Vrije Universiteit Amsterdam


Honours Programme, Project Report

---

# Job Failure Characterization and Prediction in HPC Datacenter Using Deep Learning

---

**Author: Yizhen Zang (2721673)**

| | |
|---|---|
| *1st supervisor:* | prof. dr. ir. Alexandru Iosup |
| *daily supervisor, if different:* | ir. Xiaoyu Chu |
| *2nd reader:* | supervisor name |

*A report submitted in fulfillment of the requirements for the Honours Programme, which is an excellence annotation to the VU Bachelor of Science degree in Computer Science/Artificial Intelligence/Information Sciences*

April 7, 2024

# Contents

## List of Figures



## List of Tables

# Abstract

Datacenters serve as essential infrastructures of the digital society. Numerous services, including search engines, social media networks, e-commerce websites, online gaming, and trading platforms, rely on extensive datacenter infrastructure. As a result, there is a growing demand for the reliability and availability of the datacenter operations. Meanwhile, past decades have seen an emerging trend in machine learning (ML) based services. This trend encompasses not only traditional ML applications such as image recognition, recommendation systems, and machine translation but also ad-hoc generative AI services such as ChatGPT and Midjourney. Since ML shows powerful abilities in various domains, more and more researchers and users are turning to ML-based models or services to conduct their research and tasks. However, although fault-tolerant systems have been implemented, job failures, especially ML job failures, still occur. This is due to the distinctive attributes and complexity of ML jobs compared to general jobs. These failures can result in a serious loss of data and revenues, and decrease the user experience of services. Therefore, understanding and predicting ML job failures are significant for improving the quality and reliability of data center service.

This study is dedicated to the characterization and prediction of job failures, focusing particularly on ML tasks. We collected and cleaned a long-term job dataset in an HPC cluster in the Netherlands. Through statistical analysis, we characterized different aspects of ML job failures, including arrival patterns, temporal correlations, and failure peaks. Our analysis yields 18 pertinent observations, highlighting key insights into ML job failures. Our findings reveal that ML jobs have a significantly higher failure rate (24.75%) compared to generic jobs (16.57%). Additionally, the mean time between failures (MTBF) of ML jobs (412.70 s) is notably longer than that of generic jobs (69.17 s), indicating differences in the reliability and stability of these job types.

We implemented a hybrid LSTM-TCN model to predict the occurrence of failed jobs across varying time granularity, leveraging insights from failed ML jobs. Our predictive model, the LSTM-TCN hybrid, outperforms both individual LSTM and TCN models across different time granularities. This superior performance highlights its effectiveness in capturing temporal patterns associated with ML job failures. However, it is important to note that this enhancement is accompanied by longer inference times and increased memory usage, emphasizing the trade-off between predictive accuracy and computational resources.

Our study investigates ML job failures in an HPC cluster, providing unique insights for enhancing datacenter reliability in the era of ML-based services.



**Keywords:** job failure, datacenter, failure analysis, failure prediction, machine learning, time series analysis, system modeling

# 1 Introduction

In this section, we present the context, research questions, and contributions of our study, focusing on machine learning (ML) job failures within HPC datacenters. We address the challenges posed by these failures and outline our strategy for understanding and predicting them.

## 1.1 Context

Datacenters stand as essential infrastructures in our digital society [3]. As the demand for digital applications grows, the functionality and complexity of datacenters also increase. Although mechanisms like hardware redundancy have been built into datacenters to achieve high service reliability and availability, task and job failures persist [4, 5], leading to data loss, resource wastage, and financial implications [6]. The evolving landscape of digital services demands continuous operation and seamless user experiences, amplifying the impact of failures. In particular, the distinct power consumption behaviors observed in ML nodes within datacenters highlight the significance of addressing ML job failures [7]. Thus, understanding and mitigating job failures within datacenters emerge as critical challenges in maintaining the smooth operation of digital services and minimizing disruptions to users and businesses.

Earlier studies have examined the analysis and prediction of job failures across different types of datacenter, including both clouds and HPC environments. These investigations have proposed disparate methods for predictive modeling, ranging from statistical learning techniques [8], to sophisticated neural network architectures [9]. Approaches like Online Sequential Extreme Learning Machine (OS-ELM) [10], and multi-layer Bidirectional Long Short Term Memory (Bi-LSTM) [11] have been explored. Research has also delved into the analysis of cloud failures at different levels, such as low-level failure traces in distributed systems [12] and statistical analysis of multi-source data archives [13]. Despite these endeavors, there remains a gap in understanding ML job failures, which exhibit distinct characteristics in terms of runtimes and energy consumption compared to generic workloads [7].

To bridge this gap, our study focuses on ML job failures within the SURFLisa scientific cluster in the Netherlands (details in Section 2). Our primary research question revolves around understanding the characteristics of ML job failures and designing a predictive model for them. Leveraging a long-term job dataset collected from the SURFLisa cluster, we conducted a comprehensive statistical analysis to delineate various aspects of ML job failures, including arrival patterns, temporal correlations, and failure peaks. Our investigation unveils significant insights into the distinct characteristics of ML job failures, highlighting their higher failure rates compared to generic jobs.

In this study, we examined three categories of jobs: ML, generic, and course-related. Initially, we divided all jobs into ML and generic jobs. However, upon closer examination, we identified a subset of machine learning jobs conducted exclusively by students, which might exhibit distinct failure patterns. Therefore, we introduced a third category for these course-related jobs. By including this third category, we are able to capture the unique challenges and outcomes associated with student-led ML projects. Figure 1 shows the distribution of generic, ML, and course jobs by their exit states (details in 4.1).

Furthermore, we developed and evaluated three deep learning models – LSTM, TCN, and a hybrid LSTM-TCN model – to forecast the occurrence of job failures across varying temporal resolutions. Our predictive modeling shows the effectiveness of the LSTM-TCN hybrid model in capturing time-related patterns in ML job failures.

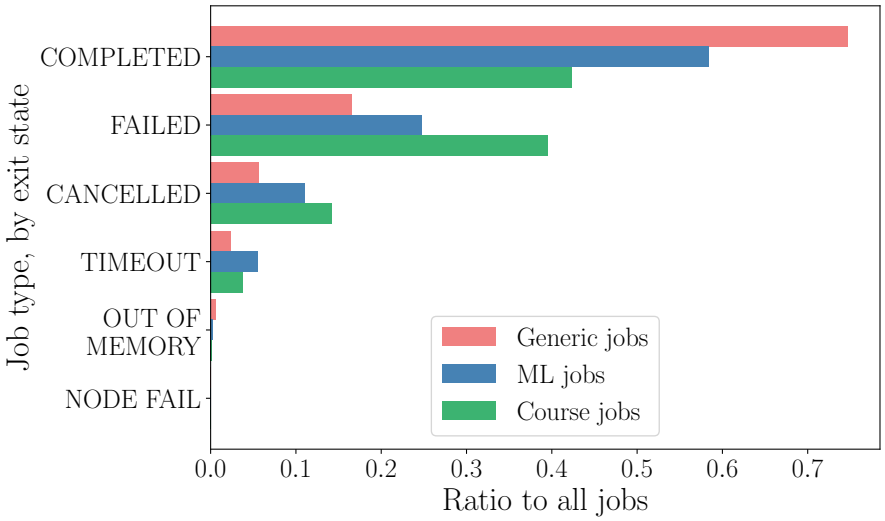



**Figure 1: Distribution of generic, ML, and course jobs by their termination states.**

## 1.2 Research Questions

Our main research question is: **How can we comprehensively understand the unique characteristics of machine learning job failures and develop an effective deep learning model to accurately predict them?** It could be divided into the following three research questions (RQ):

**RQ1.** In light of the impractical assumptions underlying many existing systems, how can we obtain a clean dataset of job-level metrics suitable for analysis and modeling?

**RQ2.** What are the unique characteristics of ML job failures, and how do they compare to generic and course job failures?

**RQ3.** What methodologies can be employed to design a deep learning model to predict ML job failures with high accuracy?

With these research questions answered, we aim to offer insights into job failures, especially into ML-specific job failures in large-scale datacenters, ultimately leading to the reduction of wasted time and resources.

## 1.3 Research Contributions

Our contributions are threefold:

(1) We collected a long-term job dataset from a large-scale HPC cluster, and we developed a toolbox to clean and analyze them. All analysis code is available at `https://github.com/yyzangg/honours-programme-project`.

