Flexible Payload Configuration for Satellites using Machine Learning

Marcele O. K. Mendonça, Flor G. Ortiz-Gomez, Jorge Querol, Eva Lagunas,
Juan A. Vásquez Peralvo, Victor Monzon Baeza, Symeon Chatzinotas and Bjorn Ottersten
Interdisciplinary Centre for Security Reliability and Trust (SnT) - University of Luxembourg, Luxembourg.
Corresponding Author: marcele.kuhfuss@uni.lu

Abstract—Satellite communications, essential for modern connectivity, extend access to maritime, aeronautical, and remote areas where terrestrial networks are unfeasible. Current GEO systems distribute power and bandwidth uniformly across beams using multi-beam footprints with fractional frequency reuse. However, recent research reveals the limitations of this approach in heterogeneous traffic scenarios, leading to inefficiencies. To address this, this paper presents a machine learning (ML)-based approach to Radio Resource Management (RRM).

We treat the RRM task as a regression ML problem, integrating RRM objectives and constraints into the loss function that the ML algorithm aims at minimizing. Moreover, we introduce a context-aware ML metric that evaluates the ML model's performance but also considers the impact of its resource allocation decisions on the overall performance of the communication system.

Index Terms—Radio Resource Management, Satellite Communications, Machine Learning.

I. INTRODUCTION

Satellite networks offer an appealing solution for delivering ubiquitous connectivity across diverse domains such as the maritime and aeronautical markets and communication services to remote regions [1]. Current Geostationary (GEO) broadband satellite systems use a multibeam footprint strategy to enhance spectrum utilization. In these systems, both power and bandwidth resources are typically allocated uniformly across the various beams. While this uniform allocation simplifies resource management, it may lead to inefficiencies in scenarios with varying traffic demands. Some beams may experience high demand, exceeding their available capacity, while others may have underutilized resources. This challenge has prompted research into more adaptive and dynamic resource allocation methods. In this regard, flexible payloads have emerged as an enabling technology to manage limited satellite resources by dynamically adapting the frequency, bandwidth, and power of the payload transponders according to users' demand [2].

Existing approaches aim to minimize the difference between offered and required capacity while adding constraints in terms of power [3], [4], and co-channel interference [5]. The power allocation derived in [3] is solved using water-filling, whereas a sub-optimal complexity game-based dynamic power allocation (AG-DPA) solution is proposed in [4]. A modified simulated annealing algorithm, as presented in [5], outperforms conventional payload designs in matching requested capacity across beams, emphasizing its effectiveness. However, the intricate computational complexities associated with these algorithms can significantly limit their practical applicability within real-world systems. Moreover, these approaches do not adequately consider the dynamic nature of capacity requests that change over time. In this context, Machine learning (ML) algorithms emerge as a more favorable alternative, as they are able to learn from varying capacity request scenarios.

ML algorithms have gained popularity in satellite communications, particularly in resource allocation [6]. Some studies explored reinforcement learning (RL) techniques [7] to cope with the time-varying capacity; however, they introduced additional delays due to online payload controller training. Also, the RL exploration phase, aimed at discovering optimal strategies through action exploration, can occasionally result in system outages or disruptions when untested actions are selected. In contrast, [8] adopts a multi-objective optimization approach using supervised learning, offering an alternative perspective.

In this work, we extend the ML-based method in [8] which originally employed a convolutional neural network (CNN) for solving the RRM task as a classification problem. In this approach, the ML model's objective is to select the best payload configuration from a discrete set of power and bandwidth combinations, treated as distinct classes. This technique considers 8 beams with 12 configurations each, giving a total of 4.3×10^8 potential payload

configurations. We expand to 10 beams with 9 configurations each, totaling 3.5×10^9 configurations. Although the number of configurations decreases after applying the system constraints, incorporating more beams inevitably increases the number of classes. Having many classes complicates the ML model evaluation as traditional metrics like accuracy can be inadequate, and metrics like recall may not fully depict system performance. The situation worsens when dealing with imbalanced class distributions since the models may favor dominant classes, leading to bias. To address this, we reframe the RRM task as a regression problem, incorporating RRM objectives and constraints into the ML loss function. We also introduce a new metric to assess the ML model's performance, offering an alternative and insightful way to evaluate its effectiveness in the context of RRM.

