POLITECNICO DI TORINO
Repository ISTITUZIONALE

ExaMon-X: A Predictive Maintenance Framework for Automatic Monitoring in Industrial IoT Systems

Original
ExaMon-X: A Predictive Maintenance Framework for Automatic Monitoring in Industrial IoT Systems / Borghesi, A., Burrello, A., Bartolini, A.. - In: IEEE INTERNET OF THINGS JOURNAL. - ISSN 2327-4662. - 10:4(2023), pp. 2995-3005. [10.1109/JIOT.2021.3125885]

Availability:
This version is available at: 11583/2978557 since: 2023-05-16T15:15:44Z

Publisher:
IEEE-INST ELECTRICAL ELECTRONICS ENGINEERS

Published
DOI:10.1109/JIOT.2021.3125885

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

24 June 2026

ExaMon-X: a Predictive Maintenance Framework for Automatic Monitoring in Industrial IoT Systems

Andrea Borghesi, Alessio Burrello, and Andrea Bartolini

***Abstract*—In recent years, the Industrial Internet of Things (IIoT) has led to significant steps forward in many industries, thanks to the exploitation of several technologies, ranging from Big Data processing to Artificial Intelligence (AI). Among the various IIoT scenarios, large-scale data centers can reap significant benefits from adopting Big Data analytics and AI-boosted approaches since these technologies can allow effective predictive maintenance. However, most of the off-the-shelf currently available solutions are not ideally suited to the HPC context, e.g., they do not sufficiently take into account the very heterogeneous data sources and the privacy issues which hinder the adoption of the cloud solution, or they do not fully exploit the computing capabilities available *in loco* in a supercomputing facility. In this paper, we tackle this issue, and we propose an IIoT holistic and vertical framework for predictive maintenance in supercomputers. The framework is based on a big lightweight data monitoring infrastructure, specialized databases suited for heterogeneous data, and a set of high-level AI-based functionalities tailored to HPC actors' specific needs. We present the deployment and assess the usage of this framework in several in-production HPC systems.**

***Index Terms*—High Performance Computing, Industrial IoT, Industry 4.0, Predictive Maintenance, Artificial Intelligence**

I. Introduction

High-Performance Computing (HPC) systems are large and complex industrial plants, which are gaining importance in today's society and industry [1]. Recent reports quantify the Return on Investment (ROI) produced by applying HPC in an industrial environment: in Europe, each Euro invested in HPC generates, on average, 69€ in profit, while in the US, a single dollar spent in HPC generates, on average, 43$ of profit [1]. HPC systems are hosted in computing rooms, each containing multiple racks that pack tens/hundreds of computing nodes, each composed of various computing elements (CPUs/GPUs) based on multi/many-core processors. The high complexity, diversity, and heterogeneity of components of HPC systems pose enormous management challenges to maintain them operational. The complexity, scale, and cost make HPC infrastructure a perfect candidate for IIoT and industry 4.0 applications.

A. Borghesi, Alessio Burrello and A. Bartolini are with the University of Bologna, Department of Computer Science and Engineering and Electrical Engineering, Bologna, Italy. Copyright (c) 2021 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

With increasing scale, the complexity of the maintenance rises as well (e.g., the sheer number of monitored components can overwhelm human operators), together with the operating costs[2], [3]. Fortunately, the wealth of data stirred the interest in the adoption of Artificial Intelligence (AI) methods for predictive maintenance, especially from the fields of Machine and Deep Learning (ML & DL)[4], [5]. The term *predictive maintenance* indicates a set of practices to intervene on IIoT systems *before* critical conditions might arise. This new approach is in contrast with the normal state of practice, which is based on either reactive strategies (fix a problem after its detection, risking to cause disservice) or planned maintenance interventions. However, these methods tend to result in too conservative policies (incurring in higher-than-necessary costs). Deploying AI methods in an environment with strict data security and access control requirements, such as an HPC system, imposes the end-users to develop and deploy themselves the IIoT software (SW) tools and data analytics functionalities. Hence, existing big data analytics tools are not well suited for IIoT and HPC scenarios, albeit recent research directions started to address the challenge [6], [7].

In previous works, we described a vertical and holistic IIoT monitoring framework for HPC systems called ExaMon, aimed at collecting a large amount of data from a variety of heterogeneous sources. The lower layers of ExaMon framework dealt with data gathering and storage[8]. In this paper, we present a new set of functionalities to abstract the low-level data engineer and analysts' steps into domain-specific and more comfortable to understand functionalities. The end-users of these functions are the system administrators, the facility managers, and user support managers, who have heterogeneous backgrounds and may lack in-depth data analysis knowledge. The set of functionalities offered by ExaMon-X differ from other off-the-shelf solutions since they are tailored to the specific data center use case; in addition, they are completely and seamlessly integrated with an existing monitoring infrastructure deployed in several production systems. This constitutes a necessary first step towards the creation of a *digital twins* of the data center. The digital twin is a model that represents a system and its performance, enabling the description of the system itself and its dynamics, predicting its evolution, and optimising its operation, management, and maintenance. The tight integration of the low-level monitoring infrastructure and data management middleware and of the high-level AI-boosted data analytics tools is a key element for this

purpose (as highlighted by related works in the industrial domain [9]); to the best of our knowledge, very few research works adopt this vertical approach in data center and supercomputing facilities.

These functionalities were derived from the lessons learned while deploying AI methods to production HPC systems for predictive maintenance and extend ExaMon. This suite of additional functionalities is referred to as ExaMon-X. We show a set of end-to-end applications of IIoT for predictive maintenance on a datacenter. We are the first to demonstrate the deployment of predictive maintenance on a real, in-production datacenter leveraging an IIoT infrastructure.

The main contributions of this paper are:

- an empirical analysis of the predictive maintenance requirements on a real in-production datacenter;
- domain-specific functionalities for IIoT predictive maintenance on datacenters, called ExaMon-X, selected and adapted to the datacenter context after years of preliminary analysis on real production systems, e.g., identifying the most suited techniques for the common use cases and taking into account specific characteristic (heterogeneous data sources, high-performance computing capability, etc.);
- an implementation of these functionalities as an extension of ExaMon framework;
- a demonstration of the deployment and usage of ExaMon-X on an in-production tier-0 datacenter, in multiple scenarios, hence the first application of a vertical and holistic approach in a real supercomputer.

The paper is organized as follows. Section II fleshes out the IIoT and HPC context. Sec. III overviews related works Sec. IV introduces ExaMon-X, and Sec. V reports the empirical evaluation on different use cases, and Sec. VI concludes the paper.

II. IIoT and HPC Systems

Data centers and supercomputing systems are large-scale systems composed of numerous components (up to millions) to run parallel applications (or jobs) with high computational demands, such as scientific and industrial simulations[10]. Supercomputers are IIoT systems where a huge set of hardware (HW) and SW resources can be monitored to characterize their behavior and, potentially, create digital twins for predictive maintenance tasks[11], [12], [9].

HPC systems host several actors with different roles, each with various requirements and use cases, often belonging to other organizations[13]. In this work, we classify four main categories of actors: i) system administrators, that are the people in charge of the correct functioning of the data center (from determining the job scheduling policies to fixing broken components, etc.); ii) facility managers, concerned with system-wide issues, such as energy consumption, mortgage costs, and thermal/cooling problems[14]; iii) system accountants, who need to manage and keep track of the different accounts and projects, granting access to users, and providing reports and statistics on the usage of the machines[15]; iv) HPC users, industrial and academic partners who submit jobs, and are typically interested in fast completion times and fair prices[16]. Finally, the diversity of roles implies different use-cases, which often require *ad-hoc* analytics solutions[17].

