

# A Reliability Score for Video-Analytics Edge-IoT Devices in Urban Environments

Marçal Garcia<sup>\*1</sup>, Joan Oliveras Torra<sup>\*2,1</sup>, David Aguilera-Luzón<sup>1</sup>, Peini Liu<sup>2</sup>,  
Mario José Diván<sup>3</sup>, Josep Lluís Berral<sup>1,2</sup>  
*Universitat Politècnica de Catalunya<sup>1</sup>, Barcelona Supercomputing Center<sup>2</sup>, INTEL Corp.<sup>3</sup>*  
{marcal.garcia, david.aguilera.luzon, josep.ll.berral}@upc.edu, {joan.oliveras, peini.liu}@bsc.es,  
mario.jose.divan.koller@intel.com

**Abstract**—Performing video-analytics in Edge and IoT devices towards urban mobility allows computing near-data, providing fast and autonomous management from streets and roads. Thanks to advances in artificial intelligence methods, real-time classification of camera-provided image feeds can be performed in low-power devices to be deployed next to urban signaling and city services. However, devices deployed “in the wild” are affected by environmental factors such as temperature, either ambient or retained by its container, humidity or voltage changes, coming from sensorized urban furniture. Deciding where to deploy video-analytics workload requires a minimal device performance reliability, assuring that load can be processed in-place, before moving it towards the Fog/Cloud or other nearby devices. Here we present a simplistic factorization of a Reliability Score to be quickly computed in Edge nodes, towards orchestrating workload according to the node surrounding environment. The score takes into account ambient temperature and humidity, also received voltage. The scoring is evaluated performing video-analytics workloads on real devices under physical stress, observing the performance degradation related to the external and internal factors applied onto the device. From the study, we observe how the different factors affect the functioning mode and quality of service, aside of indicating an estimation of the reliability of the urban-placed devices.

**Index Terms**—Edge Computing, Reliability, Environmental Stress, Urban Mobility Video Analytics

## I. INTRODUCTION

Pervasive computing networks are expanding rapidly as the Edge and the Internet of Things are opening new opportunities for processing data and analytics near users. These hyper-distributed networks are composed by low-power machines, traditionally used as data-retrieval nodes, off-loading analytics into Cloud or High-Performance Computing (HPC) infrastructures. But such approach presents

This work is financed by the INTEL UFunding #14780, the EU-HORIZON and MSCA programmes under GA.101092646 and GA.101086248, by Generalitat de Catalunya (AGAUR) GA.2021-SGR-00478, and the Spanish Ministry of Science (MICINN), the Research State Agency (AEI) and European Regional Development Funds (ERDF/FEDER) PID2021-126248OB-I00, MCIN/AEI/10.13039/501100011033/FEDER, UE.

challenges on scalability, autonomy, reliability and availability, as those networks do not have always assured connectivity towards the Cloud. Furthermore, off-loading data outside of devices might pose a risk to users’ privacy. For this, recent approaches look towards computing at the same Edge and IoT nodes, adapting applications for low-power resources, amortizing the consumed power, and reducing the need for continuous connectivity.

Current scenarios implement Edge-Cloud architectures in which Edge nodes retrieve data to be transmitted to Cloud nodes, where resource-demanding applications ingest and transform leveraging virtually infinite scalable infrastructures. Edge nodes are considered *low-power* due to low computing capabilities in comparison to Cloud nodes. However, data and video analytics applications are evolving towards efficiency, reducing the resources demand, allowing such processing to be performed on the same Edge. The principal problem becomes the uncertain reliability of those nodes, deployed in “exposed to the elements” environments with physical stress, e.g., managing urban mobility through road-side placed nodes and video-cameras. Physical stress comes from the nature elements, such as insolation, temperature, humidity, even changes in supplied voltage. Such stress must be considered when deploying or migrating load from those nodes, considering the current status of the Edge node. External factors and internal load degrade the Quality of Service (QoS) of those applications running inside, delaying or slowing their execution and throughput.

In this paper we propose a Reliability Scoring metric *RS* considering physical stress factors in *low-power* devices, result of temperature, humidity and variable voltage exposure towards device and workload Quality of Service. Focusing on Video-Analytics (VA) applications as one of the most present HPC-original applications pushed to the Edge, we present a simple characterization of the impact of physical factors, along with the resource usage and QoS indicators in workload performance

for VA. Along with the reliability metric, we study the impact of the physical stress in constrained devices, to evaluate and discuss the relevance of such towards a the *RS* metric, providing a representation of the node under the different mentioned stress dimensions, to be used in digital-twin mechanisms to make decisions on placement and migration of workloads in Edge and IoT devices. Through the estimation of the application and device status, schedulers and load orchestrators will leverage critical information towards deciding whether to keep applications in the Edge devices until their QoS is degraded, place applications in selected devices promising better reliability and potential QoS, migrate applications towards same-level devices in less constrained situations or off-load to the Cloud when no available device is reliable enough.