This paper is organized as follows. Section II introduces the flexible payload architecture and outlines the RRM task. Section III compares regression and classification-based ML methods for flexible payload. In Section IV, we present metrics for evaluating model performance, including a new ML metric for RRM. The methods are evaluated in Section V. Finally some conclusion remarks are included in Section VI.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a GEO satellite system with a single multi-beam GEO satellite that covers a wide Earth region via B spot-beams. We focus on the forward link, considering U single-antenna user terminals (UTs) distributed across the satellite's coverage area. We assume that the considered payload can adaptably handle per-beam power and bandwidth resources.

A. Link-budget analysis

The offered capacity C_b can be written as

$$C_b = \text{BW}_b \cdot \text{SE}_b, \qquad (1)$$

where SE_b is the spectral efficiency (SE) for beam b in bps/Hz [9]. The SE is a function of the carrier to interference plus noise ratio (CINR) in the b-th beam CINR_b. The CINR in dB can be written as

$$10^{\frac{-\text{CINR}_b}{10}} = 10^{\frac{-\text{CIR}_b}{10}} + 10^{\frac{-\text{CNR}_b}{10}}, \qquad (2)$$

where CIR_b is the carrier to interference ratio and CNR_b is the carrier to noise ratio for beam b in dB. The CIR represents the ratio of the power allocated

at b-th beam (P_b, in dBW) to the interference power at b-th beam (I_b, in dBW). The CNR (in dB) can be obtained

$$\text{CNR}_b = \text{EIRP3dB}_b + G/T - A - k - \text{BW}_b, \quad (3)$$

where $\text{EIRP3dB}_b = P_b + G_b$ is the effective isotropic radiated power in dBW, G_b is the beam gain that depends on the half power beamwidth $\theta_{3\text{dB}}$ in dBi, G/T is the merit figure of the user terminal, A is the free space attenuation in clear sky conditions in dB, and k is the Boltzmann constant.

With a particular P_b, we determine the CINR_b and subsequently the SE_b. Then, we obtain the capacity as in equation (1) using SE_b and BW_b.

B. Traffic demand

To generate instances of satellite traffic demand at specific time instances, we employ the SnT Traffic Emulator [10]. This emulator utilizes three distinct input datasets: population data, aeronautical data, and maritime data. The data are processed to create a matrix representing the traffic demand. In this matrix, each position i, j corresponds to a geographic location, and the value $r_{i,j}$ denotes the traffic demand in that specific geographic location in bits per second (bps). From $r_{i,j}$, the requested capacity R_b is calculated by aggregating all the (i,j) points within the coverage region of beam b.

C. RRM task

The RRM aims to effectively allocate the available satellite resources such as power P_b and bandwidth BW_{b_c} so that C_b matches R_b for each beam $b = 1, \cdots B$ over time t, avoiding resource waste. The RRM task can be formulated as the following minimization problem [9]

$$\min_{P_b(t), \text{BW}_{b_c}(t)} \quad \frac{\beta_1}{B} \sum_{b=1}^{B} |C_b(t) - R_b(t)| + $$
$$+ \frac{\beta_2}{B} \sum_{b=1}^{B} P_b(t) + \frac{\beta_3}{B} \sum_{c=1}^{N_c} \sum_{b_c=1}^{B_c} \text{BW}_{b_c}(t) \qquad (4)$$

$$\text{s.t:} \quad C_b(t) \geq R_b(t) \text{ if } \{P_b(t) < P_{\max,b}$$
$$\text{and } \mathrm{BW}_{b_c} < \mathrm{BW}_{\max,b}\} \tag{5}$$
$$C_b(t) = C_{\max}(t) \text{ if } \{P_b(t) = P_{\max,b}$$
$$\text{and } \mathrm{BW}_{b_c} = \mathrm{BW}_{\max,b}\} \tag{6}$$
$$\sum_{b=1}^{B} P_b(t) \leq P_{\max,\mathrm{T}} \tag{7}$$
$$\sum_{b_c=1}^{B_c} \mathrm{BW}_{b_c} \leq \mathrm{BW}_{\max,c}. \tag{8}$$

The cost function in equation (4) aims to simultaneously minimize three terms: the difference between C_b and R_b, the power P_b (in W), and the bandwidth BW_{b_c} (in Hz) across all beams for each time instant t. The weights β_1, β_2, and β_3 are assigned to each term to indicate its relative importance.

