Future Generation Computer Systems 179 (2026) 108311

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Elevating Datacenter Resilience with ThermADNet: A Thermal Anomaly Detection System

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

ARTICLE INFO

Keywords:
Datacenter
Thermal Anomaly Detection
Deep Learning
LSTM
Autoencoder

ABSTRACT

In the era of digital transformation, datacenters and High Performance Computing (HPC) Systems have emerged as the backbone of global technology infrastructure, powering essential services across various industries, including finance and healthcare. Therefore, ensuring the uninterrupted service of these datacenters has become a critical challenge. Thermal anomalies pose a significant risk to datacenter operation, potentially leading to hardware deterioration, system downtime, and catastrophic failures. This threat is exacerbated by the growing number of datacenters, increased power density, and heat waves fostered by global warming. Detecting thermal anomalies in datacenters involves several challenges. Large-scale data collection is difficult, requiring diverse monitoring signals from thousands of nodes over long periods. The absence of labeled data complicates the identification of normal and abnormal states. Establishing accurate classification thresholds to minimize false positives and negatives is another significant hurdle. Traditional statistical methods often fail to capture temporal dependencies and complex correlations in monitoring signals. Additionally, finding anomalies at both the system and subsystem levels adds to the complexity. Deploying machine learning models in production environments presents technical and operational challenges, making real-time anomaly detection a demanding task. This paper introduces ThermADNet, a Thermal Anomaly Detection framework that combines statistical rules-based methods with Deep Neural Network (DNN) techniques for thermal anomaly detection in datacenters. ThermADNet utilizes a semi-supervised learning approach by training on a "semi-normal" dataset, addressing the challenges of large-scale data collection, semi-normal dataset identification, and classification threshold establishment. This framework's efficacy is validated by its success in identifying real physical thermal failure events within a Tier-0 datacenter, pinpointing anomalies at both the system and subsystem levels, including compute nodes and datacenter infrastructure. In the critical evaluation window covering the July 28 failure, ThermADNet achieves precision and recall up to 0.97, with F1-scores as high as 0.97. By providing detailed information about anomalies, the framework clarifies the characteristics and reasoning behind the DNN outputs, thereby building trust in the AI model and ensuring that users can understand and rely on the system's decisions. By offering a sophisticated method for thermal anomaly detection, ThermADNet significantly contributes to enhancing datacenter reliability and efficiency. This advancement supports the uninterrupted operation of critical HPC systems, averting considerable economic and societal losses.

1. Introduction

Datacenters serve as pivotal assets driving scientific, technological, economic, and industrial progress. Datacenters leverage the combined computing power of thousands of individual computing nodes. This aggregation of computational resources enables supercomputers to deliver significantly enhanced performance, critical for tackling complex computational tasks [1]. The increasing demand for computing resources comes with a corresponding rise in energy consumption. Projections suggest that by 2030, the Information and Communication Technology (ICT) sector's energy usage will constitute approximately 20 % of the global demand, with datacenters contributing a third of that consumption [2]. The growth in datacenter size and complexity, coupled with rising power densities as exemplified by the transition from the Summit

* Corresponding author.
E-mail addresses: mohsen.seyedkazemi@unibo.it (M. Seyedkazemi Ardebili), andrea.acquaviva@unibo.it (A. Acquaviva), luca.benini@unibo.it, lbenini@iis.ee.ethz.ch (L. Benini), a.bartolini@unibo.it (A. Bartolini).

https://doi.org/10.1016/j.future.2025.108311
Received 4 April 2025; Received in revised form 16 November 2025; Accepted 27 November 2025
Available online 19 December 2025

supercomputer's 13 MW peak power consumption in 2017 to the Frontier supercomputer's 29MW in 2021, exacerbates these risks, highlighting the importance of efficient thermal anomaly detection [3]. Experts predict that the computational needs for AI will continue to grow exponentially. This trend suggests that future AI datacenters will require supercomputer-like architectures to handle the increasing demands for processing power and efficiency [4–6].

Investment in supercomputing infrastructure is substantial. For instance, the EuroHPC program has allocated approximately ε650 million in capital and operational expenditures for procuring three pre-exascale systems [7]. In late 2024 and into 2025, the European Union launched a complementary initiative to bolster AI-focused computing infrastructure on a scale comparable to its High Performance Computing (HPC) efforts. The European Commission pledged approximately ε750 million (about $820 million) in 2025 to establish and operate a network of AI-optimized supercomputers, referred to as AI Factories, across the EU. This EU contribution represents half of a ε1.5 billion joint investment, with the remaining funds provided by member states [8,9]. Consequently, datacenter outages can be extremely costly. In the business sector, an outage of Amazon.com in year 2016 would have resulted in an average revenue loss of $15 million, highlighting the significant societal and financial risks associated with disruptions in datacenter operations [10]. The consequences of datacenter outages extend beyond financial implications, affecting critical services and infrastructure. Such events emphasize the critical importance of robust datacenter infrastructure and management practices to mitigate potential losses and ensure the uninterrupted functioning of essential services.

A datacenter is a sophisticated environment housing thousands of computing nodes with power consumption reaching into the megawatt range, all of which generate considerable heat as a byproduct of operation. The authors of [11–18] showed that significant spatio-temporal power-thermal heterogeneity is present during datacenter production, despite the advanced cooling methods used (further detailed in Section 4).

The common practice in cooling system control involves using room-level thermal control. This approach utilizes the average room or average room coolant temperature (air/water) as the input variable for the thermal control loop. However, this method is inadequate as it may obscure crucial thermal irregularities. In 2019, a study [16] unveiled that the Marconi supercomputer encountered significant thermal challenges during its production: *(i)* The usage of hybrid (water/air) cooling technologies and rear-door heat exchanger within the datacenter created a multitude of thermal zones and hotspots [16,17]. *(ii)* Contrary to expectations, spatial proximity does not guarantee thermal coupling, with significant temperature deviations of up to $\sim 6°C$ observed between neighbor nodes. *(iii)* Significant temperature deviations, up to approximately $11°C$, have been observed among compute nodes at the same height in racks but located in different areas of the datacenter. This indicates horizontal spatial thermal heterogeneity. *(iv)* Thermal decoupling between the top and bottom of racks has been observed, with central racks experiencing an average temperature difference of $\sim 15°C$ between their upper and lower sections. These findings emphasize that averaging temperatures at the room level for thermal control is inadequate. This approach can cause localized thermal abnormalities, impairing the optimal performance of specific compute nodes or leading to thermal hotspots.

Minor thermal hotspots can trigger a cascade of events, leading to an imbalance between heat generation by computing nodes and heat dissipation by the cooling system. This can result in thermal anomalies and, in severe conditions, physical thermal failure. These anomalies pose significant risks to datacenter operations, potentially causing damage to IT and facility equipment, as well as datacenter outages.

To illustrate the negative repercussions of thermal anomalies, we conducted an analysis of computing capacity reduction at the CINECA [1] supercomputer, a Tier-0 PRACE facility, following two thermal anomaly incidents. Our findings reveal (i) a 20 % decrease in computing capacity over four days, starting from June 27, 2019, and (ii) a 50 % reduction in computing capacity that lasted nearly two days, beginning on July 28, 2021.

Thermal anomalies may stem from: *(i)* Malfunctioning cooling systems, *(ii)* Abnormal power fluctuations or increased computing demands, *(iii)* Reduced cooling capacity due to abnormal ambient temperatures, *(iv)* Variations in response times between computing and cooling components in response to workload changes, and are expected to increase in the future due to the ongoing trends in electronics that exacerbate computing device power densities, complicating datacenter cooling requirements, as well as the escalating frequency and intensity of heat waves due to global warming [19]. This poses additional challenges, particularly for aging datacenter facilities whose infrastructures were designed decades ago and may struggle to cope with these environmental changes.

Challenges. Given the operational and reliability impact of thermal failures, robust anomaly detection is essential. However, adapting anomaly detection to datacenters entails distinctive challenges, including:

(I) Large dataset: Accumulating diverse monitoring signals from thousands of computing nodes during normal and abnormal operational states necessitates extensive data collection efforts over long periods, considering anomalies are infrequent occurrences.

(II) Annotated dataset: The absence of labeled data poses a significant obstacle. Thermal anomalies may manifest in various ways, such as elevated electronics or coolant temperatures, or suboptimal cooling system performance. Additionally, self-regulating mechanisms within computing units may obscure signs of cooling deficiencies, making visual inspection of monitoring signals by facility managers insufficient for detection. Consequently, latent thermal anomalies, albeit not catastrophic, can undermine supercomputer performance, impacting metrics like energy-to-solution, time-to-solution, and node reliability.

(III) In-production deployments: Deploying machine learning (ML) models in production environments presents several challenges across technical, organizational, and operational domains. Key issues include data quality, continuous availability, model scalability, computational cost considerations, integration complexity, and versioning and experimentation. Addressing thermal anomaly detection thus becomes a challenging and expensive task in datacenter management.

State of the art (overview). Prior work on datacenter/HPC anomaly detection falls into three broad tracks (a detailed discussion of related work is provided in Section 2).

(1) *Statistical rule-based monitoring* flags threshold violations or sudden derivatives on node/facility telemetry and has been used from cloud Service Level Objective monitoring to Tier-0 HPC rooms, with fast, transparent alerts but limited ability to capture cross-metric and temporal dependencies [18,20–22].

(2) *ML-based approaches* span supervised, semi-/unsupervised methods over network, node, and job telemetry, reporting strong accuracy on fault attribution and performance anomalies (e.g., LightGBM, autoencoders, VAEs) [23–27]. Some studies further combine rule-derived labels or priors with ML to reduce annotation needs. [18].

(3) *Thermal, CFD and control studies* use measurements and CFD (or Artificial Neural Networks surrogates) to optimize airflow and predict inlet temperatures, informing design/operations but not typically integrated with room-scale anomaly labeling [28–30].

Despite these advances, several critical gaps remain: (i) scarce validation on *real physical thermal failures*, with most studies relying on synthetic

[1] Cineca is a non-profit consortium, made up of 69 Italian universities, 27 national public research centres, the Italian Ministry of Universities and Research (MUR) and the Italian Ministry of Education (MI), and was established in 1969 in Casalecchio di Reno, Bologna [67]. It is the most powerful supercomputing centre for scientific research in Italy, as stated in the TOP500 list of the most powerful supercomputers in the world: Marconi100, is ranked at the 18th position of the list as of November 2021, with about 30 P/FLOPS [71].

or simulated anomalies; (ii) narrow sensing scopes (node-only or few facility metrics) rather than multi-layer node + cooling + power + ambient views; (iii) insufficient modeling of spatio-temporal dependencies across racks, Computer Room Air Conditioning (CRAC) units, and RDHX loops (see Section 4 for details); (iv) heuristic thresholding of reconstruction errors, lacking principled conversion into actionable alerts; (v) limited reporting of operational metrics such as computational overhead, inference latency, and (vi) absence of production-ready deployment pipelines, including Machine Learning Operations (MLOps) practices for continuous integration and monitoring.

To bridge these gaps, we present *ThermADNet*, a thermal anomaly detection framework tailored for HPC datacenters. ThermADNet integrates statistical cues with deep learning, operates on heterogeneous monitoring signals, and is validated against real physical thermal failure data. Our main contributions are as follows:

- Design of a robust Thermal Anomaly Detection framework, primed for seamless integration into production systems alongside existing monitoring infrastructure. Utilization of an extensive array of features, encompassing monitoring signals sourced from diverse sensors within computing nodes, various cooling facilities (both water and air-based), power distribution system (Modbus) as well as meteorological data obtained from online weather forecasting services. Introduction of a semi-supervised Machine-Learning-based Method (Autoencoder), merging the strengths of rule-based and ML methodologies to enhance thermal anomaly detection efficiency. Dataset compositions for training and testing purposes, ensuring robust evaluation and benchmarking of the proposed framework's performance. Introducing a methodology for defining the reconstruction error thresholds. This has been used to convert real numbers into binary classification, facilitating practical implementation and decision-making processes.
- Implementation of complete process of machine learning operations, including development, deployment, and Continuous Integration/Continuous Deployment (CI/CD). Additionally, we report the processing time latency at different stages of the machine learning pipeline, as well as the computation costs for the datacenter.
- Validation of the framework's effectiveness through the examination of reported real-world physical thermal anomalies. Identification and characterization of anomaly locations within the datacenter environment.

To contextualize these contributions, the next section reviews related work on anomaly detection approaches in datacenters.

2. Related work

In the state-of-the-art (SoA), various methodologies have been used to study anomalies in datacenters. By approaching exascale computing systems, the importance of anomaly detection research topics in datacenters and HPC systems increases [24,31,32]. The main objective of anomaly detection is to identify abnormal patterns or behaviors in the data generated by these systems. This could include any unusual occurrences related to resource usage in computing nodes and datacenter facilities, system log activity, network traffic flow, etc. In the datacenter system, anomalies reduce the performance and increase the cost by affecting the computing capacity and energy of the datacenter. The aim is to detect such anomalies as soon as possible, allowing for prompt scrutiny and addressing to minimize downtime of the supercomputers [26,33].

2.1. Datacenter anomaly studies

Anomalies are reported due to network contention [34], shared resources contention [35,36], hardware-level problems [37], memory [38], CPU [39], and cooling system failure [16,40]. These anomalies are then flagged as potential aberrations for further inspection by the system administrator [41].

2.1.1. Statistical rule-based approach

Several studies rely on rule-based logic, where anomalies are flagged once system metrics exceed fixed or statistically derived thresholds.

In [20] (2015), the authors focused on application/runtime, resource, and Quality of Service/Service Level Agreement anomalies in cloud systems. They monitored CPU, memory, threads, and logs, and applied either manual fixed thresholds or seasonal statistical bounds. Detected anomalies triggered resolution via an Event–Condition–Action (ECA) policy engine. No quantitative accuracy was reported; validation was qualitative and based on prior results.

