/cluster-trace-microservices-v2021

[9] Alibaba. 2022. Alibab MicroServices Traces V2. Retrieved from https://github.com/alibaba/clusterdata/tree/master/cluster-trace-microservices-v2022

[10] Saeed M. Alqahtani and Robert John. 2017. A comparative analysis of different classification techniques for cloud intrusion detection systems' alerts and fuzzy classifiers. In *Proceedings of the Computing Conference*. 406–415. https://doi.org/10.1109/SAI.2017.8252132

[11] George Amvrosiadis, Michael Kuchnik, Jun Woo Park, Chuck Cranor, Gregory R. Ganger, Elisabeth Moore, and Nathan DeBardeleben. 2018. The Atlas cluster trace repository. *Usenix Mag* 43, 4 (2018).

[12] Beloglazov Anton. 2011. Planetlab VM Traces. Retrieved 2011 from https://github.com/beloglazov/planetlab-workload-traces

[13] Ali Anwar. 2018. IBM Docker Registry Traces. Retrieved 2018 from https://dssl.cs.vt.edu/drtp/

[14] Ali Anwar, Mohamed Mohamed, Vasily Tarasov, Michael Littley, Lukas Rupprecht, Yue Cheng, Nannan Zhao, Dimitrios Skourtis, Amit S. Warke, Heiko Ludwig et al. 2018. Improving docker registry design based on production workload analysis. In *Proceedings of the 16th USENIX Conference on File and Storage Technologies (FAST'18)*. 265–278.

[15] ATLAS. 2011. ATLAS Cluster Traces. Retrieved 2011 from https://ftp.pdl.cmu.edu/pub/datasets/ATLAS/

[16] Azure. 2024. Azure LLM Inference Dataset. Retrieved from https://github.com/Azure/AzurePublicDataset/blob/master/AzureLLMInferenceDataset2024.md

[17] Francisco J. Baldan, Sergio Ramirez-Gallego, Christoph Bergmeir, Francisco Herrera, and Jose M. Benitez. 2018. A forecasting methodology for workload forecasting in cloud systems. *IEEE Trans. Cloud Comput.* 6, 4 (Oct. 2018), 929–941. https://doi.org/10.1109/TCC.2016.2586064

[18] Paras Bhagtya, S. Raghavan, K. Chandraseakran, and Usha D. 2021. Workload classification in Multi-vm cloud environment using deep neural network model. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing (SAC'21)*. Association for Computing Machinery, New York, NY, 79–82. https://doi.org/10.1145/3412841.3442068

[19] H. S. Bhathiya. 2016. Autoscale Analyser. Retrieved from https://github.com/hsbhathiya/AutoscaleAnalyser/tree/master/datasets/cloud_traces

[20] Muhammad Bilal, Marco Serafini, Marco Canini, and Rodrigo Rodrigues. 2020. Do the best cloud configurations grow on trees? An experimental evaluation of black box algorithms for optimizing cloud workloads. *Proc. VLDB Endow.* 13, 12 (July 2020), 2563–2575. https://doi.org/10.14778/3407790.3407845

[21] Marc Carrascosa. 2020. Stadia Cloud Gaming Dataset. Retrieved from https://github.com/wn-upf/Stadia_cloud_gaming_dataset_2020

[22] Italian National Supercomputing Center. 2020. Marconi100. Retrieved from https://zenodo.org/search?q=Marconi100&l=list&p=1&s=10&sort=bestmatch

[23] Kennedy Space Center. 1995. NASA HTTP Traces. Retrieved 1995 from https://www.kaggle.com/datasets/adchatakora/nasa-http-access-logs/download?datasetVersionNumber=1

[24] Milan Cermak, Tomas Jirsik, Petr Velan, Jana Komarkova, Stanislav Spacek, Martin Drasar, and Tomas Plesnik. 2018. Towards provable network traffic measurement and analysis via semi-labeled trace datasets. In *Proceedings of the Network Traffic Measurement and Analysis Conference (TMA'18).* IEEE, 1–8.