The proposed metrics and data retrieval environment are evaluated against Quality of Service in Video-Analytics applications, involving telemetry retrieval towards obtaining the status and performance of devices, with load generated through experimental and real workload patterns, in physical stress scenarios through variable overheating, humidity and voltage upon the device. The generated data corpus obtained from telemetry and device sensors is modeled as a Reliability Score, indicating the capacity of the device to hold VA workloads given its status. The experiments have been performed using RaspberryPi v.4b as commodity Edge-IoT devices, exposed to temperatures up to +100 degree Celsius, humidity from 8% to 82%, and voltages limiting the specification for physical experimentation, and using TorchServe with ResNet-18 [1] application, as a reduced image classification neural network for small devices, recreating demand with experimental patterns to study the range of stress supported by the device and its impact on the QoS, also generating test load using user-demand patterns from the Alibaba Public Dataset as a realistic user behavior against on-line services. For this, the presented work introduces the following contributions:

- A Reliability Score metric estimates the physical stress in Edge-IoT devices, considering external conditions such as temperature, humidity and voltage variation. The proposed metric is designed to estimate the health of the device given its conditions towards potential scheduling and application placement.
- A technical methodology and testbed to stress Edge devices towards retrieving data and studying the effects of physical stress on Edge devices, allowing the reproduction and simulation of such devices, for orchestration algorithms on Edge Video Analytics.
- An open dataset reproducing a range of experiments, used to validate the proposed metric, displaying the quality of service in relation to external and system conditions, towards further design of digital twinning of Edge and IoT devices for VA or image-processing services<sup>1</sup>.

In this study we have observed the impact and magnitude of the physical stressors upon a proof-of-concept device and VA workloads, observing the relation between temperature and humidity, also the operation modes with respect voltage, against the device performance. The data obtained from sensors, telemetry and application, is to be published as OpenData for the community, first towards reproducibility, and second towards reusing for simulation and digital-twin building in further research in the field of Edge-IoT management.

This paper is organized as follows: Section II presents the state-of-art, followed by the modelling methodology in Section III, Experiment Design in Section IV, validation experiments in Section V, discussion on results in Section VI, and finally the conclusions and next steps in Section VII.

## II. RELATED WORK

Environmental stress factors affect hardware aging, moreover on devices exposed to the elements. *Extreme temperatures* affects IoT devices, from computing to SiC devices [2]–[4]. *Humidity and dust* are also important factors, as they cause corrosion and oxidation, increasing thermal resistance and leading to malfunction [5], [6]. Operational stress such as *voltage fluctuation* must be taken into account [7], also mechanical stress and material degradation, moreover on devices with continuous vibration or exposed to corrosive or oxidant materials [8], [9]. Such approaches attempt to predict in advance the remaining life of devices, towards active maintenance, providing indicators towards the current quality of each Edge/IoT device [10].

There are many works modelling and predicting resources and applications, to optimize application placement, resource provisioning, impact of application co-location, and quality of service forecasting, e.g. [11]–[13]. Estimating the impact of resource consolidation is already a classical problem on scheduling and autonomic computing, with lots of techniques using either hand-crafted models or machine learning towards fast and smart decision-making. Such techniques feed from system telemetry to predict the *fitness* of a placement or resources quota. Latest methods towards placement go from classic Machine Learning approaches [14] to modern Deep Reinforcement Learning [15]–[17], and

<sup>1</sup>Dataset publicly available at [http://recerca.ac.upc.edu/cromai/datasets/dataset\\_rs\\_v1.0\\_20250320.tar.xz](http://recerca.ac.upc.edu/cromai/datasets/dataset_rs_v1.0_20250320.tar.xz)

time-series modelling for modelling and pattern recognition for resources usage [18]. Generation of logical stress scenarios require either scaling up workloads towards reaching the resources limit, or generation of co-placed applications reducing the availability or generating resources competition in the device. Most often, existing traces are used to indicate load levels or simulate availability reduction, such as Alibaba, Azure or Google Cloud public traces [19], [20], while other approaches propose characterizing existing traces and generate synthetic ones [21].

