

# Convolutional Neural Networks for Flexible Payload Management in VHTS Systems

Flor G. Ortiz-Gomez , Daniele Tarchi , *Senior Member, IEEE*, Ramón Martínez ,  
Alessandro Vanelli-Coralli , *Senior Member, IEEE*, Miguel A. Salas-Natera , and Salvador Landeros-Ayala 

**Abstract**—Very high throughput satellite (VHTS) systems are expected to have a large increase in traffic demand in the near future. However, this increase will not be uniform throughout the service area due to the nonuniform user distribution, and the changing traffic demand during the day. This problem is addressed using flexible payload architectures, enabling the allocation of the payload resources in a flexible manner to meet traffic demand of each beam, leading to dynamic resource management (DRM) approaches. However, DRM adds significant complexity to the VHTS systems, which is why in this article, we are analyzing the use of convolutional neural networks (CNNs) to manage the resources available in flexible payload architectures for DRM. The VHTS system model is first outlined, for introducing the DRM problem statement and the CNN-based solution. A comparison between different payload architectures is performed in terms of DRM response, and the CNN algorithm performance is compared by three other algorithms, previously suggested in the literature to demonstrate the effectiveness of the suggested approach and to examine all the challenges involved.

**Index Terms**—Bandwidth allocation, beamwidth allocation, convolutional neural network (CNN), deep learning (DL), dynamic resource management (DRM), flexible payload, machine learning (ML), power allocation.

## I. INTRODUCTION

VERY high throughput satellite (VHTS) systems have a key role in supporting future 5G and broadcast terrestrial networks [1], [2]. VHTS systems overcome the capacity of traditional systems that provide fixed and mobile satellite services using contoured regional footprints. The VHTS objective is to achieve Terabit/s data rate per satellite in the near future

Manuscript received April 15, 2020; revised July 30, 2020; accepted August 25, 2020. Date of publication September 10, 2020; date of current version August 26, 2021. This work was supported in part by the Spanish Government, Ministerio de Economía, Industria y Competitividad through the National Program of Research, Development, and Innovation within the FUTURE-RADIO Project under Grant TEC2017-85529-C3-1-R and in part by a Ph.D. scholarship provided by the Universidad Politécnica de Madrid. (*Corresponding author: Flor G. Ortiz-Gomez.*)

Flor G. Ortiz-Gomez, Ramón Martínez, and Miguel A. Salas-Natera are with the Information Processing and Telecommunications Center, Universidad Politécnica de Madrid, 28040 Madrid, Spain (e-mail: fdg.ortiz@upm.es; ramon.martinez@upm.es; miguel.salas@upm.es).

Daniele Tarchi and Alessandro Vanelli-Coralli are with the Department of Electrical, Electronic, and Information Engineering, University of Bologna, 40136 Bologna, Italy (e-mail: daniele.tarchi@unibo.it; alessandro.vanelli@unibo.it).

Salvador Landeros-Ayala is with Agencia Espacial Mexicana, Mexico City 01020, Mexico (e-mail: landeros.salvador@aem.gob.mx).

Digital Object Identifier 10.1109/JSYST.2020.3020038

[3], based on multibeam coverage with polarization schemes, frequency reuse, and spectrum optimization [4].

Nowadays, VHTS systems provide uniform throughput over the entire service area; however, traffic demands are expected to be nonuniformly distributed over the service area since the user distribution is not uniform within the coverage. This will result in a system where some beams do not have the required capacity, i.e., not meeting the traffic demands, whereas other beams overcome the required capacity or, simply, wasting resources [5], [6]. On the other hand, operators claim that one of the main challenges in the design of future satellite broadband systems is the way to increase the satellite revenues while addressing uneven and dynamic traffic demands. In that sense, flexible payload is a promising solution to meet changing traffic demand patterns [3], [7]–[9].

Most of the existing satellite communication (SatCom) payloads do not offer any flexibility in terms of neither bandwidth nor coverage. Power flexibility can be instead achieved by modifying the working point of the on-board amplifier according to the transponder loading. Recently, research interests have been focusing on designing a new generation of flexible satellite payloads enabling dynamic resource management (DRM) [7]–[9].

Cocco *et al.* [10] represent the problem of radio resource allocation for VHTS as an objective function that minimizes the error between the offered and the required capacity. However, extensive analysis is required for both the payload architecture design and resource management.

The next-generation VHTS systems will provide Terabit connections using advanced flexible payloads, which will allow the redirection and reconfiguration of beams, in addition to individual per-beam power and bandwidth allocation. Thus, DRM techniques for SatCom will be a key for operators [11]. While it may seem feasible to achieve a solution to this problem through optimization techniques, on a larger scale, the number of resources to be managed, the constraints coming from the system and the infinite number of traffic demand situations may lead to a problem that cannot be solved by conventional techniques [12]. To solve this problem, Liu *et al.* [13] suggested an assignment game-based dynamic power allocation (AG-DPA) to achieve suboptimal low complexity in multibeam satellite systems. The authors compare the results obtained with a proportional power allocation (PPA) algorithm, obtaining a remarkable advantage in terms of power saving; however, the management of resources is still insufficient for the required demand.

Recently, there is an increasing amount of research effort proposing to solve the problem of DRM in SatCom using machine learning (ML) techniques. In this sense, Ferreira *et al.* [14] discuss the reason why ML techniques should be used instead of traditional optimization techniques for resource allocation in SatCom. The authors comment that when the number of communication resources and the selection of a set present contradictory objectives, the use of traditional optimization techniques is limited. Alternatively, the use of ML techniques in SatComs overcame some of the limitations of other approaches to resource allocation in cognitive radios.

To this aim, the interest to use ML algorithms in satellite systems has increased recently [14]–[16]. There have been some technological advances in the use of ML on-board communications satellites (e.g., by NASA for cognitive space communications) [17]–[19]. On the other hand, a model for satellite image motion blurring using a convolutional neural network (CNN) is proposed in [20], and the results obtained confirm that the proposed method reliably eliminates more motion blurring than conventional methods. In this regard, a new method for satellite rapid positioning using neural network based beam scanning is proposed in [21].

In [22], the application of reinforced learning to manage complex space links operations occurring in LEO and Deep Space missions shows promising results. Here, authors propose the use of software-defined radio technology to enable the flexibility and configurability that future cognitive communication missions will require, thus allowing the system to adapt to changes using software updates.

Ortiz-Gomez *et al.* [23], [24] propose to solve the DRM problem using a neural network through a classification algorithm, where classes correspond to all the possible configurations of payload resource allocation. The management of the payload resources is performed in an autonomous way; the main advantage of this methodology is that the management is performed with a low computational cost, since the neural network training is performed offline. However, this methodology has several challenges; one of them is the exponential dependence of number of classes on the number of beams in addition to possible variations of power, bandwidth, and/or beamwidth, resulting in unsolvable problems due to flexibility increase.

