HazardNet: A thermal hazard prediction framework for datacenters

HazardNet: A Thermal Hazard Prediction Framework for Datacenters

Mohsen Seyedkazemi Ardebili[a], Andrea Acquaviva[a], Luca Benini[a,b], Andrea Bartolini[a]

[a] *Department of Electrical, Electronic and Information Engineering, University of Bologna, Italy*
[b] *Integrated Systems Laboratory, ETH Zurich, Switzerland*

Abstract

Modern scientific discoveries rely on an insatiable demand for computational resources. To meet this ever-growing computing demand, the datacenters have been established, which are complex controlled environments that host thousands of computing nodes, storage, high-performance communication networks, cooling systems, etc. A datacenter consumes a large amount of electrical power (in the range of megawatts), which gets completely transformed into heat, creating complex spatial and temporal thermal dissipation problems [1, 2]. Therefore, although a datacenter contains sophisticated cooling systems, minor thermal issues/anomalies can potentially trigger a chain of events that leads to an imbalance between the heat generated by computing nodes and the heat removed by the cooling system, leading to thermal hazards. Thermal hazards are detrimental to datacenter operations as they can lead to IT and facility equipment damage as well as an outage of the datacenter, with severe societal and business losses [3, 4]. So, predicting the thermal hazard/anomaly is critical to prevent future disasters. In doing so, collecting and analyzing large-scale monitoring signals and methodology for anomaly detection and prediction are challenging tasks.

In this manuscript, after providing a methodology for defining the thermal anomaly, we proposed HazardNet, a thermal hazard prediction framework that consists of a complete pipeline of deep learning models. We evaluated the proposed framework in two different scenarios. In the first scenario, we evaluated the model's performance over the entire study period, resulting in an F1-score of 0.98. In the second scenario, we enforced causality in the collected data by training and testing the model in two disjunct and consecutive periods, resulting in an F1-score of 0.87. Thanks to these promising results, HazardNet can capture the complex spatial and temporal dependency between datacenter operational parameters and thermal hazards and predict them in advance.

The dataset and code used in this study are publicly available on Zenodo (DOI: https://doi.org/10.5281/zenodo.10050368) and GitHub (https://github.com/MSKazemi/HazardNet).

Keywords: Datacenter, Thermal Hazard, Predictive Model, Thermal Anomaly Detection, Deep Learning, Temporal Convolutional Network

1. Introduction

Supercomputers are at the forefront of computing and nowadays at the heart of scientific, technological, economical, and industrial development and innovations. Datacenters provide supercomputers with high computing capacity by aggregating the computing power of thousands of computing nodes to deliver considerably higher performance than a typical desktop computer can provide.

The ICT sector's total electricity consumption is expected to reach 20% of the worldwide demand by 2030, with datacenters expected to account for one-third of that [5]. The EuroHPC program has invested ~650M€ in CAPEX and OPEX for the three procured pre-exascale systems with an estimated daily average cost of ~600k€ for a supercomputer. Therefore one-day outage of this supercomputer can cause a loss of ~600k€for the European taxpayer [3]. Whereas in the business datacenter sector, in 2016, an Amazon.com web service shortage would have cost, on average, 15M$ of revenue lost [6]. Therefore, datacenter outages or facility damages can have severe societal and financial consequences.

A datacenter is a complex controlled environment that hosts thousands of computing nodes that consume electrical power in the range of megawatts, which gets completely transformed into heat. Although datacenters contain sophisticated cooling systems (we will describe them briefly in Section 3) [7, 8, 9, 10, 11], studies in [1, 2] indicate quantitative evidence of power-thermal bottlenecks in real-life production workload, showing the presence of significant spatio-temporal power-thermal heterogeneity. For instance, the same studies for a specific Tier-0 datacenter reveal: *(i)* The different cooling technologies used in the datacenter room create heterogeneous thermal zones. *(ii)* Horizontal spatial proximity does not imply thermal coupling, and the compute nodes' inlet temperature significantly changes in the horizontal section. Measurements revealed up to $\sim 11^{\circ}C$ difference for the monthly average compute nodes' inlet temperature for nodes at the same height in the racks but at different locations in the datacenter room. *(iii)* The bottom and top of the racks are thermally decoupled. For example, the bottom and top of the racks at the center of the supercomputer room expe-

Email addresses: mohsen.seyedkazemi@unibo.it (Mohsen Seyedkazemi Ardebili), andrea.acquaviva@unibo.it (Andrea Acquaviva), luca.benini@unibo.it (Luca Benini), lbenini@iis.ee.ethz.ch (Luca Benini), a.bartolini@unibo.it (Andrea Bartolini)

rience, on average, $\sim 15°C$ thermal variation. This practically means that averaging temperatures at room and rack levels may hide important thermal hazard "symptoms".

Therefore minor thermal issues/anomalies can potentially start a chain of events that leads to an imbalance between the amount of heat generated by the computing nodes and the heat removed by the cooling system originating thermal hazards. This can be the result of the following: *(i)* an abnormal working condition of the cooling system, *(ii)* abnormal power fluctuation as well as a computing demand above the typical case, *(iii)* cooling capacity reduction due to an abnormal ambient temperature, *(iv)* different response latencies of computing and cooling elements to workload variations. Thermal hazards are detrimental to datacenter operations as they can lead to IT and facility equipment damage as well as an outage of the datacenter. In addition, (i) trends in electronics are worsening the power density of computing devices, increasing datacenter cooling complexity, and adopting free-cooling technologies while (ii) global warming produces unprecedented heat waves that are difficult to be tolerated, especially for many datacenter hosting sites whose building and cooling infrastructures were designed a few decades ago.

As a practical example of the detrimental impact of thermal hazards: CINECA's supercomputer (which is a Tier-0 PRACE supercomputing center) during the four days of reported thermal hazards in summer 2019 (from 2019-06-27 to 2019-07-01), encountered, on average, a 20% reduction in computing capacity - we will refer later in the paper to this event as a "reported-thermal-hazard".

There are three main challenges in creating a thermal hazard prediction framework capable of tracking the supercomputer's power and thermal spatio-temporal evolution within the datacenter:

1. *Large dataset:* to collect different monitoring signals of thousands of computing nodes during normal and abnormal production conditions; in general, anomalies are rare events, so capturing them requires collecting monitoring signals over a long period (i.e., seasons/year).

2. *Annotated dataset:* the collected dataset will not contain labels as not all thermal hazards get noticed nor get reported. Indeed cooling shortage conditions can manifest in different ways: with (i) an increase in the temperature of the electronics of computing nodes, (ii) an increased temperature of coolants (air/water), and (iii) the suboptimal working state of the cooling system. Moreover, computing units have a self-regulated system for managing power and temperature, which may conceal the symptoms of cooling shortage. This makes it more difficult for the facility manager to detect issues through visual inspections of monitoring signals. Therefore, in cases where thermal hazards are not catastrofic may remain invisible to the facility manager while impacting the supercomputer's performance i.e. degrading energy-to-solution, time-to-solution, reliability of the nodes, etc. Thermal hazard labeling is thus a challenging and expensive task in a datacenter. Due to the similarity between anomaly detection and labeling problems, we will refer to this as thermal (hazard) anomaly labeling.

3. *Spatio-temporal capable ML model:* The predictive model must be able to learn the supercomputer's power and thermal spatiotemporal evolution from a large dataset. Then predicting the thermal hazards in advance, giving facility managers, system administrators, or automated system software enough time to take countermeasures and limit the impact of the thermal hazard.

From this perspective, Machine Learning (ML) predictive tools are promising candidates for detecting and predicting anomalies, such as time-series forecasting and classification tools [12, 13, 14, 15, 16]. Deep Learning (DL) is a branch of Machine Learning and AI; since it shows successful functionality in learning from data, it is widely employed in different areas. Advancements in DL have enabled techniques for training models on large-scale time-series data. Recurrent Neural Networks (RNNs) are a category of Artificial Neural Networks (ANNs) that can track/show the temporal dynamic behavior of the time-series data. Long Short-Term Memory (LSTM) extends RNNs to learn long-term dependencies, which are common in time series data, but at a higher training complexity. LSTMs can learn long-term dependencies. This is possible due to the additional forget gate, beyond the basic input and output gates found in traditional RNNs. These gates control the flow of information and maintain a persistent internal state, allowing LSTMs to store information for longer periods of time. As a result, LSTMs can learn and make predictions based on long-term patterns and dependencies in time-series data [12, 17, 18]. Temporal Convolutional Networks (TCNs) are a type of neural network that are designed to process sequential data, such as time-series data or speech signals. They are similar to traditional convolutional neural networks (CNNs), but they are specifically designed to handle sequential data by using convolutional operations that are applied across time. Temporal Convolutional Network (TCN) has been proposed to overpass Long Short-Term Memory (LSTM) nets computational cost and has proved to achieve better performance[12, 19]. As a matter of fact, these models can be leveraged to create thermal hazard prediction solutions.

We should also emphasize that the spatial-temporal nature of the monitoring signals in a datacenter creates a complex correlation between the thermal and power monitoring signals. Therefore, monitoring data structure can play a meaningful role in the (two requirements mentioned earlier) thermal (hazard) anomaly labeling and predictive model's performance; in other words, the 1D or 2D array data structure can destroy some temporal information.

These challenging requirements create a meaningful perspective for research in automated approaches for thermal (hazard) anomaly labeling and prediction. We presented the preliminary results of this research in a conference poster/short paper (4 pages) [20]. The manuscript at hand represents an expanded and comprehensive version of our work, showcasing significant advancements. In this paper, we introduce HazardNet, a thermal hazard prediction framework that effectively addresses the aforementioned challenges. Our contributions can be summarized as follows:

1. *Input data structure:* To optimize the predictive model's

performance, we thoroughly explore various data structures. Through our investigation, we demonstrate that the 4D data structure proves most effective in preserving the spatio-temporal information embedded within monitoring signals.