(2) We conducted thorough analyses, comparing different job types, especially focusing on ML jobs. Our findings uncover unique patterns and correlations in ML job failures, providing valuable insights for optimizing datacenter operations.

(3) We evaluated two classical deep learning models, LSTM and TCN, alongside our own LSTM-TCN hybrid model, to predict ML job failures across different time granularities. The hybrid model showed superior performance in terms of predictive accuracy. However, it also exhibited longer inference times and higher memory usage, underscoring the trade-off between performance and computational resources.

These contributions enhance our understanding of job failures in HPC datacenters and provide researchers and practitioners with practical tools and methodologies.

### 1.4 Research Structure

The rest of this work is organized as follows. In Section 2, the system model used in this work and the structure of the SURFLisa cluster are presented. Section 3 discusses the approaches used for data processing, analysis, and prediction of job failures. In Section 4, we illustrate the results of job failures analysis, while Section 5 focuses on the formulation and evaluation of the prediction models. Section 6 introduces previous studies on job failure analysis and prediction. Finally, conclusions are summarized in Section 7.

# 2 Background

In this section, we present first the system model used in this research, and then the information about the SURFLisa cluster. Relevant concepts are also discussed to facilitate understanding.

## 2.1 System Model

Figure 2 illustrates the system model of SURF. Different types of jobs are submitted to the scheduler by users and then scheduled to different racks and nodes. There are 6 types of termination states for jobs and tasks in our dataset, i.e., completed, canceled, failed, timeout, out of memory and node fail. While jobs running, the scheduler logs that contain job-related metrics are recorded [14], from which we collected the job data for this work. Some of the jobs will be partitioned into several tasks. Jobs fail if one of their tasks fails and by default, jobs will be resubmitted automatically only if there is a system failure. In this work, we classify completed jobs as successful or completed jobs and the other five types as failed jobs.

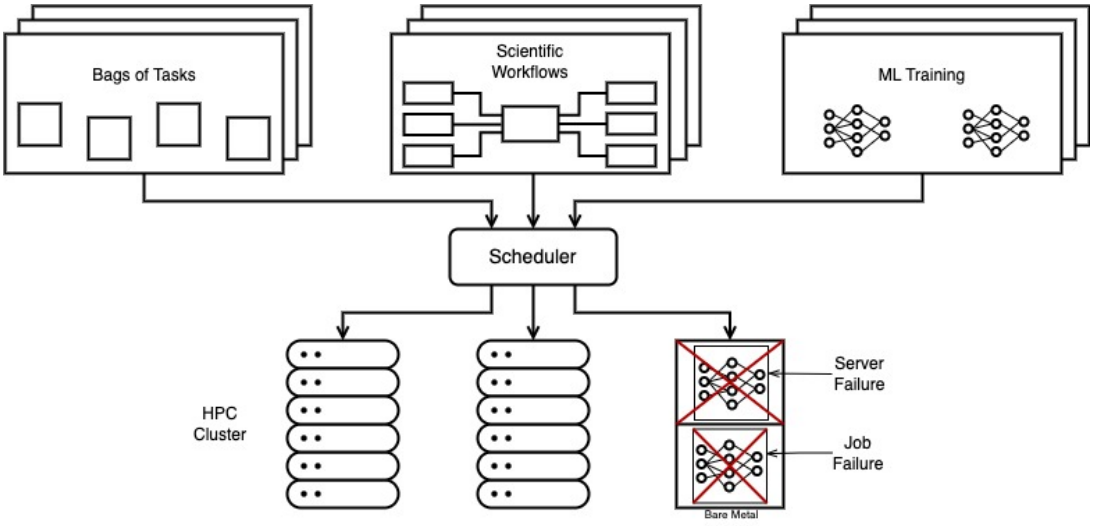



**Figure 2: System model of fail-stop HPC failures [1].**

## 2.2 Lisa Specification

Lisa is one of the largest clusters in the scientific datacenter hosted by SURF, a cooperative association of Dutch educational and research institutions [15]. The Lisa cluster consists of 349 nodes across 21 racks, with 55 GPU nodes designated for ML jobs and others are CPU-only nodes. The CPU-only nodes come mainly with 1 CPU, 16 cores, and 91 GB of system memory. While most GPU nodes have 4 GPUs, plus 2 CPUs and 24 cores. More than 90% of the jobs on the GPU nodes originate from the domain of machine learning, as verified by the datacenter administrators through analysis of the utilized libraries.

# 3 Methods for Characterizing and Predicting Job Failures

In this section, we outline the approaches for data processing, failure characterization, and failure prediction. The overall process can be seen in fig. 3. From the Slurm scheduler, we collected raw data and cleaned it to ensure its quality and usability. Following preprocessing, the cleaned data was subjected to thorough analysis to extract meaningful insights into job execution patterns and failure occurrences. The insights gained from the analysis stage informed the design and development of predictive models to forecast job failures.

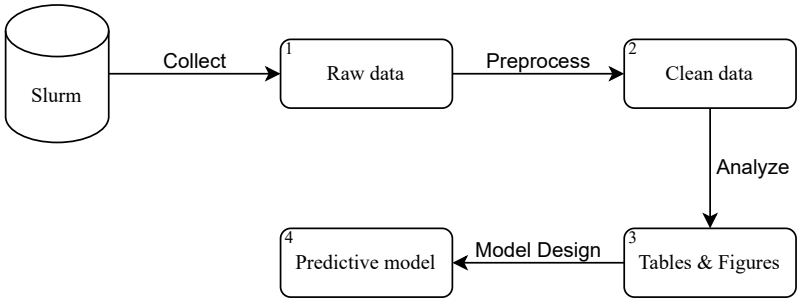



**Figure 3: Research process.**

## 3.1 Data Collection and Sanitization

We gathered job data from the Slurm scheduler logs of Lisa from March 1, 2022, to February 28, 2023. Each row in the dataset represents the execution of a single job or task and contains metrics such as submit time, end time, number of allocated CPUs, etc. The dataset was sanitized in the following procedures. First, traces with missing values in key metrics columns such as "Start" and "End" were filtered out. Then we unified the units of features about resource usage. We also split the node list for each job trace and separate generic jobs, ML jobs, and course-related jobs based on the type of node. Table 1 gives an overview of the dataset.

| | | Start Date | End Date | Metrics | Measurements |
|---|---|---|---|---|---|
| Raw Data | All Jobs | 2022-01-01 | 2023-02-28 | 21 | 2,161,579 |
| Cleaned Data | Generic Jobs | 2022-01-01 | 2023-02-28 | 32 | 1,938,630 |
| | ML Jobs | | | | 222,898 |
| | Course Jobs | | | | 9,755 |

**Table 1: Overview of the job dataset.**

## 3.2 ML Job Failure Characterization

We performed multiple types of analysis to characterize ML job failures in the SURFLisa cluster. The results of the analysis are discussed in Section 4.

First, we conducted statistical analysis on various metrics to explore the properties of ML job failures, including job arrival patterns and CPU usage. Furthermore, we used several

analytical approaches to inspect the time correlation between failures. We employed the autocorrelation function (ACF) to evaluate the level of correlation of the time series data with itself over different time lags.

Similar to Yigitbasi et al.'s work [2], we analyzed four main characteristics of failures during peak periods with the threshold value defined as $\mu + k\sigma$, where $\mu$ is the average, where $\sigma$ is the standard deviation, and $k$ is a positive coefficient. The four metrics inspected are peak duration, inter-peak time, inter-arrival time (IAT) during peaks, and duration of failures during peaks.

### 3.3 ML Job Failure Prediction

We designed a predictive model to forecast the number of failed jobs at specified time intervals, such as minutes, hours, and days. A hybrid architecture combining Long Short-Term Memory (LSTM) and Temporal Convolutional Network (TCN) is employed for this task.

The model works as follows. For instance, when considering an hourly prediction horizon, the model takes as input the data from the past 30 hours and outputs the anticipated numbers of failed jobs for the following 7 hours. This approach enables the model to capture both short-term dependencies, facilitated by the LSTM component, and long-term patterns, enhanced by the TCN architecture.

To ensure the robustness and effectiveness of our predictive model, we carefully selected hyperparameters tailored to the characteristics of our dataset and the requirements of the prediction task (details in Section 5.1.4). In Section 5.2, we present detailed analyses of model results, including comparative evaluations between the LSTM, TCN, and the hybrid LSTM-TCN model, with insights into computational aspects such as inference time and memory usage.