Liu *et al.* [16] suggest a novel dynamic channel allocation algorithm based on deep reinforcement learning (DRL-DCA) in multibeam satellite systems, where the results showed that this algorithm can achieve a lower blocking probability, compared to traditional algorithms; however, the joint channel and the power allocation algorithm is not taken into consideration. Preceding with the previous algorithms, based on the deep reinforcement learning (DRL) architectures, Ferreira *et al.* [14], [15] demonstrated that a feasible solution to real-time and single-channel resource allocation problems can be designed. However, in their proposed study, DRL architectures are based on quantizing the resources before they are allocated, whereas satellite resources, such as power, are inherently continuous. In that sense, Luis *et al.* [25] explore a DRL architecture for power allocation that uses continuous and state action spaces, avoiding the need for discretization. Nonetheless, the policy is not optimal, as some demand is still being lost.

The operation of DRL algorithms for DRM has proven to have great advantages among the various ML algorithms; however, latency has a vital role in SatCom. Hence, the great disadvantage of DRL algorithms is the additional latency due to the online processing delay. When the algorithm is implemented offline, the DRM functions as an intelligent switch for the SatCom system, reducing the added latency [23].

Some of the recent literature considers that DRM is able to take into account only two possible flexible resources [10], [11], [23], i.e., power and bandwidth, whereas other studies consider that only one flexible resources can be handled [14], [15], [25]. Differently from previous approaches, the main objective of this article is to solve the DRM problem using an ML algorithm based on CNNs.

The main contributions of this article can be listed as follows.

- 1) CNN algorithm is suggested and allows us to implement offline the system, thus avoiding the latency problems in DRL.
- 2) A novel system model and cost function are suggested that allows for an optimal solution, which determines how to match resources to a demand pattern while minimizing the resource consumption of satellites.
- 3) Differently from previous approaches, we consider three possible flexible resources for the study of DRM, i.e., power, bandwidth, and beamwidth.
- 4) In this article, a comparison is made between different flexible payloads according to the three previously identified resources.

The CNN algorithm performance is compared with three other algorithms: PPA [13], AG-DPA [13], and classification algorithm [23].

The article is organized as follows. Section II includes the considered system model and the problem statement, Section III presents the CNN-based algorithm, Section IV presents the simulation results and the analysis of the case study, and finally, Section V concludes the article. In the Appendix, the CNN design for the considered DRM problem is outlined.

## II. SYSTEM MODEL AND PROBLEM STATEMENT

In this section, the system architecture is introduced, and the DRM problem is defined.

### A. System Architecture

In Fig. 1, the high-level model of the system is depicted. The considered payload is supposed to be able to manage in a flexible way, three resource types, i.e., power, bandwidth, and beamwidth, similar to that in [26] and [27]. The flexibility in these resources is achieved using the architecture shown in Fig. 1.

From the technological point of view, the flexible power allocation can be obtained through traveling wave tube amplifiers adapting the input back-off. At the same time, the flexible payload must be able to separate the signals into frequency blocks and then rearrange them to obtain a flexible bandwidth; this process requires a channelizer on board the satellite, as mentioned in [27], able to identify the frequency plan (color

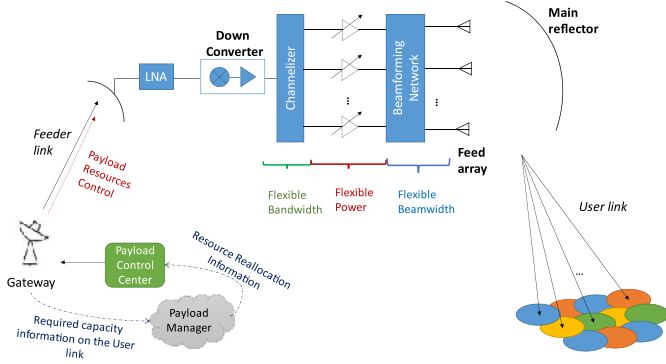


Fig. 1. System architecture.

assignment by frequency and polarization). The proposed system has a four-color frequency plan (two frequencies and two polarizations). Finally, to achieve flexibility in beamwidth, the output multiplexer of the traditional payload must be replaced by beam forming networks (BFNs). The BFN configuration change is halfway between the possibility of synthesizing any beam and the possibility of choosing from a set of configurations for the same coverage [7].

The proposed system manages communication resources in response to changes in traffic demand. The payload manager receives the input data from the gateways and the user beams, and, then, generates an optimal control through the payload control center using a DRM and sends it to the satellite. This in turn affects the downlinks of the users.

In the following, we consider that the system is composed by  $B$  beams, in which the overall bandwidth is  $BW$  and the transmission power of the  $b$ th beam is  $P_b$ .

### B. Link Budget

We assume that capacity  $C_b$  offered by the  $b$ th beam depends on the bandwidth allocated to the beam,  $BW_b$ , and the spectral efficiency in the beam,  $SE_b$

$$C_b = BW_b SE_b \quad (1)$$

$$SE_b = f_1(\text{CINR}_b) \quad (2)$$

where  $SE_b$  is the spectral efficiency of the modulation and coding scheme of a commercial reference modem used to obtain the capacity of each beam [28]. Hence, in (2), the variation of the spectral efficiency of each beam with the carrier to interference-plus-noise ratio (CINR<sub>*b*</sub>) is modeled through a generic function  $f_1(\cdot)$  as in [3] and [28].

A bent-pipe transponder architecture is considered in the satellite. The feeder link from the gateway to the satellite is not considered in the forward link because different technologies are considered for guaranteeing the total link budget, such as uplink power control, and gateway diversity [29], [30]. In this sense, the downlink link budget in the user link can be written as

$$10^{-\frac{\text{CINR}_b}{10}} = 10^{-\frac{\text{CIR}_b}{10}} + 10^{-\frac{\text{CNR}_b}{10}} \quad (3)$$

where

$$\text{CIR}_b = P_b - I_b \quad (4)$$

$$I_b = \sum_{\varphi=1}^{\Phi} P_{\text{co}}(\phi, \theta_b) \quad (5)$$

$$\text{CNR}_b = P_b + G_b + G_u - L_b - 10 \cdot \log_{10}(k \cdot T_{\text{sys}}) - 10 \cdot \log_{10}(\text{BW}_b) \quad (6)$$

$$L_b = L_{\text{FS},b} + L_{\text{atm},b} + L_{\text{RF},b} + L_{\text{RF},u} \quad (7)$$

$$G_b = f_2(\theta_b). \quad (8)$$

CINR<sub>*b*</sub>, CIR<sub>*b*</sub> (i.e., carrier to interference ratio), and CNR<sub>*b*</sub> (i.e., carrier to noise ratio) are expressed in decibel. CIR<sub>*b*</sub> in (4) 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 beam gain must be evaluated for the  $b$ th beam in the service area and it depends on  $\theta_b$ , which is the beamwidth, i.e., the 3-dB aperture angle of the  $b$ th beam. The sidelobes of the satellite antenna pattern are taken into consideration only for the calculation of the cochannel interference due to spatially separated cochannel beams (the same color in the frequency plan). The cochannel interference power ( $I_b$ ) is a function of the frequency reuse scheme and  $\theta_b$ , and can be calculated, assuming the cochannel beams set in the system (5), where  $\varphi$  represents the  $\varphi$ th interferer spot,  $\Phi$  is the total number of interfering beams of the beam  $b$ , and  $P_{\text{co}}$  is the power level (in W) of  $\varphi$ th interference inside the  $b$ th beam.