In [21] (2017), the authors used rule-/heuristic-based monitoring under a Service Level Objective framework to detect anomalies and identify bottlenecks in cloud-hosted PaaS services. They tracked end-to-end request latency and per-component service times, flagging anomalies when fewer than 95 % of requests met latency targets learned from baseline benchmarking. Root causes were inferred via majority voting among four statistical heuristics (importance regression, change-point detection, high-quantile check, tail-outlier test). They reported *counts of detected anomalies* under different Service Level Objective settings and *detection latencies of approximately 2–3 min*.

Study [22] (2023) employed monitoring signals from both computing nodes and facility-level infrastructure of Marconi-100 CINECA a Tier-0 HPC datacenter to detect thermal anomalies using statistical rule-based methods. A wide range of thermal, power, and cooling metrics were monitored across nodes, racks, cooling units, and power distribution systems. Thresholds were defined using ASHRAE recommendations where available, and otherwise derived from statistical quantiles of historical data.

Anomalies were flagged whenever monitored signals either violated predefined thresholds (constraint violations) or exhibited sudden variations exceeding derivative thresholds. Two principal categories were identified: (i) Constraint Violations - when a metric deviates significantly from its expected operating range. (ii) High Derivatives - when a metric undergoes unusually rapid changes over a short interval.

Based on these principles, the authors proposed a set of rule-based statistical methods ("flags") designed to capture abnormal patterns at the node, subsystem, system, and datacenter levels. Each rule raised a flag upon detecting suspicious behavior, and the sum of active flags at each timestamp was used as a severity indicator for anomaly assessment.

The approach successfully identified a real thermal incident in the CINECA Marconi100 Tier-0 HPC cluster, with increasing severity levels corresponding to precursor phases and the final failure. Although the method demonstrated clear qualitative alignment with operator-reported failures, no quantitative accuracy metrics (e.g., precision, recall, F1-score) were reported. Validation was instead based on temporal correlation with incident reports and spatial anomaly localization through heatmaps.

In [18] (2024), the authors analyzed three IPMI metrics-inlet temperature, outlet temperature, and node power-on CINECA's Marconi-A2 system, a Tier-0 production HPC cluster, ultimately identifying inlet temperature as the most reliable indicator of thermal hazards. They proposed a rule-based statistical labeling scheme, defining a Node-Threshold (NT) as the 0.95 quantile of each node's inlet temperature distribution to assign a binary anomaly status at the node level. Complementing this, the Spatio-Temporal Impact Threshold (STIT) extended detection beyond single nodes and moments in time by evaluating 6 h windows and requiring a quorum of anomalous nodes, thereby capturing the spatial and temporal continuity of hazards across the datacenter. These statistically derived labels were then used to train supervised deep learning models, where a Temporal Convolutional Network (TCN) achieved near-perfect F1 = 0.98 under random splits and robust F1 = 0.87 under time-separated realistic tests.

Other studies [38,39] analyze the monitoring data of the system and its components to identify correlations between various issues (such as detecting I/O congestion and out-of-memory) and their causes.

The main strength of statistical rule-based methods lies in their speed and simplicity, which makes them suitable for lightweight online monitoring in large-scale datacenters. However, they are inherently limited in capturing complex correlations across metrics: a single variable may appear normal in isolation but become anomalous when combined with others, or vice versa. Moreover, as the volume and diversity of monitoring data grow, rule-based analysis and manual root-cause inspection become inefficient. For these reasons, ML-based approaches are increasingly adopted for anomaly detection in datacenters [23–27,42].

2.1.2. ML-based approaches

Machine learning (ML) has become a central component of modern anomaly detection and diagnostic frameworks for datacenters and HPC systems. Existing studies vary widely in their monitored metrics, learning paradigms, and operational objectives, yet together they demonstrate both the diversity and increasing maturity of ML-driven solutions [23–27,42]. Below, we present these contributions in chronological order to highlight the evolution of methodologies-from early statistical techniques to modern deep and federated learning models-while emphasizing their strengths and limitations.

Early work from 2010 such as [43] explored conventional statistical techniques (thresholds, moving averages, EWMA, Naïve Bayes) for the early detection of thermal anomalies in a production datacenter. Using three months of real sensor data, the naïve Bayes classifier detected up to 18 % of anomalous events approximately 12 min before occurrence. However, anomalies were defined using simplistic rules, and the models lacked the capacity to capture the spatiotemporal dependencies inherent in thermal behavior.

Subsequent efforts expanded the focus to other datacenter subsystems. For example, Li [44] (2016) leveraged temperature-sensor telemetry to perform unsupervised anomaly detection, evaluating the approach on simulations of predictive failures such as worn-out fans and CRAC malfunctions. Around the same period, Study [23] (2016) demonstrated that ML can also support network-level diagnostics: by analyzing TCP statistics with decision trees and random forests, their lightweight endpoint-based tool attributed client, server, and network faults with up to 96 % accuracy.

Network-oriented anomaly detection was further explored in [27] (2016), which applied SVMs enhanced with Poisson Moving Average (PMA) features to cloud traffic traces. Using protocol types, port numbers, packet sizes, and packet counts, their model achieved a detection rate of 98.56 % with only 1.44 % false positives, and successfully detected a real DDoS attack within five minutes.

ML-based anomaly detection for HPC systems continued to evolve with [42] (2018), which developed a dynamic online/offline framework for identifying inefficient applications on the Lomonosov-2 supercomputer. Their LSTM-based classifier operated on 40-minute windows of 16 node-level metrics (CPU, cache misses, memory traffic, MPI and I/O counters), and suspicious jobs were re-evaluated by an offline Random Forest classifier, yielding 95 % agreement with offline ground truth for 3300 applications.

Autoencoder-based unsupervised and semi-supervised methods soon followed. Study [25] (2019) demonstrated that a semi-supervised autoencoder trained solely on normal data (CPU, memory, power, temperature) could detect misconfigurations with weighted F-scores above 0.91-improving accuracy by 12 % over other semi-supervised baselines. Complementary works [45–47] explored supervised methods for fault detection and classification, though these often relied on trivial correlations (e.g., idleness = failure) and were trained primarily on synthetic data due to the scarcity of labeled anomalies in production HPC systems. To alleviate this challenge, semi-supervised methods such as [48] (2019) were proposed, requiring only normal samples for training.

Subsequently, [49] (2020) developed an ML-based real-time predictor of node failures using log collections from four HPC systems, extracting recurring failure patterns for proactive detection.

Later works introduced holistic, multi-metric, and production-grade frameworks. Study [24] (2021) proposed an end-to-end anomaly-diagnosis pipeline that monitored 160 telemetry variables (CPU, memory, counters, interrupts), applied windowing and feature selection, and benchmarked Random Forest, XGBoost, and LightGBM models. LightGBM achieved the best performance, with macro-average F1-scores up to 0.91 and false-alarm rates below 0.05

While semi-supervised models improve label efficiency, they often suffer from high false-positive rates. Study [50] (2022) addressed this limitation by combining a semi-supervised model with a supervised classifier, supported by a custom monitoring infrastructure that enabled the generation of labelled datasets. This hybrid approach achieved an F-score of approximately 0.86 and predicted anomalies up to one hour before they were officially reported by system administrators.

Alternative paradigms have also been explored. For example, Aksar et al. [51] (2022) introduced an active learning method to discriminate performance variances across HPC nodes, reducing annotation requirements but still relying on synthetic datasets. Conversely, Molan et al. [52] (2023) demonstrated a fully unsupervised failure-detection method evaluated on real anomalies from a Tier-0 system, offering a more operationally practical solution due to its independence from labels.

A step toward large-scale production deployment is Prodigy [26] (2023), an unsupervised anomaly-detection framework tested on a 1,488-node production system (Eclipse) and a 52-node testbed (Volta). Prodigy uses LDMS telemetry sampled at 1 Hz, extracts over 700 statistical and spectral features, selects the most discriminative via a Chi-square test, and trains a variational autoencoder (VAE) on healthy samples. It achieves macro F1-scores of 0.95 (Eclipse) and 0.88 (Volta), outperforming classical unsupervised baselines. Additionally, its interpretability module (CoMTE) provides node- and job-level root-cause explanations.

More recently, federated learning has been introduced for HPC anomaly detection, as demonstrated in [53] (2024), enabling privacy-preserving model training across distributed systems.

Finally, [18] (2024) proposed HazardNet, a hybrid statistical-ML approach for thermal anomaly prediction in a Tier-0 datacenter. Statistical rule-based labeling was first used to identify thermal anomalies, and the resulting labels trained deep neural predictors that achieved an F1-score of 0.87 at the room level. Further details on the rule-based methodology are provided in Section 2.1.1.

2.2. Datacenter thermal studies

2.2.1. Design improvements

Research on datacenter thermal management spans a wide range of approaches, including airflow and cooling-architecture analysis, sensor-based thermal characterization, data-driven surrogate modeling, and IoT-enabled monitoring. These lines of work collectively aim to deepen the understanding of temperature dynamics in high-density computing facilities and to improve the effectiveness and efficiency of cooling strategies.

One of the earliest studies in this area is [54] (2006), which explored the use of mobile temperature sensors combined with a thermal computer model to reconstruct heat maps and room-level thermal evolution. Although the method provided spatial coverage, the reliance on sensor relocation produced temporally inconsistent snapshots, leading to chronologically distorted thermal models. This limitation highlighted the need for fixed, time-synchronized sensing infrastructures for accurate thermal characterization.

Subsequent efforts shifted toward analyzing airflow distribution and cooling behavior in production facilities. The study in [28] (2009) systematically compared raised versus hard-floor deployments, cooling-unit placement, hot-aisle/cold-aisle organization, and ducted versus flooded airflow configurations. Using a combination of temperature measurements and CFD simulations, the work showed that ducted supply and return airflow, combined with proper aisle separation, significantly improves thermal stability and reduces hot-air recirculation. In contrast, unguided flooded-air configurations resulted in uneven temperatures and were suitable only for smaller server rooms.

Further refinement of airflow analysis was provided in [29] (2011), which conducted extensive field measurements and CFD-based evaluation of an operational datacenter. The study identified several contributors to thermal inefficiency, including CRAC mis-provisioning, tile airflow imbalance, and hot-cold air recirculation. Multiple mitigation strategies were examined, such as cold-aisle containment, blanking of unused rack spaces, repositioning of CRAC units, and ceiling return modifications. The combination of aisle containment with overhead ducting achieved the best results, reducing the Supply Heat Index to 0.16 and decreasing peak rack inlet temperatures from 35 °C to 33.1 °C.

As modeling techniques advanced, [55] (2016) introduced a methodology based on Grammatical Evolution to generate computationally efficient models for runtime CPU and inlet temperature prediction. The approach achieved average errors of 2 °C for CPU temperature and 0.5 °C for inlet temperature, offering a lightweight alternative to full CFD simulations for real-time monitoring and control.

The integration of machine learning into thermal prediction continued with [30] (2018), which developed an Artificial Neural Network (ANN) surrogate model trained on high-fidelity CFD/HT simulation data. The ANN achieved an average error of 0.6 °C for rack inlet temperature and 0.7 % for tile airflow rate, demonstrating the feasibility of using surrogate models as fast, accurate predictors suitable for online optimization of cooling systems.

More recent efforts have focused on increasing sensing resolution and system awareness. The study in [17] (2022) presented a full-stack IoT monitoring infrastructure deployed in a production datacenter to capture fine-grained ambient temperature variations. The analysis revealed substantial spatial thermal heterogeneity and demonstrated how computing-cluster activity shapes temperature behavior across the facility. The study also emphasized the importance of leveraging internal node sensors for more detailed power and thermal characterization.

2.2.2. Thermal aware task scheduling

The authors of [56] (2009) 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 [54] (2006). 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.

2.2.3. Thermal storage system

An entirely different approach involves using a thermal storage system with chilled water tanks. This system can mitigate thermal failures while nodes continue to function using an Uninterruptible Power Supply (UPS) during thermal emergency [57] (2007).

2.2.4. Thermal studies of CINECA datacenter

Regarding the CINECA datacenter, which is the target datacenter in this study for experimental results: The study [16] (2019) 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 several factors. It includes the ambient temperature, as the chillers are located outdoors, and the use of direct free cooling, which recirculates the outside cold air into the datacenter. It also includes the outlet air or water temperature of the cooling systems, and feedback from the nodes’ power consumption, which recirculation of hot 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 [16] (2019) shows that the 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 [16] 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 W 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.*

2.3. State of the art recap and positioning of ThermADNet

Table 1 provides a consolidated summary of the reviewed research directions in datacenter anomaly detection and thermal studies, positioning ThermADNet relative to prior efforts.

Prior work spans (i) rule-based monitoring with thresholds or heuristics for rapid, transparent detection; (ii) ML approaches-supervised, semi-/unsupervised-for performance and fault anomalies on node/cluster telemetry; and (iii) thermal engineering tracks (CFD/design studies, surrogate models, thermal-aware scheduling). These lines advance detection and control but rarely integrate holistic, multi-source datacenter signals with spatio-temporal learning and production-grade deployment.

Observed gaps.

G1) *Label scarcity and realism.* Many studies rely on synthetic faults or limited labels; few are validated on real thermal failures with time-separated evaluation.
G2) *Narrow sensing scopes.* Typical inputs cover node-only or a handful of facility metrics, under-utilizing power/cooling/electrical and ambient context.
G3) *Under-modeled spatio-temporal coupling.* Cross-rack/CRAC/RDHX interactions and slow/fast dynamics are often simplified or ignored.
G4) *Thresholding rigor.* Semi-supervised AE methods frequently lack principled conversion from reconstruction error to actionable binary alerts.
G5) *Operational metrics.* Compute/ops overhead, inference latency are rarely reported for production viability.
G6) *Deployment pathway.* MLOps, CI/CD, and online integration with monitoring stacks (pub/sub, APIs) are often out of scope.

How ThermADNet addresses these gaps.

C1) Real physical thermal failure validation on an in-production Tier-0 HPC room, with time-separated tests and analysis of a physical thermal failure (Sections 6 and 6.5; Fig. 8(a)).
C2) Multi-source sensing across *241* monitored metrics from nodes, cooling (CRAC, RDHX), electrical (Modbus), and weather context (Section 5.2; Table 2; Section 6.1).

Table 1
Comparison of major research directions in datacenter anomaly detection and positioning of **ThermADNet** relative to prior work. Representative studies for each category are discussed in Section 2 [18,20–30].