# 4 Analysis of Machine Learning Job Failures

In this section, we provide a comprehensive analysis of ML job failures within the SURFLisa HPC cluster, examining various aspects including failure statistics, arrival patterns, CPU usage, time correlations, and peak analysis.

## 4.1 Failure Statistics

**O-1.** ML jobs have a relatively higher failure rate (24.75%) than generic jobs (16.57%). Among ML jobs, course-related jobs exhibit a high failure rate (39.50%) that doubles that of generic jobs.

**O-2.** The mean time between failures (MTBF) of ML jobs (412.70 s) is longer than generic jobs (69.17 s).

**O-3.** The mean user failure rate of ML jobs (48.10%) is higher than that of generic jobs (39.96%). Course-related jobs demonstrate a mean user failure rate as high as 55.75%.

To understand the characteristics of ML and generic job failures, we first compared the distribution of jobs' completion statuses. Table 2 depicts the fractions of each type of end state. The observation is that ML jobs have a relatively higher failure rate (24.75%) than generic jobs (16.57%). Among ML jobs, course-related jobs feature a high failure rate (39.50%) that doubles that of generic jobs (**O-1**). Despite the low occurrences of terminations due to "OUT OF MEMORY", "REQUEUED", and "NODE FAILURE", failures remain a prevalent issue, underscoring the significance of addressing failure rates, especially within ML workloads.

| Exit State Type | Generic Jobs | ML Jobs | Course Jobs |
|---|---|---|---|
| COMPLETED | 74.74 % | 58.39 % | 42.32 % |
| FAILED | 16.57 % | 24.75 % | 39.50 % |
| CANCELLED | 5.68 % | 11.04 % | 14.16 % |
| TIMEOUT | 2.42 % | 5.52 % | 3.81 % |
| OUT OF MEMORY | $< 1$ % | $< 1$ % | $< 1$ % |
| REQUEUED | $< 1$ % | $< 1$ % | $< 1$ % |
| NODE FAILURE | $< 1$ % | $< 1$ % | $< 1$ % |

**Table 2: Distribution of jobs by exit state.**

MTBF is an essential metric for assessing the reliability of a system. A higher MTBF value indicates greater system stability and resilience [13]. The MTBF of ML jobs (737.54 s) is seven times the number of generic jobs (103.46 s) (**O-2**). Notably, when considering course-related ML jobs separately, the MTBF further extends to 905.15 seconds. This suggests that GPU nodes supporting ML tasks exhibit greater dependability and availability compared to those handling generic workloads.

Understanding the failure rate by user can guide resource allocation and planning strategies. We inspect this metric for all types of jobs. As depicted in Table 3, ML jobs demonstrate a higher mean user failure rate (48.10%) compared to generic jobs (39.96%).

Moreover, course-related ML jobs exhibit an even higher mean user failure rate of 55.75% (**O-3**). This suggests that users with ML jobs may require additional support or resources to improve their job success rate. By identifying users with consistently low failure rates, system administrators can allocate resources more efficiently and prioritize users based on their historical performance.

| Job Type | Generic Jobs | ML Jobs | Course Jobs |
|---|---|---|---|
| MTBF [s] | 69.17 | 412.70 | 905.15 |
| Avg. User Failure Rate [%] | 39.96 | 48.10 | 55.75 |

**Table 3: Average time between failures and user failure rates of jobs.**

## 4.2 Arrival Patterns

**O-4.** Arrival is highly variable for all types of jobs. The number of submitted jobs per day varies by up to five orders of magnitude.

**O-5.** Fewer ML jobs arrive on average.

**O-6.** Job submissions of generic and ML jobs, including course-related jobs have a diurnal pattern (9 to 18), and so do job failures of ML jobs.

The analysis of job submissions by day in Figure 4 reveals a striking variability in arrival patterns across all job types (**O-4**), and significantly fewer ML jobs are submitted (**O-5**). The substantial range, spanning five orders of magnitude, indicates the dynamic nature of job submissions within the datacenter environment. Such variability can be attributed to diverse factors, including user workload, system capacity, and scheduling policies.

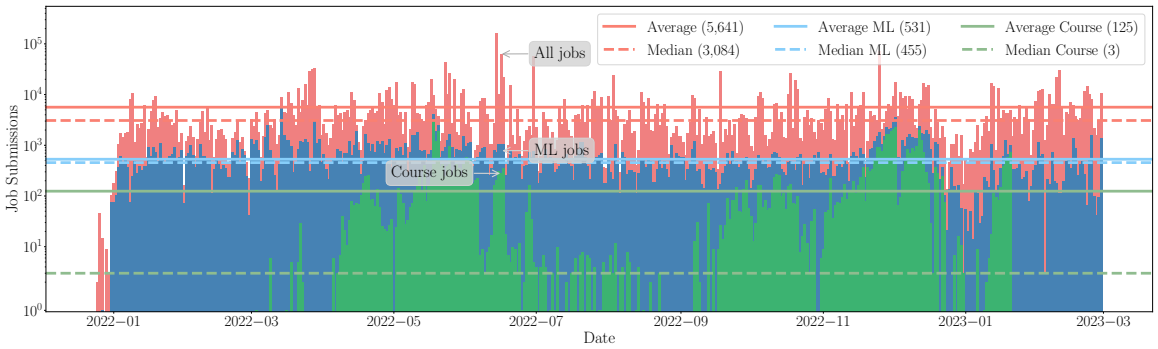
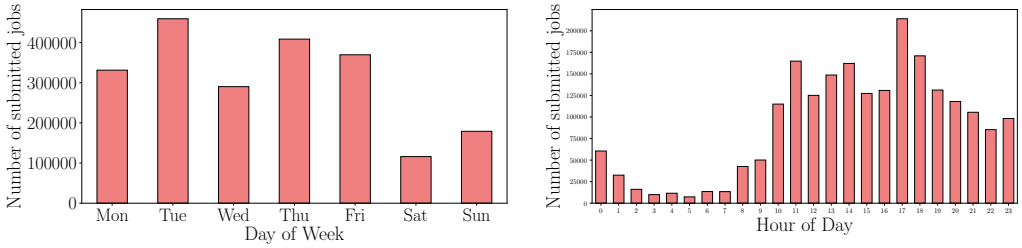


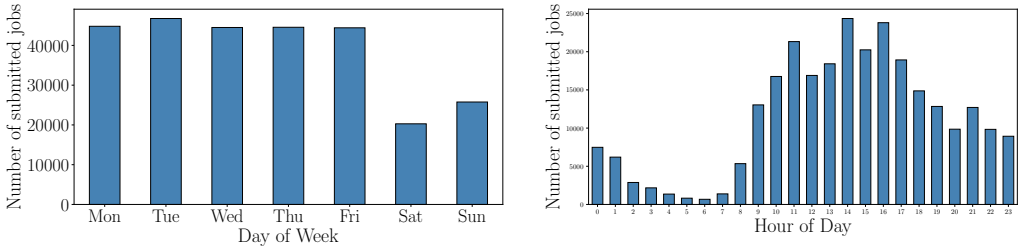

**Figure 4: Overview of the number of submitted jobs by day.**

The weekly and hourly distribution of job submissions and failures, as depicted in Figures 5 and 6, unveils distinct temporal patterns in arrival and failure rates (**O-6**). A diurnal pattern is evident in job submissions, with peaks occurring between 9:00 and 18:00. This pattern likely reflects working hours, indicating that job submissions are influenced by user behavior and operational schedules. Moreover, a diurnal pattern is also observed in ML job failures, suggesting that failure occurrences follow similar temporal trends as job submissions.

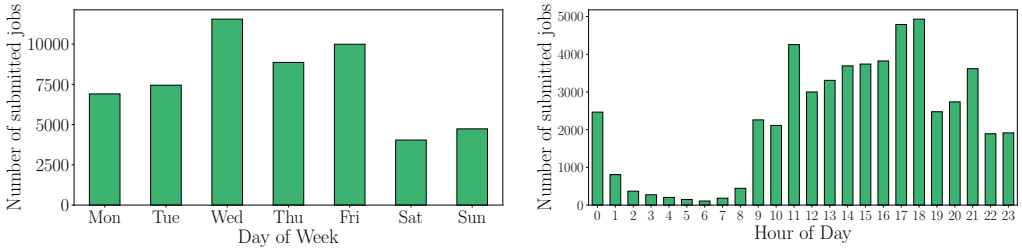
Additionally, while ML job failures exhibit a more consistent distribution across weeks and days, failures of generic jobs are more concentrated in two distinct periods, 11:00 to 14:00 and 17:00 to 18:00. This discrepancy highlights potential differences in failure triggers or environmental conditions between ML and generic workloads. Understanding these fluctuations is crucial for resource allocation, capacity planning, and workload management strategies within the datacenter.