The traditional calculation of the CNR<sub>*b*</sub> is defined as a function of the beam power ( $P_b$ , in dBW), beam gain ( $G_b$ , in dB), and bandwidth ( $BW_b$ , in Hz) assigned to each beam as presented in (6).  $G_b$  represents the beam antenna gain in the user location, whose dependence with beamwidth ( $\theta_b$ ) can be modeled through an adequate function  $f_2(\cdot)$  [3]. CNR<sub>*b*</sub> depends on the antenna gain of the user ( $G_u$ , in dB), the Boltzmann constant ( $k$ , in W/K·Hz) and the system temperature ( $T_{\text{sys}}$ , in K). In addition, free space loss, atmospheric loss [31], transmission loss, and receive loss ( $L_{\text{FS},b}$ ,  $L_{\text{atm},b}$ ,  $L_{\text{RF},b}$ , and  $L_{\text{RF},u}$  respectively, in dB) are included for total link loss ( $L_b$ , in dB).

### C. Traffic Model

In [32], traffic models representing realistic operating scenarios are introduced. The authors consider four different datasets that provide measurements for all beams and cover a sufficiently long time-window. These data were provided by SES S.A.; by exploiting these datasets, the authors aim to represent, first, the demand behavior during a typical daily operation cycle, and second, the unfrequented cases that could lead to major service failures if the algorithms do not provide adequate results.

The reference model used in [32] represents the throughput demand in a typical commercial scenario, where a higher data rate is requested during specific time intervals of the day.

It is possible to define  $x$  as a specific geographic area of 1 km<sup>2</sup> and  $A_b$  as  $b$ th beam area (in km<sup>2</sup>). In this sense,  $r_x(t)$  is defined as the required throughput density per km<sup>2</sup> (in bps/km<sup>2</sup>) inside  $x$  at time  $t$ ,  $r_b(t)$  as expected value over all the area inside  $b$ th

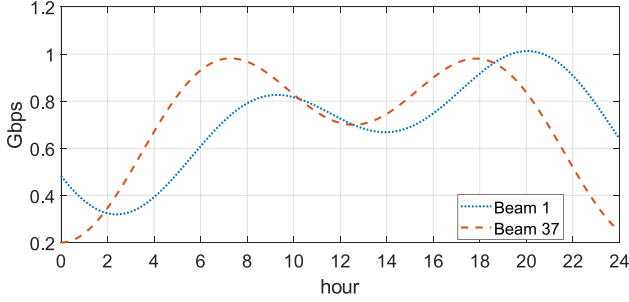


Fig. 2. One-day cycle of traffic demand per beam.

beam at time  $t$  (in bps/km<sup>2</sup>) and the requested traffic for the  $b$ th beam  $R_b(t)$  (in bps) is defined as

$$R_b(t) = r_b(t) \cdot A_b \quad (9)$$

$$r_b(t) = E[r_x(t)] \quad (10)$$

The throughput density per km<sup>2</sup> depends on the throughput per user ( $C_u$ , in bps/user), the population density ( $D_x$ , in inhabitant/km<sup>2</sup>), the penetration rate ( $F_x$ , in user/inhabitant), and the concurrence rate that depends on the time of day ( $T_x(t)$ ), hence

$$r_x(t) = C_u \cdot D_x \cdot F_x \cdot T_x(t). \quad (11)$$

To this aim, the variation of the concurrence rate throughout the day ( $T_x(t)$ ) is obtained by simulating the behavior of the data presented in [32]. The traffic demand in each beam behaves as shown in Fig. 2, in which the traffic demand varies depending on the time of day. The figure shows a one-day cycle of traffic demand in two different beams (number 1 and 37).

#### D. DRM Cost Function

The DRM must manage the available resources to minimize the error between the offered capacity in each beam ( $C_b$ ) and the required capacity ( $R_b$ ) while optimizing the used resources (power, bandwidth, and beamwidth) over time. In that sense, the DRM cost function is defined as

$$\min_{P_b(t), BW_{b_c}(t), \theta_b(t)} F_1 \rightarrow F_1 = \frac{\alpha}{B} \sum_{b=1}^B |C_b(t) - R_b(t)| + \frac{\beta}{B} \sum_{b=1}^B \text{EIRP}_b(t) + \frac{\gamma}{B} \sum_{c=1}^{N_c} \sum_{b_c=1}^{B_c} BW_{b_c}(t) \quad (12)$$

where

$$C_b(t) = f_3(P_b(t), BW_{b_c}(t), \theta_b(t)) \quad (13)$$

$$\text{EIRP}_b(t) = f_4(P_b(t), \theta_b(t)) \quad (14)$$

subject to

$$\begin{cases} C_b(t) \geq R_b(t), & \text{if } P_b(t) < P_{\max,b}, \theta_b(t) < \theta_{\max} \\ & \text{and } BW_{b_c}(t) < BW_{\max,b} \\ C_b(t) = C_{\max}(t), & \text{if } P_b(t) = P_{\max,b}, \theta_b(t) = \theta_{\max} \\ & \text{and } BW_{b_c}(t) = BW_{\max,b} \end{cases} \quad (15)$$

$$\sum_{b=1}^B P_b(t) \leq P_{\max,S} \quad (16)$$

$$\sum_{b_c=1}^{B_c} BW_{b_c}(t) \leq BW_{\max,c} \quad (17)$$

$$\theta_b(t) \in \{\theta_1, \theta_2, \dots, \theta_{\max}\} \quad \forall b, t \quad (18)$$

where the beamforming antenna generates  $B$  beams over the coverage area. The offered capacity by  $b$ th beam at the time  $t$ ,  $C_b(t)$  (in bps), must change as it depends on  $R_b(t)$  (in bps), the required capacity in the  $b$ th beam at the time  $t$ .