Aspect / Feature	Rule-based Methods	ML-based Approaches	Thermal / CFD Studies	ThermADNet (this work)
Real physical failure validation	× / partial	× / partial	×	✓ Tier-0 HPC Cluster failures
Multi-source sensing (node + cooling + power + ambient)	×	Limited	✓ (thermal only)	✓ Comprehensive (241 metrics)
Spatio-temporal modeling	×	✓ (partial)	✓ (thermal only)	✓ LSTM-AE temporal coupling
Semi-supervised learning	×	✓	×	✓ Rule-derived semi-normal dataset
Threshold definition	Fixed / heuristic	Heuristic	-	✓ Quantile-based
Explainability / localization	×	Partial	✓ (CFD visualization only)	✓ System + subsystem localization
Operational metrics (latency, overhead)	×	Rarely	×	✓ Reported and analyzed
Deployment pathway (MLOps, CI/CD, monitoring integration)	×	×	×	✓ ExaMon + Kubeflow + Kubernetes (Section 5.8, 5.4)

C3) Spatio-temporal representation via LSTM-AE and localization at system/subsystem granularity (Sections 5.5 and 6; Fig. 9).
C4) Principled alerting through quantile-based threshold strategies (T1-T4) on reconstruction-error distributions, enabling tunable sensitivity/precision (Sections 5.7 and 6.3; Fig. 7).
C5) Reporting of compute overhead for realistic operations (Section 6.6; Table 8; Table 9; Section 7).
C6) A full deployment blueprint: ExaMon integration, MQTT pub/sub, Kubernetes + Kubeflow CI/CD, and staged rollout (Sections 5,5.4 and 5.8; Fig. 5; Section 10).

Together, these elements position ThermADNet as a production-oriented, semi-supervised framework that fuses rule-derived normality cues with deep spatio-temporal learning to deliver actionable, room-aware thermal anomaly detection.

3. Autoencoder

In the context of addressing the challenge of detecting thermal anomalies, Machine Learning (ML) tools have emerged as a promising approach for anomaly detection. Notably, ML tools such as time-series forecasting, classification, and autoencoders have demonstrated potential for identifying thermal anomalies[40,58–63].

In the realm of artificial intelligence, a multilayer perceptron (MLP) is a type of ANN comprising interconnected nodes organized into multiple layers. Each node, which is a neuron, processes input data and forwards output signals to the subsequent layer of neurons. The final layer's output is then used for prediction or classification purposes. DL is a branch of AI and Machine Learning that has demonstrated success in learning from data and is, therefore, widely utilized in various fields. DL advancements have facilitated the development of methods for training models on large-scale time-series data.

Recurrent Neural Networks (RNNs) are a category of 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 [58,64,65].

Autoencoder is a sort of ANN model (Fig. 3 Deep Learning Layer) composed of three components, *Encoder*, *Code*, and *Decoder*, which reconstruct the input at the output of the model, and each of these parts can compose of multiple hidden layers. *Encoder* maps the input to the *Code* layer, and *Decoder* maps the code layer to the output of the autoencoder. Eq. (2) shows the error (reconstruction error ϵ) of the input from the reconstructed input ($\widehat{Input}$) at the output of the autoencoder.

$$\widehat{Input} = Decoder(Encoder(Input)) \tag{1}$$

$$\epsilon = |Input - \widehat{Input}| \tag{2}$$

During training, the autoencoder learns to reconstruct its input at the output by minimizing the error. The performance of the autoencoder in reconstructing the input can be measured by the reconstruction error, which indicates how well the model has learned the underlying structure of the data. The training process can be done in a supervised, semi-supervised, or unsupervised manner [66].

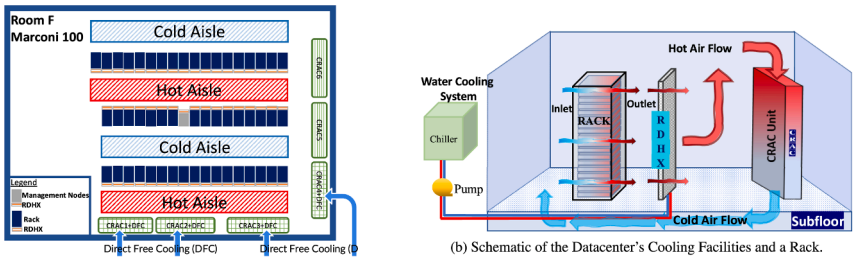
The reconstruction error of a well-trained autoencoder is a valuable metric for identifying anomalies in the anomaly detection task. By comparing the reconstruction error with a predefined threshold, we can classify samples as either normal or abnormal. In order to properly capture the temporal aspect and extract useful properties from the data while learning how different parameters correlate, it is crucial to train the model on a subset of the dataset that is as close to a normal subset (semi-normal) of the dataset as possible. However, two challenges must be addressed. Firstly, how to extract a semi-normal subset of the dataset from the complete dataset, and secondly, how to determine the optimal reconstruction error threshold ϵ_{th}. These challenges are critical to the success of the anomaly detection task, as the accuracy of the model's detections hinges on the quality of the semi-normal subset and the chosen threshold.

4. 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) [67]. It is a national supercomputing center 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-100 supercomputer and the datacenter hosting it. Marconi-100 has been in operation from April 2020 and ranked 9th (list of June 2020) and 18th (list of November 2021) in the Top500 list, which ranks the most powerful supercomputers worldwide [68].

Marconi-100 is the accelerated cluster based on IBM Power9 architecture and Volta NVIDIA GPUs. It comprises 980 nodes; each node has 2x16 cores IBM POWER9 (@3.1 GHz) processors and is empowered with 4 x NVIDIA Volta V100 GPU accelerators (16GB), and 256 GB of RAM per node. The supercomputer's peak performance is 32 PFlops. The Marconi-100 datacenter (Fig. 1(a)) hosts 55 racks (49 computing racks), arranged in 3 rows; each rack has 20 stacked chassis, each with one node [69].

Fig. 1 shows the Marconi-100 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).



(a) Racks Arrangements of Marconi100.

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

Fig. 1. Datacenter computing room.

Marconi-100 datacenter is cooled using Computer Room Air Conditioning (CRAC) units by the Direct Expansion (DX) Air-conditioning system. 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 the external cold air for cooling the datacenter. In this case, the DFC system starts to work when the outdoor temperature is lower than 18 °C. 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.

Also, there is a water cooling system for Rear Door Heat eXchangers (RDHX), with the chiller loop (cold loop) temperature around $12°C$ to $17°C$, and RDHX loop (hot loop) temperature around $23°C$ 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 to circulating water from a chiller. Thus, the compute node outlet air temperature reduces before its discharge into the datacenter. RDHX is used to augment the computing density in 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.

5. Methodology

In this section, we describe the methodology for the proposed Thermal Anomaly Detection framework (ThermADNet) for datacenters. As shown in Fig. 2, the workflow begins with data collection and preprocessing, which feed both the offline model training pipeline and the online inference pipeline. Trained weights are uploaded to the inference pipeline, which produces anomaly labels and alerts.

5.1. System under study (cooling)

The analyzed room (Marconi100, CINECA) employs a hybrid cooling architecture that combines air-based Computer Room Air Conditioning (CRAC) units with Direct Expansion (DX) and Direct Free Cooling (DFC), together with liquid-based Rear Door Heat eXchangers (RDHX) on a chiller loop (hot/cold aisles). In this setup, DFC engages when outdoor temperature is below 18 °C; RDHX operates with a chiller (cold) loop typically around 12–17 °C and an RDHX (hot) loop around 23–30 °C (see Fig. 1 in Section 4).

5.2. Input features

To train and evaluate ThermADNet, we rely on multi-source telemetry collected by the ExaMon monitoring framework. The signals span compute-node sensors, cooling and power facilities, and ambient conditions. Table 2 summarizes the main groups of variables and representative examples. Overall, the study uses 241 metrics from one rack (20 nodes) together with room-facility signals, collected over a four-month period (April 8-August 18, 2021).

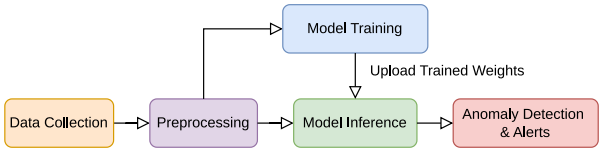


Fig. 2. High-level workflow of ThermADNet. Data collection and preprocessing feed both offline training (blue) and online inference (green). The trained model is periodically updated, enabling real-time anomaly detection and alerts (red).

5.3. Framework architecture

While Fig. 2 summarized the overall workflow of ThermADNet, Fig. 3 illustrates the detailed architecture and the interactions across its five main layers: (1) Datacenter Layer, (2) Monitoring System (Data Collection Layer), (3) Data Preprocessing Layer, (4) Deep Learning Layer, and (5) Anomaly Detection and Alerting Layer.

The datacenter layer includes approximately a million built-in sensors (We did not install any new sensors in this layer.) that collect various types of data from computing nodes and datacenter facility systems, such as power consumption, temperature, etc. (see Section 4 Background Setups). The monitoring system and data collection layer have the responsibility of continuously and efficiently gathering data from sensors and storing it in time-series databases. In this layer, we utilized ExaMon [70] (Section 5.4 Monitoring System). The data preprocessing layer is responsible for processing the raw data collected from sensors. It cleans and filters out irrelevant or erroneous data, handles missing data, and transforms it into a format that is suitable for DL models (Section 5.6 Data Preprocessing). The DL layer includes Deep Neural Network (DNN) models and is responsible for training the model to recognize anomalous patterns in the monitoring data (Section 5.5 Deep Learning (Autoencoder)). Finally, the anomaly detection and alerting layer is responsible for detecting anomalies based on the output of DL models. This layer can also generate various reports that can be helpful in identifying the source of the anomalies (Section 5.7 Label Generation). Overall, our methodology for thermal anomaly detection offers a comprehensive approach to detecting anomalies in the datacenter. It can be applied to various use cases in different domains with some modifications and adjustments to the layers, based on specific requirements.

The proposed framework consists of two separate dataflow pipelines: offline and online. The offline pipeline is designed for training the DNN model, while the online pipeline is optimized for real-time inference and generation of online labels, reports, and alerts. In both the offline and online pipelines for monitoring data extraction from the monitoring system, we used RESTful API. In the online pipeline to publish the results

Table 2
Categories of monitored signals and representative metrics collected in ThermADNet.

Group	Representative variables
Compute nodes (IPMI)	Inlet / PCIe / CPU[0,1] / GPU[0.3] Temperatures; Fan Speed; PSU Power
CRAC (DX + DFC)	Compressor Utilization; DFC State; Free-cooling Valve Position; Fan Speed; Return/Supply Air Temperature
RDHX (Water Loop)	Water Flow Rate; Inlet/Outlet Water Temperature; Three-way Valve Position; Water ΔT
Electrical (Modbus)	ICT Total Power; RDHX Pumps Power; Chillers Power; CRAC Units Power
Ambient/Weather	Outdoor Temperature (DFC context)

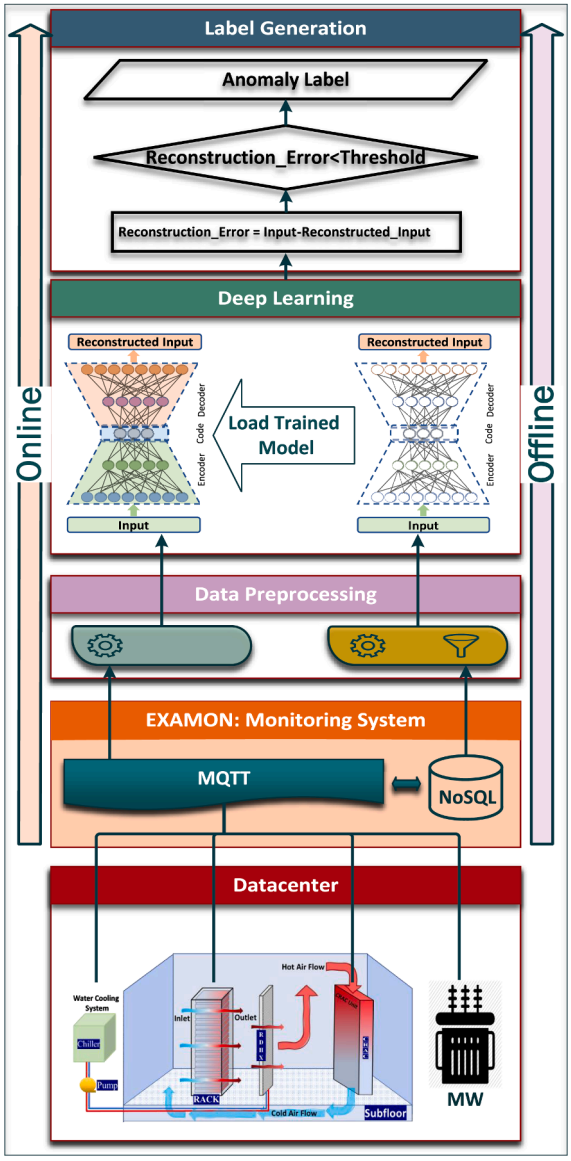


Fig. 3. Detailed architecture of the ThermADNet framework.

of inference back to the monitoring system, we utilized the publisher-subscriber communication method using the MQTT protocols.

5.4. Monitoring system

The CINECA datacenter features a holistic monitoring framework, ExaMon, which aggregates a wide set of telemetry data collected via a set of plugins (one for each monitored component) that read the sensor and communicate to the ExamonDB via MQTT messages. ExamonDB uses Cassandra and KairosDB technologies. The different monitored components are at the system level, the cooling and power provisioning equipment, while at the compute node level, IPMI for out-of-band telemetry. The ExaMon monitoring system collects sensor data and stores these data in its internal KairosDB database as time traces and is remotely accessible through REST APIs.