#### A. Beyond State of the Art

Those methods are mostly prepared for Cloud and High-Performance Computing environments, where the resources availability is firstly large and wide, thus, the impact of over- and under-provisioning can be contained within affordable safety margins, and secondly capable of handling the overhead of complex orchestration mechanisms, allowing constant and accurate models from newly arrived or dynamic applications. In contrast, in the Edge or IoT deployments, resources are constrained and therefore over-provisioning scenarios have a higher impact. Moreover, unlike the Cloud, the Edge is traditionally decentralized, thus, creating and updating models to decide where an application can be provisioned can highly reduce the resources available for current applications to be deployed, meanwhile Cloud keeps a centralized orchestration node for such tasks. An Edge orchestration agent, taking care of an Edge node or a specific application deployed in the Edge, must be capable of generating a quick decision with data to be retrieved from its node and potential neighbor nodes. Therefore, here we present a simplistic *reliability score* independent of the application profile, relying only on the directly available sensing and telemetry, to make decisions of whether the observed device is suitable for an application or not.

### III. RELIABILITY SCORE

The Reliability Score (RS) serves as a critical metric for evaluating the ability of a device to perform computational tasks under varying conditions. For a physical RS, we consider at this time measurements affecting the reliability from device-related aspects, rather than workload-related aspects, to be commented later. The device-related RS, or *physical stress*, reflects the inherent reliability of a device, considering its current health and environmental challenges, without factoring in any specific workload. For proof-of-concept purposes, we set-up a Video-Analytics application as benchmark, comparing the QoS in different scenarios.

However, the scoring does not include any input variable from the application. In a more advanced reliability score definition, we could also consider a *logical stress* from information from the workload referring to its demands and device occupation.

Such physical stress reliability provides the device's resilience under its operating conditions. The RS can be considered as a direct relation between the *Health* of the device ( $h$ ), as its capacity of providing the contained resources in their best conditions, and the *Environment* adversarial to the device ( $e$ ), including low-high temperatures, humidity, unstable power supply, etc. The relation can be depicted as  $RS = function(h, e)$ , where a value  $< 1$  implies that the device is overwhelmed by the conditions, deteriorating any level of health towards performance.

The device health  $h$  represents the condition of the available resources. Such condition includes the temperature stability of the device extracted from the processors, considering the average temperature as the average in the device in measured standard conditions (i.e. 40°C), and maximum working temperature (i.e. 105°C) as threshold resetting the device due to protection mechanisms for overheating:

$$h_t = \max\left(0, 1 - \frac{|T_{device} - T_{std.core}|}{|T_{max.working.temp} - T_{std.core}|}\right)$$

Regarding environmental factors, we can initially consider external temperature as

$$e_t = \max\left(0, 1 - \frac{|T_{ambient} - T_{std.device}|}{|T_{max.working.temp} - T_{std.device}|}\right)$$

considering the current temperature detected by a thermal sensor immediately outside the device core (close but not inside) against the maximum working temperature on the device, and average temperature in the device in measured standard conditions (i.e. 25°C). Also, humidity as

$$e_h = \max\left(0, 1 - \frac{|H_{ambient} - H_{std.device}|}{|H_{max.humidity} - H_{std.device}|}\right)$$

considering the humidity detected by a sensor against the maximum supported by the device. For cases where not specified, we can consider the average and maximum detected in the environment (8% - 82%). And finally, we include the voltage stability, considering the average voltage as the nominal, and the safety limit allowed (i.e. 4.8V-5.2V) as boundaries for an anomalous scenario, as

$$e_v = \max\left(0, 1 - \frac{|V_{device} - V_{nominal}|}{|V_{max} - V_{min}|}\right)$$

The environmental scoring becomes the aggregation of the different factors, plus the potential  $e_d$  and other factors when included, where higher  $e$  indicates lower environmental impact (i.e.,  $e$  is

how much favorable is the environment). The final combination for the physical stress reliability score needs to include  $h$  and  $e$ , as:

$$\text{Reliability Score} : RS = \sqrt[4]{h_t \cdot e_t \cdot e_h \cdot e_v}$$

Note that additional factors can be introduced in the scoring, by defining their impact boundaries and definitions. E.g., as a potentially relevant environmental factor, we can consider the amount of peripherals connected to the device  $e_d$ , feeding from the device power supply in a variable pattern, reducing the device power capacity. The demonstration of the relation between the proposed metrics and the resulting quality of service, represented by the time per requested frame (i.e., the inverse of the *frames per second*), is shown in the Experiments section.

#### IV. TESTBED DESIGN

The selected experiments focus on Video Analytics on image/frame classification, towards the use case of urban mobility environments, detecting vehicles and pedestrians through roadside cameras attached to Edge nodes exposed to the elements. Part of the design of the testing experiments is provided by the collaboration between Universitat Politècnica de Catalunya, Barcelona Supercomputing Center and Cellnex S.A., with the objective of testing the Reliability Score on small Edge devices, to later scale towards hyper-distributed devices for Next-Gen 6G technologies.