The constraint in equation (5) ensures that offered capacity either meet or surpass the required capacity for each beam, under the condition that both the power and bandwidth allocations for the b-th beam at time t do not exceed their upper bounds. On the other hand, if the required capacity in the b-th beam is greater than the system can provide, the offered capacity within the b-th beam is inevitably capped at its maximum attainable value as accounted in (6).

To manage the overall power consumption, the total power $\sum_{b=1}^{B} P_b(t)$ is constrained to not surpass the prescribed upper limit $P_{\max,\mathrm{T}}$ in equation (7). Moreover, the constraint in equation (8) imposes an upper limit on the total bandwidth allocated in each color of the frequency plan, ensuring that it does not exceed the available bandwidth per color, $\mathrm{BW}_{\max,c}$. The total bandwidth is allocated to the beams of each color c within the frequency plan comprising N_c colors where B_c is the number of beams with the same frequency and polarization defined by color c.

III. PROPOSED ML-BASED FLEXIBLE PAYLOAD METHODS

In this section, we propose a CNN model to solve the RRM task as a regression problem, which involves determining the optimal payload configuration for specific traffic demands. Subsection III-A introduces the dataset used to train and evaluate the model. In subsection III-B, we summarize the CNN model used for classification in [8] and in subsection III-C we detailed our proposed model.

A. Dataset and preprocessing

The dataset consists of M labeled examples, where each example is a data point with associated features and a target label.

The data points are matrices representing the traffic demand at each geographic location in the service area. Preprocessing steps include data reduction via Max-Pooling filters, standardization, and principal component analysis (PCA) to extract relevant features and reduce training complexity.

The target label corresponds to the optimal payload configuration for a given traffic demand matrix. This configuration minimizes a cost function (equation 4) while satisfying the constraints (equations 5-8). The nature of the target label varies depending on whether it's a classification or regression problem. In classification, the label is categorical, denoting the class to which each data point belongs. In regression, it's continuous, representing numeric values. We outline the specific target labels for each ML problem as follows.

B. ML-based flexible payload via classification

In [8], the RRM task is treated as a classification problem with L classes representing possible payload configurations. The softmax activation function is applied in the CNN's output layer to convert raw output scores (logits) into a probability distribution over these classes. The final class is obtained by selecting the class with the highest probability. The model is trained by minimizing cross-entropy error, a standard loss function for classification tasks that measures the dissimilarity between predicted class probabilities and actual labels.

In this classification approach, the original optimization problem (as represented in equation (4)) is only considered when generating the training dataset and defining the classes. That is, the optimization problem is not considered when formulating the loss function used to train the neural network. In essence, the loss function used during the neural network's training phase is designed to optimize the model's performance and is distinct from the original optimization problem associated with RRM.

C. ML-based flexible payload via regression

When approaching the RRM task as a regression problem, the target labels are the desired offered capacity values for each beam. The linear activation

function is used so that the neurons produce continuous values as their outputs: the predicted offered capacity values. In this case, the payload configuration with offered capacity values more similar to the ones obtained by the ML-model is selected. The similarity here is measured in terms of minimum mean absolute error (MAE).

The model is trained by minimizing the mean square error (MSE) between the output and the training label while considering a penalty term to account for the constraint in equation (5). This penalty term ensures that the model's predictions satisfies the constraint, thus aligning this approach with the RRM problem.

It is important to note that the MSE between the output and the training label is related to the first term being minimized in equation (4). This means that our regression-based approach consistently maintains a connection to the core optimization objectives of the RRM task, not only during the generation of the training dataset but also throughout the CNN training process.

IV. PERFORMANCE ASSESSMENT

In this section, we assess the model's performance and determine its ability to make accurate predictions when presented with unseen data.