[25] Katja Cetinski and Matjaz B. Juric. 2015. AME-WPC: Advanced model for efficient workload prediction in the cloud. *J. Netw. Comput. Appl.* 55 (Sep. 2015), 191–201. https://doi.org/10.1016/j.jnca.2015.06.001

[26] Wenyan Chen, Kejiang Ye, Yang Wang, Guoyao Xu, and Cheng-Zhong Xu. 2018. How does the workload look like in production cloud? Analysis and clustering of workloads on Alibaba cluster trace. In *Proceedings of the IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS'18).* IEEE, Singapore, Singapore, 102–109. https://doi.org/10.1109/PADSW.2018.8644579

[27] Yanpei Chen, Sara Alspaugh, and Randy Katz. 2012. Interactive analytical processing in big data systems: A cross-industry study of MapReduce workloads. *Proc. VLDB Endow.* 5, 12 (Aug. 2012), 1802–1813. https://doi.org/10.14778/2367502.2367519

[28] Yanpei Chen, Archana Ganapathi, Rean Griffith, and Randy Katz. 2011. The case for evaluating mapreduce performance using workload suites. In *Proceedings of the IEEE 19th Annual International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems.* 390–399. https://doi.org/10.1109/MASCOTS.2011.12

[29] Yujun Chen, Xian Yang, Qingwei Lin, Hongyu Zhang, Feng Gao, Zhangwei Xu, Yingnong Dang, Dongmei Zhang, Hang Dong, Yong Xu et al. 2019. Outage prediction and diagnosis for cloud service systems. In *Proceedings of the World Wide Web Conference.* 2659–2665.

[30] Zheyi Chen, Jia Hu, Geyong Min, Albert Y. Zomaya, and Tarek El-Ghazawi. 2019. Towards accurate prediction for high-dimensional and highly-variable cloud workloads with deep learning. *IEEE Trans. Parallel Distrib. Syst.* 31, 4 (2019), 923–934.

[31] Zhuo Chen, Fu Jiang, Yijun Cheng, Xin Gu, Weirong Liu, and Jun Peng. 2018. XGBoost classifier for DDoS attack detection and analysis in SDN-Based Cloud. In *Proceedings of the IEEE International Conference on Big Data and Smart Computing (BigComp'18).* 251–256. https://doi.org/10.1109/BigComp.2018.00044

[32] Yuming Cheng, Chao Wang, Huihuang Yu, Yahui Hu, and Xuehai Zhou. 2019. GRU-ES: Resource usage prediction of cloud workloads using a novel hybrid method. In *Proceedings of the IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS'19).* 1249–1256. https://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00175

[33] Hsu Chin-Jung. 2018. Scout. Retrieved 2018 from https://github.com/oxhead/scout

[34] Georgia Christofidi, Konstantinos Papaioannou, and Thaleia Dimitra Doudali. 2023. Is machine learning necessary for cloud resource usage forecasting? In *Proceedings of the ACM Symposium on Cloud Computing (SoCC'23).* Association for Computing Machinery, New York, NY, 544–554. https://doi.org/10.1145/3620678.3624790

[35] Science Clouds. 2017. Chameleon Cloud Traces. Retrieved 2017 from https://github.com/oxhead/scout

[36] Eli Cortez, Anand Bonde, Alexandre Muzio, Mark Russinovich, Marcus Fontoura, and Ricardo Bianchini. 2017. Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms. In *Proceedings of the 26th Symposium on Operating Systems Principles (SOSP'17).* Association for Computing Machinery, New York, NY, 153–167. https://doi.org/10.1145/3132747.3132772

[37] François De Keersmaeker, Yinan Cao, Gorby Kabasele Ndonda, and Ramin Sadre. 2023. A survey of public IoT datasets for network security research. *IEEE Communications Surveys & Tutorials* 25, 3 (2023), 1808–1840.

[38] Mingzhe Du, Yang Wang, Kejiang Ye, and Chengzhong Xu. 2020. Algorithmics of cost-driven computation offloading in the edge-cloud environment. *IEEE Trans. Comput.* 69, 10 (2020), 1519–1532.