1) *Physical Benchmark*: The study of the physical impact on the device is represented by the environmental external factors *Temperature*, *Humidity* and *Voltage*, also the internal factors such as *Core Temperature*. The test device (a RaspberryPi v.4B) is kept in a temperature chamber, to be heated and humidified from 15-20°C (room temperature) to ~65°C (added temperature) to ~100°C (greenhouse temperature), and relative humidity from 8% to 82%. Given that retained temperatures in urban artificial surfaces, such as pavement, can reach 65°C when ambient temperature reaching 45°C [22], [23], through experimentation we measured that CPU core temperatures reach the maximum in safety bounds ~110°C, considering that the device is enclosed in a containment box, adding greenhouse effect to the ambient temperature. Temperature and humidity are measured through a DH11 Sensor, controlled by a separate Arduino UNO v.4 monitoring the readings, with repeated experiments placing the sensor in different positions to avoid biased specific readings.

Aside, the power supply capacity provided by the grid can vary, altering the received voltage, from stable ranges where the device can retrieve

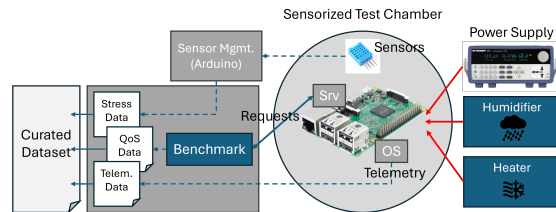


Fig. 1: Schema of the testbed, towards experiments and data retrieval. The stressors are applied into the device/test chamber, with data collected from the sensors, Operating System telemetry, and benchmarking application, composed as the dataset to perform the study.

power within safety ranges of current, to anomalous values for which the device is supposedly protected by design. Voltage is modified through a programmable Power Supply using safe voltage ranges from the RaspberryPi v.4B specifications, from 4.6V (low voltage protection triggered), to 4.8V (low voltage), to 5.0V (standard voltage) to 5.2V (high voltage). Such selection of values in every dimension have been checked and selected to avoid direct damage to the device that could alter the experiments results. Table I provides the summary of sensors and telemetry variables.

2) *Level-based Workload Benchmark*: In order to provide a benchmark, the experiments are performed using a Video Analytics application consisting of a classification neural network ResNet-18 deployed on a TorchServe server. ResNet-18 is a version prepared towards small devices, hence able to cope with the available resources on, i.e., deployed on RaspberryPi v.4B Edge nodes. The physical stress consists of testing different levels of concurrent load in a range level [1, 2, 4 ··· 64], on different temperature, humidity and voltage scenarios against the VA-server, to examine the effect of such stress factors in different load situation, as explained in the previous subsection.

The VA server is requested with a fixed size image, as an arbitrary image of size 1200x900 for classification as a standard frame, that could be obtained from a road-side camera or RSPT stream. The internal TorchServe inference process is independent of the used image, and it consists of resizing, center cropping, normalization, and conversion to a tensor. Note that the image size could be relevant on the resizing process; however, we focused on a realistic image size to be obtained from a standard road-side camera. The service (and device) return a JSON-formated response with the predicted classification, as response to the request.

3) *User-based Workload Benchmark*: Additionally, real-workload testing has been performed to

Variable	Units	Scale	Domain (Expected)	Register
Ambient Temperature	C	Continuous	25–105 °C, average device 35°C (ambient 20°C, +heat 55–65°C)	Sensor Adafruit DHT11
Ambient Humidity	%	Continuous	8.5–82% (humidifier)	Sensor Adafruit DHT11
Provided Voltage	V	Continuous	4.6–5.2 V	Power Source LABPS3060SM
CPU usage	%	Continuous	16–100% (4 core)	Library iostat
Memory usage	GB	Continuous	1.6–1.8 (server warmed up)	Library iostat
Core Temperature	C	Continuous	55–108 °C	Library iostat
Workload Completion Time	s	Continuous	272–83669 s	Library time

TABLE I: Variables measured from sensors and system telemetry

complete data retrieval, extracted from the Alibaba Public Datasets [19]. A variable range of petitions against the device is explored, covering everything from idle to overwhelmed resources usage, reaching saturation of the device and leading to a decrease in the Quality of Service of the application. Form an original dataset with 5300 machines over a period of 9 days, with Cloud services patterns, we define a distribution of clients requesting inference petitions, merged by weekday also adding random noise based on the variance at each hourly period, to obtain a *standard day-load*. Using a Mixture Gaussian process, amounts of requests are generated following the patterns, while also ensuring that experiments do not oversaturate the server.

