

PAPER • OPEN ACCESS

Including AI experiments onboard the UPMSat-3 satellite mission

To cite this article: Ángel-Grover Pérez-Muñoz *et al* 2024 *J. Phys.: Conf. Ser.* **2716** 012101

View the [article online](#) for updates and enhancements.

You may also like

- [Artificial intelligence, machine learning, deep learning, and big data techniques for the advancements of superconducting technology: a road to smarter and intelligent superconductivity](#)
Mohammad Yazdani-Asrami
- [Suprathermal Electron Transport and Electron Beam Formation in the Solar Corona](#)
Bofeng Tang, Haihong Che, Gary P. Zank et al.
- [Recent Progress in Quantum Key Distribution Network Deployments and Standards](#)
M Stanley, Y Gui, D Unnikrishnan et al.



UNITED THROUGH SCIENCE & TECHNOLOGY

 The Electrochemical Society
Advancing solid state & electrochemical science & technology

**248th
ECS Meeting**
Chicago, IL
October 12-16, 2025
Hilton Chicago

**Science +
Technology +
YOU!**

Abstract submission
deadline extended:
April 11, 2025

SUBMIT NOW

The banner features a woman in a brown blazer smiling and gesturing, set against a blue background with a network of nodes and lines. The top and bottom of the banner are decorated with a repeating pattern of circular icons containing stylized circuit traces.

Including AI experiments onboard the UPMSat-3 satellite mission

Ángel-Grover Pérez-Muñoz^{1,2}, Alejandro Alonso^{1,2}, María S. Pérez³, Juan Zamorano^{1,2}, Hugo Valente^{1,2}, Juan A. de la Puente^{1,2}, Angel Porras-Hermoso⁴ and Montserrat Bayón-Laguna⁴

¹ Universidad Politécnica de Madrid. Sistemas de Tiempo Real y Arquitectura de Servicios Telemáticos, ETSI Ingenieros de Telecomunicación, 28040, Madrid, Spain

² Universidad Politécnica de Madrid. Information Processing and Telecommunications Center, ETSI Ingenieros de Telecomunicación, 28040, Madrid, Spain

³ Universidad Politécnica de Madrid. Ontology Engineering Group, ETSI Ingenieros Informáticos, 28660, Madrid, Spain

⁴ Universidad Politécnica de Madrid. Instituto Universitario de Microgravedad “Ignacio Da Riva”, ETSI Aeronáutica y del Espacio, 28040, Madrid, Spain

E-mail: {angel.perez.munoz, alejandro.alonso, maria.s.perez, juanrafael.zamorano, hugo.svalente, juan.de.la.puente, angel.porras.hermoso, m.bayon}@upm.es

Abstract.

Artificial Intelligence (AI) techniques are being used in general-purpose industrial computing systems. There is a great interest in expanding its use across other types of systems. However, they are not immediately applicable to embedded safety-critical systems. In particular, in spacecrafts, there are subsystems with high integrity requirements, which means that their failure could affect the overall behavior of the vehicle or even the loss of the complete mission. This paper deals with the use of some relevant AI techniques onboard space systems. Machine Learning and Neural Networks are potential techniques for these systems. The objective of this paper is to evaluate its applicability, select the most appropriate tools, and determine its feasibility to place onboard the satellite. Through the analysis of standards proposals, and a thermal estimation use case, we identify the issues, challenges, and guidelines to be considered for the use of AI, specifically machine learning, in UPMSat-3.

1. Introduction

Artificial Intelligence (AI) is applied in many space applications from the National Aeronautics and Space Administration (NASA) and European Space Agency (ESA). Initially, AI was restricted to the ground segment to assist the missions autonomously, and it was isolated from the space segment due to hardware constraints and the scarce resources available on their onboard computers. Dedicated AI hardware platforms enabled the usage of AI in the space segment. For instance, the ϕ -Sat-2 CubeSat from ESA and Open Cosmos [1] included specialized hardware (Intel’s Myriad 2 Vision Processing Unit) to run AI experiments for Earth imagery such as cloud detection, image compression, and classification of maritime vessels, among others. More advanced systems like planetary rovers, such as the NASA’s Mars Pathfinder, Mars Exploration Rovers, or Perseverance Mars rover, have included AI software (Autonomous Exploration for



Gathering Increased Science, AEGIS) for autonomous planning and tactical activities without the direction of the mission operators [2].