5.5. Deep learning

The DL layer of our framework consists of two autoencoders. While these autoencoders have identical architectures, they may have different weights at certain periods. The first autoencoder is used in the offline training dataflow pipeline for the training or retraining of the model. The second autoencoder is used in the online dataflow pipeline for continuous anomaly detection. After training updates, the weight of the autoencoder in the offline pipeline automatically uploads to the online pipeline's autoencoder. This keeps the online pipeline updated with the latest training information, improving the accuracy and reliability of the anomaly detection process. We used the thermal, power, and cooling parameters of the Marconi100 datacenter as input for the autoencoder model. Well-trained autoencoders can reconstruct the input parameters at the output with minimal reconstruction error ϵ. In other words, an autoencoder can capture the characteristics of a dataset and reconstruct it. The main idea is to train the autoencoder with the normal subset of the datacenter monitoring dataset. This way, the autoencoder can capture the standard characteristics of the monitoring signals. It can then reconstruct the normal monitoring data with very low error. Conversely, it will reconstruct abnormal monitoring data with a high ϵ. Since the data does not have any normal or abnormal labels, the question is how we can define a normal subset of the data. This is discussed further in the following Data Preprocessing section.

We evaluated two different types of autoencoders (AE) for our study: MLP-AE and LSTM-AE. The MLP-AE model consists of six Multilayer Perceptron (MLP) layers and has a total of 100K trainable parameters. This model is effective in learning the normal relation of various input parameters, critical for our study. However, due to its architecture, the MLP-AE model cannot learn the temporal relation of the data, limiting its effectiveness in time-series data analysis. Conversely, the LSTM-AE model, a form of Recurrent Neural Network (RNN), can learn long-term dependencies and temporal attributes of time-series data. This model consists of two main components: the encoder and decoder layers, each containing two LSTM sublayers. Altogether, the LSTM-AE model has half a million trainable parameters, offering ample capacity to detect complex patterns in the data.

5.6. Data preprocessing

The data preprocessing layer is responsible for processing the raw data collected from sensors in both the online and offline dataflow pipelines. It performs several tasks, such as handling missing data, transforming data into a format suitable for DL models, scaling/normalization, and filtering out abnormal or suspicious data. This layer in the training pipeline, offline data flow, has a crucial filtering task that sets it apart from the online data flow. This filter is designed to allow only a subset of the dataset to proceed to the training phase of the subsequent layer, ensuring that the dataset closely matches the normal distribution of the monitoring signals of the datacenter. For filtering out the data, we use the Initial Data Annotation as described in the following.

Table 3
Initial data annotation.

Initial Label	Definition	Percentage of Dataset
Normal	$\sum Flags = 0$	13.93%
Ambiguous	$1 \leq \sum Flags \leq 25$	81.83%
Abnormal	$\sum Flags > 25$	4.24%

5.6.1. Initial data annotation

We need to extract a normal subset from the dataset. Since the dataset lacks annotations that differentiate between normal and abnormal samples, we need to identify an approach to assist us. The statistical rules-based approach proposed in paper [22] appears to be applicable for this step. As outlined in the Section 2.1.1 Statistical Rule-based Approach, this approach gives a comprehensive analysis of physical and thermal anomalies. It sets statistical rules to flag when unusual or suspicious patterns are detected in various monitored systems. These systems include computing nodes, CRAC Units, RDHX, Modbus, and more. By adding up the flags raised at each timestamp, the approach measures the severity of abnormalities in the datacenter. However, it's important to note that this method can't capture complex correlations in the monitoring signals. It also has limitations in handling time-series data, particularly with long-term dependencies.

We used the severity level of abnormalities in the datacenter as a metric to extract a subset of the dataset, which we refer to as the semi-normal subset. The intent was to filter out parts of the monitoring data that appeared suspicious. We then used this semi-normal subset to train the autoencoder. We applied statistical rules to all critical metrics of computing nodes and room facilities, including CRAC Units, RDHX, Modbus, etc. In total, we defined 281 flags for 241 metrics. We then calculated the sum of the raised flags, which represents the severity level of abnormality, for each timestamp or sample.

Ambiguous Dataset| Due to the challenges in defining a solid threshold for the sum of raised flags (severity level of abnormality) to classify a sample as normal or abnormal, we applied the statistical rule-based method conservatively, classifying only the extreme samples, those with zero raised flags, and those with a high number of raised flags. Samples with no raised flags, which indicate a severity level of zero, are likely normal. We selected these samples to create a semi-normal subset, which constitutes 13.93 % of the dataset. We defined samples with the top 4.24 % of raised flags as abnormal. Consequently, we did not classify over 81.83 % of the dataset. We referred to this subset as a "ambiguous dataset". Table 3 provides precise definitions for the normal, abnormal, and ambiguous datasets based on the statistical rule-based method.

5.6.2. Training and testing dataset compositions (A-F)

In the offline dataflow, we train the model using a subset of data that represents normal or semi-normal conditions, namely the training dataset. We then evaluate the model's performance using a different subset that includes both semi-normal and abnormal data, referred to as the test dataset. The model does not see the test data during training.

Fig. 4 illustrates how we divided the data into training and testing subsets. The left side of the figure shows data with no raised flags (green area), while the right side shows data with more than 25 raised flags (red area). The middle area (ambiguous) illustrates data with unclear labels. As we move from left to right, the number of raised flags increases. Each row represents a distinct *dataset composition* (A–F) used in our experiments, progressively including more ambiguous data in the training set.

Before discussing the different dataset compositions, we should note that we had several reasons to explore the impact of incorporating ambiguous data into the training set: (i) The samples with a few raised flags lacked clear labels, so it wasn't safe to assume they were all abnormal. (ii) The percentage of samples with zero raised flags was small (only 13.93 %), potentially limiting the model's performance. (iii) If we trained the model solely on samples with zero raised flags, it could result in overfitting and poor generalization.

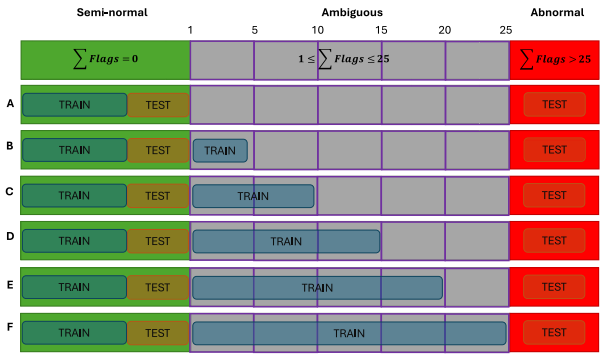


Fig. 4. Different dataset compositions of the train and test dataset.

As depicted in Fig. 4 row A, we started the train and test composition with *dataset composition A*, in which we randomly selected 75 % of the normal data for training. The remaining 25 % of the normal data, along with all of the abnormal data, were utilized for testing. Then, in *dataset compositions B–F*, we progressively increase the amount of ambiguous (gray) data included in the training set.

For instance, in *composition B*, the model was trained using gray samples with less than five raised flags, along with 75 % of randomly selected normal samples (zero flags data). Conversely, in *composition F*, the model was trained using all the gray data and 75 % of the standard samples.

5.7. Label generation and alerting

Up to this point in the methodology, we discussed how we train the autoencoder model, and it reconstructs the input at the output. Here, we will introduce a statistical approach to convert the outputs of the model, which is a continuous value, into a binary class.

5.7.1. Reconstruction error threshold

The autoencoder in the DL layer reconstructs input at the output of the model with some error ϵ, known as the Reconstruction Error. A well-trained autoencoder should reconstruct normal samples with low ϵ, while abnormal samples will yield a higher ϵ. In the label generation layer, we create a binary label for the input sample by comparing the ϵ with a threshold, referred to as the *Reconstruction Error Threshold* ϵ_{th}.

If the reconstruction error (ϵ) exceeds the threshold (ϵ_{th}), the autoencoder classifies the sample as abnormal. Otherwise, it is classified as normal. The threshold ϵ_{th} is crucial in regulating the anomaly detection sensitivity of the framework. A low ϵ_{th} results in a high anomaly detection rate, leading to an increased number of false positives. This conservative approach ensures safety but may cause unnecessary alarms and operational inefficiencies in the datacenter. Conversely, a high ϵ_{th} can reduce false positives but may increase the likelihood of false negatives, potentially overlooking critical anomalies that could have severe detrimental effects.

$Q_x(\cdot)$ denotes the x quantile function. For example, $Q_{0.99}(\cdot)$ denotes the 0.99 quantile function. ϵ_i represents the reconstruction error for the ith sample. We define the 0.99 quantile of the reconstruction error distribution, which includes both normal and ambiguous data, as the reconstruction error threshold ϵ_{th}. In the experimental results, we will demonstrate how changing the quantile x from 0.99 affects the percentage of detected anomalies. Furthermore, we show how to adjust the anomaly detection sensitivity by using the distribution of reconstruction errors from different parts of the dataset to define ϵ_{th}.

Assuming that we train the autoencoder model using samples with fewer than 10 raised flags, we propose four different threshold strategies for defining ϵ_{th}:

1. **Strategy T1:**
   The threshold ϵ_{th} is set by calculating the 0.99 quantile of the reconstruction error distribution across all samples with fewer than 10 raised flags:
   $$\epsilon_{th} = Q_{0.99}\left(\left\{\epsilon_i \mid \sum \text{flags} < 10\right\}\right)$$
2. **Strategy T2:**
   The threshold ϵ_{th} is determined by computing the 0.99 quantile of the reconstruction error distribution for samples with 10 to 25 raised flags:
   $$\epsilon_{th} = Q_{0.99}\left(\left\{\epsilon_i \mid 10 \leq \sum \text{flags} \leq 25\right\}\right)$$
3. **Strategy T3:**
   The threshold ϵ_{th} is established by calculating the 0.99 quantile of the reconstruction error distribution exclusively from the ambiguous dataset:
   $$\epsilon_{th} = Q_{0.99}\left(\left\{\epsilon_i \mid \text{ambiguous dataset}\right\}\right)$$
4. **Strategy T4:**
   The threshold ϵ_{th} is set by computing the 0.99 quantile of the reconstruction error distribution only from the semi-normal dataset:
   $$\epsilon_{th} = Q_{0.99}\left(\left\{\epsilon_i \mid \text{semi-normal dataset}\right\}\right)$$

These threshold strategies allow us to explore how different subsets of the data affect the reconstruction error threshold and the resulting anomaly detection sensitivity.

5.7.2. Publishing the label back to the monitoring system

After generating the label, we utilized the MQTT protocol (publisher-subscriber communication method) to transmit the results to the monitoring system (ExaMon), subsequently triggering an alert for the system administrator.

5.8. Development and deployment

In the methodology section, we have described our methodology and how our framework works. Now, we will focus on how we developed and deployed it in the production system, as well as the technologies we used during different phases of development, testing, and deployment.

Our ThermADNet framework consists of three main components: (i) the monitoring system, (ii) the ML models development and deployment and related applications packages, libraries, etc., and (iii) the git repository and container registry. The abstraction layers of the ML component are summarized from bottom to top as follows: (i) The on-premises cloud layer hosts all the tools needed on the higher layers (e.g. OpenStack). (ii) The software platform for automating the deployment, management, and scaling of containerized applications, namely Kubernetes. (iii) The software tools for MLOps on top of Kubernetes. This layer provides an environment for the user to develop, test, and deploy the ML models, namely Kubeflow. While ML components operate on Kubernetes, the monitoring system is housed in the HPC cloud infrastructure and does not rely on Kubernetes.

Kubeflow provides a flexible framework for data analytics development and deployment, consisting of dashboards, JupyterLab, and Kubeflow Pipeline. This is implemented in micro-services using the Kubernetes container orchestration framework.

Fig. 5 illustrates our workflow for the development, testing, and deployment of the ML model (DNN Model), detailing the workflow is represented as a sequence of steps, highlighting the use of containers, registries, and various environments (Development, Staging, and Production) in different computing infrastructures (Cloud, and HPC).

The development phase targets various infrastructures, including cloud services, HPC systems. In this phase, we utilize tools from Kubeflow, such as JupyterLab and the Kubeflow Pipeline. The workflow begins in the development environment, where we create a development branch and commit code related to different aspects of the machine learning project, such as data extraction, publishing results, preprocessing, inference, training, ML pipeline, ML models, Dockerfiles, and the main application. Code changes are pushed to a GitHub repository, emphasizing the use of version control for iterative development. We set up a GitHub Action workflow. When we push code changes to GitHub, it triggers the CI/CD pipeline. This automatically executes a series of steps that build the Docker images and push them to a Docker repository. Kubernetes and the Kubeflow pipeline utilize Docker images to create Pods. These Pods establish various stages of the Machine Learning pipeline. The pipeline is created in Kubeflow using the Kubeflow Pipeline Python SDK. This automated workflow minimizes human error and ensures that the latest, tested versions of codes and models are always available for deployment, facilitating a more agile and efficient development lifecycle.

Training is carried out using SLURM jobs on the Marconi-100 HPC system. ML models require significant computing resources for training. Supercomputers are well-suited to meet this need, whereas a cloud environment may not be as effective due to the lack of dedicated accelerators.

Once the development phase is complete, a merge request is made to the staging environment. This step involves moving the development work to a more production-like setting for further testing and integration. In the staging environment, images are pulled from registries to create Pods and ML pipelines, and continuous integration (CI) triggers the execution of unit and CI tests. This phase is crucial to ensure that all codes, including the ML models, meet quality standards before being deployed to production. Successful completion of tests in the staging environment leads to the creation of a release branch, signifying that the code is ready for deployment to the production environment.

The machine learning pipeline has been deployed in a production environment. This environment is a cloud computing system that uses Kubernetes and Kubeflow. Containers are pulled from registries and deployed in the production environment. They create Pods and Machine Learning pipelines, which facilitate the operational deployment of machine learning models. The deployed ML pipeline in the production environment includes a RESTful API for interaction, a model registry for managing different versions of machine learning models, and mechanisms for publishing results with the Pub/Sub method using the MQTT protocol.

To implement the proposed framework, we employ a cloud system hosted in the CINECA supercomputing facility (on-premise) without creating any overhead on the HPC nodes. This cloud infrastructure is based on the OpenStack version of Wallaby. The nodes of this cloud system are composed of Dual-Socket Dell PowerEdge servers, 2xCPU 8260 Intel CascadeLake processors (24 cores, 2.4GHz), 48 cores per node, hyperthreading x2, 768GB DDR4 RAM, and an internal network of Ethernet 100GbE.