The DRM cost function is proposed in (12), aiming at minimizing three parameters for each time instant  $t$ . The first parameter is the error between the offered capacity and the required capacity, where  $\alpha$  (in s/b) is the weight of the error in the cost function. The second parameter to be minimized is total effective isotropic radiated power (EIRP), which is assigned to all the beams (in W), where  $\beta$  (in 1/W) is the weight of the total EIRP; the third parameter to be minimized is the total bandwidth (in Hz) that is assigned to the beams of each color ( $BW_c$ ) within the frequency plan ( $N_c$  is the number of colors in the frequency plan), and where  $\gamma$  (in 1/Hz or s) is the weight of the total bandwidth assigned in each color of the frequency plan.

From (13), the offered capacity in the  $b$ th beam at time  $t$ ,  $C_b(t)$  is a function  $f_3(\cdot)$  on the power [3], beamwidth, and the bandwidth assigned to the  $b$ th beam at time  $t$  ( $P_b(t)$ ,  $\theta_b(t)$ , and  $BW_{b_c}(t)$ , respectively). On the other hand, EIRP of the  $b$ th beam at instant  $t$ ,  $\text{EIRP}_b(t)$  is a function  $f_4(\cdot)$  on the power and beamwidth assigned to the  $b$ th beam at time  $t$  (14) [3].

Equation (15) introduces the minimum capacity constraint, stating an important limitation of the proposed cost function, where the offered capacity must be greater than or equal to the required capacity for each beam, with the condition that the power, beamwidth, and bandwidth allocated to the  $b$ th beam at time  $t$  are less than the maximum allowed for each beam ( $P_{\max,b}$ ,  $\theta_{\max}$  and  $BW_{\max,b}$ ). In case the offered capacity cannot satisfy the beam requirement constraint, the capacity offered on the  $b$ th beam shall be the maximum possible value.

Equations (16) and (17) represent the other two constraints of the cost function. These constraints are that the total power used (i.e.,  $\sum_{b=1}^B P_b(t)$ ) should not be greater than the maximum total system power ( $P_{\max,S}$ ), and the total bandwidth allocated in each colour of the frequency plan (i.e.,  $\sum_{b_c=1}^{B_c} BW_{b_c}(t)$ ) should not be greater than the available bandwidth per color ( $BW_{\max,c}$ ). In addition, the beamwidth of the  $b$ th beam must belong to the set of possible configurations previously established (18). The selected beamwidths must meet the requirement of completely covering the entire service area.

### III. CNN-BASED DRM ALGORITHM

A CNN algorithm handling resource allocation is the main part of the proposed DRM. The CNN determines how to match resources to a demand pattern while minimizing the resource consumption of satellites.

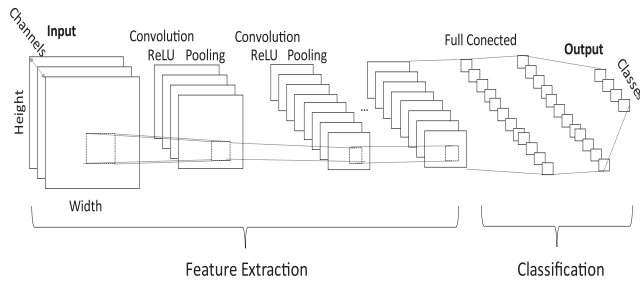


Fig. 3. CNN architecture.

Other ML algorithms are capable of providing solutions for time-variant data; however, a CNN architecture has been chosen because the distribution of traffic demand in the service area can be represented by a spatial dependence and CNN networks have demonstrated good performance in exploiting the features of spatial distributions [33], [34].

The training of the CNN is performed offline so that from the SatCom system point of view, the CNN will be operating as an intelligent switch performing the DRM.

In this section, we introduce the proposed CNN algorithm for managing the dynamic resources. For more details on CNN operations, the CNN architecture is introduced in the Appendix, whereas more general details on CNN can be found in [33]–[35].

#### A. CNN Architecture

A CNN is a deep learning (DL) algorithm that can take an input and assign importance (weights) to various aspects or objects in the input in order to distinguish one from another. The preprocessing required in a CNN is much less compared to other classification algorithms [33]–[35]. In that sense, CNN can successfully capture the spatial and temporal dependencies in the input through the application of relevant filters. The CNN architecture is optimized in comparison with other classification algorithms in order to improve the processing performance of a dataset due to the reduction of the number of parameters and the reuse of the weights. Thus, the CNN can be trained to better understand the complexity of the neural network input.

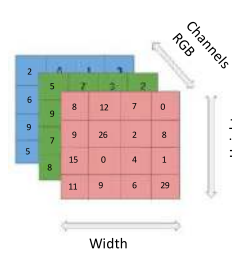
Fig. 3 presents a typical CNN architecture. In order to understand its behavior, it is divided into four main CNN operations:

- 1) convolution;
- 2) nonlinearity with a rectifier linear unit (ReLU);
- 3) pooling or subsampling;
- 4) classification (full-connected layer).

One of the most typical CNN applications is image classification. In image classification, the channel is a conventional term used to refer to a certain component of an image, e.g., the red, green, and blue channels of an image taken from a standard digital camera. In other words, an image from a standard digital camera has three channels: red, green, and blue. The channels can be seen as three 2-D matrices stacked together (one for each color), meaning at the CNN input, we have a tensor of matrices [35].

CNNs derive their name from the "convolution" operator. The main goal of convolution in the case of a CNN is to extract features from the input image (tensor of matrices). Convolution

- For Image Classification (Traditional CNN Application)



- For Resource Allocation (Proposal)

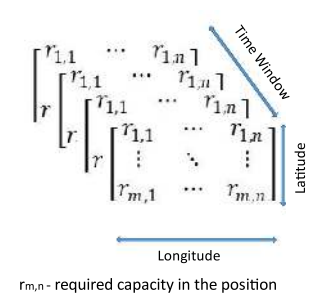


Fig. 4. Adaptation in the input layer.

preserves the spatial relationship between pixels by learning the features of the image using small input data squares.

The ReLU is an element-based operation and replaces all negative element values in the characteristics map with zero. The purpose of ReLU is to introduce nonlinearity into the CNN [33].

On the other hand, the spatial clustering (also called pooling or subsampling) reduces the dimensionality of each feature map but retains the most important information. The result of the convolutional and pooling layers represents high-level features of the input. The purpose of the full-connected layer is to use these features to classify the input image into various classes according to the training dataset [33]–[35].

CNN belongs to the collection of supervised DL algorithms, so two sets of data are required for its functioning: the training data and the test data. Observations in the training set form the experience used to learn by the algorithm. In supervised learning problems, each observation consists of one observed output variable and one or more observed inputs, whereas test data are a set of observations used to evaluate model performance using some efficiency metrics, such as accuracy [33]–[35].