[39] Nosayba El-Sayed, Hongyu Zhu, and Bianca Schroeder. 2017. Learning from failure across multiple clusters: A trace-driven approach to understanding, predicting, and mitigating job terminations. In *Proceedings of the IEEE 37th International Conference on Distributed Computing Systems (ICDCS'17).* 1333–1344. https://doi.org/10.1109/ICDCS.2017.317

[40] Brad Everman, Narmadha Rajendran, Xiaomin Li, and Ziliang Zong. 2021. Improving the cost efficiency of large-scale cloud systems running hybrid workloads—A case study of Alibaba cluster traces. *Sustain. Comput.: Inform. Syst.* 30 (June 2021), 100528. https://doi.org/10.1016/j.suscom.2021.100528

[41] Facebook. 2009. SWIM Workload. Retrieved 2009 from https://github.com/SWIMProjectUCB/SWIM/wiki/Workloads-repository

[42] DG Feitelson. 2005. Parallel Workloads Archive. Retrieved 2005 from https://www.cs.huji.ac.il/labs/parallel/workload/logs.html

[43] Canadian Institute for Cybersecurity. 2012. ISCX IDS 2012. Retrieved 2012 from https://www.unb.ca/cic/datasets/ids.html

[44] Canadian Institute for Cybersecurity. 2014. ISCX-Bot-2014. Retrieved 2014 from https://www.unb.ca/cic/datasets/botnet.html

[45] Canadian Institute for Cybersecurity. 2016. ISCX. Retrieved 2016 from https://www.unb.ca/cic/datasets/index.html

[46] Canadian Institute for Cybersecurity. 2016. ISCX-URL2016. Retrieved 2016 from https://www.unb.ca/cic/datasets/url-2016.html

[47] Canadian Institute for Cybersecurity. 2016. ISCXTor2016. Retrieved 2016 from https://www.unb.ca/cic/datasets/tor.html

[48] Canadian Institute for Cybersecurity. 2016. ISCXVPN2016. Retrieved 2016 from https://www.unb.ca/cic/datasets/vpn.html

[49] Peter Garraghan, Paul Townend, and Jie Xu. 2013. An analysis of the server characteristics and resource utilization in Google cloud. In *Proceedings of the IEEE International Conference on Cloud Engineering (IC2E'13)*. 124–131. https://doi.org/10.1109/IC2E.2013.40

[50] Google. 2009. Google Cluster Data Version 1. Retrieved from https://github.com/google/cluster-data/blob/master/TraceVersion1.md

[51] Google. 2011. Google Cluster Data Version 2. Retrieved from https://github.com/google/cluster-data/blob/master/ClusterData2011_2.md

[52] Google. 2019. Google Cluster Data Version 3. Retrieved from https://github.com/google/cluster-data/blob/master/ClusterData2019.md

[53] Google. 2024. Google Power Data. Retrieved from https://github.com/google/cluster-data/blob/master/PowerData2019.md

[54] hanghai Artificial Intelligence Laboratory. 2023. Acme. Retrieved from https://github.com/InternLM/AcmeTrace

[55] Mahesh Hariharasubramanian. 2018. *Improving Application Infrastructure Provisioning Using Resource Usage Predictions from Cloud Metric Data Analysis*. Ph.D. Dissertation. Rutgers University-School of Graduate Studies.

[56] Tianzhang He and Rajkumar Buyya. 2023. A taxonomy of live migration management in cloud computing. *Comput. Surveys* 56, 3 (2023), 1–33.

[57] Kyo Kang, Sholom Cohen, James Hess, William Novak, and A. Peterson. 1990. Feature-oriented domain analysis (FODA) feasibility study. Carnegie Mellon University, Software Engineering Institute's Digital Library. Retrieved Feb 28, 2025 from https://insights.sei.cmu.edu/library/feature-oriented-domain-analysis-foda-feasibility-study/

[58] 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 *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–15.

[59] Qinghao Hu, Zhisheng Ye, Zerui Wang, Guoteng Wang, Meng Zhang, Qiaoling Chen, Peng Sun, Dahua Lin, Xiaolin Wang, Yingwei Luo et al. 2024. Characterization of large language model development in the datacenter. In *Proceedings of the 21st USENIX Symposium on Networked Systems Design and Implementation (NSDI'24)*. 709–729.