(a) Generic jobs.




(b) ML Jobs




(c) Course jobs

**Figure 5: Daily and hourly submissions for generic, ML, and course jobs.**

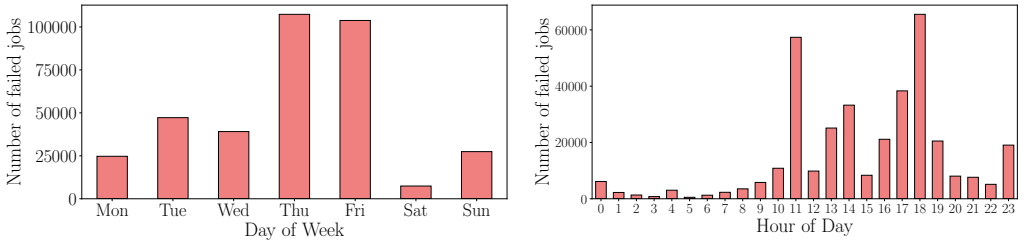
## 4.3 CPU Usage

**O-7.** Compared with generic jobs, failed ML jobs use more CPUs. Failed course-related jobs are allocated with even more CPUs.

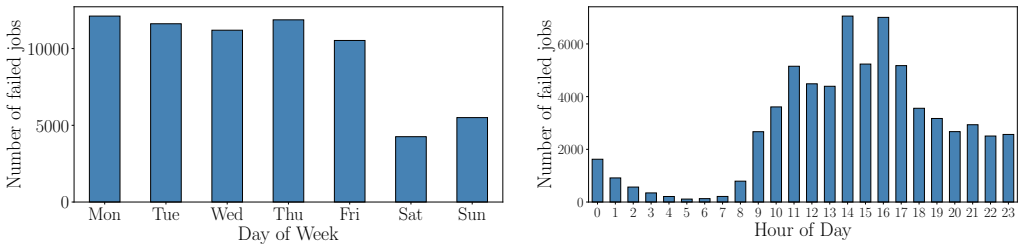
**O-8.** For ML jobs, including course-related jobs, there is no significant difference in terms of the distribution of CPUs consumed by failed and completed jobs.

**O-9.** CPU time occupied by failed ML jobs is higher than by failed generic jobs.

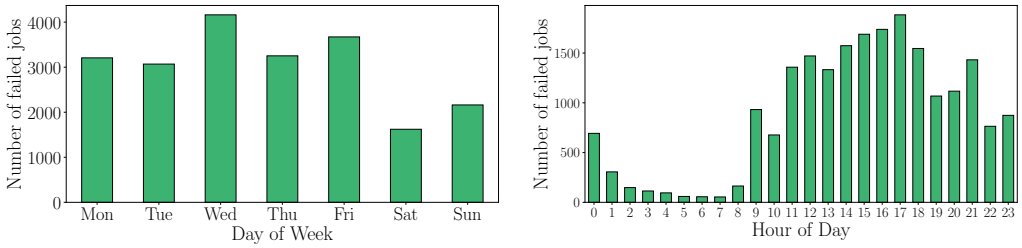
We examine the number of CPUs used in failed jobs and CPU time occupied to gain insights into the resource requirements of different job types. Figure 7 illustrates the cumulative distribution of the number of CPUs used in failed jobs, revealing distinct usage patterns across different job types (**O-7**). ML job failures are consistently allocated




(a) Generic jobs.




(b) ML Jobs




(c) Course jobs

**Figure 6: Daily and hourly failures for generic, ML, and course jobs.**

with a higher number of CPUs compared to generic jobs, indicating potentially higher computational demands or resource requirements associated with machine learning workloads. Notably, course-related jobs exhibit an even higher allocation of CPUs, suggesting specialized resource needs and computational complexities to these tasks.

Despite the differences in CPU allocation between job types, Figure 7 indicates that, for ML jobs, including course-related jobs, there is no significant disparity in the distribution of CPUs consumed by failed and completed jobs (**O-8**). This observation suggests that the number of CPUs allocated is not directly related to the failure of ML jobs but may be attributed to other factors such as computational complexity, algorithmic inefficiencies, or resource contention. Understanding these nuances in CPU utilization patterns is essential for identifying potential performance bottlenecks.

Figure 8 presents the cumulative distribution of CPU time consumed by failed jobs, highlighting differences in CPU utilization between ML and generic job failures (**O-9**). We observe that ML job failures occupy CPUs for a longer duration compared to generic job failures, indicating potentially higher computational demands or longer execution times associated with ML workloads.

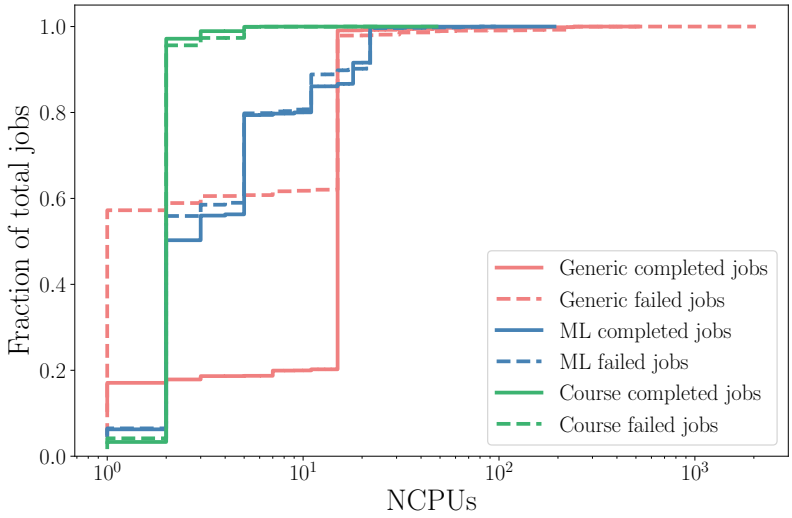



**Figure 7: The number of CPUs used in jobs, CDF plot.**

### 4.4 Time Correlations of Failures

**O-10.** At the week and day granularity, medium autocorrelations exist for only small time lags.

**O-11.** There is almost no autocorrelation at the hour granularity.

**O-12.** Neither generic job failures nor ML job failures show a linear trend.

**O-13.** There exists weekly patterns in ML and generic job failures.

**O-14.** The is little noise for failed ML jobs and generic jobs.

Previous research shows that some HPC clusters have strong autocorrelation at hourly and weekly lags, suggesting a considerable level of predictability. We intend to examine if failures in our dataset exhibit a repetitive pattern. For this purpose, we investigate the time-varying characteristics of failure events in Lisa. Figure 9 displays the autocorrelation functions at the time lag of week, day, and hour. We observe only medium correlations at the time lag 1 and 2 over the weekly granularity and at low time lags over the daily granularity (**O-10**). The presence of autocorrelation highlights the potential predictability of failure events at these time scales.

Contrary to the week and day granularities, Figure 9 indicates minimal autocorrelation at the hour granularity (**O-11**). The absence of autocorrelation suggests that failure events do not exhibit a repetitive pattern at this fine time scale, indicating a higher level of randomness or unpredictability in the occurrence of failures on an hourly basis.

Figure 10 depicts the trend, seasonal, and residual components of failure events, providing insights into the underlying temporal patterns (**O-12**). We observe fluctuations in the trend component for both ML job failures and generic job failures, indicating no discernible linear trend in the occurrence of failure events over time. This observation suggests that failure events do not exhibit systematic increases or decreases over the observation period, highlighting the importance of adaptive monitoring and response strategies capable of accommodating fluctuating failure rates within the HPC environment.