#### B. CNN for DRM

In this article, we propose to adapt CNN architecture, usually considered in image classification problems, for solving the DRM problem. The adaptation of the traditional CNN architecture has been performed in the input layer and in the output layer to solve the DRM problem.

Fig. 4 represents the adaptation in the input layer where there is a tensor of matrices, and each matrix represents the required capacity at each geographical position in the service area. The service area does not have a regular geometrical shape, so the geographical coordinates contained in each matrix outside the service area are padded to zero. With this adaptation, the depth of the matrix tensor does not represent the channels in an image, though it stands for the time instants (states) in which the system is evaluated. That is, the depth of the matrix tensor is given by the vector  $\{t, t-1, t-2, \dots, t-T\}$ , where  $t$  is the current time instant and  $T$  is the time window size.

Depending on these features; constraints and flexibility of the payload; there is a set of possible configurations for allocating resources in the beams. These possible configurations are coded

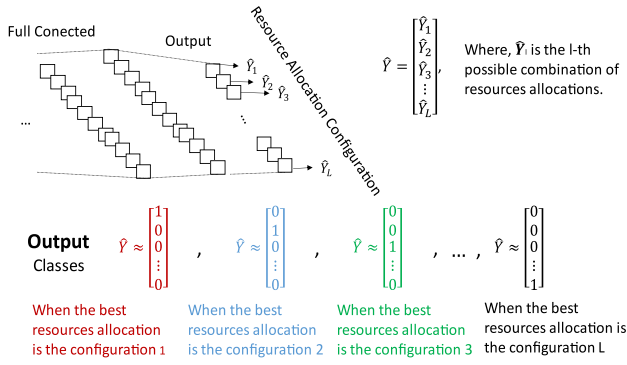


Fig. 5. Adaptation in the output layer.

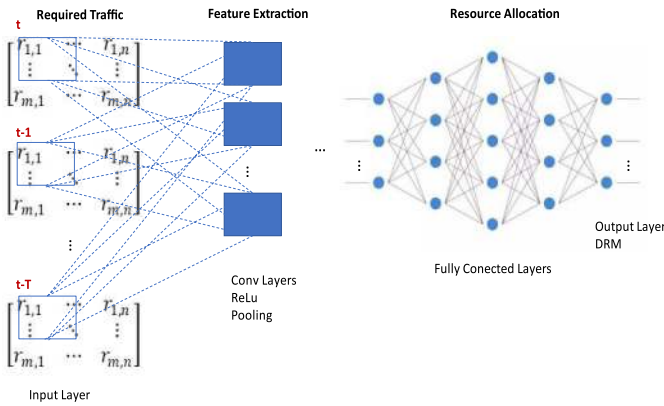


Fig. 6. Scheme of the different layers of the CNN for DRM.

in a vector of size  $L$ , representing the number of possible configurations of the payload resources. By exploiting this, the CNN has, in the output layer, the configuration that minimizes the cost function in (12) for the conditions of the input layer. This is shown in Fig. 5.

Fig. 6 shows the scheme of the different layers of the CNN. The input layer is a matrix tensor with the required capacities in each geographic coordinate. The convolution layers are used to obtain the main features of the traffic demand in the geographic coordinates. The full-connected layers are used to allocate the resource configuration with the features obtained in the convolution layers. The output layer results in a resource allocation that minimizes the DRM cost function (12).

In convolutional layers, single neurons in a perceptron are replaced by matrix processors, thus they perform an operation on the input matrix data, rather than on a single numerical value. For a better understanding, in the Appendix, the CNN architecture and related parameters are explained with additional details. The output of each convolutional neuron is calculated as [33]

$$Y_j = g_1 \left( p_j + \sum_i K_{ij} \otimes Y_i \right) \quad (19)$$

where  $Y_j$ , the output of the  $j$ th neuron, is a matrix calculated as linear combination of the outputs  $Y_i$  of the neurons in the previous layer, each operated with the convolutional  $K_{ij}$  kernel

corresponding to that connection; this amount is added to a  $p_j$  connection and then passed through a nonlinear activation function  $g_1(\cdot)$ . The convolution operator has the effect of filtering the input matrix with a previously trained kernel.

After the convolution layers, the data finally reach the full-connected layers, where the data are then debugged; thus, these layers implement the resource configuration that minimizes the DRM cost function (12). The neurons in these layers work identically to those in a multilayer perceptron, where the output of each perceptron of each layer is calculated as [33]

$$y_j = g_2 \left( p_j + \sum_i w_{ij} \cdot y_i \right). \quad (20)$$

The output  $y_j$  of the  $j$ th neuron is a value that is calculated as the linear combination of the outputs  $y_i$  of the neurons in the previous layer, each multiplied with a weight  $w_{ij}$  corresponding to that connection. This amount is added to an influence term  $p_j$  and then passed through a nonlinear activation function  $g_2(\cdot)$ .

The CNN cost function is represented by the error between the expected and the obtained value

$$F_2 = \frac{1}{D} \sum_{d=1}^D Q(\tilde{y}, \Omega) \quad (21)$$

where

$$\Omega = f_5(W, K) \quad (22)$$

where  $\Omega$  is used to denote the CNN configuration. That is,  $\Omega$  is a function  $f_5(\cdot)$  of  $W$  and  $K$ , the values of the weights and biases, whereas  $D$  is the size of the training set,  $Q(\cdot)$  is a function to be determined by CNN configuration, and  $\tilde{y}$  represents the expected values [33].

Training data are generated with the traffic demand model defined in (9) and (10), and the labels designated ( $Y$ ) for training are obtained with the DRM cost function (12).

CNN training is performed with the backpropagation algorithm [34]. An accuracy parameter is defined for this purpose, where accuracy is a metric for measuring the performance of CNN and is defined as ratio of the total number of times the correct resource configuration was allocated ( $RA_{\text{correct}}$ ) to the total number of times a resource configuration was allocated ( $RA_{\text{total}}$ ), i.e.,

$$\text{Accuracy} = \frac{RA_{\text{correct}}}{RA_{\text{total}}}. \quad (23)$$

The proposed system manages the communication resources in response to changes in traffic demand. The payload manager receives the input data from the gateways and the user beams, generates an optimal control through the DRM, and sends it to the satellite. This in turn affects the downlinks of the users.

A CNN algorithm that handles resource allocation is at the kernel of the DRM (see Fig. 7). The CNN determines how to match resources to a demand pattern while minimizing the resource consumption of satellites. The training of the network is performed offline so that for the SatCom system the CNN represents an intelligent switch that works as a DRM.