[60] Jesus Omana Iglesias, Philip Perry, Nicola Stokes, James Thorburn, and Liam Murphy. 2013. A cost-capacity analysis for assessing the efficiency of heterogeneous computing assets in an enterprise cloud. In *Proceedings of the IEEE/ACM 6th International Conference on Utility and Cloud Computing*. 107–114. https://doi.org/10.1109/UCC.2013.32

[61] IWikipedia. 2007. IWikipedia Web Traces from WikiBench. Retrieved 2007 from https://wikitech.wikimedia.org/wiki/Analytics/Archive/Data/Pagecounts-raw

[62] K. Jairam Naik. 2020. A dynamic ACO-based elastic load balancer for cloud computing (D-ACOELB). In *Proceedings of the 3rd International Conference on Data Engineering and Communication Technology (ICDECT'19)*. Springer, 11–20.

[63] J. Jayalath, E. Chathumali, K. R. M. Kothalawala, and N. Kuruwitaarachchi. 2019. Green cloud computing: A review on adoption of green-computing attributes and vendor specific implementations. In *Proceedings of the International Research Conference on Smart Computing and Systems Engineering (SCSE'19)*. IEEE, 158–164.

[64] Sima Jeddi and Saeed Sharifian. 2020. A hybrid wavelet decomposer and GMDH-ELM ensemble model for network function virtualization workload forecasting in cloud computing. *Appl. Soft Comput.* 88 (Mar. 2020), 105940. https://doi.org/10.1016/j.asoc.2019.105940

[65] 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 USENIX Annual Technical Conference (USENIX ATC'19)*. 947–960.

[66] Zhuangwei Kang, Zhuo Zhen, and Kate Keahey. 2021. *Chameleon and HTC: A Match Made in Heaven.*

[67] Kuljeet Kaur, Sahil Garg, Georges Kaddoum, and Neeraj Kumar. 2021. Energy and SLA-driven MapReduce job scheduling framework for cloud-based cyber-physical systems. *ACM Trans. Internet Technol.* 21, 2 (June 2021), 1–24. https://doi.org/10.1145/3409772

[68] Ghazal Khodabandeh, Alireza Ezaz, and Naser Ezzati-Jivan. 2024. Network analysis of microservices: A case study on Alibaba production clusters. In *Proceedings of the of the 15th ACM/SPEC International Conference on Performance Engineering*. 67–71.

[69] Hisham A. Kholidy and Fabrizio Baiardi. 2012. CIDD: A cloud intrusion detection dataset for cloud computing and masquerade attacks. In *Proceedings of the 9th International Conference on Information Technology—New Generations*. 397–402. https://doi.org/10.1109/ITNG.2012.97

[70] Dionatrã F. Kirchoff, Miguel Xavier, Juliana Mastella, and César A. F. De Rose. 2019. A preliminary study of machine learning workload prediction techniques for cloud applications. In *Proceedings of the 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP'19)*. 222–227. https://doi.org/10.1109/EMPDP.2019.8671604

[71] Dalibor Klusacek. 2016. CERIT-SC Workloads. Retrieved 2016 from https://jsspp.org/workload/index.php

[72] Dalibor Klusáček and Boris Parák. 2018. Analysis of mixed workloads from shared cloud infrastructure. In *Job Scheduling Strategies for Parallel Processing (Lecture Notes in Computer Science)*, Dalibor Klusáček, Walfredo Cirne, and Narayan Desai (Eds.). Springer International Publishing, Cham, 25–42. https://doi.org/10.1007/978-3-319-77398-8_2

[73] Siddhant Kumar, Neha Muthiyan, Shaifu Gupta, A. D. Dileep, and Aditya Nigam. 2018. Association learning based hybrid model for cloud workload prediction. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'18)*. 1–8. https://doi.org/10.1109/IJCNN.2018.8488996

[74] MIT Lincoln Labs. 1998. KDD Cup 1999's Dataset. Retrieved 1998 from https://kdd.ics.uci.edu/databases/kddcup99/kddcup99

[75] Chuan Lei, Zhongfang Zhuang, Elke A. Rundensteiner, and Mohamed Eltabakh. 2015. Shared execution of recurring workloads in MapReduce. *Proc. VLDB Endow.* 8, 7 (Feb. 2015), 714–725. https://doi.org/10.14778/2752939.2752941