A specialized IIoT infrastructure for data collection and storage is required to overcome these issues, accompanied by HPC-tailored analytics and modeling functionalities[18], [19], [20]. The scope of this paper is to introduce such an infrastructure, described in Sec. IV. We will also see how we employed "standard" IIoT components and remodeled them for the HPC use-case.

Scenarios: We have identified four main use-cases that greatly benefit from holistic monitoring and big data analytics: 1) anomaly detection, 2) fault prediction, 3) job power models, and 4) thermal models for the computing resources. Anomaly detection and fault prediction are fundamental tasks for system administrators since understanding which components are broken among thousands of different sources of errors can be extremely difficult – being capable of forecasting the faults themselves – would allow scheduling maintenance operations with minimal disruption of the productivity. In the HPC field, these tasks are hugely complicated by the scarcity of labels (annotation is a costly process) and by the lack of anomalous events: datacenters are built to have as little downtime as possible, and faults are sporadic. Hence, data imbalance must be considered while creating predictive maintenance models. System administrators keep track of the system's status in real-time; this is currently done via a disparate set of text-based status monitoring services. A graphical interface offering aggregate views of the live data enriched with fault-related information would be a great boon for their daily work in terms of ease of use and capacity to effectively convey a significant amount of information.

Facility managers are interested in faults as well. Additionally, their task is to efficiently take care of the whole infrastructure, devoting a keen interest in thermal issues that could impact the cooling systems. In this regard, thermal models describing the real-time situation (via live data graphs) of the room hosting the computing nodes and the processing elements inside each compute node to predict the evolution of the thermal dynamics would have tremendous value. Job power models can estimate HPC applications' power (and energy) before their actual execution. This would be of great interest for system administrators (to improve job dispatching strategies [21]) and to users alike, as new pricing policies could be adopted [16]. To obtain such models, different data sources must be combined (job dispatcher information and power consumption of HW devices) and possibly aggregated values should be computed (e.g., mean values); a uniform point of access capable to abstract the multiple data sources and databases (DBs) would be beneficial.

III. Related Works

The interest in data analytics tools and ML/DL models for predictive maintenance in the IIoT field has skyrocketed in recent years. As the deluge of data collected from various sensors has been increasing, the usage of such data has gained even more importance. For this purpose, AI models showed great promise, and, in turn, big players in the market started offering a cloud-based solution to analyze big data and build ML/DL models on top of it, such as Amazon Web Service, Microsoft Azure, Google Cloud AI, Apache Spark [22], etc. We sum up the main features of these companies' cloud in Table I. Note that all the providers have their main advantages in the hardware as a service (HaaS) and Infrastructure as a Service (IaaS), while providing little flexibility on deploying personal resources and the integration with a custom data collection system. The aim of this paper radically differs from the general-purpose solutions (Table I), since we are rather explicitly focused on the IIoT and datacenter areas, with the integration of data collection of an HPC center and predictive maintenance applications (Software as a "HPC" Service – SaaHS). In this context, ad-hoc techniques and tailored models allow offering more specific solutions suited for the challenges faced by HPC systems managers, administrators, and users – facilitating their adoption through improved ease of access and interpretation.

Thus, the most appropriate comparisons for ExaMon-X are methods for data collection and analysis targeting IIoT and datacenters. In this area, the SoA is still mostly focused on the data collection infrastructure rather than developing SW tools to analyze the data, neglecting building predictive maintenance models. Very few holistic, vertical solutions encompassing all layers (from data gathering and storage to processing and analysis) have been proposed. Agelastos et al. [23] describe Lightweight Distributed Metric Service (LDMS), a distributed data collection, transport, and storage tool, which has been recently enhanced by Izadpanah et al. [24] to integrate streaming analysis capabilities. These approaches focus on offering analytics capabilities without implementing actual models. Bautista et al. [25] describe an infrastructure for extreme-scale operational data collection, known as OMNI; they focus almost exclusively on the backbone of the architecture, and in terms of higher-level services, they consider the only visualization.

In recent years, many HPC systems have started to adopt Operational Data Analytics (ODA) techniques, which extract knowledge from massive amounts of monitoring data and use it to create "digital twins" of the supercomputer improved maintenance and optimization. ODA techniques have been applied in many areas. For instance, when dealing with complex fault situations, e.g., systemic anomalies involving multiple components that cannot be detected (let alone predicted) simply by looking at threshold values and expected ranges. To this end, in recent years, several research works have attempted to build more sophisticated failure detection and prediction models. DL models are incredibly successful in contexts where anomalous behaviours might arise in a wide variety of domains [26], [27], [28]. These problems are generally framed as supervised learning tasks and trained to distinguish between abnormal and normal points. For example, Tuncer et al.[29], [30] and Netti et al.[31] both use Random Forest classifier to detect synthetic anomalies injected on historical data set collected from real supercomputers. When labels are absent or scarce, semi-supervised learning has been proposed; for instance, Borghesi et al. [32] use a type of neural network called autoencoder to learn the expected behaviour of a supercomputer and discern anomalous states due to their difference. This approach is analogous to other outlier detection techniques, such as Baseman et al. [33].

Similarly, ODA has been applied for extensive monitoring of storage systems based on hard drives. Noteworthy, in huge HPC centers, the number of failures per month can be significant and impair the computation's reliability. In Xiao et al [34], a Random Forest has been employed, reaching 81.55% in predicting the hard disk failure on a time horizon of 7 days. An LSTM is used in [35] for the same task, reaching an 81.61% accuracy but with a much more complex network. Another area where ODA has shown benefits in the supercomputing context is power consumption. To implement effective policies to handle power consumption, it is generally required to have information about job power consumption before job execution. For this purpose, supervised ML models trained on historical data sets have been shown to produce very accurate results. Different types of regression models have been employed, from Support Vector Machines and linear models (see Sirbu et al.[36], [37]) to Random Forest (see Borghesi et al.[38]).

In the direction of holistic monitoring, Netti et al. propose *Wintermute* [39], a generic framework for online data analytics large-scale in HPC installations. Wintermute is composed by I) a manager handling the communication with data-gathering backbone, II) a query engine that exposes the available sensors to the plugins; III) the operator plugins that perform analysis; when a new analytics task needs to be performed, a new plugin needs to be developed. The main disadvantage of Wintermute is that users have to develop the plugins to perform the desired analysis and specify the sensors to be fed to the plugin, limiting Wintermute adoption to users with expert knowledge on both HPC domain and analytics models. On the contrary, ExaMon-X was created to assist inexpert users facing predictive maintenance tasks. Its functionalities can be used without developing additional components, nor it requires detailed knowledge of the underlying data collection framework.

IV. Framework Description

We identify two main blocks forming the infrastructure for the holistic monitoring of a datacenter: 1) ExaMon, the low-level data collection and storage backbone, and

	AWS	Azure	Google	Our Work
API & Depl.	G.P.	G.P.	G.P.	HPC Spec.
Hardware	Cloud/Buy	Cloud/Priv.	Cloud	Priv.
Data	Re./Loc.	Re./Loc.	Re.	Loc.
Cloud C.	✓	✓	✓	✗
Local C.	✓(Outpost)	✓(Stack)	✗	✓

TABLE I: Comparison of big clouds providers. Abbreviations. G.P.: General Purpose, Priv.: Private, Re.: Remote, Loc.: Local, C.: Computing.