2. *Automated dataset labeling [20]:*
   (a) Feature selection: Early works have analyzed the different signals which could be collected from a datacenter suggesting a subset of the parameters (inlet, outlet, and power consumptions of compute nodes) for the datacenter's room thermal characterization [1, 2]. In our short paper [20], we showed that the inlet temperature of computing nodes is the most effective metric for identifying thermal (hazard) anomalies.
   (b) Label Generation Method: In [20], we propose a statistical tool for thermal (hazard) anomaly labeling and validate it using reported thermal hazards.
3. *Predictive model:*
   (a) Input feature selection: We carried out an empirical study to select the best parameter (which are inlet temperature of computing nodes) for training the model.
   (b) Leveraging both classical Machine Learning (ML) and Deep Learning (DL) tools, including Long Short-Term Memory (LSTM) and Temporal Convolutional Networks (TCN), we conducted experiments to evaluate their performance. Our findings show that DL models consistently outperform classical ML approaches. The DL model achieves an F1 score that is 22% higher than the best results obtained by classical ML methods. Furthermore, within the DL models, TCN outperformed LSTM with a 17% higher F1-score [20].
   (c) Building upon these insights, we propose a TCN architecture with 3D convolutional layers. This architecture outperforms other models, resulting in a significantly higher F1-score (more than 18% higher than the initial 1D TCN architecture).
   (d) In addition, we compare a conventional implementation of the DL model with depthwise separable convolutions.

By expanding upon our previous conference publication [20], this paper presents novel findings and significant advancements in the development of HazardNet for thermal (hazard) anomaly labeling and prediction. Notably, in contrast to [20], we have employed sophisticated data structures and models to expand our investigation from 72 computing nodes (representing a single rack) to encompass the entirety of the datacenter, comprising a total of 3312 nodes.

The rest of the paper is organized as follows: first, we give an overview of the 2 Related Work and current state of the art, and next, we describe the 3 Background Setup. Then in the 4 Methodology, we introduce the proposed thermal hazard framework as well as other methods that we used for this study. After that, as proof of the approach, we report the 5 Experimental Results, and we discuss the performance of the different ML/DL models and then we discuss the 6 Framework Portability and finally conclude the study with a 7 Conclusion and Future Work.

In this study, we aim to provide transparency and reproducibility by making our dataset and code publicly available. The dataset is accessible on Zenodo with the (DOI: `https://doi.org/10.5281/zenodo.10050368`), and the code can be found on GitHub at (`https://github.com/MSKazemi/HazardNet`).

2. Related Works

In the SoA, various methodologies have been used to study thermal hazards in datacenters.

1. *Design Improvements:* A set of studies uses Computational Fluid Dynamics (CFD) to detect the thermal anomaly or undesired hotspots in the computing room of the datacenter. They propose design parameters to improve the performance of the cooling system [21, 22]. Authors of [23, 24] proposed to use sensors with a thermal computer model (a software that uses sensor data to create a thermal model) to create the room's heat map or thermal evolution model to use in the online predictor. This approach is not practical since the data are collected by moving the sensors in the room, which create different thermal snapshots for different room locations, and the final complete dataset is chronologically distorted, and the room's final thermal model is invalid. In [25], authors introduced the methodology for generating temperature models for datacenters and the runtime prediction of CPU and inlet temperature by using Grammatical Evolution techniques. They achieved average errors of $2°C$ for CPU temperature and $0.5°C$ for inlet temperature. Authors of the paper [22], after collecting the in-production datacenter's thermal data, evaluated the datacenter's thermal behavior by numerical analysis of flow and temperature fields. They found several troublesome hot spots, and they tried to reduce the air conditioning power demand by offering some solutions (such as cold-aisle containment, blocking of blank spaces between the racks, Computer Room Air Conditioning (CRAC) displacement, and alternative airflow distribution system). In [2], authors proposed a full-stack IoT system to collect the ambient temperature of the datacenter, and with data analysis, they studied the effect of cluster activity on the temperature of various locations of the datacenter. This study shows the significant thermal heterogeneity of the datacenter, and this study proposes using computing nodes' internal and onboard sensors for a more profound analysis of the power and thermal characteristics of the datacenter. Authors in [1] analyzed the thermal properties of a Tier0 datacenter deploying advanced hybrid cooling technologies. The study of spatial-temporal heterogeneity during production and cooling emergency hazards gives quantitative evidence of thermal bottlenecks in real-life production workloads, showing the presence of significant spatial thermal heterogeneity, which could be exploited by thermal-aware job scheduling and datacenter-room run-time workload adaptation and distribution.

2. *Thermal Aware Task Scheduling:* The authors of [26] proposed a mathematical model for thermal aware task scheduling, which in view of the complexity and computing time is a

trade-off between the complex CFD approach and sensor-based fast thermal evaluation model [24]. They utilized mathematical models for datacenter resources and workloads to create a Thermal Aware Scheduling Algorithm (TASA) that by scheduling the "hot" jobs on "cold" compute nodes, it reduces the temperatures of compute nodes. The simulation shows the reduction of 3.4°C of datacenter temperature by increasing 13.9% job response time.

3. *Thermal Anomaly Prediction* In [27], the authors discuss the performance of statistical techniques (thresholds, moving averages-based methods, EWMA-based algorithms, and Naïve Bayesian classifier) for early detection of thermal anomalies, using the data collected over three months from a real datacenter. The best method (naïve Bayes) reaches up to 18% detection of the anomalous events at an average of around 12 minutes before their happening (Based on our discussion with the facility manager of a datacenter, this is not sufficient time to take counteractions for preventing the thermal anomaly). This study used a very simplistic approach for the definition of the anomaly; moreover, the ML model has not enough complexity to capture spatio-temporal characteristics of monitoring signals. The authors of [28] exploit temperature sensor data in unsupervised anomaly detection. Authors benchmark their approach by simulation experiments for two different predictive failures: worn-out fans, and identifying Computer Room Air Conditioning (CRAC) failures before hotspots arise. Authors of [29] proposed a semisupervised autoencoder-based approach for anomaly detection in datacenter which improved detection accuracy of SoA (accuracies around 80%) by around 12%.

4. *Temperature Forecasting for Efficient Control of the Cooling System:* An effective control framework with a real-time prediction of server inlet temperature and tile flow rate is essential to optimize cooling system power consumption. Although Computational Fluid Dynamics and Heat Transfer (CFD/HT) are accurate and common tools to model thermal transport and airflow, the high computational costs and time make such simulations impractical for real-time simulations for datacenters. Therefore, the authors of the paper [30] developed an Artificial Neural Network (ANN)-based model to train with the offline generated dataset with CFD/HT for real-time prediction. Compared to CFD simulation, the ANN model in rack inlet temperatures can predict with an average error of 0.6 °C and 0.7% in the tile flow rate.

5. *Studies About Marconi-A2 Datacenter:* Considering the Marconi-A2 datacenter (which is the target datacenter in this study for experimental results): This study [1] characterizes the Marconi-A2 thermal distribution by considering the node's inlet, outlet temperatures, fan speed, and power consumption by the compute nodes.

The inlet temperature of the compute nodes is a complex combination of the ambient temperature (since the chillers are in the outdoor environments and due to the utilization of direct free cooling, which recirculates the outside cold air in the datacenter) and outlet air/water temperature of cooling systems and feedback from nodes power consumption (due to the recirculation of the hot air and cold air in the datacenter). The outlet temperature of the compute nodes, is a combination of the inlet temperature, and the dissipated heat by the compute elements of the nodes. The power consumption of compute nodes, is the main source of the generated heat in the datacenter.

Analysis in [1] shows that the Marconi-A2 compute nodes have a heterogeneous power and thermal map. The inlet temperature of the nodes increases vertically with an average difference of 6.5°C between the top and bottom nodes. This can be more severe in racks located in the center of the supercomputer room (15°C). Moreover, the bottom nodes face a higher variability of the inlet temperature than the top nodes in the rack. The inlet temperature significantly changes in the horizontal section plane. This study [1] measured up to 11°C difference for the monthly average compute nodes temperature for nodes at the same height in the racks. Interestingly the monthly average hotspot position in the horizontal section plane is correlated with the height. Measured data confirm that fans of bottom compute nodes work with a lower speed (RPM) and consume 15.8 Watts less (~ 6%) than top compute nodes. Therefore *due to the heterogeneity of different parameters of the datacenter, keeping the spatio-temporal information/relation of monitoring signals inside data structures is essential.*

6. *Thermal Storage System:* A completely orthogonal approach is to use a thermal storage system with chilled water tanks to overcome system failure while the nodes keep producing with Uninterruptible Power Supply (UPS) during thermal hazards [31].

In summary, simulators, thermal data collection with sensors, and mathematical-based or ML-based predictive models are utilized to (1) design improvements [21, 22, 23, 24, 25, 22, 2, 1] or, (2) thermal aware task scheduling to reduce the temperature and improve power efficiency [26, 24] or, (3) Thermal anomaly prediction to take action before the disaster [27, 28, 29] (as mentioned, perdition of the 18% of anomalies just 12 minutes in advance is not sufficient, this study used a very simplistic approach for the definition of the anomaly; moreover, the ML model has not enough complexity to capture spatio-temporal characteristics of monitoring signals [27]), or (4) online prediction of temperature for the efficient control of the cooling system [30]. To the best of our knowledge, no one has leveraged the large data available from holistic monitoring systems to study the statistical thermal hazard distribution nor proposed a data-driven Big Data (BD) and DL model for predicting them.

3. Background Setups

CINECA is a non-profit consortium of 69 Italian universities, 27 national public research centers, the Italian Ministry of Universities and Research (MUR), and the Italian Ministry of Education (MI) [32]. It is a national supercomputing centre for scientific research in Italy and one of the few Tier-0 supercomputing centers worldwide. CINECA hosts different supercomputers. This study focuses on the Marconi-A2 supercomputer and the datacenter hosting it. Marconi-A2 was in operation from January 2017 to January 2020 and ranked 19th (list of 2019) and 22nd (list of 2020) in the Top500 list, which ranks the most powerful supercomputers worldwide [33].

Figure 1 shows the Marconi-A2 datacenter. From the figure, we can recall three main components, the ICT elements (Racks/Compute nodes) arranged in hot/cold aisles, the air cooling circuit (raised floor and Computer Room Air Conditioning (CRAC)), and the liquid cooling circuit (Rear Door Heat eXchangers (RDHX), pump and chiller).

Marconi-A2 comprises 3312 nodes, each with one 68-cores Intel Xeon Phi 7250 CPU Knights Landing running at 1.4 GHz. Nodes has a 16 GB/node MCDRAM and a 96 GB/node DDR4. The internal network is Intel OmniPath Architecture 2:1. The supercomputer's peak performance is 11 PFlop/s [34]. The Marconi-A2 datacenter hosts 46 racks, plus 1 rack of switches, arranged in 3 rows; each rack has 18 stacked chassis, each with 4 nodes, totaling 3312 compute nodes.