4) *Retrieved Dataset*: The experimental datasets to be retrieved include the physical stressor values, telemetry and quality of service, composes a multi-dimensional time-series indicating the behavior of the device at different levels of stress. The generated datasets includes the following data:

- **The sensors data**: Temperature, Humidity, and provided Voltage.
- **The OS telemetry**: CPU usage (%), CPU core temperature, memory usage and total, disk usage (read, write), and network usage (received, transmitted).
- **The QoS metrics**: response time (min, max, average, st.dev, median), requests performed and failed, received client concurrency, and obtained reliability score.

Such datasets must provide fundamental data to 1) compute the Reliability Scoring from the sensor data, but also 2) model and reproduce the physical stress scenarios, towards future simulations and creation of digital twin, e.g., towards decision making in workload placement guided by the scoring.

## V. EXPERIMENTS

### A. Physical Stress Analysis

Results on physical stress application (temperature, humidity and voltage), as seen in Figure 2, shows us most of the expected effects along with

its results and magnitudes:

**Temperature**: high temperatures drive the device to CPU Throttling, increasing response time per request up to  $\times 4$  when reaching the limit of 105°C (ambient and core). Such effect was expected, and through the experiments we see that an increase of 65°C ambient in a closed chamber can lead to the limit of CPU safety levels.

**Humidity**: opposite to temperature, humidity tends to alleviate the effect of temperature increases by helping dissipation. Also, on high temperature experiments, performed within a closed container (usual on street-deployed devices), humidity is reduced also from the interior due to increased pressure inside the container.

**Voltage**: as observed, the design of the device brings different working modes depending on the provided voltage. For voltages around the safety range, current adapts to provide the required power, slightly affecting the device response time and use of resources, while affecting in higher order when the voltage is under limits. Demanded power from the device drops on under-voltage, to avoid increase current over safety limits, burning the device, and therefore affecting notably the usage of storage.

Table II shows the correlation between external stress factors and the response time for each scale experiment, as the amount of concurrent clients requesting queries. Such effects are also reflected in Figure 3, performing a Principal Components Analysis on the stress factors and sensor data, also on telemetry. As shown, the two principal eigenvectors focus on 1) Usage of resources, as indication of the device performance (as in work done), and 2) temperature and humidity stress factors. As shown in the figure, voltage has its own effect on the device operation, while examining the remaining principal components (mainly the third), it becomes relevant along with the use of I/O.

### B. Reliability Scoring Analysis

After retrieving the data from experimentation using different levels of physical stress and load, we compute the Reliability Score for each scenario,

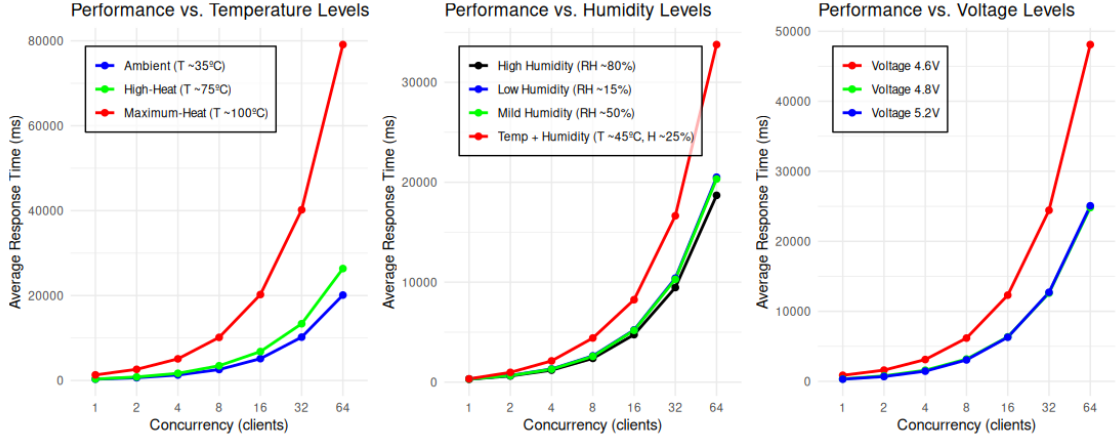


Fig. 2: Relation between Performance and levels of Ambient Temperature and Humidity, also received Voltage, with different levels of device functioning. (Note the different scales)

Scale	1	2	4	8	16	32	64	Average
Temperature	0.850790	0.832749	0.824534	0.820185	0.821068	0.818650	0.669957	0.805419
Humidity	-0.391175	-0.347681	-0.336425	-0.370744	-0.360343	-0.368527	-0.267801	-0.348957
Voltage	-0.130288	-0.111408	-0.106975	-0.101072	-0.102435	-0.100361	-0.146813	-0.114193

TABLE II: Correlation between Physical stressors versus degradation on Quality of Service (average response time), grouped by load level (scale).