[76] Habte Lejebo Leka, Zhang Fengli, Ayantu Tesfaye Kenea, Abebe Tamrat Tegene, Peter Atandoh, and Negalign Wake Hundera. 2021. A hybrid CNN-LSTM model for virtual machine workload forecasting in cloud data center. In *Proceedings of the 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP'21)*. 474–478. https://doi.org/10.1109/ICCWAMTIP53232.2021.9674067

[77] Chunlin Li, Hezhi Sun, Yi Chen, and Youlong Luo. 2019. Edge cloud resource expansion and shrinkage based on workload for minimizing the cost. *Future Gen. Comput. Syst.* 101 (Dec. 2019), 327–340. https://doi.org/10.1016/j.future.2019.05.026

[78] Jianpeng Lin, Wenjun Lin, Weiwei Lin, Tianyi Liu, Jiangtao Wang, and Hongliang Jiang. 2024. Multi-objective cooling control optimization for air-liquid cooled data centers using TCN-BiGRU-Attention-based thermal prediction models. In *Building Simulation*. Springer, 1–17.

[79] Weiwei Lin, Wentai Wu, Haoyu Wang, James Z. Wang, and Ching-Hsien Hsu. 2018. Experimental and quantitative analysis of server power model for cloud data centers. *Future Gen. Comput. Syst.* 86 (Sep. 2018), 940–950. https://doi.org/10.1016/j.future.2016.11.034

[80] Xiao Ling, Yi Yuan, Dan Wang, and Jiahai Yang. 2016. Tetris: Optimizing cloud resource usage unbalance with elastic VM. In *Proceedings of the IEEE/ACM 24th International Symposium on Quality of Service (IWQoS'16)*. 1–10. https://doi.org/10.1109/IWQoS.2016.7590395

[81] Qixiao Liu and Zhibin Yu. 2018. The elasticity and plasticity in semi-containerized co-locating cloud workload: A view from Alibaba trace. In *Proceedings of the ACM Symposium on Cloud Computing (SoCC'18)*. Association for Computing Machinery, New York, NY, 347–360. https://doi.org/10.1145/3267809.3267830

[82] Toni Mastelic, Ariel Oleksiak, Holger Claussen, Ivona Brandic, Jean-Marc Pierson, and Athanasios V. Vasilakos. 2014. Cloud computing: Survey on energy efficiency. *ACM Comput. Surveys* 47, 2 (2014), 1–36.

[83] Wiem Matoussi and Tarek Hamrouni. 2022. A new temporal locality-based workload prediction approach for SaaS services in a cloud environment. *J. King Saud Univ. Comput. Info. Sci.* 34, 7 (July 2022), 3973–3987. https://doi.org/10.1016/j.jksuci.2021.04.008

[84] Rizwan Mian, Patrick Martin, Farhana Zulkernine, and Jose Luis Vazquez-Poletti. 2013. Towards building performance models for data-intensive workloads in public clouds. In *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering (ICPE'13)*. Association for Computing Machinery, New York, NY, 259–270. https://doi.org/10.1145/2479871.2479908

[85] Microsoft. 2017. Azure Public Dataset. Retrieved 2017 from https://github.com/Azure/AzurePublicDataset/tree/master

[86] Martin Molan, Andrea Borghesi, Luca Benini, and Andrea Bartolini. 2022. Semi-supervised anomaly detection on a Tier-0 HPC system. In *Proceedings of the 19th ACM International Conference on Computing Frontiers (CF'22)*. Association for Computing Machinery, New York, NY, 203–204. https://doi.org/10.1145/3528416.3530867

[87] Piotr Nawrocki. 2020. Fruktus. Retrieved 2020 from https://github.com/Fruktus/CloudPredictionFramework

[88] Piotr Nawrocki and Wiktor Sus. 2022. Anomaly detection in the context of long-term cloud resource usage planning. *Knowl. Info. Syst.* 64, 10 (Oct. 2022), 2689–2711. https://doi.org/10.1007/s10115-022-01721-5

[89] DARPA Intrusion Detection Evaluation Group of MIT Lincoln Laboratory. 2012. Cloud Intrusion Detection Dataset. Retrieved 2012 from http://groups.di.unipi.it/~hkholidy/projects/cidd./