Marconi-A2 datacenter is hybrid cooled, which combines (i) Computer Room Air Conditioning (CRAC) units by the Direct Expansion (DX) Air-conditioning system and (ii) a water cooling system based on Rear Door Heat eXchangers (RDHX).

In DX Air-conditioning, the air used for cooling the datacenter is directly passed over the cooling coil. There are six CRAC units in the room, and four of these CRAC units support the Direct Free Cooling (DFC) system, which is referred to by the CRAC+DFC in this study. The DFC system is designed to reduce energy dissipation and improve the carbon footprint by utilizing external cold air for cooling the datacenter. In this case, the DFC system starts to work when the outdoor temperature is lower than 18^oC. Without the DFC system, the CRAC units work in standard air recirculation mode with refrigeration-based cooling. Empowering the CRAC units with a DFC system can reduce the compressor's operation and reduce energy consumption.

The water cooling system features a chiller loop (cold loop) with water temperature from $12°C$ to $17°C$, and RDHX loop (hot loop) temperature from $23°C$ to $30°C$. The RDHX device is placed in front of the hot outlet airflow of the compute node. During operation, the compute node's hot airflow is forced through the RDHX device by the compute node fans and exchanges heat from the hot air into the circulating water from the chiller. Thus, the compute node outlet air temperature reduces before being released into the datacenter. RDHX is used to augment the computing density in an air-cooled computing room.

The hot/cold aisle approach is employed to cool the datacenter. Six CRAC units support two cold aisles. The cold airflow moves under the raised floor and gets to the loaded areas; then, the hot air returns to the CRAC units above the raised floor. All racks are equipped with RDHX, and RDHX of racks are in the hot aisle.

3.1. Monitoring System

The CINECA datacenter features a holistic monitoring framework, namely ExaMon, which aggregates a wide set of telemetry data [35]. ExaMon is one of the SoA datacenter monitoring systems [36]. For each node and its associated components, such as voltage regulators and fans, the Intelligent Platform Management Interface (IPMI) provides remote telemetry access to the built-in sensors [37]. The ExaMon monitoring system collects sensor data with the IPMI interface with 20 seconds sampling rate, and it stores these data in its internal KairosDB database as time traces and remotely accessible through RESTfull APIs [35]. These are the low-level components having the task of reading the data from several sensors scattered across the system and delivering them, in a standardized format, to the upper layer of the stack. These software components are composed of two main objects, the MQTT API and the Sensor API object.

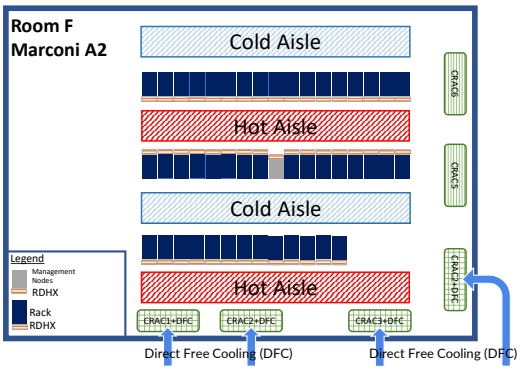
We used the data collected by ExaMon from the monitored signals of the Marconi-A2 compute nodes for the entire year 2019 in our study. We focus this work solely on node's level sensor data (IPMI data) as they provide a detailed spatio-temporal sampling of the datacenter temperature and power evolution. In a datacenter, also facility data (which includes power switchboard) are usually collected but with different granularity and different tools. Future works will consider these sources as well.

3.2. Reported Thermal Hazard

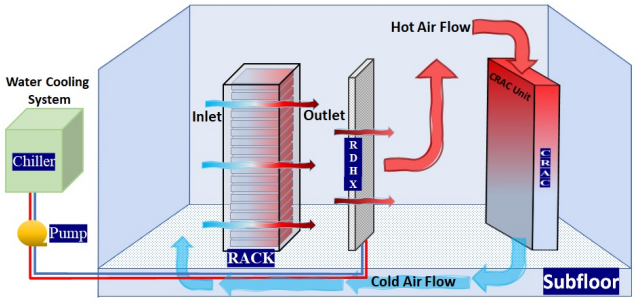
Based on the CINECA datacenter facility manager reports to the users, in 2019, the Marconi-A2 supercomputer faced a series of thermal-hazard events. A detailed study can be found in [1]. Based on that study, the Marconi-A2 had two known *reported-thermal-hazard* events in 2019: one on 28th June (peak from 16:00 to 19:00), and one on 1st July (peak from 14:30 to 17:00). We will refer in the remainder of the paper to these two recorded system failures as *reported-thermal-hazard*.

Figure 2 shows the total power consumption of the compute nodes in the reported thermal hazard period. The blue line shows the power consumption of all supercomputer nodes, and the horizontal green dashed line shows the average power consumption of all supercomputer nodes during 2019. We highlighted in red a measured abnormal supercomputer power consumption which significantly differs from the normal power profile. Given that, in general, the total power consumption of the compute nodes can represent the compute load of the supercomputer, on average, during the 4 days around thermal hazards (from 2019-06-27 to 2019-07-01), the supercomputer's compute capacity decreased by 20% due to the thermal hazard.

As introduced early in this manuscript, we aim to design a thermal hazard prediction framework. A key question is — how soon the thermal hazard should be predicted to have practical value? Which one is the target Prediction Horizon? To answer this question, we carried out an interview with the facility manager. The indication was that a 6-hour *Prediction Horizon* (PH) would provide enough time for the facility manager to take corrective action since, in the target datacenter, an operator reviews the facility settings every 6 hours. The *Prediction Horizon* (PH) is defined as the label's time distance since the last input data. For instance, if the input is the temperature of time window 00:00:00 to 05:59:59 and PH = 6 hours, the task is to predict the state of the time interval 06:00:00 to 11:59:59.



(a) Racks Arrangements of Marcon-A2.



(b) Schematic of the Datacenter's Cooling Facilities and a Rack.

Figure 1: Datacenter Computing Room.

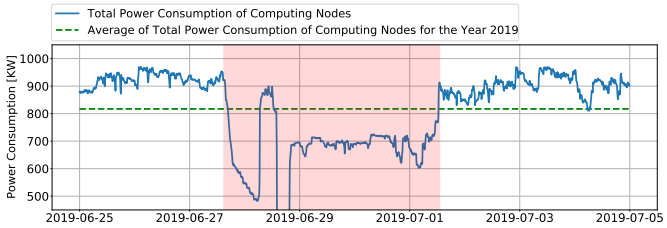


Figure 2: Total Power Consumption of Compute Nodes Around the Thermal Hazards.

4. Methodology

This section describes the proposed methodology: *(i)* First, we analyze thermal hazards periods in Subsection 4.1 to identify relevant patterns and characteristics. Based on this study, *(ii)* we then define a rule-based statistical method for binary classifying the monitoring signals collected from the nodes' sensors into datacenter-level thermal hazards in Subsection 4.2. We then apply this method to generate ground-truth labels for the entire study period of 2019. *(iii)* Next, we propose a framework for thermal hazard prediction in Subsection 4.5. The proposed framework handles the imbalanced datasets — which is typical when anomalies are scarce Subsection 4.3. *(iv)* We compare different Machine Learning/Deep Learning models for thermal hazard prediction, which serve as the framework's "brain." We also introduce different data structures to preserve the spatio-temporal information of the monitoring signals in the datacenter. As previously mentioned, the thermal and power spatio-temporal heterogeneity of the compute nodes in the datacenter creates a complex 4D dataset [1, 2]. The chosen data structure is a crucial hyperparameter that affects the size and architecture of the ML/DL model, which is discussed in Subsection 4.6. *(v)* Finally, we discuss model performance evaluation methods in Subsection 4.7 and input feature selection in Subsection 4.8.

4.1. Thermal Hazard Analysis

We started our thermal hazards analysis by studying the distribution of temperatures of the nodes during the two reported thermal hazards (on June 28th from 16:00 to 19:00 and on July 1st from 14:30 to 17:00) and comparing it against the temperature distribution of non-thermal-hazard periods. As a non-hazard distribution, we use the temperatures of the nodes during the months of June and July. We choose a large period to minimize the impact of outliers in the temperature distribution (with more than 88K samples) and periods with similar environmental conditions.

As reported in Section 2, the metrics used to capture the temperature and heat generated by the compute nodes in the datacenter are the compute node's inlet and outlet temperatures [1, 2]. Therefore, we have chosen these two metrics for conducting our initial analysis.

Figure 3 reports the inlet (top) and outlet (bottom) temperature distributions for one node for the three cases: non-hazard, hazard on 28th June, and hazard on 1st July. By comparing the two figures, we can notice that the inlet temperature non-hazard and hazards distributions are distinguishable, while this is not the case for the outlet temperature distributions[1]. We can thus conclude that inlet temperature can be used to isolate hazard nodes' temperatures from non-hazard ones. This finding may seem counterintuitive, but it is not — today's compute servers are thermally regulated, filtering out the effect of cooling shortages on the outlet temperature. The inlet one is instead directly linked to the room temperature and the cooling system.

Figure 3-top reports with a dashed black line the quantile 0.95 of the node's non-hazard distribution. As it is visible from the figure, this quantile (0.95) can be used as a threshold to separate the non-hazard and hazard node temperatures.

Based on this analysis, we concluded that the quantile 0.95 of the inlet temperature of each compute node during non-hazard periods is a good parameter to discriminate between hazard and non-hazard.

4.2. Thermal (Hazard) Anomaly Labeling Method

This section extends Section 4.1 by introducing a rule-based statistical method (referred to as $\mathcal{G}$) for generating labels for

[1] We evaluated this property also for other randomly chosen nodes.