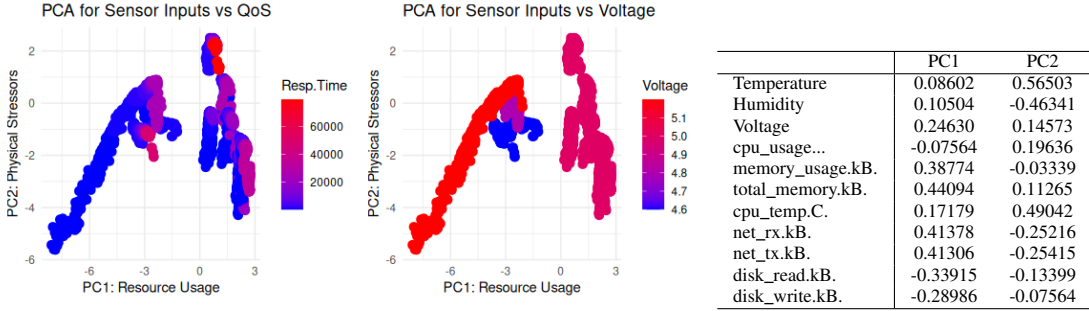


Fig. 3: PCA showing the principal components, mainly composed by the used resources vs the physical stressors. Comparison with Voltage operation mode, defining a third latent dimension (relevant in PC3).

testing if the obtained score corresponds to the obtained Quality of Service.

**Details:** As indicated previously, after experiments we detected that *humidity* is inversely correlated to the QoS, therefore, we adjusted properly the  $e_h$  factor as  $e_h(adjusted) = 1 - e_h(old)$ , inverting the direction of the specific scoring.

**Results:** Figure 4 shows the comparison between minimum Response Time (representing the baseline for the provided quality of service) and the Reliability Score, indicating how high reliability is centered on low minimal response times, while low scoring tends to indicate higher times. Remember that the scoring is computed from factors independent to the workload, in a diverse load scenario. There can be low scoring for low-load scenarios, as such load can still be completed in a stressed scenario;

however, we don't observe high reliability on high latency scenarios. This shows that the bias of the model is towards *false positives*, where the scoring is conservative indicating low reliability in still capable scenarios (depending on the load), but no *false negatives*, where we overtrust the device and end having low quality results. In order to quantify the results, the correlation between the minimum response time and the Reliability Score is of  $-0.6891$  (with respect the mean and maximum, it is  $-0.3604$ , considering that levels of concurrency affect the quality independently of the physical stress).

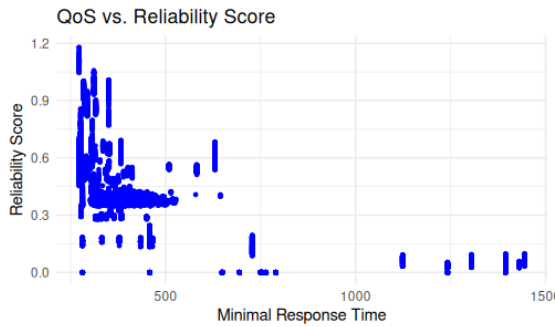


Fig. 4: Relation Reliability Score vs. Response Time, under physical stressors at all tested levels

## VI. DISCUSSION

### A. Decision Making, Weights and Thresholds

As indicated, the Reliability Score does not provide a decision but a metric to estimate how available is a node, device or resource, given a potential workload to be deployed. The resulting QoS of an application provisioned on a low-scoring device will depend on the application itself, and its characterization and forecasting will depend on the desired Service Level Agreement with users and clients. E.g., a specific application might require a minimum Reliability Score of  $M$  according to the maximum risk to be taken when placing the applications. Also, an application might prioritize the availability of certain resource, weighting it in the  $h_R$  Logical Reliability aggregation. One of the capabilities of the provided scoring method is that can be easily customized for tailored scenarios.

### B. Scheduling Algorithms

The Reliability Score is presented as an input for schedulers and decision-making algorithms, as one of the many inputs that can be used. An example of direct use for  $RS$  is as a quality metric in a *First-Fit* scheduling algorithm [24]. First-Fit is a near-optimal quadratic greedy method searching for the first node where an application is expected to fit following a quality metric, and provisioning applications decreasingly ordered by requirements. Other classic simplistic algorithms are *best-fit* as cubic greedy, or exhaustive planners. However, considering that  $RS$  might be performed in constrained nodes, it is recommended to avoid complex algorithms or untractable sets.