[90] Delft University of Technology. 2003. GWA-T-3 NorduGrid. Retrieved 2003 from http://gwa.ewi.tudelft.nl/datasets/gwa-t-3-nordugrid

[91] Delft University of Technology. 2005. GWA-T-1 DAS2. Retrieved 2005 from http://gwa.ewi.tudelft.nl/datasets/gwa-t-1-das2

[92] Delft University of Technology. 2005. GWA-T-10 SHARCNet. Retrieved 2005 from http://gwa.ewi.tudelft.nl/datasets/gwa-t-10-sharcnet

[93] Delft University of Technology. 2006. GWA-T-4 AuverGrid. Retrieved 2006 from http://gwa.ewi.tudelft.nl/datasets/gwa-t-4-auvergrid

[94] Delft University of Technology. 2015. Business Critical Workloads. Retrieved 2015 from http://gwa.ewi.tudelft.nl/datasets/Bitbrains

[95] OpenCloud. 2010. OpenCloud Hadoop Workload. Retrieved 2010 from https://ftp.pdl.cmu.edu/pub/datasets/hla/dataset.html

[96] Daon Park, Hyunsoo Kim, Youngsu Cho, Changyeon Jo, and Bernhard Egger. 2021. Can VM live migration improve job throughput? Evidence from a real world cluster trace. In *Economics of Grids, Clouds, Systems, and Services (Lecture Notes in Computer Science)*, Konstantinos Tserpes, Jörn Altmann, José Ángel Bañares, Orna Agmon Ben-Yehuda, Karim Djemame, Vlado Stankovski, and Bruno Tuffin (Eds.). Springer International Publishing, Cham, 17–26. https://doi.org/10.1007/978-3-030-92916-9_2

[97] Eva Patel and Dharmender Singh Kushwaha. 2020. Clustering cloud workloads: K-Means vs gaussian mixture model. *Procedia Comput. Sci.* 171 (Jan. 2020), 158–167. https://doi.org/10.1016/j.procs.2020.04.017

[98] Pratyush Patel, Esha Choukse, Chaojie Zhang, Aashaka Shah, Íñigo Goiri, Saeed Maleki, and Ricardo Bianchini. 2024. Splitwise: Efficient generative llm inference using phase splitting. In *Proceedings of the ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA'24)*. IEEE, 118–132.

[99] Anelis Pereira-Vale, Gastón Márquez, Hernán Astudillo, and Eduardo B. Fernandez. 2019. Security mechanisms used in microservices-based systems: A systematic mapping. In *Proceedings of the XLV Latin American Computing Conference (CLEI'19)*. IEEE, 01–10.

[100] Prasad Purnaye. 2021. OpenNebula Virtual Machine Profiling Dataset. Retrieved 2021 from https://ieee-dataport.org/documents/opennebula-virtual-machine-profiling-intrusion-detection-system#files

[101] Feng Qiu, Bin Zhang, and Jun Guo. 2016. A deep learning approach for VM workload prediction in the cloud. In *Proceedings of the 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD'16)*. 319–324. https://doi.org/10.1109/SNPD.2016.7515919

[102] R. R. Sathiya, Adiraju S. S. Aditya, Arikatla Harsha Vardhan Chowdary, and Inukonda Sudheepthi. 2023. Analysis of clustering effects in cloud workload forecasting. In *Proceedings of the 14th International Conference on Computing Communication and Networking Technologies (ICCCNT'23)*. 1–8. https://doi.org/10.1109/ICCCNT56998.2023.10307905

[103] Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training. Retrieved from https://api.semanticscholar.org/CorpusID:49313245

[104] Microsoft Research. 2017. Philly. Retrieved from https://github.com/msr-fiddle/philly-traces

[105] Andrea Rossi, Andrea Visentin, Steven Prestwich, and Kenneth N. Brown. 2022. Bayesian uncertainty modelling for cloud workload prediction. In *Proceedings of the IEEE 15th International Conference on Cloud Computing (CLOUD'22)*. 19–29. https://doi.org/10.1109/CLOUD55607.2022.00018

[106] Varun Sakalkar, Vasileios Kontorinis, David Landhuis, Shaohong Li, Darren De Ronde, Thomas Blooming, Anand Ramesh, James Kennedy, Christopher Malone, Jimmy Clidaras et al. 2020. Data center power oversubscription with a medium voltage power plane and priority-aware capping. In *Proceedings of the 25th International Conference on Architectural Support for Programming Languages and Operating Systems*. 497–511.