CNN-BASED ALGORITHM	DYNAMIC	RESOURCE	MANAGEMENT
<b>Offline</b>			
1. Generate training data: Traffic demand in geographical coordinates over time			
2. The geographical coordinates of the matrix outside the service area have $r_{m,n}=0$ .			
3. Class assignment to training data: Resource configurations that minimizes de DRM cost function			
$\min_{P_b(t), BW_{bc}(t), \theta_b(t)} F_1 \rightarrow F_1 = \frac{\alpha}{B} \sum_{b=1}^B  C_b(t) - R_b(t)  + \frac{\beta}{B} \sum_{b=1}^B EIRP_b(t) + \frac{\gamma}{B} \sum_{c=1}^{N_c} \sum_{b_c=1}^{B_c} BW_{b_c}(t)$			
4. Define CNN settings			
5. While the accuracy is less that the minimum required			
$Accuracy = \frac{RA_{correct}}{RA_{total}}$			
6. Train			
7. Test			
8. End While			
<b>Online</b>			
1. Input: Traffic demand per geographic coordinate over $T$ times			
2. The geographic coordinates of the matrix outside de service area have $r_{m,n}=0$ .			
3. DRM receives the Input data			
4. DRM selects the optimal resources configuration			
5. The Payload Control Center sends the information to the satellite to adjust the resources			

Fig. 7. CNN-based DRM algorithm.

TABLE I  
ARCHITECTURE OF THE CNN

Layer	Input	Kernel	Activation	Output
Conv1	256 x 256 x T	10 x 10, 16	ReLu	247 x 247 x 6
Pool1	247 x 247 x 6	NA	NA	123 x 123 x 16
Conv2	123 x 123 x 16	8 x 8, 16	ReLu	116 x 116 x 16
Pool2	116 x 116 x 16	NA	NA	58 x 58 x 16
Conv3	58 x 58 x 16	5 x 5, 32	ReLu	54 x 54 x 32
Pool3	54 x 54 x 32	NA	NA	27 x 27 x 32
Conv4	27 x 27 x 32	3 x 3, 32	ReLu	25 x 25 x 32
Pool4	25 x 25 x 32	NA	NA	12 x 12 x 32
Flatten	12 x 12 x 32	NA	NA	4608
FC1	4608	NA	tanh	512
FC2	512	NA	tanh	512
Class	512	NA	Softmax	L

The software tool chain used to implement CNN consists of a Jupyter development environment using Keras 2.0. CNN, which has been defined using four convolutional layers (Conv) and two full-connected layers (FC), as shown in Table I. The first convolutional layer, Conv1, consists of 16 kernels, each of them  $10 \times 10$  in size. The second convolutional layer, Conv2, consists of 16 kernels, each with a size of  $8 \times 8$ . The third convolutional layer, Conv3, consists of 32 kernels, each with a size of  $5 \times 5$ , and Conv4 consists of 32 kernels, each with a size of  $3 \times 3$ . The first full-connected layer, FC1, remodels the output of Conv4 using a flatten layer. The second layer (FC2) is connected to the classification layer. The input in the Conv1 layer represents the size of the input layer tensor, with matrices of  $256 \times 256$  and a depth equal to  $T$  (i.e., time window size). One pooling layer is added after each convolutional layer, using the maximum value of  $2 \times 2$  size to reduce the size of the convolutional layers. The

TABLE II  
DIFFERENT PAYLOAD ARCHITECTURES

Payload	Features	Beamwidth	Bandwidth	Power
1	Traditional Payload	fixed	fixed	fixed
2	Flexible Beamwidth	flexible	fixed	fixed
3	Flexible Bandwidth	fixed	flexible	fixed
4	Flexible Power	fixed	fixed	flexible
5	Beamwidth and Power	flexible	fixed	flexible
6	Flexible Beamwidth and Bandwidth	flexible	flexible	fixed
7	Flexible Bandwidth and Power	fixed	flexible	flexible
8	Full-Flexible	flexible	flexible	flexible

classification layer provides the  $L$  probabilities that, given the received input, the  $l$ th resource configuration corresponds.

### C. Performance Evaluation

To evaluate the performance of the DRM algorithm, two key performance indicators (KPIs) are proposed with which an important tradeoff can be observed.

The first KPI is defined as the normalized mean error of the DRM algorithm

$$KPI_1 = \frac{1}{B \cdot \rho} \sum_{b=1}^B |C_b - R_b| \quad (24)$$

where  $\rho$  is the normalization parameter. The normalization parameter allows the comparison of the different DRM algorithms performance when used in different traffic demand scenarios.

The second KPI is the power saving [13], defined as

$$KPI_2 = \frac{P_{Total,UPA}}{P_{Total,Alg}} \quad (25)$$

where  $P_{Total,UPA}$  is the total payload power when using a uniform power allocation (UPA) and  $P_{Total,Alg}$  is total payload power when using the power allocation using the proposed algorithm.

## IV. NUMERICAL RESULTS AND ANALYSIS

In this section, the numerical results obtained in a reference scenario are presented, comparing the performance of the proposed CNN approach with the PPA, AG-DPA, and classification algorithms [13], [23]. The selected reference scenario corresponds to a multibeam coverage for Europe and the Mediterranean basin with 82 beams, similar to the coverage currently provided by KA-SAT [36].

Table II shows the characteristics of the eight different payload architectures that have been evaluated using a CNN for DRM. In order to analyze the advantage of the proposed CNN approach, we considered different payload architectures achieved by exploiting three parameters (i.e., beamwidth, bandwidth per beam, and power per beam) in a flexible way.

The flexibility obtained by changing the beamwidth provides the possibility of generating irregular beams coverage, allowing the adaptation to changing requests. However, the flexibility of the beamwidth size depends on the definition of the BFN [7]. In this sense, three possible beamwidth sizes are considered (see Table II). Each payload has fixed or flexible parameters according to the features presented in Table II.

These parameters represent the resource allocation per beam. The fixed parameters are power with 15 dBW, bandwidth with 250 MHz, and beamwidth with  $0.60^\circ$ . The flexible parameters are: power 8–15 dBW with steps of 0.5 dB, bandwidth 100, 200, or 250 MHz, and beamwidth  $0.55^\circ$ ,  $0.60^\circ$ , or  $0.65^\circ$ . The three possible beamwidth values comply with the constraint set in (17).

In a multibeam system, the power allocation is usually a continuous variable [25]; however, in this article, the power allocation is considered to be performed by selecting one value among several in a set of possible power levels per beam, due to existing technologies that allow power to be modified in 0.5-dB steps [7].