A. Traditional ML metrics

In this subsection, we evaluate the performance of the CNN models using ML metrics. These metrics offer quantitative measures of model effectiveness across various tasks, such as classification and regression. The choice of metric depends on the specific problem being addressed. For regression, we employ metrics like MSE, MAE, and R-squared (R2) to evaluate the model's predictive accuracy in estimating continuous numeric values. On the other hand, classification tasks rely on metrics such as accuracy, precision, recall, and F1-score to assess the model's ability to correctly categorize data into discrete classes.

The accuracy, for instance, is defined as

$$\eta = \frac{T_N + T_P}{T_N + F_P + T_P + F_N},\qquad(9)$$

for a binary classification problem with positive and negative classes, where T_N and T_P are the true negative and true positive, and FN and FP are the false negative and false positive.

When the dataset is imbalanced, recall and balanced accuracy are most suitable to evaluate the

model. The recall is a type of accuracy per class defined as

$$\sigma = \frac{T_P}{T_P + F_N},\qquad(10)$$

By averaging the recall for each class, we obtain the balanced accuracy which is defined as

$$\phi = \frac{1}{2}\left(\frac{T_P}{T_P + F_N} + \frac{T_N}{T_N + F_P}\right).\qquad(11)$$

B. Proposed ML metric for RRM task

Our main goal is to fulfill the capacity demand for each beam, prioritizing capacity compliance over the perfect prediction of payload configurations. In this regard, we introduce the concept of flexible accuracy per payload configuration, a metric inspired by the recall equation (10). This flexible accuracy per payload configuration

$$\theta_l = \frac{1}{B}\sum_{b=1}^{B}\frac{S}{M_l},\qquad(12)$$

evaluates the performance of flexible payload models, where S is the number of instances in which the offered capacity in beam b was sufficient, and M_l is the number of samples is class l. As in equation (11), we obtain the average flexible accuracy or balanced flexible accuracy

$$\bar{\theta} = \frac{1}{L}\sum_{l=1}^{L}\theta_l.\qquad(13)$$

C. System performance metrics

When evaluating the model in terms of the system performance, we are interested in finding a payload configuration that ensures that the offered capacity satisfies the requested capacity for each beam. The offered capacity $\mathbf{c}_m = [C_{1,m}, C_{2,m}, C_{B,m}]$ is obtained after acquiring the payload configuration using the ML model. We then use the normalized mean square error (NMSE)

$$\nu_m = \frac{\sum[(\mathbf{c}_m - \mathbf{r}_m)^2]}{\sum[(\mathbf{r}_m)^2]}, m = 1 \cdots M_{\text{test}}\qquad(14)$$

to measure the similarity between the offered $\mathbf{c}_m$ and requested capacity $\mathbf{r}_m = [R_{1,m}, R_{2,m}, R_{B,m}]$ for each sample $m = 1, \cdots M_{\text{test}}$ in the test dataset. Then the average NMSE can be computed as

$$\nu_{\text{avg}} = \frac{1}{M_{\text{test}}}\sum_{m=1}^{M_{\text{test}}}\nu_m.\qquad(15)$$

V. SIMULATION RESULTS

A. Simulation Setup

The center frequency is 19 GHz, the satellite is positioned at 13 E, and the satellite altitude is 35786 km. The merit figure is $G/T = 17$ dB/K. The number of beams in the system is $B = 10$ and the beam centres are at

$$\phi_{\text{lat}} = [39.3, 42, 44.7, 47.4, 51, 53.7, 56.4, 39.5, 42.2, 49]$$
$$\phi_{\text{long}} = [-5.3, 0, 5.3, 10.6, -0.5, 6, 12.3, 11.4, 16.7, 17.4].$$
$$(16)$$

By varying bandwidth and power, as described in subsection II-A, we obtained the capacity per beam options shown in Table I.

TABLE I: Possible resource allocations in a beam

Index	BW_b [MHz]	P_b [dBW]	$EIRP3dB_b$ [dBW]	$CINR_b$	SE_b [bps/Hz]	C_b [Mbps]
1	150	10	49.93	6.6670	1.9246	288.6844
2	250	10	49.93	4.4741	1.5187	379.6639
3	500	10	49.93	1.4831	0.9650	482.5084
4	150	12	51.94	8.6396	2.2897	343.4547
5	250	12	51.94	6.4615	1.8865	471.6312
6	500	12	51.94	3.4817	1.3350	667.4827
7	150	14	55.94	12.4904	3.0025	450.3720
8	250	14	55.94	10.3705	2.6101	652.5215
9	500	14	55.94	7.4357	2.0668	1033.4

The dataset comprises $M = 30,000$ samples, with 70% allocated for training and 15% each for validation and testing. The stochastic gradient descent (SGD) optimizer with learning rate $\mu = 0.01$ is used to minimize the loss function.