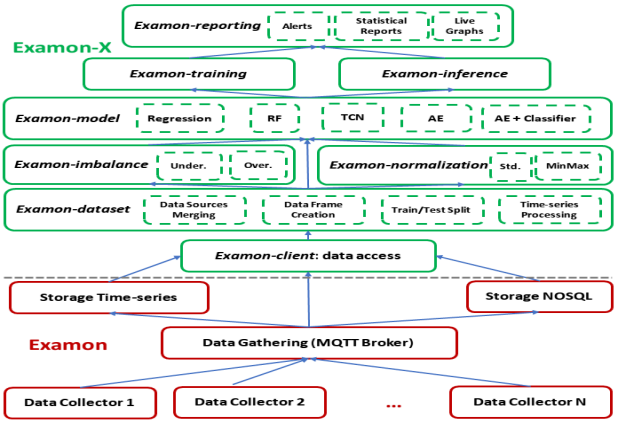

Fig. 1: The infrastructure architecture

2) ExaMon-X, the mid- and high-level components with the goals of retrieving and processing the stored data, providing a unified point of access to multiple data sources, and building ML/DL models for predictive maintenance. As the description of ExaMon has already been published (see [8]), we offer a synthetic summary to highlight its salient features. Instead, we will focus on the data processing and analytics component, ExaMon-X. Fig. **??** portrays the overall infrastructure. ExaMon is a IIoT holistic framework for HPC facility monitoring and maintenance, designed very large scale computing systems, such as supercomputers. It has been developed for the Exascale, thus stressing the capability to handle big data from many heterogeneous sources. At the lowest level, there are *collector* components to read the data from several sensors scattered across the system and deliver them, in a standardized format, to the upper layers of the stack. There are collectors with direct access to HW resources and collectors that sample data from other applications, such as batch schedulers and SW diagnostic tools. The infrastructure is built using the MQTT protocol[40]. MQTT is not the only communication protocol available for IoT scenarios; for instance, possible alternatives are AMQP (Advanced Message Queuing Protocol)[41], DDS (Data Distribution Service) [42], CoAP (Constrained Application Protocol) [43], OPC UA (OPC Foundation Unified Architecture) [44], etc. ; for a more comprehensive survey we refer to the work of Dizdarevic et al.[45]. However, we chose MQTT protocol for a variety of reasons: 1)it is lightweight, 2) it is highly scalable and guarantee high availability, 3) it is an ISO standard, 4) highly maintained as it is among the most popular IoT protocol (if not the most popular one), and it is deployable on all infrastructures (edge, data center, etc.). Silva et al.[46] recently surveyed a variety of communication protocols for IoT and experimentally demonstrated that MQTT is the one with the lowest time-to-completion (albeit with some variability), while guaranteeing sufficient quality of service (e.g., MQTT guarantees the message arrival while others such as OPC UA are best efforts). MQTT implements the publish-subscribe messaging pattern and requires three different agents: (i) the publisher, that sends data on a specific topic; (ii) the subscriber that receives data from the appropriate topic; (iii) the broker, which handles the communication between publishers and subscribrers. In ExaMon collector agents have the role of publishers.

The collected metrics are stored on a distributed and scalable time series DB, KairosDB, built on top of a NoSQL DB, Apache Cassandra as back-end. A specific MQTT subscriber (MQTT2Kairos) is implemented to bridge the MQTT protocol and the KairosDB data insertion mechanism. The bridge leverages the particular MQTT topics structure to compose the KairosDB insertion statement automatically. This gives a twofold advantage: first, it lowers the computational overhead of the bridge since it is reduced to a string parsing operation per message; and secondly, it makes it easy to form the DB query starting only from the knowledge of the matching MQTT topic.

A. ExaMon-X

The paper's main contribution is a suite of functionalities for predictive maintenance, which can exploit both the data stored in the DB and the online data stream – ExaMon-X. The functionalities are explicitly tailored to HPC requirements, such as handling large amount of heterogeneous data. ExaMon-X users can use the proposed functionalities independently to perform simple tasks (e.g., retrieving data from the DB or processing it) or combining them to perform macro-activities composed of a sequence of functions (e.g., anomaly prediction with a DL model). The use cases described in Sec. V show the combination of multiple simpler functions for high-level goals. We implemented ExaMon-X in Python.

We grouped the proposed functionalities into eight different subgroups. 1) A "client" interface for the data collection and storage back-end (ExaMon), devoted to retrieving data and exposing it to other services – *Examon-client* offers three access modes: a) noSQL queries, b) SQL-like queries, and c) time traces queries; the different types of data access are helpful as some tasks require a large historical data set (e.g., DL models training), while others need fast access to streaming data (e.g., real-time thermal prediction based on regression models). 2) *Examon-dataset* functionalities deal with data management and representation, providing services to merge multiple data sources (data collected in HPC systems usually comes from thousands of different sensors monitoring different components of the machines), create data frames, and split the data in training, validation, and test sets; additionally, specific pre-processing functions fix potential issues from the removal of invalid values and gaps filling (e.g., time-series pre-processing through value interpolation) to

categorical values encoding. As seen in Sec. II, imbalanced data sets are a common problem in the HPC context; 3) *Examon-imbalance* functions address this issue via data augmentation (oversampling, i.e., the creation of synthetic data to increase the number of samples in the minority class) or reduction (undersampling, i.e., removal of samples from the majority class).

As data collected in HPC systems come from many heterogeneous sources, they tend to have different orders of magnitude, depending on the sensor type (e.g., clock frequency measured in Hertz or temperature measured in Celsius degrees) and calibration. ML/DL models are easier to train if data share the same scale – 4) *Examon-normalization* functions provide this functionality by applying standardization (subtract the mean value and scaling to unit variance to obtain normally distributed data. Then 5) *Examon-model* functionalities create the actual ML/DL models. At the moment, the models offered are the following: a) regression models (linear and non-linear), b) Random Forest (RF), c) Temporal Convolutional Neural Networks (TCN), d) semi-supervised autoencoder networks (AE), e) autoencoder networks plus a supervised classifier. These models are implemented using Python modules Scikit-learn and Tensorflow and were selected as the most suited to the HPC context, as demonstrated in years of experimental evaluation with IIoT and datacenters. The details of the ML models used will be provided in the section describing the use cases (Sec. V).

After having created the model, 6) the training takes place using the services in *Examon-training*; the training can either use streams or batches of online data or historical data sets stored in ExaMon long-term DBs. The same three options are also available at inference time, when the trained model is put into use; 7) inference functionalities are grouped in the *Examon-inference* area. In most scenarios, training will employ historical data, as a large amount of data leads to more accurate DL models; the inference is typically used on streams or batches of online data. 8) Finally, *Examon-reporting* regards the presentation of the results, done via statistical (textual) reports, live graphs (for instances with tools such as Grafana), or alerts reporting services.

Table II summarizes ExaMon-X functionalities, together with their algorithmic complexity. The leftmost column indicates the macro-area. The second column lists the functions implemented in each category; the third column reports the complexity of the used technique; the last column maps the functions to the four use cases (using the acronym for saving space). As complexity, we report that of the general case. Consider for instance *Examon-client*. The complexity of database queries strongly depends on the operation (e.g. read or write) and on the presence of specialized support structures (e.g., indexes of various types). As it would have been extremely space-consuming (and probably not useful for a reader), we report only the complexity of the read operation in the general case, that is $O(log(N))$, with N being the number of rows/tuples. For some functionalities, no complexity was reported (indicated with the string "n.a." in the corresponding row), as it would be trivial or not relevant, e.g., producing a report or the creation of a DL model, which corresponds to simply *defining* the model.