Despite the absence of a linear trend, Figure 10 reveals the presence of weekly patterns

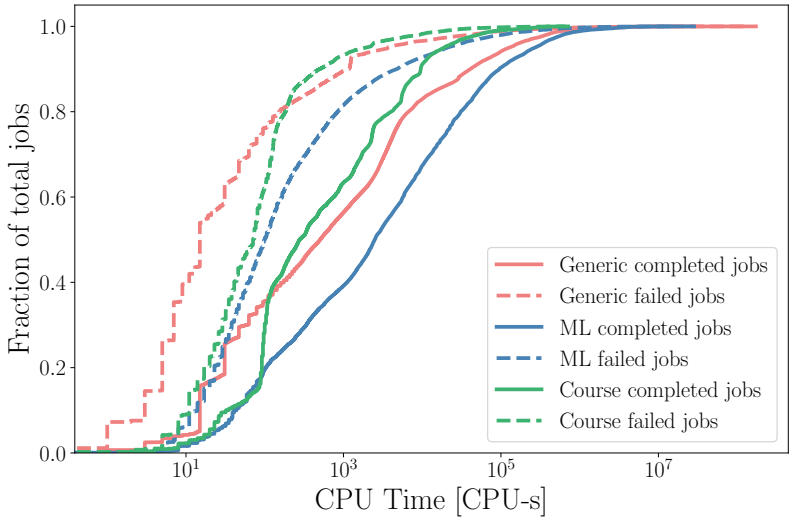
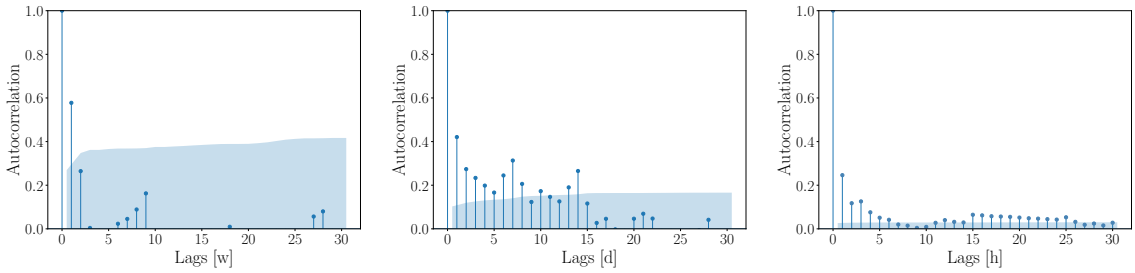



**Figure 8: CPU time consumed by jobs, CDF plot.**




(a) Autocorrelation at week granularity.

(b) Autocorrelation at day granularity.

(c) Autocorrelation at hour granularity.

**Figure 9: Autocorrelation functions with the data aggregated at different time granularities.**

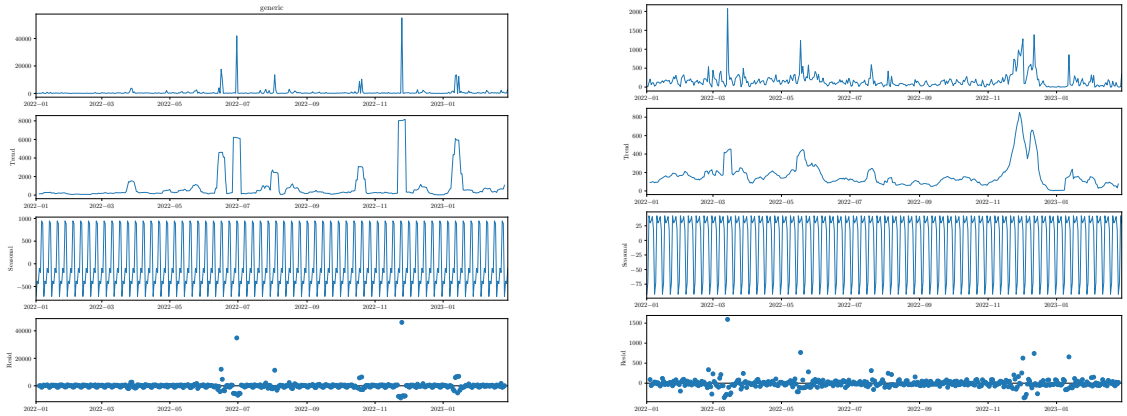
in both ML and generic job failures (**O-13**). Regular peaks and valleys repeat at intervals of seven days, indicating a cyclicality in the occurrence of failure events. However, it's noteworthy that the patterns in ML job failures and generic job failures exhibit different shapes, suggesting variations in the underlying temporal dynamics or contributing factors between job types.

The residual component captured in Figure 10 indicates minimal noise for both failed ML jobs and generic jobs (**O-14**). The absence of discernible patterns in the residual component suggests that the predictable components of the data, such as trend and seasonality, effectively explain the variability in failure events.

## 4.5 Peak Analysis

**O-15.** There is no significant difference between ML job failures (0.53 h) and generic job failures (0.56 h) in the average peak duration.

**O-16.** The average time between peaks of ML jobs (33.67 h) is higher than that for



(a) Generic jobs. (b) ML Jobs

**Figure 10: Trends, seasonality, and noise of failed jobs.**

generic jobs (7.38 h).

**O-17.** The average inter-arrival time during peaks for ML jobs (18.47 s) is higher than that for generic jobs (5.56 s).

**O-18.** ML jobs feature a higher mean failure duration (1.98 h) during peaks than generic jobs (1.47 h).

| Failure Type | Avg. Peak Duration [h] | Avg. Inter-Peak Time [h] | Avg. Failure IAT During Peaks [s] | Avg. Failure Duration During Peaks [h] |
|---|---|---|---|---|
| Failed ML Jobs | 0.53 | 33.67 | 18.47 | 1.98 |
| Failed Generic Jobs | 0.56 | 7.38 | 5.56 | 1.47 |

**Table 4: Average peak duration, inter-peak time, inter-arrival time, and failure duration for ML job failures and generic job failures.**

Understanding peaks of job failure rate can be instrumental for performance optimization and fault-tolerant design [2]. Thus, we follow the methodology described in Section 3.2 and present the four parameters of failed ML jobs and generic jobs in Table 4.

We observe that the average peak duration is 0.53 h for ML job failures and 0.56 h for generic job failures, indicating no substantial difference in the duration of peaks between the two types of jobs (**O-15**). This suggests that the duration of peak failure periods remains consistent regardless of job type, highlighting a uniformity in the temporal patterns of job failures across the dataset.

ML jobs have a significantly longer average time between peaks (33.67h) compared to generic jobs (7.38h). This substantial difference suggests that ML jobs experience peaks in failure rate with a considerably lower frequency compared to generic jobs (**O-16**). It could be attributed to various factors, such as the specific computational requirements, algorithm complexity, or resource utilization patterns associated with GPU nodes inside this cluster.

The average inter-arrival time during peaks for ML jobs (18.47 s) is also higher than that for generic jobs (5.56 s) (**O-17**). This suggests that failures in ML jobs occur less frequently during peak periods, indicating a more sporadic or intermittent failure behavior. Such insights into the frequency and timing of failures are valuable for developing targeted fault tolerance mechanisms and resource allocation strategies that can better accommodate the specific characteristics of ML workloads within HPC systems.

ML jobs have a higher mean failure duration (1.98 h) during peaks than generic jobs (1.47 h) (**O-18**), This observation underscores the longer-lasting impact of failures within ML workloads during peak periods. The extended duration of failures in ML jobs highlights the potential challenges in maintaining system reliability and performance under such conditions.

Compared with the two datasets SDSC and LANL from other HPC systems (details in Table 5), the most significant difference between our dataset and them lies in the average failure duration during peaks. Both kinds of jobs in our system have a much shorter average failure duration, which may stem from variations in system configurations, workload characteristics, or fault tolerance strategies employed across different HPC environments. Understanding these differences can help to design tailored failure mitigation strategies, resource allocation policies, and scheduling mechanisms to better support ML workloads and improve the overall reliability and performance of the system.

| System | Avg. Failure IAT During Peaks [s] | Avg. Failure Duration During Peaks [h] |
|---|---|---|
| LDNS | 8.61 | 8.39 |
| LANL | 5.88 | 5.89 |

**Table 5: Average failure duration and inter-arrival time during peaks for failures in LDNS and LANL systems. [2]**

# 5 Prediction of Machine Learning Job Failures

In this section, we introduce our LSTM-TCN hybrid model with the Attention mechanism for predicting ML job failures. First, we summarize the experimental setup, including evaluation metrics and hyperparameters. Second, we analyze model results, comparing this hybrid model with standalone LSTM and TCN models across multiple time granularities. Furthermore, we provide insights into computational aspects including inference time and memory usage, offering a comprehensive view of the model's performance and resource requirements.