The qualification standards for software systems, such as the DO-178C/ED-12C used in avionics, or the ECSS-Q-ST-80C [3] used in ESA projects, are not appropriate for AI or Machine Learning (ML) systems as the available techniques do not satisfy some of their safety requirements. ESA has recently published the first draft of a handbook to qualify ML systems in space (ECSS-E-HB-40-02A DIR1 [4]). Although it is still in progress and has not been integrated into the official software qualification standard, it is a first step towards AI certification [5]. Besides, researchers are actively working on new areas to develop tools and methodologies that ensure safety properties for AI, known as “AI Safety”. Some research works propose testing techniques for AI, as the ExtendAIST framework which includes methods and metrics to evaluate the robustness, stiffness, and behavior consistency of ML and Deep Learning (DL) models like adversarial attacks or neuron level coverage [6]. Other works proposed reference architectures to improve the reliability of ML-based systems based on N-Version programming [7], or temporal and space partitioning [8].

This article presents the study and analysis of the viable and safe usage of AI/ML experiments in the UPMSat-3 mission. The remainder of this paper is structured as follows. Section 2 describes the challenges and properties of ML for safety-critical systems. Section 3 discusses the candidate AI-based applications for the UPMSat-3 satellite. Section 4 introduces a use case on temperature estimation based on thermal data, which would serve as the basis for AI experiments in UPMSat-3. Finally, section 5 draws the conclusions and future lines of research.

2. ML techniques in space systems

AI is an umbrella term that includes many sub-fields and techniques. Throughout this paper, AI is classified based on the ECSS-E-HB-40-2A handbook [4]. This taxonomy divides AI in two groups, data-driven and knowledge-based AI. Firstly, **knowledge-based AI**, also known as Symbolic AI, involves systems whose functionalities are based on human-written *deterministic code*. That is key because deterministic code allows developers to follow the traditional practices to test and validate these systems. On the other hand, **data-driven AI**, also known as ML, acquires its knowledge by extracting patterns from experience, that is data. ML includes its sub-field DL and involves a stochastic process as it involves randomness in its training process. Therefore, traditional certification methods for ML models are not applicable, and it is the main concern in this paper. For the same reason, data-driven AI is the main focus of the EASA AI roadmap and the ECSS handbook for ML.

According to space system standards, current ML-based systems are not adequate to ensure safety requirements. As a result, it is not possible to deploy these systems in safety critical systems due to their stochastic behavior, and nondeterminism against scenarios unused during the training phase. In the following subsections, we present our findings on AI safety to address these problems and also outline the challenges encountered in doing so.

2.1. Safety properties for ML-based systems

Traditional certification standards, like the ECSS-Q-ST-80C [3], include diverse characteristics to qualify the software system such as reliability, maintainability, security, safety, and effectiveness, among others. These characteristics (widely known as non-functional requirements) are problematic to be reached due to the stochastic nature of ML. Therefore, there are different aspects that must be considered to achieve safety in ML systems.

The first property is Robustness which guarantees the continuous operation of the system providing similar responses for similar inputs. This may require the inclusion of functionality to detect hazardous situations so that it can be reverted, or notified to a human operator. The next one is Explainability [9] which aims to convert ML models from black to white boxes. This

is of special interest for safety critical systems as the model's decisions can be explained for its certification. Finally, Quality is required in source code, on the ML model and on the datasets used for testing, training, and validating the data. It should be noted that poor quality in data used for training is likely to make convergence to a poor model, as well. The ECSS-E-HB-40-02A handbook draft delves into these quality attributes.

2.2. Safety challenges for ML-based systems

As previously discussed, there are different properties that ML applications must obey to be included in safety-critical systems. Different challenges arise in doing so, for instance, the verification and validation (V&V) techniques recommended in safety-critical systems, like data and code analysis through code coverage, are not applicable as they are black boxes and, the obtained source code usually lacks complexity. This subsection summarizes the safety challenges and issues that must be addressed in ML-based systems.