Ex-Abstr.	Ex-API	Complexity	Use-Cases
Ex-client	noSQL query	O(log(N))	AD
	Time traces query	O(log(N))	AP, TP
	SQL-like	O(log(N))	JP
Ex-dataset	Data frame	n.a.	AD, JP
	Sources merge	O(N×log(N) +M×log(M))	AD
	Time-series proc.	O(N)	AP, TP
	Train / test split	O(1)	AD, AP, JP
Ex-imb.	Undersampling	O(N)	
	Oversampling	$O(\sum_{i=0}^{Ti}(N_i \times f \times k))$	AP
Ex-norm.	Standardization	O(N×f)	AD
	MinMax	O(N×f)	AP, JP
Ex-model	Regression Models	n.a.	TP
	RF	n.a.	JP
	TCN	n.a.	AP
	AE	n.a.	AD
	AE + Classifier	n.a.	AD
Ex-train	RealTime	*	TP
	Batch	*	
	Historical	*	AD, AP, JP
Ex-infer	RealTime	*	AD, TP
	Batch	*	AD, AP, JP
	Historical	*	JP
Ex-report	Alerts	n.a.	AD, AP
	Statistical Report	n.a.	AD, AP, JP
	Live graphs	n.a.	AD, TP

∗ Complexity is directly related to the machine/deep learning model employed.

TABLE II: The components of ExaMon-X and their intersection with the considered use cases. Legend: N: number of samples, M: number of second source samples, Ti: number of classes, f: features, k: neighbours, AD: Anomaly Detection, AP: Anomaly Prediction, JP: Job Power Prediction, TP: Thermal Prediction.

V. Experimental Use Cases

ExaMon-X has been deployed in different production supercomputers hosted by CINECA[1] and used to investigate multiple research directions. In this section, we describe the different machines where ExaMon-X is adopted; we discuss applications and insights gained through its execution, together with the HPC maintenance improvement. Fig.2 shows the workflows for the four use cases, highlighting the usage of ExaMon-X functionalities.

A. Anomaly Detection in HPC Systems

As HPC facilities comprise several thousands of parts, a critical issue for operating ever-increasing machines is the identification of faulty, broken, or miss-behaving components, namely the *detecting anomalies*. This is a non-trivial challenge that is typically performed by system administrator with non-negligible costs, hence the research of automated methods for anomaly detection has been a really hot topic. ExaMon and ExaMon-X have been deployed on two supercomputers hosted at CINECA, and used to detect faulty node states, performing the detection both on the cloud and on the edge; the supercomputers are D.A.V.I.D.E.[48] (where ExaMon was deployed since November 2017 until December 2019, at its dismissal) and Marconi[2], the tier-0 system currently in production (where ExaMon started to be operative in mid-2018).

[1]The largest Italian computing centre, residing in Bologna. ExaMon has been deployed on CINECA machines since 2017[47].

[2]https://www.hpc.cineca.it/hardware/marconi

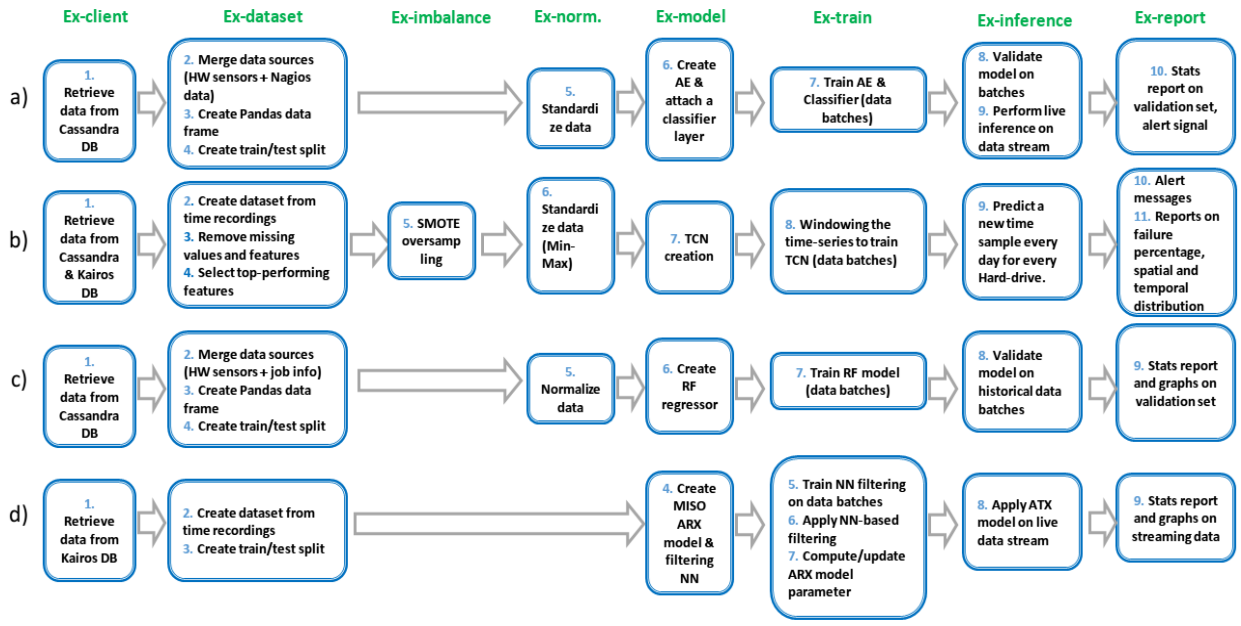

Fig. 2: ExaMon-X workflows for the four use cases: a) AD, b) AP, c) JP, d) TP.

System	*# nodes*	*# metrics*	*DB rate (# metrics/s)*
Marconi	3448	633344	8934.05
D.A.V.I.D.E.	45	6750	75.32
Marconi100	968	175572	11005.43
Total	4461	815666	20014.8

TABLE III: Data collected by ExaMon

D.A.V.I.D.E. is an HPC system based on Power architecture, composed by 45 nodes with a total peak performance of 990 TFlops and an estimated power consumption of less than 2 Kwatt per node. The tier-0 Marconi system has been the main production supercomputer since 2016; the current system is composed by 3188 nodes, each equipped with two 24-cores Intel Xeon 8160 (SkyLake) processors and 196GB of RAM memory. The total peak performance of the overall system is around 20PFlops, with 17PB available storage space.

On both D.A.V.I.D.E. and Marconi we collect: i) physical data measured with sensors; ii) workload information obtained from the job dispatcher (SLURM). The sampling rate on both supercomputer was 5 seconds (every five seconds a MQTT packet is built and sent to the broker; ExaMon low-level plugins are in charge with aggregating sensors measurements with higher sampling rates). On Marconi there is also an additional class of information, namely iii) information about the state of the system and its services collected by Nagios[49], a tool to provide alerts for system administrators – we use Nagios to annotate the collected data classifying the samples as describing HPC nodes in normal or anomalous state. Given the difference in available data, two DL approaches were applied: I) a semi-supervised method (on D.A.V.I.D.E.), where only partially labeled data is required, and a II) supervised method (Marconi), where labels are available.

1) D.A.V.I.D.E.: In the D.A.V.I.D.E. case there exist already several published works detailing the effectiveness of the approach[32]. To summarize, it is based on a autoencoder neural network, trained *using only normal data* (one network for supercomputing node); in this way the autoencoder learns the correct behaviour of a node. The reconstruction error is then used to recognize anomalous data points. The semi-supervised approach was experimentally proven to be very effective, with accuracy in the 90%-95% range – on par with most methods from the state-of-the-art. In particular, the average weighted F-score over all tested nodes is 0.935, with standard deviation equal to 0.072.