The OpenStack virtual machine executes the ExaMon production, which we extended with additional ones for the Kubeflow and Kubernetes Pods needed by the MLOps. The computational resources available for the ExaMon monitoring systems are 300GB of RAM and 40 vCores.

To implement the MLOps framework, we used Kubernetes version 1.24 in our framework for automated deployment, scaling, and management of containerized applications. Our Kubernetes cluster has 48 vCPUs and 360 GB of RAM available. For Kubeflow, we used the canonical Charmed Kubeflow version 1.6. We collect standard Kubernetes metrics in the monitoring system to analyze the computational cost of our framework.

6. Experimental results

In the methodology section, we proposed a framework for thermal anomaly detection. We introduced two types of autoencoders and a method for selecting the training and testing datasets through six different *dataset compositions* (A–F). Additionally, we defined the

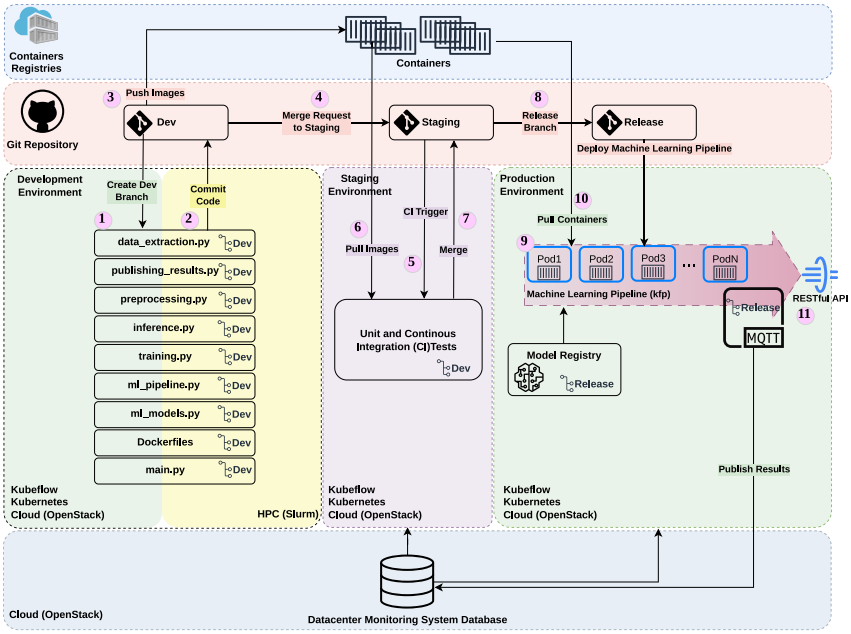


Fig. 5. Machine learning operations (MLOps) phases.

reconstruction-error threshold ϵ_{th} using four distinct *threshold strategies* (T1–T4).

In this section, we discuss our experimental results. We start by describing the dataset used for the experiments. We then assess the effectiveness of various autoencoder architectures in capturing the normal characteristics of datacenter monitoring signals. We also examine different *dataset compositions* (A–F) for the training dataset and different *threshold strategies* (T1–T4) for ϵ_{th}. Furthermore, we evaluate ThermADNet's performance in thermal anomaly detection. This includes an in-depth study of actual physical thermal failure in the datacenter. We demonstrate the framework's capability in identifying thermal anomalies and pinpointing their locations within the datacenter at both the system and subsystem levels. Finally, we reported the computational overhead of deployment on the cloud system and the inference rate.

Based on the experimental results, we aim to determine the optimal autoencoder, training *dataset composition* (A–F), and ϵ_{th} *threshold strategy* (T1–T4). Success in detecting real reported thermal anomalies (physical thermal failures) and the statistical results will guide this selection, allowing us to implement the chosen autoencoder, dataset composition, and threshold strategy into the production version of the framework.

6.1. Dataset

We utilized a holistic monitoring system, ExaMon [70], to collect monitoring signals over a four-month period (from April 8, 2021 to August 18, 2021). The data was gathered from the datacenter room hosting the Marconi100 HPC cluster at the CINECA datacenter. During the monitoring period, the system experienced a physical thermal failure on 28-07-2021.

For a general view of the monitored nodes and facilities, we have provided Fig. 1(a), which depicts the schematic of the datacenter room, facilities, and a rack. Regarding the computing nodes, we analyzed various metrics such as inlet, PCIe, CPU [0,1], GPU [0,1,2,3] temperatures, fan speed, and power supply. The racks are equipped with RDHX, and for this cooling system, we studied different essential metrics, such as water flow rate, inlet, and outlet water temperature, the position of the three-way valve, and delta temperature of the water. Furthermore, the room contains six CRAC units. For CRAC units, we studied metrics like compressor utilization, free cooling, free cooling valve open position, fan speed, return, and supply air temperature. From the main electrical power distribution system (from Modbus), we extracted the metrics: total power consumption of ICT, total power consumption of RDHX pumps, total power consumption of chillers, and total power consumption of CRAC units. Overall, we collected a total of 241 metrics from a single rack containing 20 nodes and room facilities.

6.2. Evaluation of autoencoders and dataset compositions (A–F)

LSTM outperforms MLP, and dataset composition C *is appropriate for the training dataset.* In this section, our goal is to identify the optimal autoencoder and training dataset composition for the framework, based on experimental results. We conducted these experiments using the first two months of a four-month dataset, reserving the unseen data for future steps. We trained autoencoder models (MLP-AE and LSTM-AE) using either 75 % of the semi-normal dataset or 75 % of the semi-normal dataset combined with parts of the ambiguous dataset. We used various dataset compositions as described in the Section 5 Methodology section. The remaining 25% of the normal dataset and the entire abnormal dataset are used for the test dataset, which remains fixed in all experiments. During training, samples do not have any labels, making it a form of semi-supervised learning. Fig. 6 displays the box plots of the reconstruction error for MLP-AE and LSTM-AE on the test dataset, according to six different training dataset compositions. The red boxes represent the reconstruction error of the abnormal subset of the test dataset, while the blue boxes represent the semi-normal subset of test dataset. For the sake of clarity, outliers have been eliminated. The objective is to distinguish normal from abnormal samples in the test dataset, using the autoencoder's reconstruction error. Ideally, the boxplots of the reconstruction errors for the two datasets should have minimal overlap. If the boxplots for the reconstruction errors of the normal and abnormal datasets substantially overlap, it makes classification based on these errors impractical.

Table 4 provides a quantitative basis for the visual assessment in Fig. 6. We computed the overlap coefficient (OVL) between the

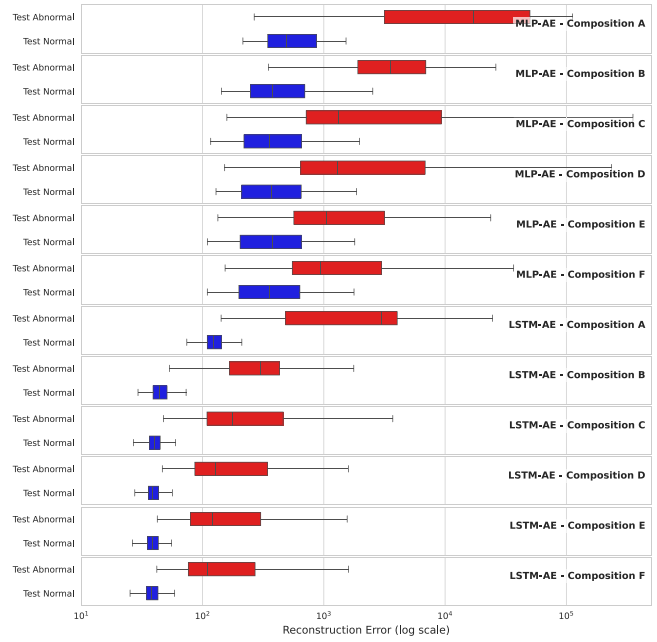


Fig. 6. Reconstruction error of MLP-AE and LSTM-AE on the test dataset for the six *dataset compositions* (A–F) used in training.

reconstruction error distributions of normal and abnormal samples for each *dataset composition* (A–F). The OVL measures the shared probability mass between two distributions, where values closer to 0 indicate better separation and values closer to 1 indicate higher overlap.

As shown in Table 4 and illustrated in Fig. 6, the MLP-AE exhibits relatively high OVL values (0.66-0.88), confirming the larger overlap observed in its boxplots. In contrast, the LSTM-AE consistently achieves very low OVL values (0.04-0.09), reflecting much clearer separation between normal and abnormal samples. This quantitative and visual evidence highlights the advantage of LSTM-AE, which can exploit the temporal dependencies present in the dataset more effectively. Consequently, LSTM-AE was selected as the deep learning model in the proposed framework.

Although *dataset compositions* B, C, and D all yield identical OVL values (0.04), the median reconstruction error separation highlights important differences. *Composition B* achieves the largest separation (Δ median ≈ 259), but its abnormal distribution is relatively narrow, which may reduce robustness to diverse anomaly patterns. *Composition D* yields the smallest separation (Δ median ≈ 89), limiting its discriminative power. *Composition C* strikes a balance: it achieves a clear separation (Δ median ≈ 138) while maintaining a broader abnormal error distribution (IQR up to 461). This balance between separation and representativeness motivated the selection of *dataset composition C* for the preprocessing stage in the offline data flow pipeline.

6.3. Reconstruction error threshold strategies (T1–T4)

In Section 6.2, we chose the LSTM-AE as the autoencoder and *dataset composition C* for the training dataset. The next step involves defining the *threshold strategy* for ϵ_{th}. We conducted nine different sets of experiments, training the LSTM-AE using nine different periods of the monitoring dataset, as reported in Table 5. The tests were conducted on a one-week monitoring dataset following the training period.

As outlined in Section 5.7, we defined four distinct *threshold strategies* (T1–T4) for ϵ_{th}. After training the LSTM-AE, we use it for inference. This involves reconstructing the input at the model's output for the test dataset. We then create a binary label for each sample by comparing the reconstruction error of the test sample with ϵ_{th}. We applied this method to nine different training periods using the four *threshold strategies* (T1–T4) to generate binary labels for test samples. The results of this experiment are presented in Fig. 7.

Table 4
Overlap coefficient (OVL) between the reconstruction error distributions of normal and abnormal samples for each *dataset composition* (A–F). Lower values indicate less overlap and thus better separation between classes.

Dataset Composition	Model	Overlap Coefficient
A	MLP-AE	0.66
B	MLP-AE	0.88
C	MLP-AE	0.77
D	MLP-AE	0.80
E	MLP-AE	0.86
F	MLP-AE	0.85
A	LSTM-AE	0.07
B	LSTM-AE	0.04
C	LSTM-AE	0.04
D	LSTM-AE	0.04
E	LSTM-AE	0.07
F	LSTM-AE	0.09

Table 5
Experiments training periods.

Experiment	Start Train	Stop Train
1	2021-06-15	2021-07-15
2	2021-04-08	2021-07-01
3	2021-04-08	2021-07-15
4	2021-04-08	2021-05-22
5	2021-04-08	2021-07-22
6	2021-06-22	2021-07-22
7	2021-04-08	2021-07-27
8	2021-04-08	2021-08-02
9	2021-04-08	2021-08-18

In this figure, the x-axis represents different quantiles, ranging from the median to the maximum value (quantile 1). The ϵ_{th} value can be calculated based on different quantiles of reconstruction errors of the semi-normal or ambiguous training dataset. The y-axis shows the average anomaly percentage for the test weeks across nine training-testing experiments. The four lines represent the results obtained with the four *reconstruction-error threshold strategies* (T1–T4). While the maximum ϵ_{th} can be attained by setting the quantile to 1, a minimum average anomaly boundary of 4 % still persists for the test weeks. This is due to the fact that a real physical thermal failure is present in the test week of three out of the nine experiments, signifying the existence of anomalies in the datacenter. As depicted in Fig. 7, T2 ($10 \leq \sum \text{Flags} \leq 25$) exhibited the lowest percentage of anomalies across quantiles, suggesting improved control over the anomaly percentage and fewer false positives. Therefore, it is a suitable candidate for defining ϵ_{th}. However, before finalizing this *threshold strategy* to compute ϵ_{th}, we should verify its ability to detect the real failure that occurred on 28-07-2021. In the subsequent section, we will verify whether the LSTM-AE with *dataset composition C* for the training dataset can accurately detect severe real physical failures as well as low-severity anomalies, by computing ϵ_{th} using *threshold strategy T2* ($10 \leq \sum \text{Flags} \leq 25$).

6.4. Evaluation of anomaly detection

Fig. 8(a) shows the results gathered from nine experiments, corresponding to the training periods outlined in Table 5. The training dataset was processed using *dataset composition C* to train the LSTM-AEs. ϵ_{th} was calculated using *threshold strategy T2*, with a quantile of 0.99. The x-axis represents the date, while the dashed red line indicates the reported real physical thermal failure that occurred on 28-07-2021. The first row displays the sum of raised flags, the second row shows the reconstruction error of the LSTM-AE for various experiments, and the third row presents the label generated by the sum of flags (The approach proposed

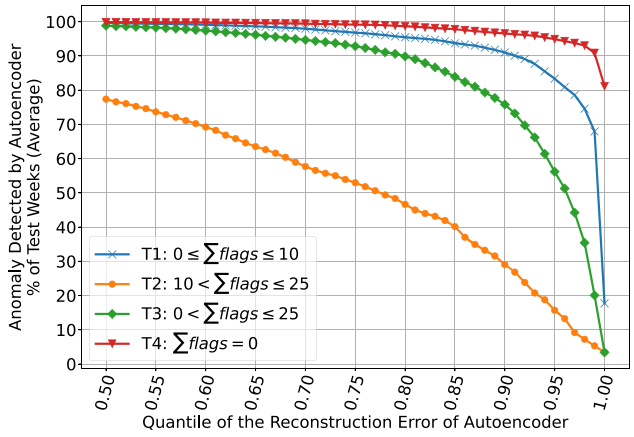


Fig. 7. Average percentage of anomalies detected across test weeks as a function of the reconstruction-error quantile. Results are shown for the four reconstruction-error threshold strategies (T1-T4).

in paper [22]). In the third row, zero flags indicate a normal condition. More than 25 flags indicate an abnormal condition, while between 1 and 25 flags fall into the ambiguous zone. The remaining rows display the waveform of the labels produced by calculating ϵ_{th} using *threshold strategy T2* ($10 \leq \sum \text{Flags} \leq 25$) for the various experiments. Two dashed black lines show the training period of each experiment, and the dashed green line shows the end of the test week, which starts just after training and lasts for one week.