B. Model evaluation and system performance

Due to the utilization of a realistic traffic model, certain payload configurations are more frequently generated within the system. Therefore, the dataset exhibits class imbalance, where some classes have a greater number of samples than others as shown in Figure 1. In cases of imbalanced datasets, recall is the preferred metric for assessing the classification model's performance, as discussed in subsection IV-A. Figure 2 presents the recall (green line) for each class when evaluating the ML-based flexible payload model using the classification approach (CNN_C).

Classes with a higher number of data samples, such as classes 5 and 25, are corrected and classified by the model with an individual accuracy higher than 95%, as indicated by the red line in Figure 2. However, many classes are misclassified when considering CNN_C. Figure 2 also includes the flexible accuracy
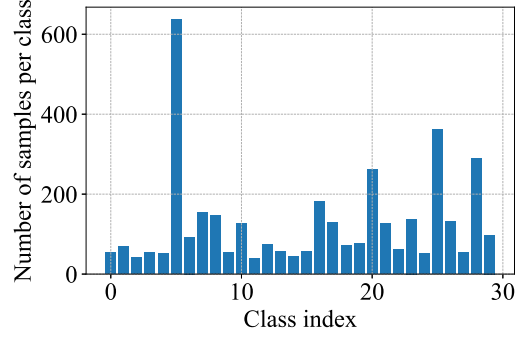

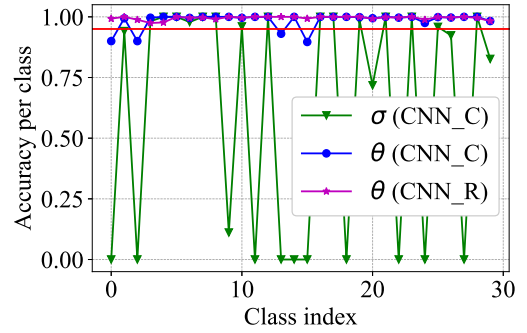Fig. 1: Number of samples per class in the validation set.

Fig. 2: Proposed CNNs.

per class for CNN_C (blue line). We can observe an improvement in terms of accuracy per class for CNN_C with this new metric, but some classes still have an individual accuracy below 95%.

The MSE results for the ML-based flexible payload model using regression are shown in Figure 3. In contrast to CNN_C, CNN_R demonstrates satisfactory performance when evaluated with a traditional ML metric. Figure 2 presents the flexible accuracy per class (in magenta) for the ML-based flexible payload model via regression (CNN_R). In such a case, all payload configurations achieve an accuracy higher than 95%. This emphasizes the effectiveness of combining an appropriate evaluation metric with a model that intricately captures the RRM task. The average offered capacity and requested capacity are compared in Figure 4. On average, both models are able to satisfactorily obtain a payload configuration that leads to an offered capacity that satisfies the demand. In Table II, we compare all the metrics used to evaluate the ML models. The traditional ML metrics used to evaluate the classification model CNN_C failed to effectively capture the model's abil-
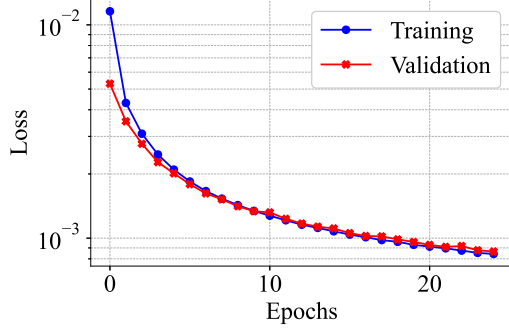
Fig. 3: MSE obtained using the ML-based flexible payload model via regression.