[107] Deepika Saxena, Ishu Gupta, Rishabh Gupta, Ashutosh Kumar Singh, and Xiaoqing Wen. 2023. An AI-Driven VM threat prediction model for multi-risks analysis-based cloud cybersecurity. *IEEE Trans. Syst. Man Cybernet.: Syst.* 53, 11 (Nov. 2023), 6815–6827. https://doi.org/10.1109/TSMC.2023.3288081

[108] SenseTime. 2020. Helios. Retrieved from https://github.com/S-Lab-System-Group/HeliosData

[109] Ohad Shai. 2012. Intel Netbatch Logs. Retrieved 2012 from https://www.cs.huji.ac.il/labs/parallel/workload/l_intel_netbatch/index.html

[110] R. S. Shariffdeen, D. T. S. P. Munasinghe, H. S. Bhathiya, U. K. J. U. Bandara, and H. M. N. Dilum Bandara. 2016. Workload and resource aware proactive auto-scaler for PaaS cloud. In *Proceedings of the IEEE 9th International Conference on Cloud Computing (CLOUD'16)*. 11–18. https://doi.org/10.1109/CLOUD.2016.0012

[111] Siqi Shen, Vincent Van Beek, and Alexandru Iosup. 2015. Statistical characterization of business-critical workloads hosted in cloud datacenters. In *Proceedings of the 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. 465–474. https://doi.org/10.1109/CCGrid.2015.60

[112] SPEC. 2008. SPEC power_ssj2008. Retrieved 2008 from https://www.spec.org/power_ssj2008/results/

[113] SPEC. 2017. SPEC CPU. Retrieved 2017 from https://www.spec.org/cpu2017/

[114] SPEC. 2018. SPEC Cloud IaaS. Retrieved 2018 from https://www.spec.org/cloud_iaas2018/

[115] Jovan Stojkovic, Chaojie Zhang, Íñigo Goiri, Josep Torrellas, and Esha Choukse. 2024. Dynamollm: Designing llm inference clusters for performance and energy efficiency. Retrieved from https://arXiv:2408.00741

[116] SURFsara. 2019. SURFsara. Retrieved 2019 from https://github.com/sara-nl/SURFace

[117] Venkat Tadakamalla and Daniel A. Menascé. 2022. Autonomic elasticity control for multi-server queues under generic workload surges in cloud environments. *IEEE Trans. Cloud Comput.* 10, 2 (Apr. 2022), 984–995. https://doi.org/10.1109/TCC.2020.2992949

[118] Queen's University. 2012. Amazon Resource Cost. Retrieved from https://research.cs.queensu.ca/home/mian/index_files/Page485.htm

[119] Guido Urdaneta, Guillaume Pierre, and Maarten van Steen. 2009. Wikipedia workload analysis for decentralized hosting. *Comput. Netw.* 53, 11 (July 2009), 1830–1845. https://doi.org/10.1016/j.comnet.2009.02.019

[120] Ruben Van den Bossche, Kurt Vanmechelen, and Jan Broeckhove. 2013. Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds. *Future Gen. Comput. Syst.* 29, 4 (June 2013), 973–985. https://doi.org/10.1016/j.future.2012.12.012

[121] Laurens Versluis, Mehmet Cetin, Caspar Greeven, Kristian Laursen, Damian Podareanu, Valeriu Codreanu, Alexandru Uta, and Alexandru Iosup. 2021. A Holistic Analysis of Datacenter Operations: Resource Usage, Energy, and Workload Characterization—Extended Technical Report. Retrieved from https://arXiv:cs/2107.11832

[122] Zhijing Wan, Zhixiang Wang, CheukTing Chung, and Zheng Wang. 2022. A survey of dataset refinement for problems in computer vision datasets. Retrieved from https://arXiv:2210.11717

[123] Zhijing Wan, Zhixiang Wang, Cheukting Chung, and Zheng Wang. 2024. A survey of dataset refinement for problems in computer vision datasets. *ACM Comput. Surv.* 56, 7 (April 2024). DOI: https://doi.org/10.1145/3627157

[124] 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 *Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI'22)*. 945–960.