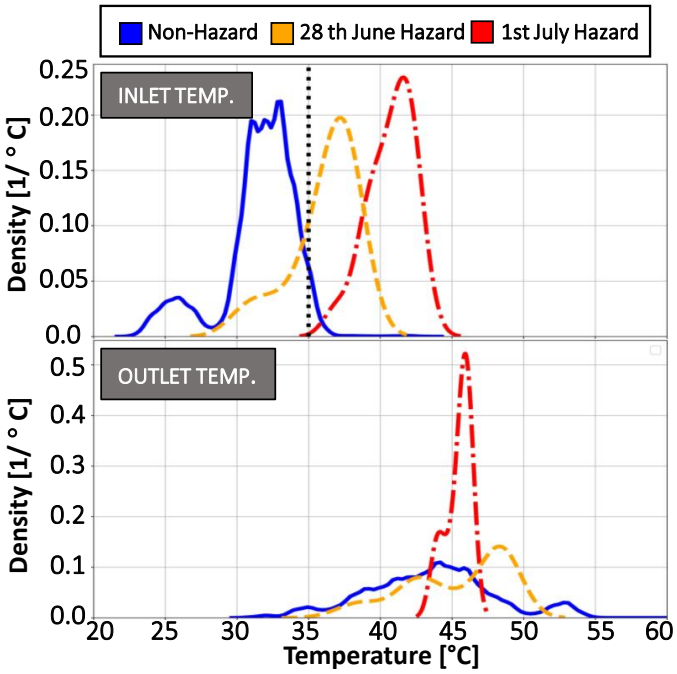


Figure 3: Temperature Distributions for Marconi-A2's One Compute Node in June-July 2019 [20].

Figure 4: Time Windowing and Labeling [20].

	Time	Node_1	...	Node_3311	Node_3312
6 HOURS	2019-01-25 00:00:00	30°C	...	41°C	42°C
	...	...	...	...	...
	2019-01-25 05:58:00	29°C	...	39°C	40°C
	2019-01-25 05:59:00	28°C	...	39°C	40°C

(a) Inlet Temperature dataset

	Time	Node_1	...	Node_3311	Node_3312
6 HOURS	2019-01-25 00:00:00	FALSE	...	TRUE	TRUE
	...	...	...	...	...
	2019-01-25 05:58:00	FALSE	...	FALSE	FALSE
	2019-01-25 05:59:00	FALSE	...	FALSE	FALSE

(b) True-False table

thermal anomalies (hazards). The method is based on the statistical analysis of two reported thermal hazards. The tool is used to generate ground-truth labels (referred to as $\mathcal{L}$)) for the datacenter during the studied period. $\mathcal{L} = \mathcal{G}(\mathcal{N})$, $\mathcal{G}$ maps the monitored inlet temperature of all 3312 compute nodes $\mathcal{N}$ to thermal hazard labels $\mathcal{L}$.

4.2.1. Node-threshold (NT)

Based on the characterization of thermal hazards described in Section 4.1, we introduce the *node-threshold* to assign a binary thermal hazard label for a specific compute node and timestamp. (if a node features a thermal hazard? True: If a node in a timestamp experiences an inlet temperature greater than the node threshold, False: otherwise) We defined the *node-threshold* individually for each compute node as the 0.95 quantile of its inlet temperature distribution over the entire dataset (covering the whole of 2019). Therefore, different nodes can have different *node-threshold*. Figure 4a summarizes a 6-hour time window (TW) of the inlet temperature dataset. We applied the *node-threshold* to assign to each (node, time) cell a `True`/`False` label indicating sample-by-sample thermal trouble, as shown in Figure 4b. We chose TW = 6 hours which is equal to the prediction horizon.

4.2.2. Spatio-temporal-impact-threshold (STIT)

To assign hazard/non-hazard labels to Time Windows (TW)s (Figure 4a), we introduce a *spatio-temporal-impact-threshold* that takes into account the spatial and temporal continuity of thermal hazards. A TW with 3312 nodes and a duration of 6 hours, with a sample rate of 1 sample per minute, would have a total of 1192320 true/false values (as shown in Figure 4b). The *spatio-temporal-impact-threshold* determines the minimum percentage of "true" values required within the TW to classify the datacenter as being in a thermal hazard. A higher threshold will result in the selection of thermal hazards that are more widespread, i.e. that involve more nodes for a longer period of time. The *spatio-temporal-impact-threshold* is a general answer to the following question. How much thermal hazard spread in time and different nodes in the datacenter (in a TW)? The final setup of this threshold will be discussed in section 5.1.

4.3. Imbalanced Dataset

In general, anomalies are rare events, making thermal hazards a minority class within the dataset. The dataset exhibits a significant disparity in the number of instances between the minority and majority classes. Due to the imbalanced nature of the dataset, the trained model will be biased towards the class that is overrepresented in the dataset. Several strategies have been developed to address the challenges posed by imbalanced datasets in machine learning training. These strategies can be categorized into three main groups: *(i) Resampling Techniques*, *(ii) Generating Synthetic Samples*, and *(iii) Cost-sensitive Learning*.

(i) Resampling Techniques: In this approach, with upsampling, which increases the number of instances in the minority class by randomly duplicating them, and downsampling, which decreases the number of instances in the majority class by randomly removing instances, we create a balanced dataset. This can be achieved using a weighted random sampler method, which involves assigning weights to items of different classes based on their class distribution. Then, samples can be selected through random sampling, where the probability of selecting an item is proportional to its weight [38, 39].

(ii) Generating Synthetic Samples: It generates synthetic samples for the minority class by interpolating between existing minority class samples. We used SMOTE techniques SMOTE, an acronym for Synthetic Minority Oversampling Technique, is a strategic oversampling methodology designed to rectify class distribution disparities within datasets characterized by imbalanced classes. SMOTE can mitigate this imbalanced classes challenge by generating novel instances for the minority class. Instead of straightforwardly replicating existing instances, SMOTE employs a sophisticated approach wherein synthetic instances are produced by interpolating between extant minority class data points within the feature space. This augmentation process serves as a powerful tool for enhancing the performance of machine learning models when confronted with imbalanced datasets [40].

(iii) Cost-sensitive Learning or Weighted Loss Function:

Modify the loss function [2] to penalize misclassifying the minority class more than the majority class.

4.4. Regularization Techniques

To prevent overfitting and improve the generalization of the model, we used regularization techniques such as dropout and L2 regularization. During training, dropout randomly ignores or "drops out" a certain number of layer outputs. This process approximates training multiple neural networks with different architectures in parallel. By implementing dropout, the network is forced to learn more robust features that are useful in conjunction with various random subsets of other neurons. Dropout is a simple yet effective regularization method that reduces overfitting and enhances the performance of deep neural networks. L2 regularization is a technique used in machine learning to prevent overfitting. It involves adding a penalty term to the loss function, which encourages the model to have smaller weights. In other words, it helps prevent overfitting by penalizing complex models.

4.5. Thermal Hazard Prediction Framework

We propose a framework for predicting thermal hazard, which encompasses data query and preprocessing, model training, and final model inference, which provides the prediction. The thermal hazard predictor is a model that, using time series data of compute nodes' sensors, predicts if a thermal hazard will happen in the datacenter within the next hours. The input data are the time series of nodes' temperature (and power consumption), and the output is a binary classification: likely forthcoming hazard or not. As mentioned, Prediction Horizon = 6 hours was chosen after consulting with the facility manager.

Figure 5 illustrates the architecture of our proposed thermal hazard predictor, which is composed of three main components: the data collection, and storage architecture based on ExaMon (described in Subsection 3.1), the thermal hazard analysis including the data extraction, preprocessing (e.g., missed data handling, time alignments), label generator and data loader, and the Artificial Intelligence (AI)-powered thermal hazard prediction system (training and inference). Different classical ML and DL tools are candidates for operation as the Al model.

The AI model's input is a TW of data extracted from the database. In the off-line training stage, a large set of TWs is extracted (training dataset), and preprocessed to generate the ground-truth labels with the rule-based statistical approach of section 4.2. Inferences with the trained model are the predictions of thermal hazards.

4.6. Machine Learning Tools

To determine the most appropriate ML/DL model for the thermal hazard prediction framework, we evaluated different classical ML and DL tools in predicting thermal hazards in CINECA's Marconi-A2 system. To keep the model's size and training time in the acceptable range (tens of thousands, rather than tens of millions), in some experiments, we use a subset of the compute nodes monitoring data as input features for the ML/DL models.

4.6.1. Classical ML-Learning Tools

0) Last Value Predictor (LVP): a minimum baseline for any time-series task; the prediction $\hat{y}$ is simply a copy of the present observation y_{true}: $\hat{y}(t + 6\text{H}) = y_{\text{true}}(t)$, with 6 hours (6H) prediction horizon as stated in Subsection 3.2.
1. *Support Vector Machine (SVM):* SVM with either linear or Radial Basis Function (RBF) kernels. SVMs produce decision boundaries with margins to improve generalization.
2. *Stochastic Gradient Descent (SGD)-classifier:* linear SVM trained with SGD instead of convex optimization, enabling larger train set size.

SVMs and SGD-classifier were implemented in Scikit-learn 0.23.

4.6.2. Deep Learning Methods

1. *Long Short-Term Memory (LSTM)*: a type of Recurrent Neural Network (RNN) that can learn long-term dependencies. Our LSTM has 2 layers of hidden and output size 16, followed by a dense layer. The LSTM model was implemented in Keras 2.4.
2. *Temporal Convolutional Network (TCN):* The common TCN is a sequence to a sequence modeling tool. However, our framework requires a classification tool, so we modified the TCN to suit our needs by adding a classification block at the end of the model. Figure 6a depicts the proposed TCN model. We propose the different architectures of the TCN model by modifying the convolutional layers and adjusting the input data structure. While keeping the model's size appropriate, using complex models will allow for expanding the number of input features. Additionally, a more complex architecture is expected to better learn the complex spatio-temporal relations of the monitoring signals. The input data structure also plays a crucial role in maintaining the spatio-temporal relations of the monitoring signals. By using complex TCN architectures, it becomes possible to use more efficient input data structures that still retain the spatio-temporal information of the monitoring signals.

The training configuration of the TCN model is presented in Table 1.

Table 1: Training Configuration of the TCN Model.

Hyperparameter	Conf.
`loss_function`	CrossEntropyLoss
`optimizer`	SGD
`optimizer_step_size`	20
`learning_rate`	0.1
`learning_rate_decay`	0.5

- *TCN with 1D Conv. Layers:* The proposed TCN has two blocks: (a) a *Feature Learning Block* of seven 1D convolutional layers with average pooling (14k parameters, given that we used 72 compute nodes monitoring data as input features of the model); (b) a *Classification Block* of 4 dense layers of 15, 6, 4, 3, 2 units. All layers present batch normalization and ReLU activation. It is designed utilizing 1D convolutional (1DConv)

[2] A loss function (also known as a cost function) is used to measure the difference between the predicted output of a model and the actual output. The goal of training a neural network is to minimize the loss function.