The first one is related to the development methodologies since the behavior of an ML model depends on its parameter values which are not traceable directly to requirements. Hence, requirements-based testing traditionally adopted in safety-critical systems are not applicable. To tackle the testing challenge in ML, alternative approaches can be used like metamorphic testing [10], where the ML model is further tested with transformations of the original testing datasets. Finally, as identified in reference [11], the deployment of artificial intelligence in embedded devices raises different concerns like the selection of a platform with hardware accelerators such as field programmable gate arrays (FPGAs) or general purpose graphics processing units (GPGPUs). Such complex architecture raises issues related to real-time constraints like temporal behavior determinism.

3. AI experiments and considerations for the UPMSat-3

The UPMSat-3 is a university satellite project led by the "Instituto Universitario de Microgravedad Ignacio Da Riva" (IDR/UPM), a research institute from the Universidad Politécnica de Madrid (UPM). It is being developed in collaboration with the Real-Time Systems research group from UPM (STRAST, Sistemas de Tiempo Real y Arquitectura de Servicios Telemáticos). The UPMSat-3 is characterized as a 12U CubeSat with $0.2 \times 0.2 \times 0.34$ cm dimensions. The satellite is scheduled for launch in mid-2024 aboard the Spectrum launcher from ISAR Aerospace.

The primary mission objective is to serve as an in-orbit demonstrator and qualification of different payloads and technologies in space. Besides, as a university satellite, it will serve as an experimental platform for UPM students and researchers. Work is underway to include experiments using AI techniques in the space and ground segments. AI in the space segment is being evaluated to be used in subsystems with physical control functionalities like the Attitude Determination and Control Subsystem (ADCS), Thermal Control Subsystem (TCS), and for the Failure, Detection, Isolation and Recovery (FDIR) functionalities.

Data is the main driver to design ML models. That is why we are considering (i) synthetic data automatically generated with models of the environment and (ii) real data obtained from similar missions. The use case presented in this paper focuses on the application of ML in the sensing activities of the thermal control system and it is discussed in the next section.

4. Use Case: Temperature estimation of PT1000 thermistors

The HERCULES mission was successfully launched in the north of Sweden, Kiruna and operated for about three hours in the stratosphere, at 30 Km above the sea level [12]. This experiment included 28 PT1000 thermistors to read temperatures, four silicon heaters to actively control the temperature, and several sensors for environmental measurement as radiometers and barometers. The objective of the mission was to characterize the thermal environment at

such altitudes where, as in space, heat is transferred mainly due to radiation and it is under near-vacuum conditions. This way, future space missions (like UPMSat-3) would leverage the mission's results for its thermal design.

Onboard the satellite, the TCS is mainly in charge of assuring its correct temperature by means of active and passive control. Thus, from the software perspective, the thermal system can be categorized as a large control system. Typically, these systems are hard to model mathematically due to their non-linearity and complex dynamics. The availability of data allows us to characterize the system of interest and to design its controller. Therefore, the use case presented in this paper focuses on the application of ML on the TCS.

The objective of this use case is to analyze the capabilities and weaknesses of using an ML model as a temperature estimator of the 28 PT1000 thermistors located across the HERCCULES gondola. A temperature prediction system would be useful in the UPMSat-3 satellite to perform model predictive control (MPC). MPC requires the system model (temperature estimator) to be fast so that different conditions can be simulated before performing the actual control.

4.1. Datasets

The datasets used to train and test the model were obtained from real measurements gathered at 30 Km of altitude in the HERCCULES stratospheric balloon mission. The complete mission lasted about 3.5 hours (12600 s), from which 3 hours (10800 s) comprised the ascent and floating phases. Thermal data was acquired cyclically in those phases with a period of 10 seconds, thereby giving us approximately a total of 1080 samples for both training and testing. This amount of data may seem to be scarce. However, previous studies like [13] achieved satisfactory results in ML thermal modeling with 707 samples and a longer sampling period of 60 seconds.

The prediction of temperature mainly depends on their evolution through time and on several environmental characteristics such as the dissipated heat through active control, or the emitted and received radiation. Therefore, the selected parameters are as follows:

- Temperatures in $^{\circ}C$ captured from 28 PT1000 thermistors distributed in different locations inside of the gondola.
- Power dissipated in *Watts* (W) by four silicon heaters. Four of the thermistors (THERM-0 up to THERM-3) are embodied in these heaters.
- Direct solar radiation in W/m^2 measured by one pyranometer and one pyrgeometer placed facing towards the sky.
- Outgoing longwave radiation in W/m^2 measured by one pyranometer and one pyrgeometer placed facing towards the Earth's surface.