2) Marconi: As in D.A.V.I.D.E. case, in Marconi as well we consider node-level anomalies, but now the faults are not only due to wrong configurations but can be generated by a variety of causes. To train the models in a supervised setting, we exploit the labels collected via Nagios. The workflow adopted in Marconi is depicted in Figure 2a. The initial step is to retrieve the data from ExaMon backbone, using *Examon-client*. The raw data is then processed to be made more suitable for learning tasks: first, data coming from multiple sources is aggregated, merging the measurements coming from physical sensors on Marconi computing nodes with the the node status registered by Nagios (the labels in this task). The merged data is then formatted as Pandas data frame, which are in turn divided in training, validation, and test sets; all these actions were performed with *Examon-dataset* functions. After data preprocessing, the most suitable DL model is chosen and created, namely an undercomplete autoencoder network for feature extraction; this autoencoder is pre-trained in an unsupervised manner (minimizing the reconstruction error). On top of the autoencoder two classification layers have been added, culminating in a softmax layer; these latter layers are trained using the labels (provided by Nagios) by minimizing the categorical cross-entropy. Fig. 3 portrays the scheme of the autoencoder network plus the

classifier layers. The user can either specify the desired hyperparameters that characterize the DL model (number of layers, batch size, etc), or can rely on the fine-tuning mechanism implemented by *Examon-model*; the fine-tuning is performed on validation data and it is based on Bayesian Optimization (we exploit the Python library *hyperopt* [50]). After the model creation, it has to be trained using batches of historical data (*Examon-train*) and then validated and used, both on batches of data or directly on data streams (*Examon-inference*.

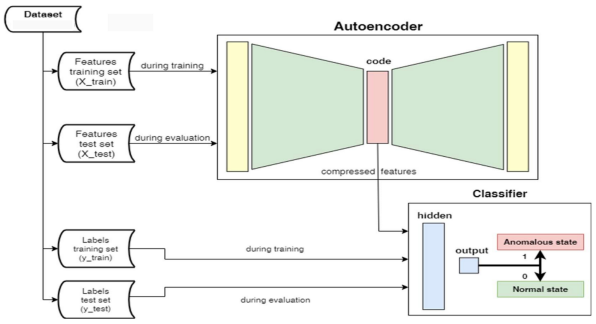

Fig. 3: Autoencoder network plus classifier scheme

Finally, the classifier is tested on a subset of the data set collected in each node. Table IV reports the results of the experimental evaluation conducted on data collected from 20 nodes of Marconi in two distinct months of production, January and May 2020. For comparison, we report as well the results obtained with the semi-supervised autoencoder ("AE Semi-sup."), to highlight the benefits brought by the labels in terms of detection accuracy. Overall, the results are very good, with an F-score equal to 0.85, considering both time periods. These numbers are slightly worse compared with other supervised methods from the literature, such as [30], [31], where accuracy is in the 95%-98% range. However, we are the first to demonstrate the potential of the supervised ML approach on a real supercomputer, with actual (and not injected) anomalies.

B. Failure Prediction of Hard Disks

The reliability of Hard Disk (HD) in HPC centers is a critical problem. Despite single HDs having very low failure probability (corresponding to a high Mean Time To Failure (MTTF)), given the high number of HDs in an HPC center, the likelihood that "at least one HD fails

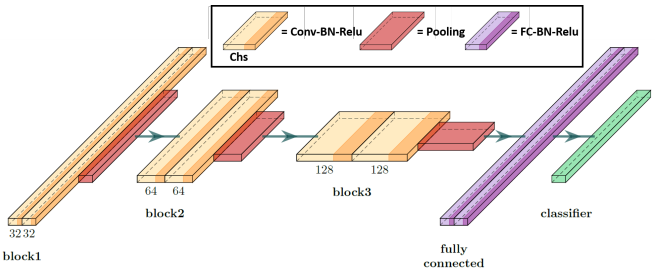

Fig. 4: TCN structure used for HDD failure prediction.

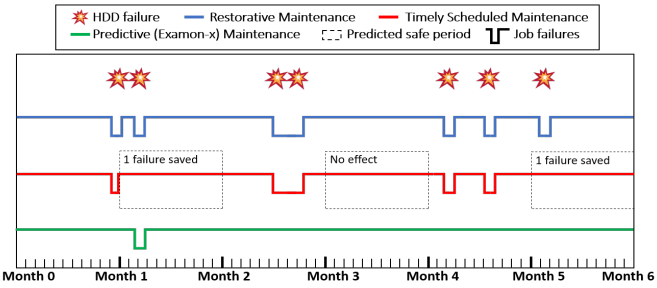

Fig. 5: Job failure under different scheduling policy.

in this week" could be an order of magnitude higher. In particular, if Y indicates the failure event, $\widetilde{X}$ the probability of an HD to be in a normal state, and n the number of HDs in the HPC center, the probability above can be computed as $P[Y] = 1-(P[\widetilde{X}])^n$. Although $(P[\widetilde{X}])$ is ~ 1, with $n \to \infty$, we have $(P[\widetilde{X}])^n \to 0$.

Current solutions for this problem are based on the redundancy of HDs, using different levels of Redundant Array of Independent Disks (RAID) to store multiple copies of data, and on hot-spare disks, unused HDs that can replace the broken ones. While these systems are expensive and can lead to a shut down of the services, modern solutions include predictive maintenance techniques that couple a continuous monitoring framework of disks with ML predictive algorithms to anticipate HDs failure. A flag is raised when the algorithm predicts an incoming failure, allowing to replace the HD, copy the data, or schedule a maintenance intervention [51]. To monitor the health status of drives, the producers provide Self-Monitoring Analysis and Reporting Technology (S.M.A.R.T.), a series of features including disk temperatures, failures in reading/writing sensors, and many others. In our case, to improve the current HD monitoring, we collect these SMART features utilizing ExaMon, that daily stores the S.M.A.R.T. features from all the HDs. On top of this, ExaMon-X is used to process these data (workflow depicted in Fig.2b). It provides a series of flags for each HD and daily periodicity, to alert a human controller to check the HD status. Notably, using ExaMon-X we can automatically pair the data collection of S.M.A.R.T. features stored in a time-sequential DB, with the predictive algorithm.

The core of the pipeline is constituted by a Temporal Convolutional Network depicted in Fig. 4, a new network topology based on Convolutional Neural Networks (CNN) that has demonstrated excellent performance for time series processing [52]. The main difference compared to classical CNNs is the introduction of the *Dilation*, which increases the receptive field of 1D convolutions without increasing the filters sizes by applying a fixed step d between the input samples processed by each filter. We deploy on ExaMon-X a TCN with 3 convolutional blocks for our failure prediction. Each block contains two convolutional layers and one pooling layer each, which expand the channels to 32, 64, and 128, respectively, while reducing

Month	Method	TP	TN	FP	FN	TNR	TPR	Prec.	Recall	F-score
Jan. 2020	AE Semi-sup.	7.4	5489.0	199.56	15.1	0.96	0.33	0.04	0.33	0.06
Jan. 2020	AE + Classr.	19.7	5669.6	9.1	2.8	1.0	0.88	0.68	0.88	0.77
May 2020	AE Semi-sup.	47.9	6732.4	237.6	146.9	0.97	0.25	0.17	0.25	0.2
May 2020	AE + Classr.	186.4	6932.3	50.4	8.3	0.99	0.96	0.79	0.96	0.86
Jan. & May	AE Semi-sup.	55.3	12221.4	437.16	162.0	0.97	0.25	0.11	0.25	0.16
Jan. & May 2020	AE + Classr.	206.1	12601.9	59.5	11.1	1.0	0.95	0.78	0.95	0.85

TABLE IV: Experimental results for the anomaly detection task performed in Marconi supercomputer. The results obtained on two months of data collected are reported; all values are average computed over 20 computing nodes. Jan. & May rows are obtained by summing the two periods. TP: True Positives; TN: True Negatives; FP: False Positives; FN: False Negatives; Prec.: Precision; TNR: True Negative Rate; TPR: True Positive Rate

the time dimension by a factor of 2. On top of these blocks we added three fully-connected layers for failure prediction. Since the beginning of this study, we have not experienced any failure in the subset of monitored HDs. We thus decided to train our TCN using a benchmark for HD failures – Blackblaze DB[3]. The data processing executed previously to TCN application are the same of [53]. The benchmark results are very promising, with a failure detection rate (FDR) of 89.1%.