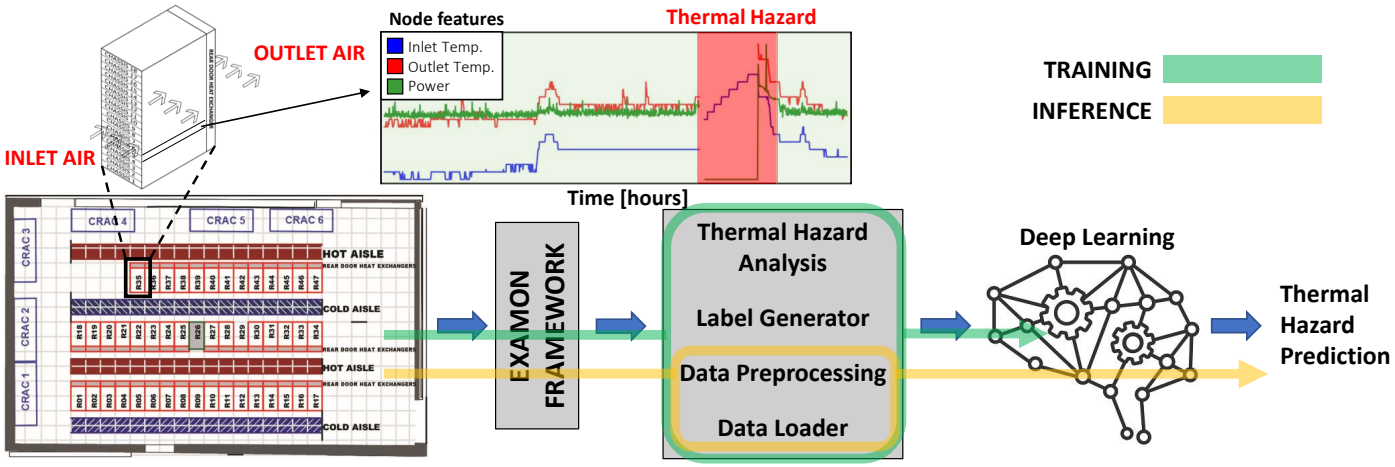


Figure 5: HazardNet: a Framework for Thermal Hazard Prediction [20].

layers, and Figure 6a, and Figure 6b show this TCN model and its data structure. The first layer of the model has N input channels, one channel for each node. Moreover, we used one dimension of the 1DConv layers for the time dimension of monitoring signals. This data structure considerably destroys the spatial information of monitoring signals.

- *TCN with 2D Conv. Layers:* We modified the TCN (Figure 6a) model by using 2DConv (2D Convolutional) layers instead of 1DConv layers. The input data structure is depicted in Figure 6c, which is a 2D array. The first layer of the model has one input channel, and from two dimensions of the 2DConv layers, we used one dimension for time and the second for the sensors without considering their location (x,y,z). This data structure considerably destroys the spatial information of monitoring signals. The size of all models are reported in Table 4.

Power Consumption as a Second Input Channel of TCN with 2DConv Layers: The power consumption of the compute nodes is the primary source of heat generation in the datacenter, so we study the TCN model's performance by adding the power consumption as a new channel of the input layer of the TCN model. This TCN model (Figure 6a) employs the 2DConv layers, and its input layer contains 2 channels, one used for nodes' inlet temperature and the other for power consumption. Its 3D data structure (or two arrays of 2D data structure) is depicted in Figure 6d.

- *TCN with 3D Conv. Layers:* We created a TCN model (Figure 6a) with 3DConv layers, and the input data structure is modified to 4D (x,y,z,t) to fit this new model, as depicted in Figure 6e. In this TCN model with 3DConv layers, we used 3 dimensions of the model's input to specify 3 axes of nodes' locations in the datacenter (nodes x, y, z-axis). So we created the input data structure considering the location of the nodes which provided monitoring data. For the time dimension of data, we used the input channels of the first layer, i.e., since the data is time-series data, each input sample of the TCN model is a sequence of the data, and for each element of the sequence, one input channel is used (e.g., with a sampling rate of 10 minutes, for Time Window of six hours (TW=6 Hours), 36 input channels are utilized). This new architecture of the TCN model allows for increasing the input features of the model while keeping the model's size acceptable range (tens of thousands and NOT tens of millions).

Outlet Temperature of Nodes Interleaved to Inlet Dataset: In addition to the inlet sensors of the nodes, which measure the inlet air temperature, each node also has a sensor that measures the outlet air temperature. In the model previously described, the inlet temperatures of the nodes are used as input for the model. In this method, we augmented the outlet temperature to the input data structure, i.e., the outlet temperature of nodes is interleaved with the inlet temperature dataset.

As mentioned in Section 2, the inlet temperature of the compute nodes is a complex combination of the ambient temperature, outlet air/water temperature of cooling systems, and feedback from nodes' power consumption. On the other hand, the outlet temperature of the compute nodes is a result of the inlet temperature, and the heat dissipation by the computing elements of the nodes [1].

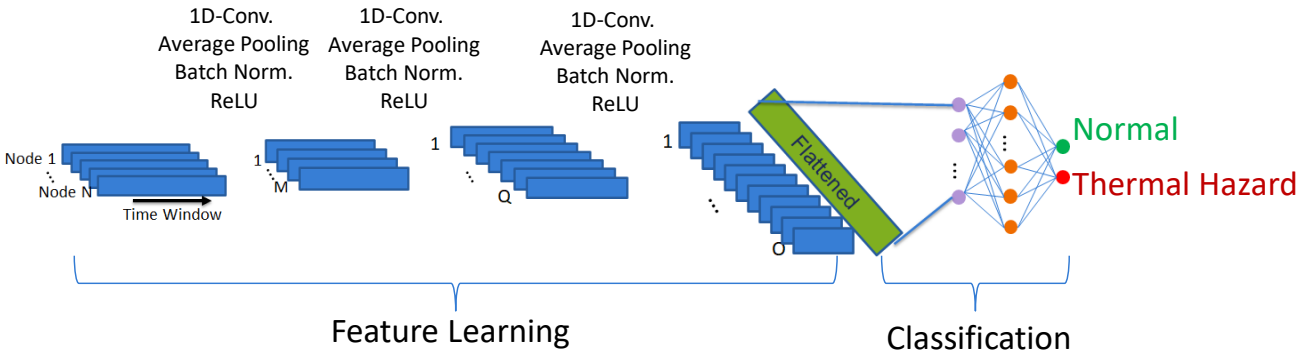
Inlet Dataset Augmented with Power Consumption of Nodes: Furthermore, we investigated the impact of including the power consumption of the nodes, in addition to the inlet temperature, as inputs to the model. This factor was examined because it represents the main source of heat within the node.

- *Depthwise Separable Convolutions:* Instead of using normal convolutional layers in the TCN model, we used depthwise separable convolution layers (PyTorch [38] implementation). These layers reduce the model's size (number of trainable parameters) and computation. The TCN models were implemented in PyTorch 1.5.

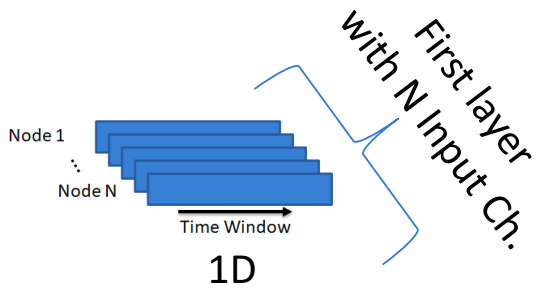
4.7. Performance Evaluation of the AI Models

We evaluated the prediction performance of different AI models using two different test approaches. The main distinction between these two approaches is the selection method for the test and training datasets.

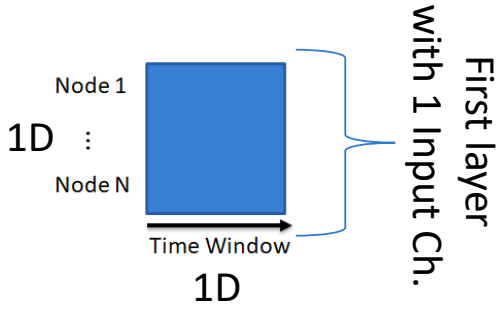
1. *Random Test Dataset:* In this approach, we randomly selected 20% of the 2019 data as the test dataset, and trained the models on the remaining 80%. However, we find two concerns about this approach: *i)* There is much overlap between each successive sample, meaning that each consecutive sample has



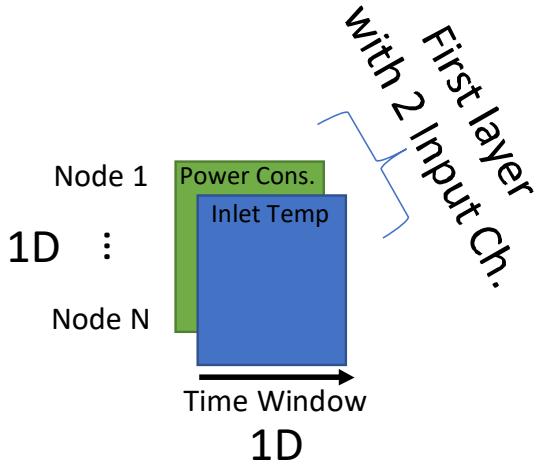
(a) TCN Model with 1DConv. Layers.



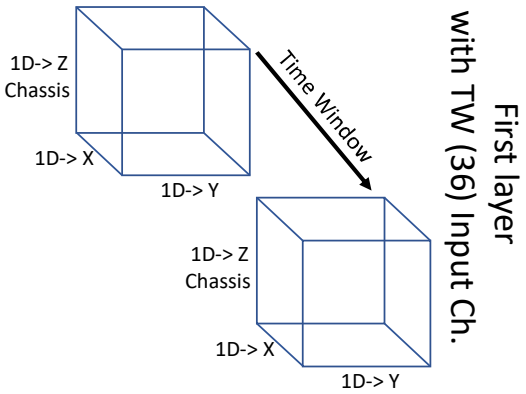
(b) Input Data Structure of the TCN Model with 1DConv. Layers.



(c) Input Data Structure of the TCN Model with 2DConv. Layers.



(d) Input Data Structure of the TCN Model with 2DConv. Layers Power Consumption as a Second Input Channel.



(e) Input Data Structure of the TCN Model with 3DConv. Layers.