In Table II, each payload has specific features depending on the defined flexibility parameters. Those have a direct effect on the DRM cost function parameters ( $\alpha$ ,  $\beta$ , and  $\gamma$ ). The Payload 1 has all values set to zero because it represents a traditional payload (no flexibility), so it has no parameters to optimize. To minimize the error of the capacity offered and to save resources, the parameters  $\alpha$ ,  $\beta$ , and  $\gamma$  are used to normalize each term of the cost function;  $\alpha$  is equal to  $(C_{T,P1})^{-1}$  for all payloads that have some flexibility factor (Payload 2–8), where  $C_{T,P1}$  represents the total capacity provided by Payload 1 (traditional payload).  $\beta$  is equal to  $(\text{EIRP}_{T,P1})^{-1}$  for all payloads that have flexibility in power and/or beamwidth otherwise is equal to zero, where  $\text{EIRP}_{T,P1}$  represents the total EIRP provided by Payload 1 (traditional payload). And finally  $\gamma$  is equal to  $(\text{BW}_{T,P1})^{-1}$  for all payloads that have flexibility in bandwidth otherwise it is equal to zero, where  $\text{BW}_{T,P1}$  represents the total bandwidth provided by Payload 1 (traditional payload).

#### A. CNN Training Analysis

The first step to be addressed is related to the CNN training, which allows us to set up the CNN parameters for the following real-time usage.

Algorithm convergence can be observed during training and testing with accuracy as a performance measure. The CNN algorithm accuracy as defined in (23) allows us to obtain the relationship between the correctly predicted values and the desired values, obtained from the training data, during each iteration; we notice that the accuracy has a value very close to 1 both in training and in the test.

Among other influential parameters, the time window size is the parameter that influences more the CNN training. Fig. 8 shows the performance of CNN during training for four different time window sizes, where time window size  $T = 1$  corresponds to only one state, time window size  $T = 2$  corresponds to two states, time window size  $T = 3$  corresponds to three states and time window size  $T = 4$  corresponds to four states.

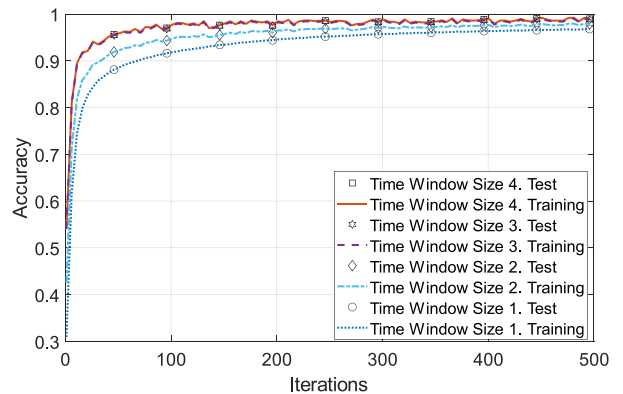


Fig. 8. Accuracy during CNN training and test for different time windows size using fully flexible payload architecture (Payload 8, see Table II).

The Payload 8 configuration was used to compare the impact of the different time window sizes on the training and test accuracy while the minimum required accuracy value is set at 0.97 [34].

Fig. 8 shows how the algorithm converges for the four time window sizes; however, the main differences are observed in the number of iterations required for the algorithm to converge and the maximum accuracy achieved. After 350 training iterations, the CNN converges regardless of the time window size showing an accuracy of at least 0.98 (in the case of Time Window Size 1). However, only 200 iterations are needed for Time Window Sizes 3 and 4 to converge.

The performance during training between using Time Window Sizes 3 and 4 is almost the same. Assuming that the larger the time window size, more delay can be added to the system; it is concluded that time window size  $T = 3$  is the most suitable for the proposed CNN approach.

#### B. CNN Performance for DRM and Comparison With Benchmark Algorithms

Once convergence is guaranteed at the training step, the DRM will work online as an intelligent switch that manages resources according to the required capacity (see Fig. 7). The DRM will attempt to minimize  $\text{KPI}_1$  in (24) while maximizing  $\text{KPI}_2$  in (25). In this sense, the performance of the algorithm used for the DRM is evaluated by exploiting a joint KPI, defined as

$$F_3 = A_1 \cdot \text{KPI}_1 + \frac{A_2}{\text{KPI}_2} \quad (26)$$

where  $A_1$  and  $A_2$  are weights parameters allowing to give different importance to the two KPIs. Better an algorithm performs lower is  $F_3$  in (26).

By assuming that  $\text{KPI}_1$  and  $\text{KPI}_2$  have the same importance, we set both  $A_1$  and  $A_2$  to 0.5; four algorithms have been evaluated for DRM using the Payload 4 architecture (see Table II), a time window size  $T = 3$ , and a normalization parameter  $\rho = R_{\max}$ , where  $R_{\max}$  represents the maximum capacity required per beam for each scenario where the algorithm was evaluated.

The evaluated algorithms are as follows:

- 1) PPA: Proportional power allocation algorithm [13];



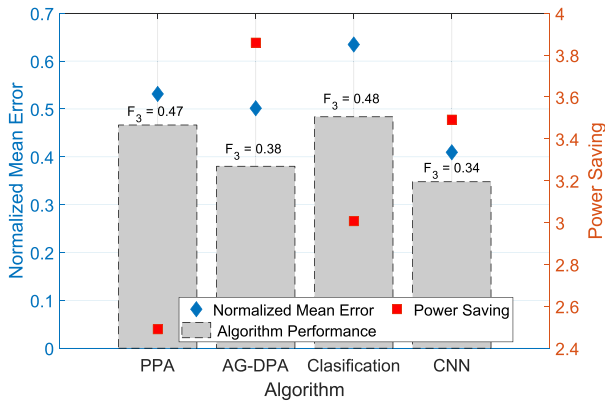


Fig. 9. Performance comparison of different algorithms for DRM using Payload 4 architecture (see Table II).

- 2) AG-DPA: Assignment game-based dynamic power allocation algorithm [13];
- 3) Classification: Neural network for classification algorithm [23];
- 4) CNN: Convolutional neural network algorithm.

Fig. 9 shows the performance comparison of the four algorithms in terms of power management. The figure shows the tradeoff between the defined KPIs. In the case of the PPA algorithm ( $F_3 = 0.47$ ), it is demonstrated that it is not a suitable algorithm since the way it manages power is proportional to traffic demand without taking into account interference between beams of the same color [13]. On the other hand, resource management using a classification algorithm with neural networks loses its effectiveness by increasing the possible situations of traffic demand [23], for that reason, it is the algorithm that presents the worst performance ( $F_3 = 0.48$ ).

The algorithm that obtains the lowest normalized mean error, i.e.,  $KPI_1$ , is the CNN algorithm (0.4096), whereas the algorithm that obtains the highest power saving is the AG-D (3.86).

AG-D focuses more on optimizing power savings [13] and, therefore, neglects the normalized error of capacity by setting its  $F_3$  to 0.38. Therefore, based on (26), the algorithm that has the best performance is the CNN algorithm that obtains  $F_3$  value of 0.34 since the proposed cost function (12) achieves a balance between minimizing  $KPI_1$  and maximizing  $KPI_2$ .