By applying ExaMon-X to collect and process the S.M.A.R.T. features for predictive maintenance of HDs, we could achieve a substantial reduction of maintenance cost. In Fig. 5, we report the difference between three maintenance policies in a simulated scenario: restorative, timely-scheduled, and predictive. Considering a FDR of 89.1%, we can prevent 9 out of 10 disk failures without reducing the system redundancy or interrupting its service, as it always happens in restorative maintenance. On the other hand, we have a non-null probability of predicting a false alarm, resulting in an unnecessary substitution. However, HDs vendors freely provide new HDs to replace end-of-life HDs, which are tested and possibly commercialized again. Thus false alarms are not too expensive.

C. Job Power Prediction

As energy consumption is a widely recognized issue for supercomputers, there exist several techniques to reduce it[47]. Software-based techniques usually operates at the job scheduling level, by balancing the workload according to the estimated power consumption. Thus, it is very important to predict the power consumption of HPC jobs *before* their execution; this can be done with ExaMon-X by merging different data sources (e.g., job dispatcher and HW sensors) to build predictive models. We collected 12 months of data from Marconi and used it to train ML models and to test their predictive capabilities; *Examon-client* was used to retrieve data stored in Cassandra DB via SQL-like queries. For every completed job ExaMon stores information available at submission time, such as job user, account, requested duration and HW resources, etc; the metrics collected during the lifetime of the job on the execution nodes were also collected and stored in an aggregated fashion (mean value over the whole job duration). Hence, we have the average power consumption

[3]https://www.backblaze.com/b2/hard-drive-test-data.html

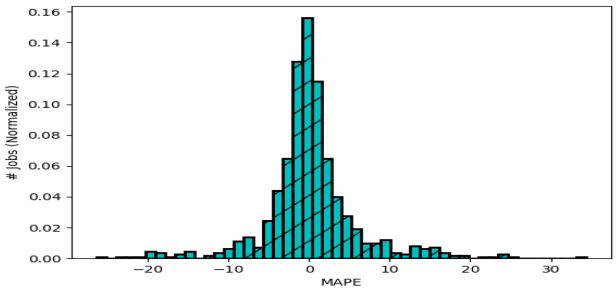

Fig. 6: Job power prediction use case: errors histogram

for each completed job, and we used this data to train and test a ML model. We opted for a Random Forest regressor, as previous works in the HPC settings revealed it to be a good choice, in terms of model's accuracy and ease of training [38].

The accuracy of the power estimation is by the prediction error. The Absolute Percentage Error is computed as $\frac{pred\_value - real\_value}{real\_value} * 100$. The Mean Absolute Percentage Error (the average value computed considering all jobs in the test set), or MAPE, is equal to 7.35%, quite a good value. Figure6 plots the histogram of the estimation errors. The x-axis specifies the percentage error while in the y-axis there is the number jobs predicted with such error. The histogram agrees with the results of the numerical analysis, with the vast majority of jobs having a percentage error smaller than 5%.

D. Thermal Prediction in HPC Systems

Optimizing and capping a data center's power consumption tackles only the first half, namely the power distribution, of the compute nodes' power dissipation issue. The second half consists of the thermal dissipation problem. Each core in the processors of the data center's computing node needs to operate below a critical temperature, after which the silicon starts to wear out. By leveraging the proposed ExaMon-X interface, we aim to predict each core's temperature inside the processors, relying on the dissipated power and the previous temperature of the core and of its neighboring cores. Unlike the last use case, here we aim to extract a compact but physically valid, predictive model by means of numerical regression. For this purpose, we choose to implement an Auto-Regressive with eXogenous input model (ARX), which matches the regression model of an input-output dynamic system. Previous works ([54]) revealed that the ($\pm 1^o C$) quantization

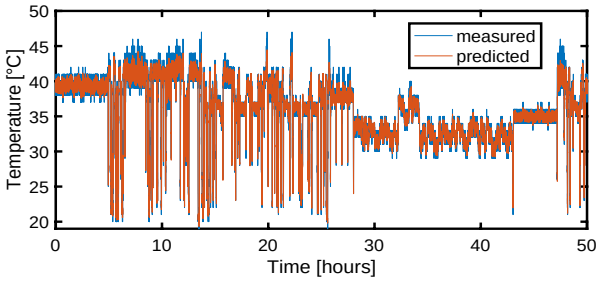

Fig. 7: Marconi system: prediction vs. actual temperature.

error in the temperature readings of the core temperature's integrated sensors induces a bias in the ARX model. To overcome this limitation, an approach proposed in the SoA is to use a MISO ARX model with additive output noise, which we will refer to as ARX+noise in the remainder[55]. The MISO model, which represents the thermal dynamics of a single core of a node, and links the core's temperature (model output) to the powers of all the node's cores (model inputs), is reported in Eq.(1)

$$\bar{T}(t) + \sum_{i=1}^{n} a_i \, \bar{T}(t-i) = \sum_{k=0}^{N_c} \sum_{i=1}^{n} b_{ki} \, P_k(t-i) + w(t) \quad (1)$$

$$T(t) = \bar{T}(t) + v(t), \quad (2)$$

where N_c is the number of cores of the node, n is the model order, $\bar{T}(t)$ is the actual (unknown) core temperature, $P_1, \ldots, P_{N_c}$ are the dissipated powers of all the cores of the node, and P_0 denotes the power dissipated by other components which are not cores. $w(t)$ is the equation error (process noise), assumed to be a zero-mean white process with variance σ_w^2. $T(t)$ is the measured core temperature, $v(t)$ is the additive measurement error, assumed to be a zero-mean white process with variance σ_v^2, uncorrelated with $w(t)$. The parameters a,b are the regression parameters that compose each core's learned compact thermal model in the system.

Albeit the ARX+NOISE model matches the physics of the HPC thermal dissipation problem (as shown in fig.7), obtaining a estimation error below $\pm 1^oC$ requires a training set with input traces with a persistent excitation condition[55]. Unfortunately, it is impossible to guarantee that each segment of a user application generates a white input stress pattern in the core's power consumption. It is then required to divide the input-output monitored trace into sub-segments and to apply a pre-filtering action to each of them; the filter consists of a convolutional networks (CNN) trained to detect if a given segment fulfils the persistent excitation constraints and could lead to robust parameters for the regression model. The architecture is made of four 1D convolutional layers, each one doubling the number of channels of its input, followed by max-pooling. Finally in the last layer there is an adaptive average pooling to bring down the dimension of the time trace to one followed by a softmax classification layer. In the training phase, a dropout layer was also inserted before the average pooling, with a drop probability of 0.5. As loss function we used binary cross-entropy.

Fig.2d depicts the block diagram of the proposed data processing pipeline in ExaMon-X, where the NN-based filtering is done before updating the a,b regression parameters used to forecast the processor temperatures (implemented using the Ex-inference Real-Time functionality).