As this ML model follows a supervised learning strategy, the selected parameters were used to define the samples consisting of inputs (also known as features) and their corresponding expected outputs. Regarding inputs, the i -th sample is denoted as $x^{(i)}$ and was composed of 207 elements. The first 28 elements (from $x_1^{(i)}$ to $x_{28}^{(i)}$) contained the PT1000 measurements, the next 4 elements (from $x_{29}^{(i)}$ to $x_{32}^{(i)}$) included the power dissipated by the heaters, the next 4 elements (from $x_{33}^{(i)}$ to $x_{36}^{(i)}$) contained the radiometers measurements, and the last elements (from $x_{37}^{(i)}$ to $x_{207}^{(i)}$) included the past six measurements from each of the 28 PT1000 sensors. Each sample was assigned a set of expected outputs, which corresponds to the future temperatures captured from the 28 thermistors (from $y_1^{(i)}$ to $y_{28}^{(i)}$) in the next sampling cycle.

Therefore, the dataset was composed of 707 pairs of $\mathbf{x} \in \mathbb{R}^{207}$ and $\mathbf{y} \in \mathbb{R}^{28}$. These were randomly shuffled and partitioned into 70% instances for training and 30% instances for testing the model. After that and before the training step, the data was normalized and outliers were filtered to favor the robust design of the model.

4.2. MLP-ANN model design

This use case addresses a regression problem as it needs to predict 28 continuous values (temperatures) based on a set of inputs (historic temperature, radiation, and dissipated heat). The technique applied in this use case conforms to a Fully Connected Artificial Neural Network (ANN) with a Multi-Layer Perceptron (MLP) structure. The general objective of the MLP-ANN model is to estimate a set of θ parameters that best fit the training dataset. Useful parameters are those that minimize a cost function $J(\theta)$ which is defined as follows for each i -th sample:

$$J^{(i)}(\theta) = (y^{(i)} - \hat{y}^{(i)})^2 \quad (1)$$

where $y^{(i)}$ refers to the expected output and $\hat{y}^{(i)}$ to the prediction returned by the MLP-ANN model given the vector of features $x^{(i)}$. The resulting cost given the θ parameters is the squared error from the prediction and the expected output.

In this use case, the minimization was obtained using the mini-batch gradient descent optimization technique which is defined as expressed below:

$$\theta^{t+1} = \theta^t - \frac{\eta}{|B|} \nabla_{\theta^t} \sum_{i \in B} J^{(i)}(\theta^t) \quad (2)$$

where B is defined as a random subset of the complete training dataset, η is the learning rate, and the last part involves the gradient vector for all the input features. The overall architecture of the model is depicted in figure 1 and is composed of 207 neurons in the input layer; three hidden layers with 100, 100, and 50 neurons; and the output layer with 28 neurons.