ity to meet the desired system benchmarks. On the other hand, when evaluating both the classification and regression models with the new ML metric, our expectations align more closely with the system's specific performance requirements.
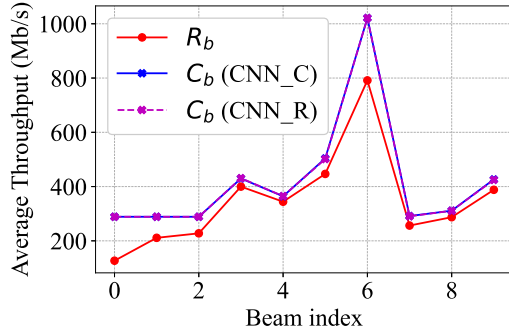

Fig. 4: Comparison between average offered capacity C_b and requested capacity R_b.

TABLE II: Comparison between evaluation metrics

Type of Metric	Metric	Value
ML	Balanced accuracy for CNN_C	61.28 %
ML	MSE + penalty term for CNN_R	0.00086
ML + System	Flexible balanced accuracy for CNN_C	98.50 %
ML + System	Flexible balanced accuracy for CNN_R	99.51 %
System	NMSE for CNN_C	0.0765
System	NMSE for CNN_R	0.0771

VI. Conclusions

In this work, we introduced a CNN to solve the RRM task considering flexible bandwidth and power. The RRM objective function and constraints were included in the ML loss function. We also proposed a new metric designed to balance traditional machine learning evaluation metrics and system performance.

The simulation results indicate that traditional ML metrics fail to capture the system's requirements, whereas the proposed metric is robust to it.

References

[1] S. K. Sharma, S. Chatzinotas, and P.-D. Arapoglou, *Satellite communications in the 5G era*. Institution of Engineering and Technology, 2018.

[2] S. Kisseleff, E. Lagunas, T. S. Abdu, S. Chatzinotas, and B. Ottersten, "Radio resource management techniques for multibeam satellite systems," *IEEE Communications Letters*, vol. 25, no. 8, pp. 2448–2452, 2020.

[3] J. P. Choi and V. W. Chan, "Optimum power and beam allocation based on traffic demands and channel conditions over satellite downlinks," *IEEE Transactions on Wireless Communications*, vol. 4, no. 6, pp. 2983–2993, 2005.

[4] S. Liu, Y. Fan, Y. Hu, D. Wang, L. Liu, and L. Gao, "Ag-dpa: Assignment game–based dynamic power allocation in multibeam satellite systems," *International Journal of Satellite Communications and Networking*, vol. 38, no. 1, pp. 74–83, 2020.

[5] G. Cocco, T. De Cola, M. Angelone, Z. Katona, and S. Erl, "Radio resource management optimization of flexible satellite payloads for dvb-s2 systems," *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 266–280, 2017.

[6] P. V. R. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Bilén, R. C. Reinhart, and D. J. Mortensen, "Multiobjective reinforcement learning for cognitive satellite communications using deep neural network ensembles," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 5, pp. 1030–1041, 2018.

[7] Y. Huang, W. Shufan, Z. Zhankui, K. Zeyu, M. Zhongcheng, and H. Huang, "Sequential dynamic resource allocation in multibeam satellite systems: A learning-based optimization method," *Chinese Journal of Aeronautics*, vol. 36, no. 6, pp. 288–301, 2023.

[8] F. G. Ortiz-Gomez, L. Lei, E. Lagunas, R. Martinez, D. Tarchi, J. Querol, M. A. Salas-Natera, and S. Chatzinotas, "Machine learning for radio resource management in multibeam geo satellite systems," *Electronics*, vol. 11, no. 7, p. 992, 2022.

[9] F. G. Ortiz-Gómez and et Al., "Supervised machine learning for power and bandwidth management in very high throughput satellite systems," *International Journal of Satellite Communications and Networking*, vol. 40, no. 6, pp. 392–407, 2022.

[10] H. Al-Hraishawi, E. Lagunas, and S. Chatzinotas, "Traffic simulator for multibeam satellite communication systems," in *2020 10th Advanced Satellite Multimedia Systems Conference and the 16th Signal Processing for Space Communications Workshop (ASMS/SPSC)*. IEEE, 2020, pp. 1–8.