### C. Other Models and Loads

The current experiments have been performed using a selected model ResNet-18, as a proof-of-concept for Video Analytics performed on Edge-IoT devices. However, other models with different

requirements can be studied, adding a new dimension on the workload characteristics and demands to be supplied by the device. Changing the model will not affect the physical stress factors in the device, but will scale the QoS provided, accordingly to the properties of the new workload. The study of how the workload characteristics affects the device logical environment remains as future work.

### D. Adding Logical Stress

Aside of physical stressors, logical health can be computed as a ratio between the resources demanded and the resources available, generally considering the main computational resources as CPU, Memory, Network and Disk I/O. Such stress will depend on the target application to be deployed, as a low-resource demanding application can easily fit into an almost fully-loaded node, also the aggregation of co-located applications can saturate resources and degrade applications on the device. An approximation could be derived from the resource  $Res$  availability and demand, as

$$h_{Res} \sim \frac{Res_{demand} - Res_{available}}{Res_{demand}}$$

where the total health becomes the geometric average of resources as an approximation to the resource criticality. Nevertheless, advancing resources demand requires a prior profiling of the application, to indicate how much it will demand, considering a request-driven application. At this time, the stress impact of co-location is considered secondary, as the current use of Edge-IoT resources focuses on controlled uses of resources, opposed to broad-user versatility-oriented Cloud resources where co-location policies aim towards consolidation of workloads in a single node. However, future uses of urban-deployed Edge devices might include multi-tenancy and multi-application use cases [25]. Remains as future work to study how to properly incorporate inference and forecasting of demanded resources from requests, towards generating a *health* factor for resources competition stress. Such studies will need consider that co-location depends on how the applications are bounded (CPU-bound, memory-bound, etc.).

## VII. CONCLUSIONS

Performing Video Analytics and AI in the Edge-IoT is impacted by the environment of the device. Moreover when the device is placed in an urban environment with physical stress factors aside of logical ones provided by the users' demand. In this work, we first study the impact of the most basic factors (temperature, humidity, voltage supply), then benchmark the device against the provided environment using AI classification applications

with different levels of load. From the study, we observed the impact of 1) temperature, as the by-design thermal protection reduces the frequency, and therefore quality of service, as ambient temperature rises; 2) humidity, as it smoothes the effect of the temperature of the device in cabinets; 3) voltage, as the different voltage supply sets the device in different operational modes. The proposed Reliability Score for Edge-IoT devices shows a correlation and indication on the expected quality of service depending on the physical stress factors detected upon the device. Towards the next study, we will focus on the impact of the workload and resources capability as logical stress factors, including different characteristics for VA models and services, also on the use of the Reliability Score into scheduling algorithms towards the placement and distribution of load in Edge-IoT device networks. Finally, we highlight that the current studies and proposed methods have potential towards other devices and environmental variables, also can be applied to high-performance systems.