Method	Performance	Notes
Anomaly Detection in HPC		
ExaMon-X	F-score: 0.85	Production Anomalies
Tuncer et al.[30]	F-score: 0.96	Simulated Anomalies
Borghesi et al.[32]	F-score: 0.93	Injected Anomalies
Failure Prediction in Hard Disks		
ExaMon-X(TCN)	Acc.: 89.1%	Blackbaze dataset
Xiao et al [34]	Acc.: 81.55%	Blackbaze dataset
Aggarwal et al [35]	Acc.: 81.61%	Blackbaze dataset
Job Power Prediction		
ExaMon-X	MAPE: 7.35	Production system
Sirbu et al.[36]	MAPE: 4.5	Pre-production system
Borghesi et al.[38]	MAPE: 8.2	Pre-production system
Thermal Prediction in HPC		
Diversi et al.[54]	Acc.: 24%	w/o NN filter
ExaMon-X	Acc.: 96%	w. NN filter

TABLE V: Comparison with state-of-the-art

E. State-of-the-Art Comparison & Time complexity

After having considered each use case singularly, we provide a synthetic overview for comparing ExaMon-X results and the state-of-the-art, shown in Table V. There are three columns: the first one indicates the method used (presenting both ExaMon-X and the top-performing approaches from the literature), the second column reports the performance metric, and the final column provides some additional notes highlighting key aspects of the data.

Although several performance metrics can be adopted, we chose the most representative one for each use case (according to the literature): 1) F-score for anomaly detection, 2) accuracy for failure prediction, 3) Mean Average Percentage Error for the power prediction, and 4) accuracy of the correctly classified windows (as percentage) for thermal prediction. As state-of-the-art, we selected the best performing techniques described in Sec. III, and we report the performance numbers presented in the papers; in the thermal prediction use case (Sec. V-D) there is no directly comparable technique, thus we report the improvement to the ARX model obtained by using the MISO ARX+noise model and the CNN-based pre-filtering. The reported numbers clearly demonstrate how ExaMon-X has a performance that is on par with the state-of-the-art while being extraordinarily generic and easy to use.

Finally, in Table VI we also report the times required to perform the different tasks which make the use cases up; the measurement unit is provided in the table. For each use case there are two columns, one reporting the time needed to perform the operation indicated by the corresponding row (see the left-most column) and one providing additional details. In the anomaly detection (AD) use case the training time refers to the time required

	AD		AP		JP		TP	
	Time	Notes	Time	Notes	Time	Notes	Time	Notes
Ex-Client	22 hours	1.2 TB of data	45 min	40.9 GB of data	4 min	3.6 GB of data	7.6 hours	411 GB of data
Ex-dataset	42 min	Dataframe (df.)	36 min	Time-windowing & df.	2 min	df.	n.a.	n.a.
Ex-norm	1 min	Standardization	1 min	Min-Max	10 s	Normalization	n.a.	n.a.
Ex-imb.	n.a.	n.a.	1 min	Oversampling	n.a.	n.a.	n.a.	n.a.
Ex-train	15 min	1 model per node	10.5 min	n.a.	6 min	n.a.	134.4 min	16 models per node
Ex-infer	12 ms	1 window	70 ms	Prediction for 1 HDD	2 s	n.a.	1.32 s	16 models per node

TABLE VI: Temporal requirements for the each use case: AD: Anomaly Detection in HPC, AP: Anomaly Prediction in HDs, JP: Job Power Prediction, TP: Thermal Prediction in HPC.

to train a node-specific model, as in this case each HPC node has an associated model; in the thermal prediction (TP) case we report the time required to train the 16 models relative to a specific HPC node - in this case there is a model for each core.

VI. Conclusion

This paper introduces ExaMon-X, a suite of functionalities for predictive maintenance in HPC systems, developed using ML techniques. ExaMon-X is built on top of ExaMon, a framework for data gathering and processing built for IIoT and big data capabilities. As the HPC context poses unique challenges, general-purpose tools (e.g., cloud and data analysis providers) are not well suited. On the contrary, the computational resources available in supercomputers allow for local deployment of data analytics capacities. The final result of our work is a vertical and open-source infrastructure that enables the adoption of tailored solutions for optimizing the management of HPC systems and predictive maintenance. We describe the functionalities and their implementation and then show how they were deployed and used in real in-production supercomputers. We demonstrate the effectiveness of ExaMon-X in tackling various challenges in the HPC context, considering real, production HPC machines.

Several challenges to the wide adoption in the HPC context still need to be addressed. In the short term, the DL models presented in this paper need to be enhanced and extended. For instance, the failure prediction is currently restricted to specific components of the supercomputing nodes (the hard disks). At the same time, we plan to study how to forecast/anticipate the insurgency of node-wide anomalies (i.e., moving from anomaly detection to anomaly prediction). Another critical aspect in IIoT scenarios is the capability to estimate the Remaining Useful Life of components. We plan to add to ExaMon-X new DL models for this purpose (in particular, we started preliminary experiments with Long-Short Term Memory Networks). Additionally, so far we have not fully considered two broad classes of faults: I) bugs in the applications submitted by users and II) internal network problems; in both cases, we should potentially need to install additional sensors in the monitoring infrastructure (ExaMon) to create a more accurate digital twin of the supercomputer. In the long term, we will move from predictive maintenance to *prescriptive* maintenance. The predictive models will be used within an optimization framework to assist the decision making of system owners and administrators. For instance, the models predict when an HW component is going to break and are used to plan a maintenance schedule that minimizes the risk of downtime while keeping costs under a given budget.

Acknowledgements

This research was partly supported by the EU H2020-ICT-11-2018-2019 IoTwins project (g.a. 857191), the H2020-JTI-EuroHPC-2019-1 Regale project (g.a. 956560)re.

References

[1] ETP4HPC. (2017) Strategic research agenda. [Online]. Available: http://www.etp4hpc.eu/sra.html

[2] X. Yang, Z. Wang, J. Xue, and Y. Zhou, "The reliability wall for exascale supercomputing," *IEEE Transactions on Computers*, vol. 61, no. 6, pp. 767–779, 2012.

[3] L. A. Parnell, D. W. Demetriou, V. Kamath, and E. Y. Zhang, "Trends in high performance computing: Exascale systems and facilities beyond the first wave," in *2019 18th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, 2019, pp. 167–176.

[4] S. Heldens, P. Hijma, and et al., "The landscape of exascale research: A data-driven literature analysis," *ACM Comput. Surv.*, vol. 53, no. 2, Mar. 2020.

[5] A. Kanawaday and A. Sane, "Machine learning for predictive maintenance of industrial machines using iot sensor data," in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 2017, pp. 87–90.

[6] S. Caíno-Lores, J. Carretero, B. Nicolae, O. Yildiz, and T. Peterka, "Toward high-performance computing and big data analytics convergence: The case of spark-diy," *IEEE Access*, vol. 7, pp. 156 929–156 955, 2019.

[7] A. Netti, D. Tafani, M. Ott, and 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.

[8] F. Beneventi, A. Bartolini, C. Cavazzoni, and L. Benini, "Continuous learning of hpc infrastructure models using big data analytics and in-memory processing tools," in *Proc. of the Conference on Design, Automation & Test in Europe*. European Design and Automation Association, 2017, pp. 1038–1043.

[9] A. Borghesi, G. Di Modica, and et al., "Iotwins: Design and implementation of a platform for the management of digital twins in industrial scenarios," in *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. IEEE, 2021, pp. 625–633.

[10] P. Czarnul, J. Proficz, and K. Drypczewski, "Survey of methodologies, approaches, and challenges in parallel programming using high-performance computing systems," *Scientific Programming*, vol. 2020, 2020.

[11] D. El Baz, "Iot and the need for high performance computing," in *2014 International Conference on Identification, Information and Knowledge in the Internet of Things*. IEEE, 2014, pp. 1–6.

[12] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li, "Predictive maintenance in the industry 4.0: A systematic literature review," *Computers & Industrial Engineering*, p. 106889, 2020.

[13] A. Geist and D. A. Reed, "A survey of high-performance computing scaling challenges," *The International Journal of High Performance Computing Applications*, vol. 31, no. 1, pp. 104–113, 2017.