Fig. 8(a) presents the entire dataset, but its details are unclear. Therefore, we have provided a zoomed-in version around the actual failure point in Fig. 8(b) for better clarity. From Fig. 8(a), several observations can be made: (i) The sum of flags reaches its maximum value at the reported real physical thermal failure. (ii) All nine experiments show peak reconstruction errors at the moment of reported physical thermal failure. (iii) All conducted experiments successfully detect the actual system failure.

Experiments 5, 6, and 7 are of particular importance due to the short distance between the training period and the reported failure and the test period in these experiments includes the reported failure. These three experiments can accurately identify the actual failure with an acceptable percentage of anomalies.

6.4.1. Validation with human-labeled anomalies

For this analysis, a human-labeled reference was compiled in collaboration with facility experts by manually inspecting monitoring signals and operator logs. These labels were produced only for the evaluation window 25–30 July 2021, which includes the real thermal failure of 28 July.

Table 6 reports the precision, recall, and F1-score of ThermADNet (LSTM-AE, threshold strategy T2, composition C) across nine training windows, benchmarked against the human labels.

Among the nine runs, Experiment 5 achieves the highest F1-score (0.97) with balanced precision and recall. Its training period ended just days before the failure, allowing the model to capture representative operating conditions. Experiments 2, 3, and 9 also perform strongly (F1 $\geq$ 0.93). In contrast, Experiments 1 and 4 show weaker results (F1 = 0.66 and 0.53, respectively), due to shorter or earlier training coverage that led to domain shift.

6.4.2. Operational day-level evaluation: stressed-day policy (baseline P90)

Fig. 8(c) reports the *daily anomaly rate* for each experiment across the full study window (10-min inference cadence). Typical days are quiet (median ≈0 % for most experiments), with occasional bursts on a small subset of days. To operationalize review, we adopt a day-level threshold defined as the *baseline P90* of the daily anomaly rate, computed on non-incident days; days exceeding this threshold are labeled *stressed* and earmarked for inspection .[2]

Table 6
Precision, Recall, and F1-score of ThermADNet (LSTM-AE, threshold strategy T2, composition C) across nine training windows, evaluated against human-labeled reference anomalies in the period 25–30 July 2021 (covering the 28-Jul thermal failure).

Experiment	Precision	Recall	F1
Experiment 1	0.49	1.00	0.66
Experiment 2	0.96	0.97	0.96
Experiment 3	0.89	0.97	0.93
Experiment 4	0.36	1.00	0.53
Experiment 5	0.97	0.97	0.97
Experiment 6	0.66	0.95	0.78
Experiment 7	0.98	0.82	0.89
Experiment 8	0.98	0.80	0.88
Experiment 9	0.97	0.95	0.96

The resulting operator workload is summarized in Table 7. Across eight of nine experiments, the mean daily anomaly rate is ∼2-4 % (≈3-6 alerts/day at a 10-min cadence), which is manageable for manual triage. At the baseline-P90 operating point, approximate false-positive fractions are in the single- to low-teens-percent range and ≈10-13 % of calendar days are classified as stressed, providing a practical workload/coverage trade-off. Experiment 4 is an outlier with higher rates, consistent with its earlier training cutoff and larger domain shift (Table 7).

The nine training windows in Table 5 were selected to probe sensitivity to recency and training-set size and to span distinct operational regimes. This choice yields materially different alert profiles-for example, Experiment 4 (earlier cutoff) shows a higher 70th-percentile daily anomaly than recency-matched windows-whereas Experiments 5-7 (short gap to the failure) represent the most realistic deployment scenario (Table 7; Fig. 8). For completeness, a sensitivity sweep of the day-threshold (percentile from P0-P100) for all nine experiments is provided in Appendix A, Fig. A.1.

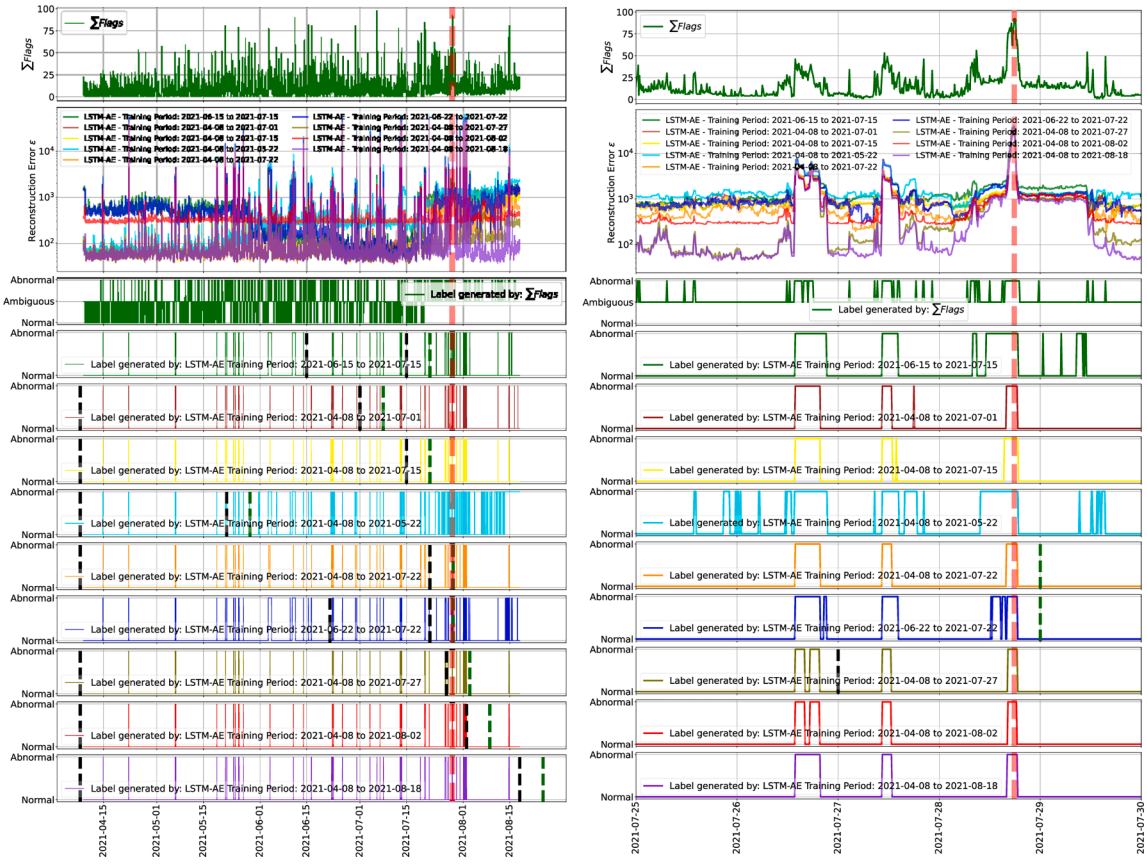
6.5. Analyzing anomalies

In this section, we provide a summary of the events surrounding the reported real physical thermal anomaly [3] on 28-07-2021. We analyzed monitoring signals and consulted with experts to compile this information.

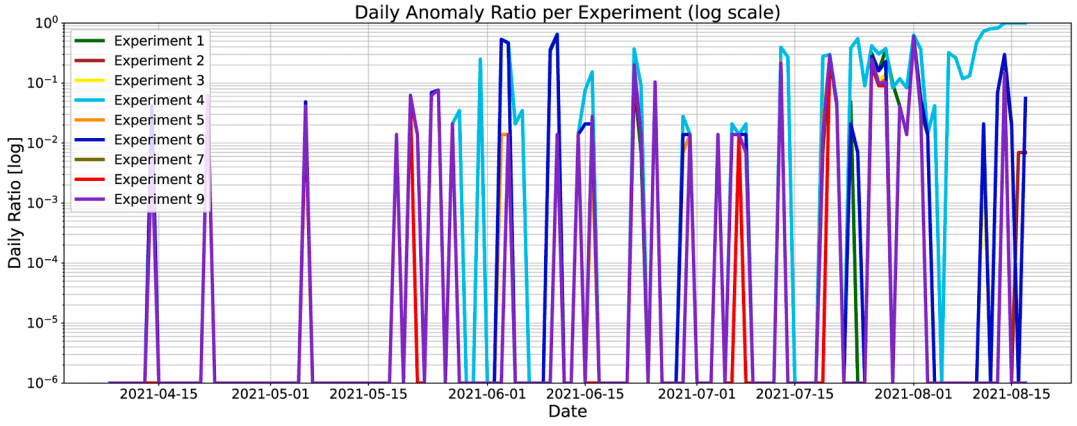
We summarized the study in three crucial snapshots before the reported failure, which had a high reconstruction error. Point A was almost a day (30 h) before the failure, point B was 12 h before the failure, and point C was the reported physical thermal failure. The heatmap in Fig. 9 shows various room and node level parameters of the datacenter on the y-axis, and the annotations within the heatmap are normalized numbers. Therefore, this figure illustrates the severity and location of the issues or anomalies that the autoencoder identified at these three points.

[2] *Baseline P90* denotes the 90th percentile of the per-day anomaly rate computed on baseline days; the 26–30 Jul window is excluded.

[3] The paper [22] presents a detailed study on the reported failure that occurred on 28-07-2021. It analyzes both node-level and room-level monitoring signals. Node-level signals include CPU, GPU, PCIe, inlet temperatures, fan speed, and power consumption for each node. Room-level signals encompass: (i) Total power consumption of the ICT devices, (ii) CRAC units' total power consumption, fan speed, compressor utilization, free cooling valve open position, outlet and inlet temperature, (iii) RDHX: total power consumption of the chillers, total power consumption of the pumps, inlet and outlet water temperature, the position of the three-way valve, and the delta temperature of outlet and inlet water, (iv) Outside temperature (ambient temperature).



(a) Entire Period. (b) Zoom-in.



(c) Daily anomaly rate per experiment (10-min inference cadence; log-scale *y*-axis). Each curve shows the fraction of 10-min intervals detected as anomalous per day across the study window. Typical days are near zero for most experiments, with experiment-dependent bursts. This motivates a data-driven day policy: mark a day as *stressed* when its daily anomaly rate exceeds the *baseline P90* threshold computed on non-incident days (26–30 Jul excluded). Summary operating metrics at P90 are reported in Tab. 7.

Fig. 8. Results of the nine experiments for computing the reconstruction-error threshold ϵ_{th} using *threshold strategy T2* ($10 \leq \sum \text{Flags} \leq 25$).

A day before the reported physical thermal failure, nodes experienced a normal inlet temperature, suggesting that the cooling systems were functioning properly and the room temperature was within the normal range. As the computing demands (GPU and CPU) increased, power consumption of the nodes began to rise. This resulted in a higher temperature within the nodes (CPU, GPU, and PCIe), causing the node fans to speed up, leading to a further increase in power consumption due to the fans. Despite the CRAC units increasing compression utilization and reducing their outlet temperature, it was insufficient to alter the inlet temperature of the nodes. Consequently, the computing loads were promptly reduced. At point A, the autoencoder identifies 73 out of the 241 sub-anomalies in various zones of node level temperatures like CPU, GPU, and PCIe and also the power consumption of nodes, and in the room level facilities, it discovers some issues mostly on water cooling system (RDHX). *Something is going to happen in the RDHX water cooling system.*

The node-level parameters at point B are normal. However, some suspicions arise at the room-level parameters. Before this point, the free cooling was activated, involving two and then three CRAC units out of four, which is the primary source of signal fluctuations in the other parts of the two cooling systems. This situation is controlled by deactivating the free cooling as well as a reduction in the computing load of the

Table 7
Per-experiment alert statistics (10-min inference cadence).

Experiment	Start Train	Stop Train	#Days	Overall Anomaly Rate (%)	Mean Daily Anomaly Rate (%)	SD of Daily Anomaly Rate (%)	Median Daily Anomaly (%)	70th-percentile Daily Anomaly (%)	Day-threshold at baseline P90 (%*)	Approx. False-positive Fraction at P90 (%)	Percent of Days Stressed at P90 (%)
Experiment 1	2021-06-15	2021-07-15	132	3.9	3.9	11.3	0	0	6.3	12.4	11.4
Experiment 2	2021-04-08	2021-07-01	132	2.0	2.0	6.8	0	0	2.6	9.1	12.9
Experiment 3	2021-04-08	2021-07-15	132	2.1	2.0	6.9	0	0	2.6	8.3	12.9
Experiment 4	2021-04-08	2021-05-22	132	12.3	12.2	23.9	0	6.25	42.6	39.9	9.8
Experiment 5	2021-04-08	2021-07-22	132	2.0	2.0	6.8	0	0	2.6	9.4	12.9
Experiment 6	2021-06-22	2021-07-22	132	3.9	3.9	11.1	0	1.39	6.5	13.7	12.1
Experiment 7	2021-04-08	2021-07-27	132	1.8	1.8	6.3	0	0	2.4	6.3	12.9
Experiment 8	2021-04-08	2021-08-02	132	1.8	1.8	6.3	0	0	2.4	6.2	12.9
Experiment 9	2021-04-08	2021-08-18	132	2.0	2.0	6.8	0	0	3.1	8.7	12.9

Table 8
Processing time and latency of deployment.