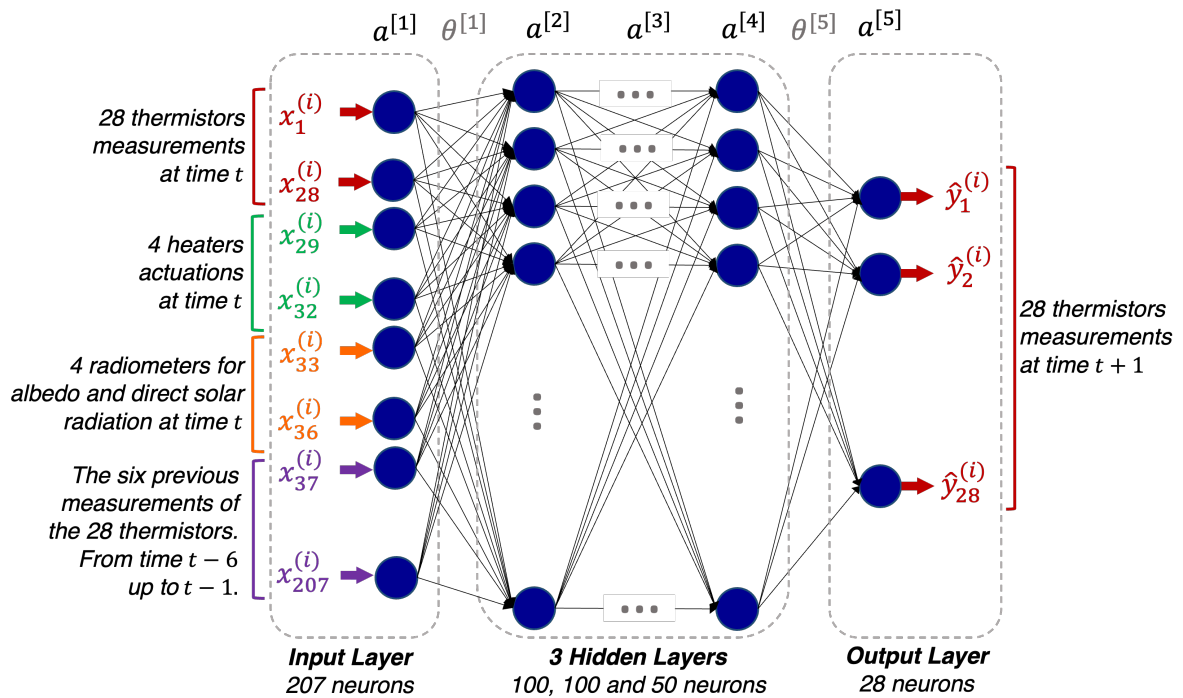


Figure 1: MLP-ANN model for temperature prediction.

The parameters were randomly initialized before the training phase. At each training iteration or epoch i , where $i = 1 \rightarrow m$, the sensors information x^i were used as the data for the input layer, hence $a^{[1]} = x^i$. The vectors of the remaining neurons $a^{[j]}$, where $j = 2 \rightarrow 4$ for hidden layers and $j = 5$ for the output layer were calculated as follows:

$$\mathbf{a}^{[j]} = \text{ReLU}(\boldsymbol{\theta}^{[j]} \cdot \mathbf{a}^{[j-1]}), \quad (3a)$$

$$\mathbf{a}^{[5]} = \hat{\mathbf{y}} = \boldsymbol{\theta}^{[5]} \cdot \mathbf{a}^{[4]}. \quad (3b)$$

where, as expressed in (3a), the neurons from the hidden layers used the rectified linear unit activation function $\text{ReLU}(x) = \max(0, x)$, providing a non-linearity relationship between inputs and outputs of the model.

The learning rate η for this use case was set to 0.001 to ensure a smoother update of the network parameters. The downside of such a small value is the slower rate of learning. That is why the training was conducted with a total of 2500 epochs (training iterations). Finally, the batch size for each training was set to 600 samples, about 80 % of the entire training dataset.

4.3. Testing and results

The validation of this temperature estimator was performed adopting a one-step-ahead forecast validation process [14] where the predictions (or forecasts) are not performed based on previous predictions, but on real and unprocessed historical measurements. It is important to note that as in previous studies [9], the testing partition could not be used because it was shuffled and did not include sequential and equally-spaced samples. Besides, we decided to use the unprocessed (not normalized and unfiltered) dataset to obtain the results in a more realistic environment. Therefore, the whole unprocessed dataset was used for this process. The results are presented in Figure 2 for thermistors 0, 1, 2, 3, 19 and 20, which were placed in different compartments.

The thermistors 0 up to 4 were located near the silicon heaters and their evolution mainly depended on their dissipated power. The predictions, blue plot, followed the same tendency as the expected values throughout the evaluation. Thermistors 19 and 27 were of interest for these studies as the spikes found near 0.5 h and 1.2 h corresponded to erroneous values which were successfully handled by the model. The results of the latter also demonstrate the absence of over-fitting, otherwise, the model would have tried to follow the erroneous trends.

The validation process of this model is also supported by two quantitative metrics. For this use case, we have applied the Root Mean-Squared-Error (RMSE) and Mean-Absolute-Error (MAE) that quantify the difference between the model's output and the expected value used as ground truth over a set of estimations given by the model. These metrics are defined as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}, \quad (4a)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|. \quad (4b)$$

where N refers to the size of the dataset. The RMSE obtained with the training dataset had a value of 0.0387. The RMSE and MAE obtained from the thermistors 0, 1, 2, 3, 19 and 27 after the one-step ahead forecast validation are presented in table 1. As presented, the higher RMSE values were obtained for THERM-19 and THERM-27. These sensors showed spurious behaviors during the mission, especially in the spikes of $7^\circ C$. It is therefore not surprising that the values are higher compared to other sensors.