[14] M. Maiterth, G. Koenig, and et al., "Energy and power aware job scheduling and resource management: Global survey—initial analysis," in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 2018, pp. 685–693.

[15] Q. Wang, Y. Shen, and J. Li, "User-level workload analysis for supercomputers," in *The 4th International Conference on SW Engineering and Information Management*, 2021, pp. 68–73.

[16] A. Borghesi, A. Bartolini, and et al., "Pricing schemes for energy-efficient hpc systems: Design and exploration," *The International Journal of High Performance Computing Applications*, vol. 33, no. 4, pp. 716–734, 2019.

[17] A. Netti, M. Ott, C. Guillen, D. Tafani, and M. Schulz, "Operational data analytics in practice: Experiences from design to deployment in production hpc environments," *arXiv preprint arXiv:2106.14423*, 2021.

[18] Z. Zong, R. Ge, and Q. Gu, "Marcher: A heterogeneous system supporting energy-aware high performance computing and big data analytics," *Big data research*, vol. 8, pp. 27–38, 2017.

[19] B. Xie, Y. Huang, and et al., "Predicting output performance of a petascale supercomputer," in *Proc. of the 26th International Symposium on High-Performance Parallel and Distributed Computing*, 2017, pp. 181–192.

[20] A. L. d. C. D. Lima, V. M. Aranha, C. J. de Lima Carvalho, and E. G. S. Nascimento, "Smart predictive maintenance for high-performance computing systems: a literature review," *The Journal of Supercomputing*, pp. 1–20, 2021.

[21] A. Borghesi, A. Bartolini, and et al., "Scheduling-based power capping in high performance computing systems," *Sustainable Computing: Informatics and Systems*, vol. 19, pp. 1–13, 2018.

[22] M. Zaharia, R. S. Xin, and et al., "Apache spark: a unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[23] A. Agelastos, B. Allan, and et al., "The lightweight distributed metric service: a scalable infrastructure for continuous monitoring of large scale computing systems and applications," in *SC'14: Proc. of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2014, pp. 154–165.

[24] R. Izadpanah, N. Naksinehaboon, and et al., "Integrating low-latency analysis into hpc system monitoring," in *Proc. of the 47th International Conference on Parallel Processing*, 2018.

[25] E. Bautista, M. Romanus, and et al., "Collecting, monitoring, and analyzing facility and systems data at the national energy research scientific computing center," in *Proc. of the 48th International Conference on Parallel Processing*, 2019, pp. 1–9.

[26] R. Dwivedi, S. Dey, C. Chakraborty, and S. Tiwari, "Grape disease detection network based on multi-task learning and attention features," *IEEE Sensors Journal*, 2021.

[27] A. Abbasi, A. R. Javed, C. Chakraborty, J. Nebhen, W. Zehra, and Z. Jalil, "Elstream: An ensemble learning approach for concept drift detection in dynamic social big data stream learning," *IEEE Access*, vol. 9, pp. 66 408–66 419, 2021.

[28] A. Kishor, C. Chakraborty, and W. Jeberson, "A novel fog computing approach for minimization of latency in healthcare using machine learning," *Int J Interact Multimed Artif Intell*, vol. 1, no. 1, 2020.

[29] O. Tuncer, E. Ates, and et al., "Diagnosing performance variations in hpc applications using machine learning," in *International Supercomputing Conference*. Springer, 2017.

[30] ——, "Online diagnosis of performance variation in hpc systems using machine learning," *IEEE Transactions on Parallel and Distributed Systems*, 9 2018.

[31] A. Netti, Z. Kiziltan, and et al., "A machine learning approach to online fault classification in hpc systems," *Future Generation Computer Systems*, 2019.

[32] A. Borghesi, A. Bartolini, and et al., "A semisupervised autoencoder-based approach for anomaly detection in high performance computing systems," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 634–644, 2019.

[33] E. Baseman, S. Blanchard, and et al., "Interpretable anomaly detection for monitoring of high performance computing systems," in *Outlier Definition, Detection, and Description on Demand Workshop at ACM SIGKDD. San Francisco (Aug 2016)*, 2016.

[34] J. Xiao, Z. Xiong, and et al., "Disk Failure Prediction in Data Centers via Online Learning," in *Proc. of the 47th International Conference on Parallel Processing*, ser. ICPP 2018. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: https://doi.org/10.1145/3225058.3225106

[35] K. Aggarwal, O. Atan, and et al., "Two birds with one network: Unifying failure event prediction and time-to-failure modeling," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 1308–1317.

[36] A. Sîrbu and O. Babaoglu, "Power consumption modeling and prediction in a hybrid cpu-gpu-mic supercomputer," in *European Conference on Parallel Processing*. Springer, 2016, pp. 117–130.

[37] ——, "Predicting system-level power for a hybrid supercomputer," in *2016 International Conference on High Performance Computing & Simulation (HPCS)*. IEEE, 2016, pp. 826–833.

[38] A. Borghesi, A. Bartolini, and et al., "Predictive modeling for job power consumption in hpc systems," in *International conference on HPC*. Springer, 2016, pp. 181–199.

[39] A. Netti, M. Mueller, C. Guillen, M. Ott, D. Tafani, G. Ozer, and M. Schulz, "Dcdb wintermute: Enabling online and holistic operational data analytics on hpc systems," *arXiv preprint arXiv:1910.06156*, 2019.

[40] O. Standard, "Mqtt version 3.1. 1," *URL http://docs. oasis-open. org/mqtt/mqtt/v3*, vol. 1, 2014.

[41] ——, "Oasis advanced message queuing protocol (amqp) version 1.0," *International Journal of Aerospace Engineering Hindawi www. hindawi. com*, vol. 2018, 2012.

[42] J. Yang, K. Sandström, T. Nolte, and M. Behnam, "Data distribution service for industrial automation," in *Proc. of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012)*. IEEE, 2012, pp. 1–8.

[43] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (coap)," Tech. Rep., 2014.

[44] T. J. Burke, "Opc unified architecture: Interoperability for industrie 4.0 and the internet of things," *Opc Foundation,[si]*, vol. 1, pp. 01–44, 2017.

[45] J. Dizdarević, F. Carpio, A. Jukan, and X. Masip-Bruin, "A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–29, 2019.

[46] D. Silva, L. Carvalho, and et al., "A performance analysis of internet of things networking protocols: Evaluating mqtt, coap, opc ua," *Applied Sciences*, vol. 11, no. 11, p. 4879, 2021.

[47] A. Bartolini, F. Beneventi, and et al., "Paving the way toward energy-aware and automated datacentre," in *Proc. of the 48th International Conference on Parallel Processing: Workshops*, 2019, pp. 1–8.

[48] W. A. Ahmad, A. Bartolini, and et al., "Design of an energy aware petaflops class high performance cluster based on power architecture," in *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, May 2017, pp. 964–973.

[49] W. Barth, *Nagios: System and network monitoring*. No Starch Press, 2008.

[50] J. Bergstra, D. Yamins, and D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *International conference on machine learning*, 2013, pp. 115–123.

[51] F. D. S. Lima, F. L. F. Pereira, and et al., "Remaining Useful Life Estimation of Hard Disk Drives based on Deep Neural Networks," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–7.

[52] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[53] A. Burrello, D. J. Pagliari, and et al., "Predicting hard disk failures in data centers using temporal convolutional neural networks," in *Accepted at Euro-Par 2020: Parallel Processing Workshops*, 2020.

[54] R. Diversi, A. Bartolini, and L. Benini, "Thermal model identification of computing nodes in high-performance computing systems," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 9, pp. 7778–7788, 2020.

[55] F. Pittino, R. Diversi, and et al., "Robust identification of thermal models for in-production high-performance-computing clusters with machine learning-based data selection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2042–2054, 2020.