Figure 6: TCN Model's Architecture and Input Data Structures for Different Types of Convolutional Layers (1DConv., 2DConv., and 3DConv.).

a lot of replicated data. As a result, if one of the two consecutive samples is in the training dataset and the other in the test dataset, due to the high overlap of the two samples, the model is indirectly trained by the test sample. *ii)* Random selection training and test datasets destroy the chronological order of the training and test samples, which is important for time-series data because it destroys the causality of the data. i.e., in the test dataset, some samples are chronologically before the training samples. However, in the objective case implementation, the model is trained with past data to predict the future.

2. *Time-separate Test Dataset:* To address the issues of the first test approach, in this approach, we simulate a real-case scenario by training the model with data from May 2019 and testing the model in the first week of June 2019.

We should highlight that using a random selection approach for validating machine learning models, even with time series datasets, is a widely accepted practice in the field. In our study, the dataset was partitioned into sequences of 6-hour periods, where each sample represents a distinct and individual data point. Although there might be similarities between consecutive samples, they are inherently unique. Utilizing random selection for test samples offers several advantages. It enables a comprehensive evaluation of the performance of the model (LVP, SVM, RBF-SVM, SGD-classifier, LSTM, and TCN) across the entire dataset. This approach captures the properties and characteristics of the dataset more effectively than using a time-separated approach. By doing so, it ensures a robust assessment of the model's generalization capabilities, particularly in scenarios where the data distribution may vary over time. We conducted tests using the random test dataset selection approach. However, we were aware of the technical challenges associated with this approach during the implementation in a real system. The actual in-production system should train the model using historical data and utilize it to make predictions for the future. The second test approach offers a perspective on the performance of a model trained on a small portion (1/12) of the dataset, but in a more realistic scenario. By using these two test approaches in conjunction (as we will discuss in Section 5), we can obtain a comprehensive evaluation of both the ML/DL tool selection and the overall performance of the framework.

4.8. Input Features Selection

For most of the ML/DL models, the training time and/or models' size (number of trainable parameters) will be substantial if all compute nodes' monitoring data (3312 compute nodes) are used as model input. For example, the TCN model, which uses 1D convolutional layers, will have several million parameters. Therefore, we studied the model's performance using a subset of the compute nodes. Selecting a subset of the compute nodes can be done using different approaches, such as:

(i) Selecting compute nodes of one random rack or a rack located in the center of the datacenter as indicative of the datacenter. One rack contains 18 chassis while each chassis encloses 4 compute nodes, so a rack hosts 72 compute nodes (${}^{72}\nu$ shows a subset of the compute nodes with 72 nodes). (ii) Selecting compute nodes from 24 racks in the datacenter, three compute nodes from each rack, one from chassis 1(bottom), chassis 9(center), and chassis 18(top). (iii) Selecting compute nodes whose inlet temperatures have the highest correlation with the datacenter's thermal hazard ground truth labels. (iv) Dividing the compute nodes of the datacenter based on the height of chassis/nodes into three groups: bottom(chassis 1 to 6), center(chassis 7 to 12), and top(chassis 13 to 18), and from inside each group, randomly selecting nodes. (v) Completely random selecting of a subset of compute nodes. etc.

4.8.1. Most Informative Subset of Nodes

In input feature selection, it is essential to select the *most informative subset of nodes* when using monitoring data from a subset of compute nodes instead of all cluster nodes.

We defined it as a subset of nodes that creates labels similar to the datacenter's original ground truth labels, which are generated utilizing all 3312 nodes. $\Omega = \{{}^{72}\nu_i | {}^{72}\nu_i \subset \mathcal{N}\}$, ${}^{72}\nu_i$ is the ith subset of $\mathcal{N}$ compute nodes with a cardinality of 72, and Ω is a set that composes of all subsets of compute nodes with a cardinality of 72. The *most informative subset of nodes* is the subset ${}^{72}\hat{\nu}$ which maximizes the F1-score metric in the following Equation 1. As mentioned in Section 4.2, the $\mathcal{G}$ is a rule-based statistical method that maps the monitoring data of nodes to the thermal hazard label.

$$ {}^{72}\hat{\nu} = \arg\max_{{}^{72}\nu_i \in \Omega} \mathcal{F}1\text{-}score\Big(\mathcal{G}(\mathcal{N}), \mathcal{G}({}^{72}\nu_i)\Big) \tag{1} $$

5. Experimental Results

In this section, we report the summary of the experimental results of this research activity on the Marconi-A2.

5.1. Validation of Datacenter Thermal Anomaly (Hazard) Labeling Method

It is essential to note that although this statistical labeling approach is based on real information extracted from the reported thermal hazard distribution, this statistical labeling approach is artificial and must be confirmed by comparing it with the reported thermal hazards. As made evident in Figure 7 (x-axis is the date), if we set the *spatio-temporal-impact-threshold* to 5%, our statistical approach captures the reported thermal hazards while detecting additional thermal hazards, which were unnoticed by the system administrators. These are conditions in which the compute nodes' temperatures have drastically increased without causing immediate damage but still potentially damaging the nodes. Our statistical labeling approach can capture these events which are unnoticed by humans.

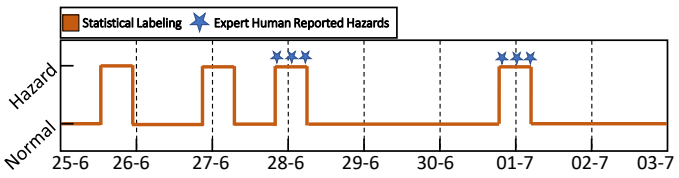


Figure 7: Thermal Hazards [20]

If we increase the *spatio-temporal-impact-threshold* (*STIT*) quorum to 25%, the statistical labeling approach could only detect the second hazard, thus making it too restrictive in identifying abnormal states. For the selected *STIT* of 5%, the datacenter is labeled as being in thermal hazard for 19.5% of the time in 2019. When we raise the threshold to 15%, the thermal hazard category reduces to 3.8%, while still detecting both hazards. This quantifies the rarity of extensive thermal hazards compared to narrow ones. Both of these thresholds can correctly capture real thermal hazards, but with different levels of sensitivity. Since, in the production scenario, there will be an operator that will react to the alarm, we prefer to train the model to be skewed toward higher sensitivity (5% threshold). Still, the operator can change this threshold with site conditions.

Table 2: Thermal Hazard Percentage.

	Spatio-Temporal-Impact-Threshold		
	5%	**10%**	**15%**
Node-threshold 95%	19.5%	8.0%	3.8%

5.2. Dataset

Considering the thermal and power characteristics of the datacenter [1], in this study, we conduct experiments utilizing the inlet, outlet temperature, and power consumption time series data of computing nodes in the Marconi-A2 datacenter. It hosts 46 racks, each rack containing 18 chassis, and each chassis include 4 nodes, so in total sensory data of 3312 nodes for entier 2019 (the year we have the monitoring data in ExaMon database) being used in this study. The datacenter experienced two thermal hazards on June 28th and July 1st. We collected the original data with a sampling rate of 20 seconds (in ExaMon), and then we downsampled it to 1 minute in the preprocessing step. We generated the ground-truth labels with the rule-based statistical method described in Section 4.2, with *node-threshold* = 0.95 and *spatial-temporal-impact-threshold* = 0.05 as motivated. With these values, 19.5% of the data is labeled as a thermal hazard, which is sufficient for training our algorithms.

5.3. Results of Input Features Selection

In Table 3, we report the results of our experiments to find the best method to select the most informative subset of nodes. We investigated different methods, such as selecting nodes with the highest correlation with the datacenter's label, dividing the nodes into three groups (bottom, center, and top), and randomly selecting nodes from each group, etc. Considering the F1-score, we found that given the most informative subset of the nodes definition, the completely random selection of nodes with an F1-score of 0.91 is the best input feature selection method. Since it collects the different nodes' monitoring data spread in different locations of the room, this random selection creates a better representative of the entire cluster than nodes of just one random rack.

5.4. Baseline (LVP):

By treating the prediction horizon, PH = 6 hours as a time lag, the hazard/non-hazard binary ground-truth labels have autocorrelation 0.65 over the year 2019 and identifying ground-truth labels 6 hours apart as the output and target of a Last-Value Predictor (LVP) yield an F1-score of 0.72. Being the LVP, the simplest (non-)model, F1-score = 0.72 is a baseline any proposed model must be compared against.

5.5. Random Test Dataset

In this set of experiments (Exp.1 to Exp.6 in Table 4), we used the random test dataset method (20% of the dataset for testing, remaining 80% for training the model), as stated in Section 4.7. For the input feature of the models, we used the monitoring data of 72 nodes of a central rack to control the size of the ML/DL models. It is worth noting that the ground truth labels were generated using the inlet temperature of all nodes of the Marconi-A2 supercomputer.

This test approach evaluates the overall performance of the models throughout the study period. We used it to find the best ML/DL tools to serve as the AI brain of HazardNet. Table 4 reports the results of these experiments. The linear SVM yields an F1-score of 0.55, which is essentially random and worse than the LVP-baseline. This is due to the linear models' poorness and to the training dataset reduction made necessary by computational complexity. The RBF ranks better, with an F1-score of 0.86, which is also 0.17 above the SGD-classifier. Both DL models outperform the non-deep ones: the LSTM reaches an F1-score of 0.91, and our TCN ranks best, with an F1-score of 0.98. Therefore, we have empirically shown that DL models work better than classical machine learning tools in the thermal hazard prediction framework. Among DL models, the TCN outperforms the LSTM model; therefore, we selected the TCN for continuing the study. For this experiment, the TCN model (Figure 6a) employed 1D Convolutional layers, and the input data structure is depicted in figure 6b.

5.6. Time-separate Test Dataset

In this set of experiments, we used the time-separate method for selecting the test and training dataset, as stated in Subsection 4.7. We trained models using monitoring data from May 2019, and conducted the test on data from the first week of June 2019. This testing approach allows us to measure the model's performance in a realistic scenario, where training and testing are performed on limited portions of the dataset.

We evaluated various models with sizes ranging from a thousand to a few million parameters, using different architectures. The prediction results of models are reported in Table 4. Although based on the random test dataset, it has been determined that TCN outperforms other approaches. To validate the accuracy of this result on the time-separated test dataset, we also assessed the performance of SVM and LSTM models on this dataset. Notably, Table 4 does not encompass the exhaustive list of all conducted experiments. To improve conciseness and focus, we opted to discuss certain experiments in the text, highlighting the impact of specific modifications/tools/techniques

Table 3: Different Approaches for Input Features (Compute Nodes) Selection.