MLOps Pipeline Stage Execution Time [s]					
Data Extraction [s]	Preprocessing [s]	Inference [s]	Publishing Results [s]	Total [s]	#Inference/Hour
10.33	0.15	0.014	0.002	10.496	343

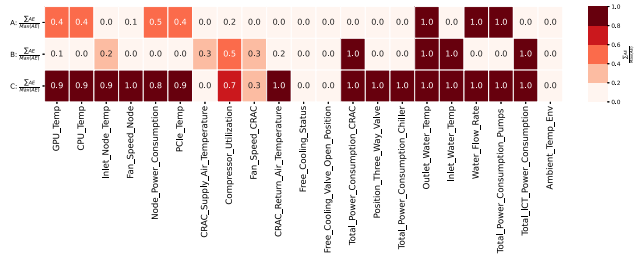


Fig. 9. Severity and zone of the anomaly in the datacenter.

room, and as it is explicit, it is successful, and there is no rise in the node level temperature. Regarding point B, the autoencoder identified high reconstruction errors in 25 out of 241 metrics. It detected some issues with the cooling system, but all systems remain under control.

After reducing point B (parameters such as the total power consumption of the ICT and CRAC units), the power consumption of the CRAC units continues to increase, peaking at point C. Before C, the free cooling was activated for four out of four CRAC units meanwhile, by activating free cooling, the power consumption of the chillers of the RDHX was reduced (due to the failure in the water cooling system), and in the same time, the computing load increased these three action 1- increasing the computing load 2- activation of free cooling and 3- reduction in chillers cooling capacity, create thermal emergency which cause an increase in the temperature of the room and temperature of the inlet and outlet water of the RDHX and inlet and outlet temperature of the CRAC units which turn into thermal emergency in the cores of nodes and it creates out of control situation in node level and room level. For point C, representing the actual physical thermal failure, the autoencoder identified issues with 204 out of 241 metrics across nearly all parts of the system.

The immediate trigger of the July 28, event was the RDHX water cooling subsystem failure; however, its impact was exacerbated by concurrent high computational load and activation of free-cooling, which together amplified the thermal imbalance.

While anomaly scores increase 12–30 h before the July 28, failure (points A and B), we interpret these as *early warning signals* of impending system stress rather than definitive failure prediction. Developing a predictive framework would require further validation and integration with forecasting methods, which we leave as future work.

6.6. Deployment evaluation

We assess the framework's deployment based on the processing time latency, as shown in Table 8, and the computational resource requirements, as illustrated in Table 9. Table 8 shows the latency in seconds for various parts of the pipeline. The final column indicates the inference rate, which is measured as the number of inferences made per hour. In pipelines, the inference rate depends on the processing time and latency of the pipeline. Data extraction is the most time-consuming step in the pipeline. Pre-processing only takes up 2 % of the data extraction latency, while the inference and result publishing steps take negligible time. This result indicates that the proposed framework can scale to exascale system requirements. Moreover, being the pipeline bottleneck, the data extraction of more complex models can be afforded with the current system at a negligible cost.

Our evaluation includes an assessment of the resources required to deploy the thermal anomaly detection pipeline. This covers HPC monitoring and the MLOps framework. In Table 9, we present the resource usage including network, CPU, and memory usage, as well as the number of Pods used. The corresponding metrics are summarized in Table 9, which shows resource usage for the baseline setup and the NN models; the results are grouped into five sub-tables, reporting the resource usage for the monitoring system ("ExaMon" sub-table), Kubernetes management, Kubeflow management, user workload not including the NN inference ("User Namespace" sub-table), and the workload due to the NN models pipeline ("ML Production pipeline" sub-table).

To better understand the implication and cost of the proposed MLOps framework in conjunction with the monitoring system, we collected resource usage data for different parts of the monitoring system, Kubernetes, and Kubeflow without running any pipelines to determine the base load of the framework. By looking at the baseline case (Baseline in Table 9), we can notice that the ExaMon framework under normal operations (continuous data collection from the different sensors and dashboards) consumes 3 virtual cores (vcores) and 190GB of memory, while the MLOps framework while not processing any data analytics pipeline uses 75 Pods (13 used by Kubernetes, 59 Kubeflow, 3 user namespace), 0.66 vcores and almost 7 GB of memory for its micro-services - almost the 22 % more vcores and 4 % more memory than the pure monitoring framework. Interestingly, when a real-time ML model pipeline is performed, the ExaMon load increases from 3.08 to 3.59. And the MLOps load increases, from 0.66 vcores to 0.75 vcores with a relatively negligible cost for real-time inference (0.01 vcores).

As a result, supporting a real-time ML model in production on the Marconi100 supercomputer requires 30 % more vcore resources than merely monitoring it. Of this 40 % increase, 16 % is attributable to the increased load on the monitoring system, while the remainder is associated with the MLOps component. The ML inference pipeline accounts for less than 1 % of the entire overhead, making it ready to scale to larger supercomputers, like exascale systems.

Table 9
HPC monitoring and MLOps framework computation resource requirements and anomaly detection pipeline deployment overhead; the 5 main sub-tables indicate the different framework's components.

Config	ExaMon				Kubernetes					Kubeflow			
	#vcores	Mem [GB]	Net in [kB/s]	Net out [KB/s]	pods	#vcores	Mem [GB]	Net in [kB/s]	Net out [KB/s]	pods	#vcores	Mem [GB]	Net in [kB/s]	Net out [KB/s]
Baseline	3.08	189.5	6670	6739	13	0.31	0.63	1350	864	59	0.22	5.44	23	28
NN Deployment	3.59	189.5	9588	7732	13	0.31	0.63	1780	880	59	0.23	5.41	26	32

Config	User Namespace				Anomaly Prediction Pipeline				
	pods	**#vcores**	**Mem [GB]**	**Net in [kB/s]**	**Net out [KB/s]**	**pods**	**#vcores**	**Mem [GB]**	**Net in [kB/s]**	**Net out [KB/s]**
Baseline	3	0.13	0.47	7	1	–	–	–	–	-
NN Deployment	3	0.02	0.91	8	1	1	0.01	0.44	1	1

7. 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:

Connection Between Monitoring and Data Preprocessing Layers: The data preprocessing and monitoring systems are currently linked through a RESTful API designed for ExaMon. If the datacenter uses a different monitoring system, minor updates may be required.

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

Threshold Modification: Modifications in the thresholds are needed, based on the target datacenter requirements.

ML Model Deployment in Production: Modifications are required in configuration based on the target datacenter on-premise cloud system.

8. Conclusion and future work

The paper introduces ThermADNet, an innovative framework designed to address the critical challenge of detecting thermal anomalies within datacenter environments, particularly in HPC systems. Leveraging a combination of statistical rules-based methods and DNN techniques, ThermADNet offers a semi-supervised learning approach by training on "semi-normal" datasets. This methodology is specifically tailored to overcome the challenges associated with large-scale data collection, identification of semi-normal datasets, and defining classification thresholds.

ThermADNet's effectiveness is underscored by its ability to successfully identify real physical thermal failure events at a Tier-0 datacenter, demonstrating its capability to pinpoint anomalies not only at the system level but also at more granular subsystem levels. This is achieved through the utilization of comprehensive monitoring signals from a variety of sensors embedded within computing nodes and datacenter infrastructure.

In the critical evaluation window of 25–30 July 2021, which includes the real thermal failure of 28 July, the framework achieves precision up to 0.97, recall up to 0.97, and F1-scores as high as 0.97 (Table 6). At the day-level operational evaluation, the system sustains a manageable alert rate of only 3–6 anomaly alerts per day (10-min cadence), with ∼10–13 % of days flagged as stressed at the P90 threshold (Table 7), offering a practical balance between sensitivity and operator workload. Deployment analysis confirms that real-time inference introduces less than 1 % overhead to the monitoring pipeline, achieving an inference rate of 343 inferences/hour with negligible CPU and memory footprint (Tables 8–9).

These quantitative results substantiate ThermADNet's contributions: (i) reliable detection of real failures with high F1-scores, (ii) practical operator workload in production environments, and (iii) production-ready efficiency with minimal runtime cost. Collectively, they confirm that ThermADNet is a scalable, high-accuracy solution for enhancing HPC datacenter resilience. By enhancing thermal anomaly detection efficiency, ThermADNet significantly contributes to improving datacenter reliability and efficiency, ensuring uninterrupted operation of critical services and averting potential economic and societal losses.

The scope for future research and development in thermal anomaly detection and related fields includes the following key areas: *Enhanced Model Generalization:* Future research could investigate the creation of more sophisticated DNN models that exhibit better generalization across various datacenter configurations and cooling technologies. This could make ThermADNet applicable in a wider range of datacenter environments without requiring major reconfiguration. *Anomaly Prediction:* Extending ThermADNet's capabilities from anomaly detection to anomaly prediction would represent a significant advancement. By predicting

potential thermal anomalies before they occur, datacenter operators could proactively manage cooling resources and workload placement, further minimizing the risk of thermal anomalies and thermal failure. *Expanded Scope of Anomalies:* While ThermADNet primarily focuses on thermal anomalies, its fundamental principles and methodologies could be tailored to detect a wider range of anomalies within datacenters, such as power fluctuations, hardware failures, and network congestion. Exploring these areas could offer a more comprehensive approach to datacenter health and efficiency.

CRediT authorship contribution statement

Mohsen Seyedkazemi Ardebili: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Andrea Acquaviva:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization; **Luca Benini:** Writing – review & editing, Writing – original draft, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization; **Andrea Bartolini:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The study has been conducted in the context of HORIZON-CL4-2022-DATA-01 project DECICE (g.a. 101092582), EuroHPC EU PILOT project (g.a. 101034126), EU's HE SEANERGYS (g.a. 101177590), EU Pilot for exascale EuroHPC EUPEX (g.a. 101033975). This work is also supported by the Spoke "FutureHPC & BigData" of the ICSC - Centro Nazionale di Ricerca in "High Performance 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 anomaly definition and ThermADNet deployment for large-scale HPC cluster production. We acknowledge the CINECA award under the ISCRA initiative, for the availability of High Performance Computing resources and support. Grammar checking and linguistic improvements were performed with the assistance of ChatGPT.

Appendix A. Stressed-day threshold selection and sensitivity analysis

This appendix reports the stressed-day threshold-sweep analysis of the *daily anomaly rate* across all nine experiments (Fig. A.1). For each experiment, we vary the baseline percentile of the daily anomaly rate (i.e., the fraction of 10-min inference intervals detected as anomalous within a day) from P0 to P100, and show the resulting trade-off between the approximate false-positive fraction (the share of alert intervals occurring on non-stressed days) and the percentage of calendar days labeled as stressed. Baseline days exclude 26–30 Jul to avoid leakage from the incident window. A day is marked stressed when its daily anomaly rate exceeds the chosen percentile threshold. All axis scales (x, left y, right y) are kept identical across panels for cross-experiment comparison.

The sweep indicates that operating points around P90-P95 provide a balanced trade-off between alert noise and day-level coverage, while consistently capturing the 28-Jul event above the threshold. Summary operating metrics at the P90 operating point-used in the main text-are reported per experiment in Table 7.

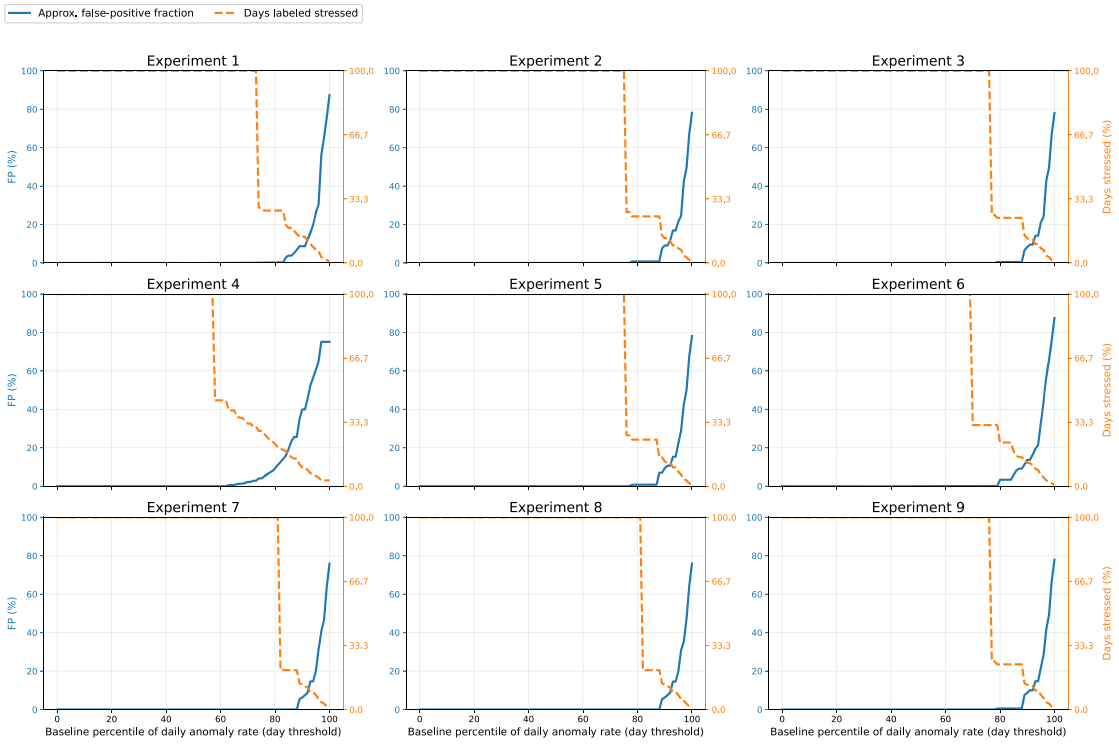


Fig. A.1. Anomaly threshold sweep across experiments. Each subplot shows the trade-off between the approximate false-positive fraction (left axis) and the percentage of days labeled stressed (right axis) as the baseline percentile of the daily anomaly rate is varied (P0-P100). The y-axis range is identical across panels.

References

[1] ACM, Getting started with HPC, 2021, https://selects.acm.org/selections/getting-started-with-hpc.

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

[3] S. Atchley, Frontier's architecture, July 12, 2022, (https://olcf.ornl.gov/wp-content/uploads/Frontiers-Architecture-Frontier-Training-Series-final.pdf). Accessed: 2024-04-06.

[4] R. Williams, The download: what's next for supercomputers, and electrifying everything, MIT Technol. Rev. (2023). https://www.technologyreview.com/2023/09/21/1079967/the-download-whats-next-for-supercomputers-and-electrifying-everything/.