5. Conclusions and future work

AI will be a major driver for space applications on both the ground and space segment. Onboard the satellite, there are different types of experiments that can leverage ML models like

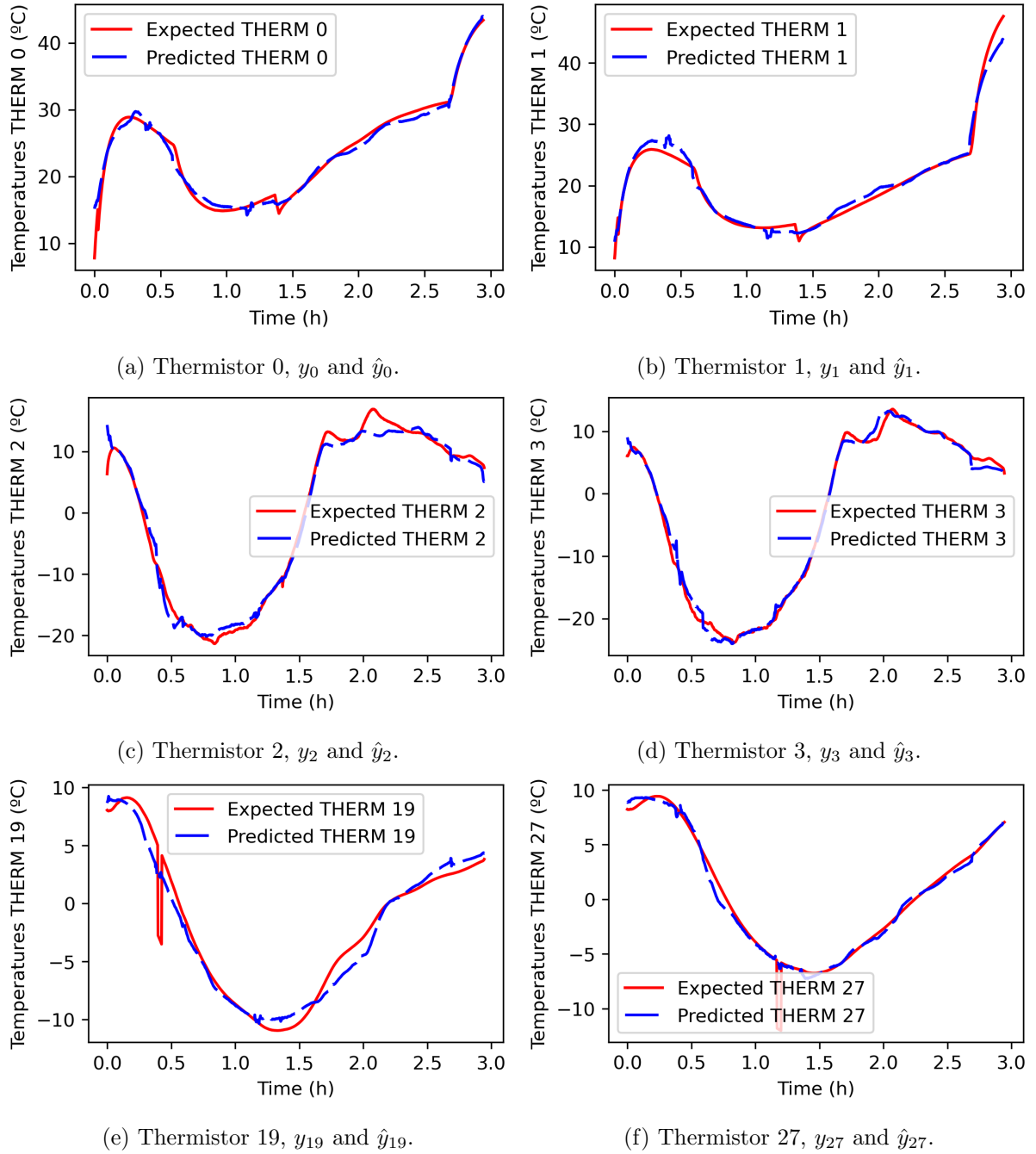


Figure 2: Comparison of predictions and estimations for thermistors 0, 1, 2, 3, 19 and 27.