## 5.1 Experiment Setup

### 5.1.1 Formulation

In this section, we provide the mathematical formulation of the LSTM-TCN hybrid model tailored for predicting job failures across various time granularities.

The model architecture comprises several key components as discussed below.

**LSTM Model**

The Long Short-Term Memory (LSTM) model is a recurrent neural network (RNN) variant known for capturing long-term dependencies in sequential data. At each time step $t$, the LSTM model updates its hidden state $h^t_{\text{LSTM}}$ and cell state $c^t_{\text{LSTM}}$:

$$h^t_{\text{LSTM}}, c^t_{\text{LSTM}} = \text{LSTM}(X^t_{\text{LSTM}}, h^{t-1}_{\text{LSTM}}, c^{t-1}_{\text{LSTM}}; \theta_{\text{LSTM}})$$

where $X^t_{\text{LSTM}}$ represents the input sequence matrix at time $t$, $\theta_{\text{LSTM}}$ denotes the parameters of the LSTM model, and the LSTM function encompasses the LSTM operations.

**TCN Model**

The Temporal Convolutional Network (TCN) model is another sequential model that employs convolutional operations to capture temporal patterns. At each time step $t$, the TCN model computes its hidden state $h^t_{\text{TCN}}$ using convolutional layers:

$$h^t_{\text{TCN}} = \text{TCN}(X^t_{\text{TCN}}; \theta_{\text{TCN}})$$

where $X^t_{\text{TCN}}$ denotes the input sequence matrix, $\theta_{\text{TCN}}$ represents the parameters of the TCN model, and TCN encapsulates the TCN operations.

**Attention Mechanism**

The Attention mechanism is deployed to enhance the model's capability to focus on relevant parts of the input sequence. At each time step $t$, the attention mechanism computes attention weights $a^t_{\text{Attention}}$ based on the hidden states of the LSTM and TCN models:

$$\text{Attention}^t = \text{Attention}(h^t_{\text{LSTM}}, h^t_{\text{TCN}})$$

These attention weights are then used to combine the LSTM and TCN outputs to produce an Attention-based representation $y^t_{\text{Attention}}$:

$$y^t_{\text{Attention}} = \text{Concatenate}(h^t_{\text{LSTM}}, h^t_{\text{TCN}}, a^t_{\text{Attention}})$$

**Hybrid Model**

Finally, the hybrid model integrates the attention-based representation $y^t_{\text{Attention}}$ to generate the final output prediction $y^t_{\text{Hybrid}}$:

$$y^t_{\text{Hybrid}} = \text{Dense}(y^t_{\text{Attention}}; \theta_{\text{Hybrid}})$$

where $\theta_{\text{Hybrid}}$ represents the parameters of the hybrid model, and Dense represents a fully connected layer.

### 5.1.2 Dataset

The dataset used for our predictive models spans a duration from July 4, 2022, to October 27, 2022, taken from a larger dataset that is enriched based on the dataset used in Section 4 and with more metrics. This dataset comprises 461,500 entries with 99 columns. Each row in the dataset represents a unique job execution, encompassing various metrics such as node performance, network statistics, and GPU-specific information. It has the same sampling frequency of 30 seconds, offering a fine-grained perspective on job execution dynamics. The 'timestamp' column records the date and time of each job submission. A brief overview is shown in Table 6.

| Dataset | Description |
|---|---|
| Source | SURFLisa |
| Start Date | 2022-07-04 |
| End Date | 2022-10-27 |
| Sampling Rate | 30 seconds |
| Metrics | 98 |
| Datapoints | 461,500 |

**Table 6: Overview of the ML job dataset for prediction.**

### 5.1.3 Evaluation Metrics

We considered several metrics to evaluate the model, with Mean Squared Error (MSE) being the primary metric, supplemented by Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). These metrics provide comprehensive insights into the model's accuracy and precision, enabling a thorough assessment of its performance.

**Mean Squared Error (MSE)**

The Mean Squared Error quantifies the average squared difference between the predicted and true values of job failures over a given period. MSE is preferred as the primary metric as it penalizes larger errors more significantly. A lower MSE value indicates better alignment between predicted and true values, signifying improved predictive accuracy. Mathematically, the MSE for the hybrid model is computed as:

$$\text{MSE}_{\text{Hybrid}} = \frac{1}{N}\sum_{t=1}^{N}(y^t_{\text{Hybrid}} - y^t_{\text{True}})^2.$$

**Mean Absolute Error (MAE)**

The Mean Absolute Error measures the average absolute difference between the predicted and true values of job failures. Similar to MSE, a lower MAE value indicates closer agreement between predicted and true values, highlighting enhanced predictive accuracy and reduced deviation. Mathematically, the MAE for the hybrid model is calculated as:

$$\text{MAE}_{\text{Hybrid}} = \frac{1}{N} \sum_{t=1}^{N} |y^t_{\text{Hybrid}} - y^t_{\text{True}}|.$$

**Root Mean Squared Error (RMSE)**

RMSE is the square root of the MSE, representing the standard deviation of the residuals. It provides a measure of dispersion similar to MSE but in the original units of the target variable, aiding in the practical interpretation of prediction accuracy. RMSE is calculated as:

$$\text{RMSE}_{\text{RMSE}} = \sqrt{\text{MSE}}.$$

### 5.1.4 Hyperparameter

Hyperparameters shape the architecture, behavior, and performance of predictive models. In this section, we discuss the specific hyperparameters tailored to our LSTM-TCN hybrid model, clarifying the rationale behind their selection.

**Sequence Length**

The sequence length is important for giving the model enough context. We carefully selected a sequence length of 30 steps to balance between capturing historical patterns thoroughly and mitigating computational complexities.

**Prediction Steps**

The prediction steps define how far ahead the model forecasts, closely linked with operational needs and the dynamic nature of ML job failures. Considering these aspects, we chose a prediction horizon of 7 time steps. This timeframe achieves a balance between promptly anticipating potential failures and managing the computational demands of the model.

**LSTM and TCN Architecture**

The architectural nuances of the LSTM and TCN models significantly impact their ability to recognize temporal patterns and extract important features from input sequences. In our hybrid model, we combined a single-layer LSTM with 20 units with the default TCN architecture.

**Learning Rate**

Learning rate is another crucial hyperparameter, influencing the optimization speed and model convergence. To ensure stable training and reliable convergence, we opted for the default learning rate with the Adam optimizer.

Overall, the hyperparameter selection process involved a balance between model complexity and predictive accuracy, guided by empirical insights from extensive experimentation with different configurations. This thorough exploration and fine-tuning of hyperparameters contributed to the robustness and efficacy of the hybrid LSTM-TCN model for job failure prediction across diverse time granularities, which is discussed in the next section.

## 5.2 Model Results

In this section, we evaluate the performance of the LSTM, TCN, and LSTM-TCN hybrid models in predicting job failures across different time granularities. We compare the performance metrics of each model variant and provide insights into their predictive capabilities and computational efficiency.

### 5.2.1 LSTM, TCN

Table 7 provides a comparative analysis of the LSTM and TCN models in terms of prediction accuracy across daily, hourly, and minute intervals.

The LSTM model exhibits varying performance metrics across different temporal resolutions. At the daily granularity, the LSTM achieves an MSE of 0.2191, RMSE of 0.4681, and MAE of 0.4355. As we transition to finer temporal resolutions, such as hourly and minute intervals, the LSTM model demonstrates notable improvements in predictive accuracy, as evidenced by substantially lower MSE, RMSE, and MAE values.

Conversely, the TCN model demonstrates more consistent performance across all time granularities, with relatively low MSE, RMSE, and MAE values compared to the LSTM model. TCN's superior performance at finer time intervals indicates its effectiveness in capturing short-term patterns and dependencies.

| | | MSE | RMSE | MAE |
|---|---|---|---|---|
| LSTM | Day | 0.2191 | 0.4681 | 0.4355 |
| | Hour | 0.0054 | 0.0736 | 0.0253 |
| | Minute | 0.0443 | 0.2105 | 0.1457 |
| TCN | Day | 0.0876 | 0.2960 | 0.1848 |
| | Hour | 0.0041 | 0.0640 | 0.0216 |
| | Minute | 0.0024 | 0.0489 | 0.0193 |

**Table 7: Performance Comparison between LSTM and TCN.**

### 5.2.2 LSTM-TCN

Table 8 presents the performance metrics of the LSTM-TCN hybrid model, showcasing its effectiveness in predicting job failures across different temporal resolutions. Compared to individual LSTM and TCN models, the LSTM-TCN hybrid model demonstrates improved performance, with reduced MSE, RMSE, and MAE values across all granularities. For instance, at the daily granularity, the LSTM-TCN model achieves an MSE of 0.1636, RMSE of 0.4045, and MAE of 0.3713, indicating superior predictive accuracy.