[5] R. Merritt, New class of accelerated, efficient AI systems mark the next era of supercomputing, November 2023, (https://blogs.nvidia.com/blog/efficient-ai-supercomputers-sc23/). Accessed: 2024-07-02.

[6] J. Langston, Microsoft announces new supercomputer, lays out vision for future AI work, Source AI (2020). https://news.microsoft.com/source/features/ai/openai-azure-supercomputer/.

[7] ETP4HPC, ETP4HPC - the European technology platform (ETP) for high-performance computing (HPC), 2022 [Online], https://www.etp4hpc.eu/.

[8] E. Commission, AI factories, 2024, Accessed 2025, https://digital-strategy.ec.europa.eu/en/policies/ai-factories.

[9] E. Commission, Seven consortia selected to establish AI factories which will boost AI innovation in the EU, 2024, Accessed 2025, https://digital-strategy.ec.europa.eu/en/news/seven-consortia-selected-establish-ai-factories-which-will-boost-ai-innovation-eu.

[10] J.L. Hennessy, D.A. Patterson, Computer Architecture, Sixth Edition: A Quantitative Approach, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 6th edition, San Francisco, CA, USA, 2017.

[11] M. Pore, Z. Abbasi, S.K.S. Gupta, G. Varsamopoulos, Techniques to Achieve Energy Proportionality in Data Centers: A Survey, Springer New York, New York, NY, 2015, pp. 109–162. https://doi.org/10.1007/978-1-4939-2092-1_4

[12] H. Shoukourian, T. Wilde, H. Huber, A. Bode, Analysis of the efficiency characteristics of the first high-temperature direct liquid cooled petascale supercomputer and its cooling infrastructure, J. Parallel Distrib. Comput. 107 (2017) 87–100. https://doi.org/10.1016/j.jpdc.2017.04.005

[13] J. Rogers, ORNL's warm water HPC facilities and control systems, 2019,

[14] A. Bartolini, C. Conficoni, R. Diversi, A. Tilli, L. Benini, Multiscale thermal management of computing systems-the MULTITHERMAN approach, IFAC PapersOnLine 50 (1) (2017) 6709–6716. https://doi.org/10.1016/j.ifacol.2017.08.1168

[15] C. Conficoni, A. Bartolini, A. Tilli, C. Cavazzoni, L. Benini, Integrated energy-aware management of supercomputer hybrid cooling systems, IEEE Trans. Ind. Inf. 12 (4) (2016) 1299–1311.

[16] M. Seyedkazemi Ardebili, C. Cavazzoni, L. Benini, A. 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, Springer, 2019, pp. 1–16.

[17] M. Seyedkazemi Ardebili, D. Brunelli, T. Polonelli, L. Benini, A. Bartolini, A full-Stack and end-to-end IoT framework for room temperature modelling on large-Scale, Available at SSRN 4075667 (2022).

[18] M.S. Ardebili, A. Acquaviva, L. Benini, A. Bartolini, HazardNet: a thermal hazard prediction framework for datacenters, Future Gener. Comput. Syst. (2024).

[19] W.M. Organization, Climate change and heatwaves, 2023, (https://wmo.int/content/climate-change-and-heatwaves). Fact Sheet.

[20] R. Ahad, E. Chan, A. Santos, Toward autonomic cloud: automatic Anomaly detection and resolution, in: 2015 International Conference on Cloud and Autonomic Computing, IEEE, 2015, pp. 200–203.

[21] H. Jayathilaka, C. Krintz, R. Wolski, Performance monitoring and root cause analysis for cloud-hosted web applications, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 469–478.

[22] M. Seyedkazemi Ardebili, A. Bartolini, A. Acquaviva, L. Benini, Rule-based thermal anomaly HPC systems, in: High Performance Computing. ISC High Performance 2022 International Workshops: Hamburg, Germany, May 29–June 2, 2022, Revised Selected Papers, Springer, 2023, pp. 262–276.

[23] B. Arzani, S. Ciraci, B.T. Loo, A. Schuster, G. Outhred, Taking the blame game out of data centers operations with netpoirot, in: Proceedings of the 2016 ACM SIGCOMM Conference, SIGCOMM '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 440–453. https://doi.org/10.1145/2934872.2934884

[24] B. Aksar, B. Schwaller, O. Aaziz, V.J. Leung, J. Brandt, M. Egele, A.K. Coskun, E2EWAtch: an end-to-end anomaly diagnosis framework for production HPC systems, in: European Conference on Parallel Processing, Springer, 2021, pp. 70–85.

[25] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, L. Benini, A semisupervised autoencoder-based approach for Anomaly detection in high performance computing systems, Eng. Appl. Artif. Intell. 85 (2019) 634–644.

[26] B. Aksar, E. Sencan, B. Schwaller, O. Aaziz, V.J. Leung, J. Brandt, B. Kulis, M. Egele, A.K. Coskun, Prodigy: towards unsupervised Anomaly detection in production hpc systems, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2023, pp. 1–14.

[27] B.L. Dalmazo, J.P. Vilela, P. Simoes, M. Curado, Expedite feature extraction for enhanced cloud Anomaly detection, in: NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2016, pp. 1215–1220.

[28] J. Cho, T. Lim, B.S. Kim, Measurements and predictions of the air distribution systems in high compute density (internet) data centers, Energy. Build. 41 (10) (2009) 1107–1115.

[29] B. Fakhim, M. Behnia, S.W. Armfield, N. Srinarayana, Cooling solutions in an operational data centre: a case study, Appl. Therm. Eng. 31 (14–15) (2011) 2279–2291.

[30] J. Athavale, Y. Joshi, M. 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), IEEE, 2018, pp. 871–880.
[31] ECP: exascale computing project, https://www.exascaleproject.org/what-is-exascale/.
[32] A. Netti, W. Shin, M. Ott, T. Wilde, N. Bates, A conceptual framework for HPC operational data analytics, in: 2021 IEEE International Conference on Cluster Computing (CLUSTER), 2021, pp. 596–603. https://doi.org/10.1109/Cluster48925.2021.00086
[33] D. Milojicic, P. Faraboschi, N. Dube, D. Roweth, Future of HPC: diversifying heterogeneity, in: 2021 Design, Automation Test in Europe Conference Exhibition (DATE), 2021, pp. 276–281. https://doi.org/10.23919/DATE51398.2021.9474063
[34] A. Bhatele, J.J. Thiagarajan, T. Groves, R. Anirudh, S.A. Smith, B. Cook, D.K. Lowenthal, The case of performance variability on dragonfly-based systems, in: 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE, 2020, pp. 896–905.
[35] A. Bhatele, K. Mohror, S.H. Langer, K.E. Isaacs, There goes the neighborhood: performance degradation due to nearby jobs, in: SC'13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, IEEE, 2013, pp. 1–12.
[36] M. Dorier, G. Antoniu, R. Ross, D. Kimpe, S. Ibrahim, CALCioM: mitigating I/O interference in HPC systems through cross-application coordination, in: 2014 IEEE 28th International Parallel and Distributed Processing Symposium, IEEE, 2014, pp. 155–164.
[37] A. Marathe, Y. Zhang, G. Blanks, N. Kumbhare, G. Abdulla, B. Rountree, An empirical survey of performance and energy efficiency variation on intel processors, in: Proceedings of the 5th International Workshop on Energy Efficient Supercomputing, 2017, pp. 1–8.
[38] A. Agelastos, B. Allan, J. Brandt, A. Gentile, S. Lefantzi, S. Monk, J. Ogden, M. Rajan, J. Stevenson, Toward rapid understanding of production HPC applications and systems, in: 2015 IEEE International Conference on Cluster Computing, IEEE, 2015, pp. 464–473.
[39] J.M. Brandt, D. DeBonis, A.C. Gentile, J. Lujan, C. Martin, D.J. Martinez, S.L. Olivier, K. Pedretti, N. Taerat, R. Velarde, Enabling Advanced Operational Analysis Through Multi-subsystem Data Integration on Trinity, Technical Report, Sandia National Lab.(SNL-CA), Livermore, CA (United States), 2015.
[40] M. Seyedkazemi Ardebili, M. Zanghieri, A. Burrello, F. Beneventi, A. Acquaviva, L. Benini, A. Bartolini, Prediction of thermal hazards in a real datacenter room using temporal convolutional networks, in: 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2021, pp. 1256–1259.
[41] G. Pang, C. Shen, L. Cao, A.V.D. Hengel, Deep learning for anomaly detection: a review, ACM Comput. Surv. 54 (2) (2021). https://doi.org/10.1145/3439950
[42] D. Shaykhislamov, V. Voevodin, An approach for dynamic detection of inefficient supercomputer applications, Procedia. Comput. Sci. 136 (2018) 35–43.
[43] M. Marwah, R. Sharma, C. Bash, Thermal Anomaly prediction in data centers, in: 2010 12th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems, 2010, pp. 1–7.
[44] C. Li, Cooling Anomaly detection for servers and datacenters with naive ensemble, in: 2016 32nd Thermal Measurement, Modeling & Management Symposium (SEMI-THERM), IEEE, 2016, pp. 157–162.
[45] A. Netti, Z. Kiziltan, O. Babaoglu, A. Sîrbu, A. Bartolini, A. Borghesi, A machine learning approach to online fault classification in HPC systems, Future Gener. Comput. Syst. 110 (2020) 1009–1022.
[46] A. Netti, Z. Kiziltan, O. Babaoglu, A. Sîrbu, A. Bartolini, A. Borghesi, Online fault classification in HPC systems through machine learning, in: European Conference on Parallel Processing, Springer, 2019, pp. 3–16.
[47] A. Netti, D. Tafani, M. Ott, M. Schulz, Correlation-wise smoothing: lightweight knowledge extraction for hpc monitoring data, in: 2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE, 2021, pp. 2–12.
[48] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, L. Benini, Anomaly detection using autoencoders in high performance computing systems, in: Proceedings of the AAAI Conference on Artificial Intelligence, 33, 2019, pp. 9428–9433.
[49] A. Das, F. Mueller, B. Rountree, Aarohi: making real-time node failure prediction feasible, in: 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE, 2020, pp. 1092–1101.
[50] A. Borghesi, M. Molan, M. Milano, A. Bartolini, Anomaly detection and anticipation in high performance computing systems, IEEE Trans. Parallel Distrib. Syst. 33 (4) (2022) 739–750. https://doi.org/10.1109/TPDS.2021.3082802
[51] B. Aksar, E. Sencan, B. Schwaller, O. Aaziz, V.J. Leung, J. Brandt, B. Kulis, A.K. Coskun, ALBADross: Active learning based anomaly diagnosis for production HPC systems, in: 2022 IEEE International Conference on Cluster Computing (CLUSTER), IEEE, 2022, pp. 369–380.
[52] M. Molan, A. Borghesi, D. Cesarini, L. Benini, A. Bartolini, RUAD: Unsupervised Anomaly detection in HPC systems, Future Gener. Comput. Syst. 141 (2023) 542–554. https://doi.org/10.1016/j.future.2022.12.001
[53] E. Farooq, M. Milano, A. Borghesi, Harnessing federated learning for Anomaly detection in supercomputer nodes, Future Gener. Comput. Syst. 161 (2024) 673–685.
[54] Q. Tang, T. Mukherjee, S.K.S. Gupta, P. Cayton, Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters, in: International Conference on Intelligent Sensing and Information Processing, IEEE, 2006.
[55] M. Zapater, J.L. Risco-Martín, P. Arroba, J.L. Ayala, J.M. Moya, R. Hermida, Runtime data center temperature prediction using grammatical evolution techniques, Appl. Soft. Comput. 49 (2016) 94–107.
[56] L. Wang, G. Von Laszewski, J. Dayal, X. He, A.J. Younge, T.R. Furlani, Towards thermal aware workload scheduling in a data center, in: 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks, IEEE, 2009, pp. 116–122.
[57] D. Garday, J. Housley, Thermal storage system provides emergency data center cooling, White Paper Intel Information Technology, Intel Corporation (2007).
[58] X. Fang, Z. Yuan, Performance enhancing techniques for deep learning models in time series forecasting, Eng. Appl. Artif. 85 (2019) 533–542. https://doi.org/10.1016/j.engappai.2019.07.011
[59] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, G.M. Ljung, Time Series Analysis: Forecasting and Control, John Wiley & Sons, Hoboken, New Jersey, Hoboken, New Jersey, 2016.
[60] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, Q. Zhang, Time-series Anomaly detection service at microsoft, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 3009–3017.
[61] C. Choy, J. Gwak, S. Savarese, 4D spatio-temporal convnets: Minkowski convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3075–3084.
[62] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, arXiv preprint arXiv:1803.01271 (2018).
[63] M.S. Ardebili, A. Bartolini, L. Benini, Poster: multi-level anomaly prediction in tier-0 datacenter: a deep learning approach, in: ACM International Conference Proceeding Series, 2022, pp. 197–198.
[64] I.H. Sarker, Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions, SN Comput. Sci. 2 (6) (2021) 1–20.
[65] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural. Comput. 9 (8) (1997) 1735–1780.
[66] B. Aksar, Y. Zhang, E. Ates, B. Schwaller, O. Aaziz, V.J. Leung, J. Brandt, M. Egele, A.K. Coskun, Proctor: a semi-supervised performance Anomaly diagnosis framework for production HPC systems, in: International Conference on High Performance Computing, Springer, 2021, pp. 195–214.
[67] ORGANIZATION, https://www.cineca.it/en/about-us/organization.
[68] The 53th edition of the TOP500 list, JUNE 2019, https://www.top500.org/.
[69] U. Tecnico, Technical documents, 2019, https://www.cineca.it/.
[70] A. Bartolini, F. Beneventi, A. Borghesi, D. Cesarini, A. Libri, L. Benini, C. Cavazzoni, Paving the way toward energy-aware and automated datacentre, in: Proceedings of the 48th International Conference on Parallel Processing: Workshops, ICPP 2019, ACM, New York, NY, USA, 2019, pp. 8:1–8:8. https://doi.org/10.1145/3339186.3339215
[71] The 58th edition of the TOP500 list, NOVEMBER, (2021). https://www.top500.org/.