Table 1: RMSE and MAE values for the thermistors 0, 1, 2, 3, 19 and 27.

Metric	THERM-0	THERM-1	THERM-2	THERM-3	THERM-19	THERM-27
RMSE	0.3705	0.4427	0.4659	0.4604	0.6972	0.5681
MAE	0.2679	0.2961	0.3405	0.3388	0.3327	0.2245

autonomous feedback control, and other decision-making functionality such as FDIR. The usage of AI in these platforms requires not only to be efficient, but also to fulfill safety requirements such as robustness, availability, reliability, among others. Efforts in the industries emerged and recognized these requirements, as a result, they are working on standards proposals. This paper presented some of the challenges, issues, and guidelines for the inclusion of AI/ML experiments in UPMSat-3, based on the handbook and standards from ESA and EASA.

In general, some issues and challenges that prevent AI usage in safety-critical systems include the low predictability of its functional and behavior due to the stochastic nature of ML models, and the difficulty in ensuring its real-time determinism due to complex heterogeneous architectures. The main conclusion is that there is a need to pave the way towards the certification/qualification of AI-based safety systems.

As for future work, we are still working and studying satellite systems that have used AI in their missions, so that similar systems could be applied in UPMSat-3. Although, ML models are deterministic from a mathematical perspective as they return the same outputs given the same inputs, the output from the model to a new input depends on the correlation between that input and the inputs from the training dataset. Further work needs to be developed to check their correlation in runtime. In any case, it is important to note that the usage of AI should not be justified just because it is the trend, but because of the value or benefits it offers.

Acknowledgments

This work was mainly supported by the STRAST research group and IDR/UPM institute under the projects OAPES-CM “Operación Avanzada de Pequeños Satélites” (Ref.: Y2020/NMT-6427) and PRESECREL (PID2021-124502OB-C43). The authors also acknowledge the financial support of the Ministerio de Ciencia e Innovación, and Comunidad de Madrid Proyectos Sinérgicos from the I+D plan (Spain).

References

- [1] Furano G, Tavoularis A and Rovatti M 2020 Ai in space: applications examples and challenges *2020 IEEE Int. Symp. on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)* pp 1–6
- [2] Estlin T A, Bornstein B J, Gaines D M, Anderson R C, Thompson D R, Burl M, Castaño R and Judd M 2012 *ACM Trans. Intell. Syst. Technol.* **3** 19
- [3] ECSS 2017 *ECSS-Q-ST-80C Rev.1. Software product assurance* (Noordwijk: ESA Requirements and Standards Section)
- [4] ECSS 2023 *ECSS-E-HB-40-02A DIR1. Space Engineering: Machine Learning Qualification Handbook* (Noordwijk: ESA Requirements and Standards Section)
- [5] EASA 2023 *EASA Concept Paper: guidance for level 1 and 2 machine learning applications* (Cologne: EASA)
- [6] Wu T, Dong Y, Zhang Y and Singa A 2020 *Appl. Sci.* **10** 518
- [7] Machida F 2019 N-version machine learning models for safety critical systems *2019 49th Annual IEEE/IFIP Int. Conf. on Dependable Systems and Networks Workshops (DSN-W)* pp 48–51
- [8] Biondi A, Nesti F, Cicero G, Casini D and Buttazzo G 2020 *IEEE Embedded Systems Letters* **12** 78–82
- [9] Nauta M, Trienes J, Pathak S, Nguyen E, Peters M, Schmitt Y, Schlötterer J, van Keulen M and Seifert C 2023 *ACM Comput. Surv.* **55** 42
- [10] Xie X, Ho J W, Murphy C, Kaiser G, Xu B and Chen T Y 2011 *J. Systems and Software* **84** 544–558
- [11] Zhang Z and Li J 2023 *Micromachines (Basel)* **14** 897
- [12] Ángel Grover Pérez-Muñoz, Gamazo-Real J C, González-Bárcena D and Zamorano J 2023 *Computers and Electrical Engineering* **111** 108953
- [13] Reis Junior J D, Ambrosio A M, de Sousa F L and Silva D F 2021 *Journal of the Brazilian Society of Mechanical Sciences and Engineering* **43** 198
- [14] Suradhaniwar S, Kar S, Durbha S S and Jagarlapudi A 2021 *Sensors* **21**