## REFERENCES

- [1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: <https://arxiv.org/abs/1512.03385>
- [2] P. Dini, G. Basso, S. Saponara, and C. Romano, "Real-time monitoring and ageing detection algorithm design with application on sic-based automotive power drive system," *IET Power Electronics*, vol. 17, pp. n/a–n/a, 03 2024.
- [3] Z. Ni, X. Lyu, O. P. Yadav, B. N. Singh, S. Zheng, and D. Cao, "Overview of real-time lifetime prediction and extension for sic power converters," *IEEE Transactions on Power Electronics*, vol. 35, no. 8, pp. 7765–7794, 2020.
- [4] F. Niknia, J.-L. Danger, S. Guilley, and N. Karimi, "Aging effects on template attacks launched on dual-rail protected chips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 5, pp. 1276–1289, 2022.
- [5] S. Elkateb, A. Métwalli, A. Shendy, and A. E. Abu-Elanien, "Machine learning and iot – based predictive maintenance approach for industrial applications," *Alexandria Engineering Journal*, vol. 88, pp. 298–309, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1110016823011572>
- [6] X. Guo, V. Verma, P. Gonzalez-Guerrero, and M. R. Stan, "When "things" get older: Exploring circuit aging in iot applications," in *2018 19th International Symposium on Quality Electronic Design (ISQED)*, 2018, pp. 296–301.
- [7] S. Yousuf, S. A. Khan, and S. Khursheed, "Remaining useful life (rul) regression using long–short term memory (lstm) networks," *Microelectronics Reliability*, vol. 139, p. 114772, 2022.
- [8] R. Muñoz, F. Nuno, J. Diaz, M. González, M. Prieto, and Menéndez, "Real-time monitoring solution with vibration analysis for industry 4.0 ventilation systems," *The Journal of Supercomputing*, vol. 79, pp. 1–25, 11 2022.
- [9] B. Steenwinkel, D. De Paepe, S. Vanden Haute, P. Heyvaert, M. Bentefrit, P. Moens, A. Dimou, B. Van Den Bossche, F. De Turck, S. Van Hoecke, and F. Ongenaes, "Flags: A methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing expert knowledge with machine learning," *Future Generation Computer Systems*, vol. 116, pp. 30–48, 2021.
- [10] A. Zimpeck, C. Meinhardt, L. Artola, and R. Reis, *Reliability Challenges in FinFETs*. Cham: Springer International Publishing, 2021, pp. 29–63. [Online]. Available: [https://doi.org/10.1007/978-3-030-68368-9\\_3](https://doi.org/10.1007/978-3-030-68368-9_3)
- [11] S. Sotiriadis, N. Bessis, C. Amza, and R. Buyya, "Elastic load balancing for dynamic virtual machine reconfiguration based on vertical and horizontal scaling," *IEEE Transactions on Services Computing*, vol. 12, no. 2, pp. 319–334, 2019.
- [12] K. Rządca, P. Findeisen, J. Świdorski, P. Zych, P. Broniek, J. Kusmirek, P. K. Nowak, B. Strack, P. Witusowski, S. Hand, and J. Wilkes, "Autopilot: Workload autoscaling at google scale," in *Proceedings of the Fifteenth European Conference on Computer Systems*, 2020.
- [13] V. Hayyolalam and A. A. Pourhaji Kazem, "A systematic literature review on qos-aware service composition and selection in cloud environment," *Journal of Network and Computer Applications*, vol. 110, pp. 52–74, 2018.
- [14] J. L. Berral, I. n. Goiri, R. Nou, F. Julià, J. Guitart, R. Gavaldà, and J. Torres, "Towards energy-aware scheduling in data centers using machine learning," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, ser. e-Energy '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 215–224.
- [15] T. Danino, Y. Ben-Shimol, and S. Greenberg, "Container allocation in cloud environment using multi-agent deep reinforcement learning," *Electronics*, vol. 12, no. 12, 2023.
- [16] Y. Ju, Y. Chen, Z. Cao, L. Liu, Q. Pei, M. Xiao, K. Ota, M. Dong, and V. C. M. Leung, "Joint secure offloading and resource allocation for vehicular edge computing network: A multi-agent deep reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 5, pp. 5555–5569, 2023.
- [17] Z. Jian, X. Xie, Y. Fang, Y. Jiang, Y. Lu, A. Dash, T. Li, and G. Wang, "Drs: A deep reinforcement learning enhanced kubernetes scheduler for microservice-based system," *Software: Practice and Experience*, 2023.
- [18] J. L. Berral, C. Wang, and A. Youssef, "Ai4dl: Mining behaviors of deep learning workloads for resource management," in *Proceedings of the 12th USENIX Conference on Hot Topics in Cloud Computing*, ser. HotCloud'20. USA: USENIX Association, 2020.
- [19] Alibaba, "Alibaba cluster data 2017-2018," <https://github.com/alibaba/clusterdata/blob/master/cluster-trace-v2017/README2017.md> [https://github.com/alibaba/clusterdata/blob/master/cluster-trace-v2018/trace\\_2018.md](https://github.com/alibaba/clusterdata/blob/master/cluster-trace-v2018/trace_2018.md) Retrieved December 24.
- [20] M. Azure, "Azure public datasets," 2024. [Online]. Available: <https://github.com/Azure/AzurePublicDataset>
- [21] S. Bergsma, T. Zeyl, A. Senderovich, and J. C. Beck, "Generating complex, realistic cloud workloads using recurrent neural networks," in *Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles*, ser. SOSP '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 376–391.
- [22] K. Guan, "Surface and ambient air temperatures associated with different ground material: a case study at the university of california, berkeley," 2011. [Online]. Available: <https://api.semanticscholar.org/CorpusID:130416359>
- [23] P. Knox, "How hot does pavement get in summer?" <https://site.extension.uga.edu/climate/2022/05/how-hot-does-pavement-get/>, 2022.
- [24] D. Johnson, *Near-optimal Bin Packing Algorithms*, ser. Massachusetts Institute of Technology, project MAC, 1973. [Online]. Available: <https://books.google.es/books?id=8pGNGAAACAAJ>
- [25] M. Yannuzzi, F. van Lingem, A. Jain, O. L. Parellada, M. M. Flores, D. Carrera, J. L. Perez, D. Montero, P. Chacin, A. Corsaro, and A. Olive, "A new era for cities with fog computing," vol. 21, no. 2, p. 54–67, Mar. 2017. [Online]. Available: <https://doi.org/10.1109/MIC.2017.25>