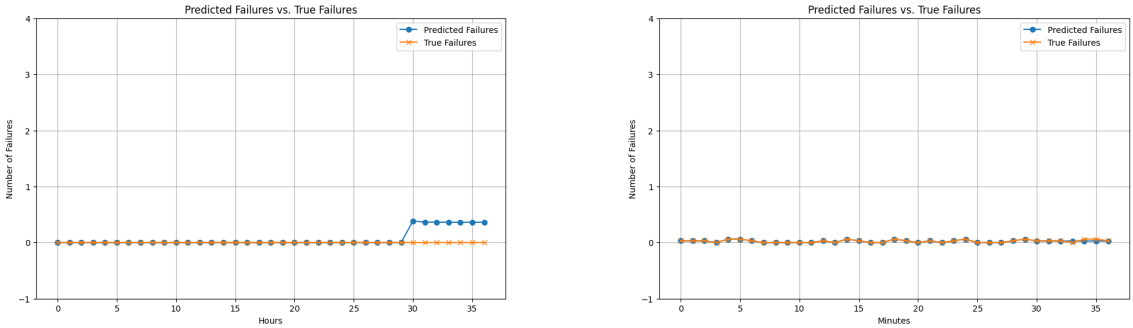
In Figure 11, the relationship between predicted failures and actual failures is depicted at hourly and minute intervals for the hybrid model, with a single set of inputs.

### 5.2.3 Comparison

A comprehensive comparison of the LSTM, TCN, and LSTM-TCN hybrid models is provided in Table 9, outlining various aspects including inference time, memory usage, and predicting time for a single set of inputs.

| | MSE | RMSE | MAE |
|---|---|---|---|
| Day | 0.1636 | 0.4045 | 0.3713 |
| Hour | 0.0121 | 0.1102 | 0.0959 |
| Minute | 0.0002 | 0.0141 | 0.0035 |

**Table 8: Performance of the LSTM-TCN Model.**




(a) Predicted vs. True at hour granularity.

(b) Predicted vs. True at minute granularity.

**Figure 11: Predicted Failures vs. True Failures at different time granularities.**

In terms of inference time, the LSTM model demonstrates the shortest inference time across all time granularities. However, as the temporal resolution becomes finer, both TCN and LSTM-TCN models exhibit slightly longer inference times, with the LSTM-TCN hybrid model having the longest inference time. This indicates that while LSTM performs faster in making predictions, the inclusion of TCN components in the hybrid model slightly increases inference time.

Regarding memory usage, all models show an increase in memory consumption with finer time granularities, as expected due to the larger number of data points being processed. Notably, the LSTM-TCN hybrid model consumes the highest amount of memory across all granularities. This can be attributed to the combined architecture of LSTM and TCN, which requires more parameters and memory allocation compared to individual LSTM or TCN models.

When considering predicting time for a single set of inputs, all models demonstrate similar performance across different time granularities, with slight variations in predicting time observed between the LSTM, TCN, and LSTM-TCN models. However, these differences are marginal, indicating that the predicting time for a single set of inputs is largely consistent across all models regardless of the architecture or granularity.

Overall, the LSTM-TCN hybrid model showcases improved performance metrics such as MSE, RMSE, and MAE compared to individual LSTM and TCN models. However, this enhancement comes at the cost of longer training time and increased memory usage due to the complexity of the hybrid architecture. Therefore, while the hybrid model offers superior predictive accuracy, it may require additional computational resources for training and deployment compared to standalone LSTM or TCN models.

| | | LSTM | TCN | LSTM-TCN |
|---|---|---|---|---|
| Inference Time [s] | Day | 0.0634 | 0.0769 | 0.0861 |
| | Hour | 0.2378 | 0.4974 | 0.5566 |
| | Minute | 8.5920 | 26.3390 | 19.2426 |
| Memory Usage [B] | Day | 2752.59 | 3221.13 | 4484.30 |
| | Hour | 2755.41 | 3240.00 | 4486.99 |
| | Minute | 2891.66 | 3444.52 | 4668.73 |
| Predicting Time for a Single Set of Inputs [s] | Day | 0.5 | 0.5 | 0.6 |
| | Hour | 0.5 | 0.6 | 0.6 |
| | Minute | 0.6 | 0.6 | 0.6 |

**Table 9: Comparison of LSTM, TCN, and the Hybrid Model.**

# 6 Related Work

In this section, we present an overview of earlier studies on the characterization and prediction of job failures in HPC systems, respectively.

## 6.1 Job Failures Characterization

Characterizing job failures is critical for understanding the underlying patterns and factors contributing to system instability. While earlier studies often relied on publicly available trace datasets such as the Google cluster traces (GCT) [16, 17, 18, 10, 19], recent efforts have diversified to include open-source trace archives and operational datacenter traces.

Studies based on the GCT have provided valuable insights into the differences between completed and failed jobs in terms of required resources [20, 21]. However, reliance solely on GCT may introduce biases and limitations, prompting the exploration of alternative trace datasets.

Works like the Failure Trace Archive (FTA) [12] and operational trace archives from datacenters like the SURF datacenter [14] have facilitated broader analyses of job failures. Kondo et al. [12] designed the (FTA) and established a standard format for failure-related data, although high-level failures like job failures were not explicitly included. Laursen et al. [14] published an operational traces archive that contains over 100 low-level metrics with the finest granularity gathered at 15-second intervals from the SURF datacenter. This archive enabled statistical analysis of the Lisa cluster within the datacenter, showcasing the significance of data science analysis. Additionally, based on this archive, Cetin [22] investigated the operational characterization of the Lisa cluster during the COVID and non-COVID periods. He also compared the differences between generic nodes (CPU nodes) and ML nodes (GPU nodes).

Moreover, while some research has explored the time correlation of failures in large-scale distributed systems [23, 1], there remains a dearth of investigation specifically focused on HPC clusters. Our study adds to this topic by examining not only the autocorrelation but also the seasonality and periodicity of failed jobs in an HPC environment. Yigitbasi et al. [2] model the peak failure periods for nineteen traces acquired from multiple large-scale distributed systems. Similarly, we compute the four parameters in the model mentioned above and compare them with the parameters of traces gathered from HPC clusters.

### 6.2 Job Failures Prediction

Predicting job failures is essential for proactive fault management and resource optimization within HPC clusters. Existing prediction models leverage both traditional machine learning algorithms and deep learning approaches, often based on datasets like the GCT.

Prior research has extensively explored the methodologies of job failure prediction within HPC cluster environments. Existing prediction methods leverage both traditional machine learning algorithms and deep learning approaches. Traditional machine learning algorithms, such as Logistic Regression, Decision Trees, Random Forest, and Extreme Gradient Boosting, have been employed to build predictive models [8, 24]. Additionally, deep learning techniques like Long Short Term Memory (LSTM) networks, including single-layer, bi-layer, and tri-layer LSTM, have also been utilized to improve prediction accuracy [11, 25].

However, the prediction of ML job failures has not been well-studied. Liu et. al. [26] propose parallel and cascade model-ensemble mechanisms and a sliding training method to address the shortcomings of classic models when predicting GPU failures. Their work demonstrates the need for specialized predictive models tailored to the characteristics of ML jobs, which often exhibit high resource consumption and complex dependencies. Our study contributes to this area by developing a predictive model specifically designed to enhance the efficiency and reliability of HPC clusters, particularly in the context of ML job failures.

Table 10 provides an overview of existing job failure prediction models, showcasing the diversity of techniques and data sources employed in this domain.

By building upon and extending prior research, our study aims to advance the understanding and prediction of job failures within HPC systems, with a particular emphasis on ML workloads.

| Data sources | Inputs | Outputs | Techniques | Reference |
|---|---|---|---|---|
| GCT 2011 | mean CPU usage, mean memory usage, unmapped page cache, mean disk I/O time, mean disk usage | termination status | RNN | Chen et al., 2014 |
| GCT 2011 | job attributes and system attributes | termination status | LDA, ELDA, QDA, LR | Rosà et al., 2015 |
| GCT 2011 | static features | termination statuses | OS-ELM | Liu et al. 2017 |
| GCT 2011 | task priority, number of task resubmissions, scheduling delay | termination status | Bi-LSTM | Gao et al., 2022 |

**Table 10: Overview of the job failure prediction models.**

# 7 Conclusion

In this study, we delved into the critical challenges of job failures within HPC datacenters, with a particular focus on ML job failures. Our investigation was motivated by the increasing reliance on datacenter infrastructures for digital services and the emergence of ML-based applications. We identified ML job failures as a significant challenge due to their distinctive attributes and the potential impact on service reliability and user experience.