Computing Node Selection Approach	Number of Nodes	F1-score	Accuracy
All nodes of one random rack	72	0.85	0.91
From 24 racks, 3 nodes, one from chassis 1(bottom), chassis 9(center), and chassis 18(top)	72	0.85	0.92
Select nodes with highest correlation with datacenter's labels	72	0.77	0.89
From 36 racks, 3 nodes, one from chassis 1(bottom), chassis 9(center), and chassis 18(top)	108	0.86	0.92
Random selection of racks, node 0 from chassis 1(bottom), chassis 9(center), and chassis 18(top)	108	0.85	0.92
Divided the nodes to three groups bottom (chassis 1 to 6), center (chassis 7 to 12), and top (chassis 13 to 18), and from inside each group, randomly selected 36 nodes.	108	0.91	0.95
Completely random selection of nodes	108	0.91	0.95

on the model's performance. By adopting this approach, we aimed to keep the table 4 concise, featuring only the most crucial results for a clearer presentation.

In Exp.7, the model trained with inlet temperature data of the central rack in the datacenter and with a time-separate test dataset method achieved an F1-score of 0.74, which is 0.24 lower than with a random test dataset method. Such degradation is due to the random selection of the test dataset in Exp.6, which includes similar samples in the training and test sets. However, the test approach of Exp.7 is closer to the real usage of the predictive model.

We suspect that the limited accuracy of Exp.7 is caused by: (1) A limited number of input features (the limited set of computing nodes monitoring data considered for the prediction). (2) The architecture of the TCN model not being complex enough to extract and learn the intricate spatiotemporal relation of the input dataset. (3) Non-stationarity in the thermal effects that are not captured if we use only past data to predict the future. We summarized our discussion of the experimental results into six main groups.

5.6.1. Input Features of the TCN Model

Point (1) implies that a single central rack is not a sufficient representation of the datacenter. By reducing the number of features to monitoring data from only 72 compute nodes instead of 3312 nodes, we lose a significant amount of information about the datacenter. Besides, we should highlight that labels are generated utilizing all the inlet temperatures of 3312 nodes. Based on Table 3, which shows the different feature selection approaches, we know that random selection of the 72 computing nodes will provide the most informative subset of the nodes. So, in Exp.8, we used monitoring data of computing nodes for training that were randomly selected from the datacenter, aiming to improve the prediction performance. Randomly selecting the input features (computing nodes) leads to a slight improvement in performance. The F1-score increased from 0.74 to 0.77. However, using all 3312 nodes' monitoring data as input for the model instead of just 72 random nodes, resulting in a model that is 215 times larger, only improves the F1-score by 1% (Exp.8, compared with Exp.9).

Including the power consumption or outlet temperature of nodes, in addition to the inlet temperature of the computing nodes, as input to the model reduces the F1-score by approximately 8% to 11% [(Exp.10, compared with Exp.11), (Exp.14, compared with Exp.17), and (Exp.14, compared with Exp.15)], depending on the model architecture. This difference can be attributed to the distinct characteristics of the input data.

5.6.2. Size of TCN Model

Increasing the size of the model (trainable parameters) makes the TCN model able to receive more input data. For example, instead of considering data from just one rack (72 nodes), it can handle data from the entire cluster (3312 nodes). This leads to a modest improvement of slightly more than 1% in the F1-score of the model (Exp.8, compared to Exp.9) and (Exp.10, compared to Exp.12). However, this is not true in general as it may also cause a performance reduction (Exp.10, compared to Exp.13) of 5%. When enlarging the model size while keeping the input data constant (Exp.12, compared to Exp.13) and (Exp.14, compared to Exp.16), an approximately 6% decrease in the F1-score is observed.

5.6.3. Architecture of TCN Model and Input Data Structure

By considering the following experiments [(Exp.8, compared with Exp.10), (Exp.9, compared with Exp.12), and (Exp.12, compared with Exp.14)], we observe that the performance of the model improved when using the same monitoring data, but with a more advanced model architecture and input data structure.

From (Exp.8, compared with Exp.10), we can observe that by replacing the 1DConv layer with 2DConv layers and modifying the data structure of the input, despite having a model that is 90% smaller in terms of the number of trainable parameters, the F1-score increased from 0.77 to 0.82.

In the 2DConv architecture, the convolution operation is performed in both the temporal and spatial dimensions of the data, unlike the 1DConv. However, similar to the 1DConv, this architecture also leads to the loss of a significant amount of spatial information from the dataset. So in Exp.14 to Exp.17 we replaced the 1D or 2D Convolutional layers in the TCN model

with 3DConv layers. Additionally, we utilized a 4D data structure that includes the dimensions of x, y, z, and time. This architectural modification resulted in a remarkable enhancement in the F1-score, with a 11% increase. Furthermore, an intriguing finding was that this modification led to a significant reduction in the model size, exceeding 99%. By a matter of fact, by leveraging 3DConv layers and the 4D data structure, we were able to capture the spatial-temporal information of the monitoring signals more effectively.

5.6.4. Depthwise Separable Convolutions

In Exp.18 and Exp.19 we compare the performance of the TCN model with depthwise convolution and the typical TCN model. We create a TCN model utilizing depthwise separable 3D convolution layers, and for the input of the model, a 4D data structure is employed. Exp.18 is comparable with Exp.14 since both use only inlet temperature as input for the TCN model. However, using depthwise convolution decreased the model's performance from an F1-score of 0.87 to 0.81. Additionally, the size of the model was reduced by more than 78%, while the training time increased by 50%.

For the augmented dataset (i.e., the outlet temperature of nodes is interleaved in the inlet temperature dataset), the depthwise separable convolution method increased the F1-score from 0.8 to 0.81 (Exp.17 and Exp.19). At the same time, the model's size was reduced by 73%, and the training time increased by 33%. Therefore, depthwise separable convolution reduces the model's size and computation. Still, the training time increases in PyTorch [38] implementation due to the increased number of convolutional layers (pointwise), i.e., these layers are sequential layers, and although they reduce the number of parameters and multiplications, they increase the serial parts of the code.

5.6.5. Training Parameters (Imbalanced Dataset and Regularization Techniques)

In our use cases, utilizing a weighted loss function does not yield satisfactory results. In fact, its F1-score is 0.74, which is approximately 15% lower than the best model. However, employing the SMOTE in conjunction with dropout and L2 regularization led to a substantial improvement. This approach resulted in an F1-score of 0.86, showcasing a notable 16% increase compared to the utilization of a weighted loss function. Furthermore, an alternative technique, the weighted random sampler method, demonstrated slightly superior performance, with an F1-score of 0.87, just over 1% higher than the SMOTE approach. However, it is noteworthy that the incorporation of regularization methods in conjunction with the weighted random sampler led to a reduction in the F1-score by approximately 3%.

Given the abundance of samples in the majority class, downsampling did not result in any loss of critical information regarding the dataset's characteristics and properties. Consequently, it appears that the weighted random sampler method, utilized to address imbalanced classes, also functions as an effective regularization technique in our context.

5.6.6. SVM and LSTM Models

In addition to the TCN modes, we also evaluated the SVM and LSTM models using *Time-separate Test Datasets*. As input of these models, we used monitoring signals from all 3312 nodes. The F1-scores we obtained for the SVM and LSTM models were 0.79 and 0.81, respectively. These scores are 10% and 7% lower than the optimal TCN model.

6. Framework Portability

Our framework is intentionally designed with modularity in mind, allowing it to be adaptable to different datacenter architectures. However, we recognize that specific adjustments might be necessary when deploying it in different settings. Specifically, the following areas may require adaptation:

Middle Layer Implementation: The middle layer, facilitating the connection between the data processing framework and the monitoring system, currently utilizes a RESTful API tailored for ExaMon. Minor updates might be needed if the monitoring system of the datacenter is different.

TCN Model Adjustments: Modifications in the TCN model, particularly in the first layer, could be necessary if the composition of the datacenter is different than the one studied.

AI model deployment in production: we envision challenges in deploying the proposed AI models in a production environment, namely versioning of the models, continuous integration and deployment CI/CD. We, however, believe that this issue is common in other domains and could benefit from technologies developed in the context of MLOps. Based on our experience, the major challenge is convincing the system administrators in installing and maintaining a new tool, for this perspective, the success of introducing a new tool depends more on ease of deployment and availability of online resources more than on the tool's performance and capabilities. Based on our experience, vendors and system integrators are a better target than the system administrator for introducing new tools in production as their availability will come with the machine.

7. Conclusion and Future Work

In this paper, we proposed HazardNet, a framework for thermal hazard prediction. The thermal hazard predictor is a model that, based on time series data from monitoring sensors in computing nodes, predicts if a thermal hazard will occur in the datacenter in the following hours. Based on the reported-thermal-hazards analysis, we defined a rule-based statistical method for the thermal anomaly (hazard) labeling/detection in the Tier-0 datacenter. We studied the most contributed parameters in thermal hazard detection and prediction and showed that inlet temperature is the most capable parameter for this goal. For thermal hazard prediction, we investigated different classical machine learning and DL tools and empirically showed that the Temporal Convolutional Network (TCN) with an F1-score of 0.98 outperforms non-deep models and LSTM. We also explored different TCN architectures and showed that TCN with 3D convolutional layers has the highest prediction performance (F1-score of 0.87).

Table 4: Results of Experiments.