[125] Qizhen Weng, Lingyun Yang, Yinghao Yu, Wei Wang, Xiaochuan Tang, Guodong Yang, and Liping Zhang. 2023. Beware of fragmentation: Scheduling GPU-Sharing workloads with fragmentation gradient descent. In *Proceedings of the USENIX Annual Technical Conference (USENIX ATC'23)*. 995–1008.

[126] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. 1–10.

[127] Tianyang Wu, Maolin Pan, and Yang Yu. 2022. A long-term cloud workload prediction framework for reserved resource allocation. In *Proceedings of the IEEE International Conference on Services Computing (SCC'22)*. 134–139. https://doi.org/10.1109/SCC55611.2022.00030

[128] Minxian Xu, Chenghao Song, Huaming Wu, Sukhpal Singh Gill, Kejiang Ye, and Chengzhong Xu. 2022. esDNN: Deep neural network based multivariate workload prediction in cloud computing environments. *ACM Trans. Internet Technol.* 22, 3 (Aug. 2022), 75:1–75:24. https://doi.org/10.1145/3524114

[129] Rahul Yadav, Weizhe Zhang, Keqin Li, Chuanyi Liu, Muhammad Shafiq, and Nabin Kumar Karn. 2020. An adaptive heuristic for managing energy consumption and overloaded hosts in a cloud data center. *Wireless Netw.* 26, 3 (Apr. 2020), 1905–1919. https://doi.org/10.1007/s11276-018-1874-1

[130] Yahoo. 2010. Yahoo Webscope Dataset. Retrieved from https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&guccounter=1

[131] Mohammad Yekta and Hadi Shahriar Shahhoseini. 2023. A review on machine learning methods for workload prediction in cloud computing. In *Proceedings of the 13th International Conference on Computer and Knowledge Engineering (ICCKE'23)*. IEEE, 306–311.

[132] Brian Zhang, Valencia Zhang, and Michael Hum. 2022. Budget in the cloud: Analyzing cost and recommending virtual machine workload. In *Proceedings of the International Communication Engineering and Cloud Computing Conference (CECCC'22)*. 12–17. https://doi.org/10.1109/CECCC56460.2022.10069750

[133] Qi Zhang and Raouf Boutaba. 2014. Dynamic workload management in heterogeneous cloud computing environments. In *Proceedings of the IEEE Network Operations and Management Symposium (NOMS'14)*. 1–7. https://doi.org/10.1109/NOMS.2014.6838288

[134] Qingchen Zhang, Laurence T. Yang, Zheng Yan, Zhikui Chen, and Peng Li. 2018. An efficient deep learning model to predict cloud workload for industry informatics. *IEEE Trans. Industr. Inform.* 14, 7 (July 2018), 3170–3178. https://doi.org/10.1109/TII.2018.2808910

[135] Nannan Zhao, Hadeel Albahar, Subil Abraham, Keren Chen, Vasily Tarasov, Dimitrios Skourtis, Lukas Rupprecht, Ali Anwar, and Ali R. Butt. 2020. DupHunter: Flexible high-performance deduplication for docker registries. In *2020*

USENIX Annual Technical Conference (USENIX ATC 20), USENIX Association, 769–783. Retrieved from https://www.usenix.org/conference/atc20/presentation/zhao

[136] Jianyong Zhu, Bin Lu, Xiaoqiang Yu, Jie Xu, and Tianyu Wo. 2023. An approach to workload generation for cloud benchmarking: A view from Alibaba trace. In *Proceedings of the IEEE 15th International Symposium on Autonomous Decentralized System (ISADS'23)*. 1–8. https://doi.org/10.1109/ISADS56919.2023.10092039

[137] Mohd Zuhair, Pronaya Bhattacharya, Vivek Kumar Prasad, Manav Barot, and Monil Modi. 2022. Analysis of boosting mechanisms in cloud-based intrusion detection systems. In *Proceedings of the 5th International Conference on Contemporary Computing and Informatics (IC3I'22)*. 961–966. https://doi.org/10.1109/IC3I56241.2022.10072683

Received 17 March 2024; revised 9 February 2025; accepted 14 February 2025