Through a comprehensive analysis of a long-term job dataset from the SURFLisa scientific cluster in the Netherlands, we characterized various aspects of ML job failures, including arrival patterns, temporal correlations, and failure peaks. Our observations reveal that ML jobs exhibit higher failure rates compared to generic jobs, with a significantly higher mean time between failures (MTBF). These insights underscore the need for tailored approaches to address ML job failures within datacenter environments.

To mitigate these failures, we proposed and evaluated a predictive modeling approach leveraging deep learning techniques, the LSTM-TCN hybrid model. Our predictive model demonstrated superior performance in capturing temporal patterns associated with ML job failures, outperforming standalone LSTM and TCN models at varying time granularities. However, it is essential to consider the trade-off between predictive accuracy and computational resources, as the hybrid model incurred longer inference times and increased memory usage.

In summary, our study contributes to a deeper understanding of ML job failures within datacenter environments and offers practical insights for enhancing datacenter reliability in the era of ML-based services. By providing methodologies for the characterization and prediction of ML job failures, we aim to empower researchers and practitioners to optimize datacenter operations and minimize disruptions to digital services.

Future research directions may include further refinement of predictive models to balance between accuracy and resource efficiency, and exploring adaptive strategies for fault tolerance tailored specifically for ML workloads within datacenter environments.

## References


[1] X. Chu, L. Versluis, S. Talluri, and A. Iosup, "How Do ML Jobs Fail in Datacenters? Analysis of a Long-Term Dataset from a HPC Cluster," 2023.

[2] N. Yigitbasi, M. Gallet, D. Kondo, A. Iosup, and D. Epema, "Analysis and modeling of time-correlated failures in large-scale distributed systems," in *2010 11th IEEE/ACM International Conference on Grid Computing*, (Brussels, Belgium), pp. 65–72, IEEE, Oct. 2010.

[3] Dutch Data Center Association, "State of the dutch data centers." `https://www.dutchdatacenters.nl/en/publications/state-of-the-dutch-data-centers-2022/`. Accessed: 2023-01-09.

[4] H. S. Gunawi, M. Hao, T. Leesatapornwongsa, T. Patana-anake, T. Do, J. Adityatama, K. J. Eliazar, A. Laksono, J. F. Lukman, V. Martin, and A. D. Satria, "What Bugs Live in the Cloud? A Study of 3000+ Issues in Cloud Systems," in *Proceedings of the ACM Symposium on Cloud Computing*, (Seattle WA USA), pp. 1–14, ACM, Nov. 2014.

[5] H. S. Gunawi, M. Hao, R. O. Suminto, A. Laksono, A. D. Satria, J. Adityatama, and K. J. Eliazar, "Why Does the Cloud Stop Computing?: Lessons from Hundreds of Service Outages," in *Proceedings of the Seventh ACM Symposium on Cloud Computing*, (Santa Clara CA USA), pp. 1–16, ACM, Oct. 2016.

[6] C. Cerin, C. Coti, P. Delort, F. Diaz, M. Gagnaire, M. Mijic, Q. Gaumer, N. Guillaume, J. L. Lous, S. Lubiarz, J.-L. Raffaelli, K. Shiozaki, H. Schauer, J.-P. Smets, L. Seguin, and A. Ville, "Downtime Statistics of Current Cloud Solutions," p. 5, 2014.

[7] L. Versluis, M. Cetin, C. Greeven, K. Laursen, D. Podareanu, V. Codreanu, A. Uta, and A. Iosup, "A Holistic Analysis of Datacenter Operations: Resource Usage, Energy, and Workload Characterization – Extended Technical Report," July 2021.

[8] A. Rosà, L. Y. Chen, and W. Binder, "Predicting and Mitigating Jobs Failures in Big Data Clusters," in *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 221–230, May 2015.

[9] A. Rosa, L. Y. Chen, and W. Binder, "Failure Analysis and Prediction for Big-Data Systems," *IEEE Trans. Serv. Comput.*, vol. 10, pp. 984–998, Nov. 2017.

[10] C. Liu, J. Han, Y. Shang, C. Liu, B. Cheng, and J. Chen, "Predicting of Job Failure in Compute Cloud Based on Online Extreme Learning Machine: A Comparative Study," *IEEE Access*, vol. 5, pp. 9359–9368, 2017.

[11] J. Gao, H. Wang, and H. Shen, "Task Failure Prediction in Cloud Data Centers Using Deep Learning," *IEEE Trans. Serv. Comput.*, vol. 15, pp. 1411–1422, May 2022.

[12] D. Kondo, B. Javadi, A. Iosup, and D. Epema, "The Failure Trace Archive: Enabling Comparative Analysis of Failures in Diverse Distributed Systems," in *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, (Melbourne, Australia), pp. 398–407, IEEE, 2010.

[13] S. Minnema, "A Statistical Analysis of Cloud Service Failures by Severity," 2022.

[14] K. V. Laursen Olason, A. Uta, A. Uta, P. Melis, D. Podareanu, and V. Codreanu, "Beneath the SURFace: An MRI-like View into the Life of a 21st Century Datacenter," June 2020.

[15] SURF, "lisa - surf user knowledge base - surf user knowledge base." `https://servicedesk.surf.nl/wiki/display/WIKI/Lisa`. Accessed: 2023-01-09.

[16] X. Chen, C.-D. Lu, and K. Pattabiraman, "Failure Analysis of Jobs in Compute Clouds: A Google Cluster Case Study," in *2014 IEEE 25th International Symposium on Software Reliability Engineering*, (Naples, Italy), pp. 167–177, IEEE, Nov. 2014.

[17] P. Garraghan, P. Townend, and J. Xu, "An Empirical Failure-Analysis of a Large-Scale Cloud Computing Environment," in *2014 IEEE 15th International Symposium on High-Assurance Systems Engineering*, (Miami Beach, FL, USA), pp. 113–120, IEEE, Jan. 2014.

[18] Y. Chen, A. Ganapathi, R. Griffith, and R. H. Katz, "Analysis and Lessons from a Publicly Available Google Cluster Trace," 2010.

[19] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," 2012.

[20] M. Alam, K. A. Shakil, and S. Sethi, "Analysis and Clustering of Workload in Google Cluster Trace Based on Resource Usage," in *2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*, (Paris), pp. 740–747, IEEE, Aug. 2016.

[21] M. Jassas and Q. H. Mahmoud, "Failure Analysis and Characterization of Scheduling Jobs in Google Cluster Trace," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, (D.C., DC, USA), pp. 3102–3107, IEEE, Oct. 2018.

[22] M. B. Cetin, "Operational Characterization of a Public Scientific Datacenter During and Beyond the COVID-19 Period," 2021.

[23] H. Li, D. Groep, L. Wolters, and J. Templon, "Job Failure Analysis and Its Implications in a Large-Scale Production Grid," in *2006 Second IEEE International Conference on E-Science and Grid Computing (e-Science'06)*, (Amsterdam, The Netherlands), pp. 27–27, IEEE, Dec. 2006.

[24] School of Computer Engineering, Suranaree University of Technology (SUT), Thailand, A. Banjongkan, W. Pongsena, N. Kerdprasop, and K. Kerdprasop, "A Study of Job Failure Prediction at Job Submit-State and Job Start-State in High-Performance Computing System: Using Decision Tree Algorithms," *JAIT*, vol. 12, no. 2, pp. 84–92, 2021.

[25] T. N. Tengku Asmawi, A. Ismail, and J. Shen, "Cloud failure prediction based on traditional machine learning and deep learning," *J Cloud Comp*, vol. 11, p. 47, Sept. 2022.

[26] H. Liu, Z. Li, C. Tan, R. Yang, G. Cao, Z. Liu, and C. Guo, "Predicting GPU Failures With High Precision Under Deep Learning Workloads," in *Proceedings of the 16th ACM International Conference on Systems and Storage*, (Haifa Israel), pp. 124–135, ACM, June 2023.