Random Test Dataset								
Exp. No	**Size**	**Model Architecture**	**Input**	**#Chnls**	**#Nodes**	**F1-score**	**Precision**	**Recall**
Exp.1[20]	**-**	**Last Value Predictor**	**Inlet Temp.**	**-**	**72 Nodes of One Rack**	0.72	0.72	0.72
Exp.2[20]	**< 1K**	**Linear SVM**	**Inlet Temp.**	**-**	**72 Nodes of One Rack**	0.55	0.56	0.55
Exp.3[20]	**< 1K**	**RBF-SVM**	**Inlet Temp.**	**-**	**72 Nodes of One Rack**	0.80	0.94	0.86
Exp.4[20]	**< 1K**	**SGD-classifier**	**Inlet Temp.**	**-**	**72 Nodes of One Rack**	0.64	0.76	0.69
Exp.5[20]	**8K**	**LSTM**	**Inlet Temp.**	**-**	**72 Nodes of One Rack**	0.84	0.98	0.91
Exp.6[20]	**14K**	**TCN**	**Inlet Temp.**	**72**	**72 Nodes of One Rack**	0.98	0.99	0.98
Time-separate Test Dataset								
Exp. No	**Size**	**Model Architecture**	**Input**	**#Chnls**	**#Nodes**	**F1-score**	**Precision**	**Recall**
Exp.7[20]	**14K**	**1D Conv, Normal**	**Inlet Temp.**	**72**	**72 Nodes of One Rack**	0.74	0.7	0.79
Exp.8	**14K**	**1D Conv, Normal**	**Inlet Temp.**	**72**	**72 Randomly Selected Nodes**	0.77	0.66	0.92
Exp.9	**3017K**	**1D Conv, Normal**	**Inlet Temp.**	**3312**	**3312 All Nodes**	0.78	0.66	0.96
Exp.10	**1.5K**	**2D Conv, Normal**	**Inlet Temp.**	**1**	**72 Randomly Selected Nodes**	0.82	0.74	0.93
Exp.11	**3K**	**2D Conv, Normal**	**Inlet Temp. & Power**	**2**	**72 Randomly Selected Nodes**	0.73	0.68	0.8
Exp.12	**636K**	**2D Conv, Normal**	**Inlet Temp.**	**1**	**3312 All Nodes**	0.83	0.78	0.9
Exp.13	**3320K**	**2D Conv, Normal**	**Inlet Temp.**	**1**	**3312 All Nodes**	0.78	0.65	0.98
Exp.14	**25.1K**	**3D Conv, Normal**	**Inlet Temp.**	**36**	**3312 All Nodes**	0.87	0.85	0.9
Exp.15	**27.1K**	**3D Conv, Normal**	**Inlet Temp. & Power**	**36**	**3312 All Nodes**	0.8	0.68	0.97
Exp.16	**901K**	**3D Conv, Normal**	**Inlet Temp.**	**36**	**3312 All Nodes**	0.81	0.79	0.83
Exp.17	**27.1K**	**3D Conv, Normal**	**Inlet & Outlet Temp.**	**36**	**3312 All Nodes**	0.8	0.69	0.97
Exp.18	**5.4K**	**3D Conv, Depthwise**	**Inlet Temp.**	**36**	**3312 All Nodes**	0.81	0.69	0.97
Exp.19	**7.4K**	**3D Conv, Depthwise**	**Inlet & Outlet Temp.**	**36**	**3312 All Nodes**	0.81	0.67	0.96

For future work on improving the performance and capabilities of this framework, we have identified two key areas for enhancement: 1. Integration of monitoring signals from datacenter cooling facilities: To enhance the framework's ability to detect and mitigate thermal anomalies, we plan to leverage monitoring signals directly from datacenter cooling facilities. Currently, our framework relies on inlet temperature measurements, which indirectly reflect the state of cooling facilities. By incorporating additional direct monitoring signals, we can enhance the framework's accuracy and robustness. 2. Implementation of more advanced techniques, such as autoencoders, for thermal anomaly detection: While our current framework is effective, we recognize the potential for further advancements in anomaly detection. By integrating more advanced techniques, like autoencoders, we can enhance the framework's ability to identify and classify thermal anomalies with greater precision and efficiency.

8. Acknowledgments

The study has been conducted in the context of EU H2020-JTI-EuroHPC-2019-1 project REGALE (g.n. 956560), EuroHPC EU PILOT project (g.a. 101034126), EU Pilot for exascale EuroHPC EUPEX (g. a. 101033975),EU DECICE project (g.a. 101092582), and CINECA. This work is also supported by the Spoke "FutureHPC & BigData" of the ICSC – Centro Nazionale di Ricerca in "High Per- formance Computing, Big Data and Quantum Computing", funded by European Union – NextGenerationEU.

During this study, we had several meetings with the CINECA's facility manager for thermal hazard definition and HazardNet deployment for large-scale HPC cluster's in-production.

References

[1] Mohsen Seyedkazemi Ardebili, Carlo Cavazzoni, Luca Benini, and Andrea Bartolini. Thermal characterization of a tier0 datacenter room in normal and thermal emergency conditions. In *International Conference on High Performance Computing in Science and Engineering*, pages 1–16. Springer, 2019.

[2] Mohsen Seyedkazemi Ardebili, Davide Brunelli, Tommaso Polonelli, Luca Benini, and Andrea Bartolini. A full-stack and end-to-end iot framework for room temperature modelling on large-scale. *Available at SSRN 4075667*.

[3] ETP4HPC. Etp4hpc – the european technology platform (etp) for high-performance computing (hpc), Apr. 7, 2022 [Online]. https://www.etp4hpc.eu/.

[4] EuroHPC JU. European high performance computing joint undertaking (eurohpc ju), June. 10, 2022 [Online]. https://digital-strategy.ec.europa.eu/.

[5] Nicola Jones. How to stop data centres from gobbling up the world's electricity. *Nature*, 561(7722):163–167, 2018.

[6] John L. Hennessy and David A. Patterson. *Computer Architecture, Sixth Edition: A Quantitative Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 6th edition, 2017.

[7] Madhurima Pore, Zahra Abbasi, Sandeep KS Gupta, and Georgios Varsamopoulos. Techniques to achieve energy proportionality in data centers: A survey. In *Handbook on data centers*, pages 109–162. Springer, 2015.

[8] Hayk Shoukourian, Torsten Wilde, Herbert Huber, and Arndt Bode. Analysis of the efficiency characteristics of the first High-Temperature Direct Liquid Cooled Petascale supercomputer and its cooling infrastructure. *Journal of Parallel and Distributed Computing*, 107:87–100, September 2017.

[9] Jim Rogers. Ornl's warm water hpc facilities and control systems, 2019.

[10] Andrea Bartolini, Christian Conficoni, Roberto Diversi, Andrea Tilli, and Luca Benini. Multiscale thermal management of computing systems-the multitherman approach. *IFAC PapersOnLine*, 50(1):6709–6716, 2017.

[11] Christian Conficoni, Andrea Bartolini, Andrea Tilli, Carlo Cavazzoni, and Luca Benini. Integrated energy-aware management of supercomputer hybrid cooling systems. *IEEE Transactions on Industrial Informatics*, 12(4):1299–1311, 2016.

[12] Xing Fang and Zhuoning Yuan. Performance enhancing techniques for deep learning models in time series forecasting. *Engineering Applications of Artificial Intelligence*, 85:533–542, 2019.

[13] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, Hoboken, New Jersey, 2016.

[14] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3009–3017, 2019.

[15] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

[16] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271*, 2018.

[17] Iqbal H Sarker. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2(6):1–20, 2021.

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[19] Yangdong He and Jiabao Zhao. Temporal convolutional networks for anomaly detection in time series. In *Journal of Physics: Conference Series*, volume 1213, page 042050. IOP Publishing, 2019.

[20] Mohsen Seyedkazemi Ardebili, Marcello Zanghieri, Alessio Burrello, Francesco Beneventi, Andrea Acquaviva, Luca Benini, and Andrea Bartolini. Prediction of thermal hazards in a real datacenter room using temporal convolutional networks. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1256–1259. IEEE, 2021.

[21] Jinkyun Cho, Taesub Lim, and Byungseon Sean Kim. Measurements and predictions of the air distribution systems in high compute density (internet) data centers. *Energy and buildings*, 41(10):1107–1115, 2009.

[22] Babak Fakhim, M Behnia, SW Armfield, and N Srinarayana. Cooling solutions in an operational data centre: A case study. *Applied thermal engineering*, 31(14-15):2279–2291, 2011.

[23] "Mark Fontecchio" "Donald Knuth". Hp thermal zone mapping plots data center hot spots.

[24] Qinghui Tang, Tridib Mukherjee, Sandeep KS Gupta, and Phil Cayton. Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters. In *2006 Fourth international conference on intelligent sensing and information processing*, pages 203–208. IEEE, 2006.

[25] Marina Zapater, José L Risco-Martín, Patricia Arroba, José L Ayala, José M Moya, and Román Hermida. Runtime data center temperature prediction using grammatical evolution techniques. *Applied Soft Computing*, 49:94–107, 2016.

[26] Lizhe Wang, Gregor Von Laszewski, Jai Dayal, Xi He, Andrew J Younge, and Thomas R Furlani. Towards thermal aware workload scheduling in a data center. In *2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks*, pages 116–122. IEEE, 2009.

[27] M. Marwah, R. Sharma, and C. Bash. Thermal anomaly prediction in data centers. In *2010 12th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, pages 1–7, 2010.

[28] Cong Li. Cooling anomaly detection for servers and datacenters with naive ensemble. In *2016 32nd Thermal Measurement, Modeling & Management Symposium (SEMI-THERM)*, pages 157–162. IEEE, 2016.

[29] Andrea Borghesi, Andrea Bartolini, Michele Lombardi, Michela Milano, and Luca Benini. A semisupervised autoencoder-based approach for anomaly detection in high performance computing systems. *Engineering Applications of Artificial Intelligence*, 85:634–644, 2019.

[30] Jayati Athavale, Yogendra Joshi, and Minami Yoda. Artificial neural network based prediction of temperature and flow profile in data centers. In *2018 17th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, pages 871–880. IEEE, 2018.

[31] Doug Garday and Jens Housley. Thermal storage system provides emergency data center cooling. *White Paper Intel Information Technology, Intel Corporation*, 2007.

[32] CINECA. Organization. https://www.cineca.it/en/about-us/organization.

[33] The 53th, 55th, 58th editions of the top500 list., JUNE 2022. https://www.top500.org/.

[34] Ufficio Tecnico. Technical documents, Sep 2019. https://www.cineca.it/.

[35] Andrea Bartolini, Francesco Beneventi, Andrea Borghesi, Daniele Cesarini, Antonio Libri, Luca Benini, and Carlo Cavazzoni. Paving the way toward energy-aware and automated datacentre. In *Proceedings of the 48th International Conference on Parallel Processing: Workshops*, ICPP 2019, pages 8:1–8:8, New York, NY, USA, 2019. ACM.

[36] Alessio Netti, Michael Ott, Carla Guillen, Daniele Tafani, and Martin Schulz. Operational data analytics in practice: Experiences from design to deployment in production hpc environments. *arXiv preprint arXiv:2106.14423*, 2021.

[37] Intel. *Intel Server Board S2600IP and Workstation Board W2600CR Technical Product Specification*. October 2013.

[38] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[40] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.