### C. Payload Architectures Performance

After having evaluated the impact of the training phase and the algorithm KPIs, the performance for different payload architectures, as presented in Table II, has been evaluated using the CNN algorithm and a 3-state time window size.

Fig. 10 shows the DRM performance for the different payload architectures in a specific time of the day (i.e., 10 A.M.) for the 82 beams in the service area. The blue bars show the required capacity in each beam at 10 A.M. (using the proposed traffic model). The offered capacity of each payload architecture is represented by the respective mark shown in Fig. 10. It is shown that Payloads 7 (flexibility in bandwidth and power) and 8 (full flexibility) are those achieving better performance, since the

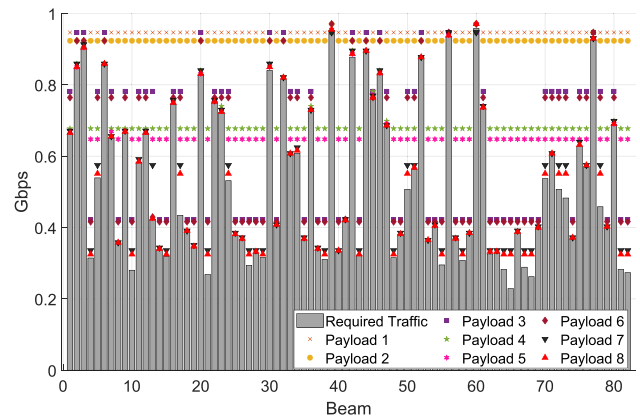


Fig. 10. DRM performance for different payload architectures at 10 A.M.

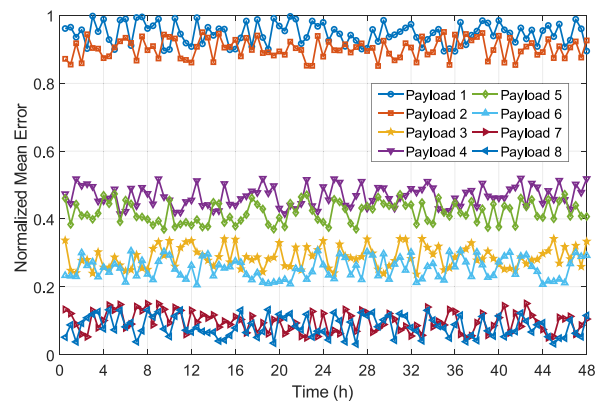


Fig. 11. Normalized mean error between the offered capacity and the required capacity of the system during 48 h.

capacity offered by these payloads have a behavior similar to the required capacity.

As expected, Payload 1 (traditional) has the worst performance since there is no flexibility in any of its parameters and the capacity offered is always the same.

The parameters that have the most notable effect on payload performance are power and bandwidth, where beamwidth has very reduced influence (Payload 2) on resource management performance (given the options in the case study). However, the beamwidth allows adjusting the cost function when accompanied by other flexibility parameters (Payload 5, Payload 6, and Payload 8).

The DRM performance of the eight payloads was evaluated for 48 h and is shown in Figs. 11 and 12. Fig. 11 shows the normalized mean error between the offered capacity and the required capacity of the system. CNN was used for all payloads and evaluated for the same distribution of traffic demand during the 48-h evaluation. For this reason, the mean error presented in Fig. 11 was normalized to  $\rho = \text{error}_{\max}$ , where  $\text{error}_{\max}$  is the maximum error obtained in all cases, and corresponding to 0.475 Gb/s.

Fig. 11 shows the better performance of Payloads 7 and 8 during the 48 h compared to the rest, reducing the mean error between the offered capacity and the required capacity by up to 89% compared to a traditional payload (Payload 1).

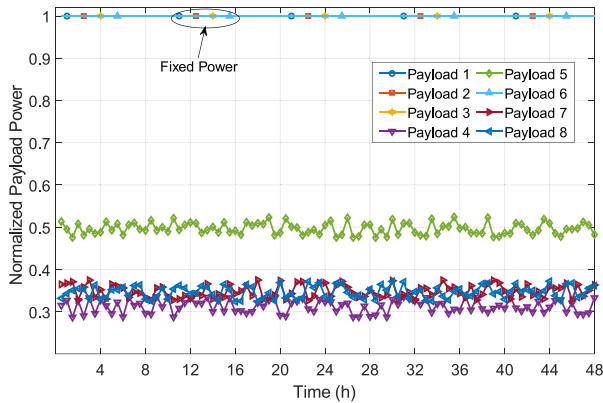


Fig. 12. Normalized payload power for 48 h.

According to the conditions of the case study, the bandwidth (Payload 3) is the parameter that has a greater influence on the error between the offered capacity and the required capacity, reducing the error up to 68% compared to a traditional architecture (Payload 1), followed by the power (Payload 4) that reduces the error up to 51%, and finally the beamwidth, reducing only 9%.

Fig. 12 shows the normalized power consumed by the payloads, representing the inverse of the power saving. It also shows the power requirement of the payload for 48 h, and it can be seen that the payloads that have a constant power value (i.e., Payloads 1, 2, 3, and 6) have the highest power requirement; this is due to the conditions of the case study (see Table II), in addition to having fixed power, the assigned value is the maximum possible in these payloads.

The power requirements of the payload are reduced by up to 50% (see Fig. 12) when the flexibility parameters are power and beamwidth (Payload 5), because the gain in each beam can also be adjusted to meet the required capacity, allowing power savings.

Power required for Payload 7 (flexibility in power and bandwidth) is similar to that required for Payload 8 (full flexibility). Fig. 12 also shows that both payloads can reduce the power by up to 65% given the conditions of the case study (see Table II).

In the case of Payload 4, the least power is required (see Fig. 11) because the only resource being managed is power, since in addition to the attempts to reduce the error between the offered capacity and the required capacity, the payload also tries to minimize the power used by the payload. However, if the priority is to reduce the error between the offered capacity and the required capacity, the Payloads 3, 5, 6, 7, and 8 have better performance (see Fig. 11).

## V. CONCLUSION

This article analyzes the use of a CNN to solve the DRM problem in SatComs by using a suitable cost function and a realistic traffic model. The suggested DRM cost function aims to minimize the error between the offered capacity and the required capacity while minimizing the amount of resources used in the satellite.

This contribution provides a tool to implement a dynamic and efficient resource management in future VHTS systems. The satellite industry increasingly expresses an interest in DL

applications for satellite systems [19], thus opening a great possibility for future works.

Compared to PPA, AG-DPA, and classification, the proposed algorithm achieves a better performance on the tradeoff to reduce the capacity error and power consumption.

One of the limitations of CNN in DRM is the dependence with the traffic model used during training. Thus, in a real system with changes in the traffic behavior noncompliant with the model, the CNN will have to be trained again.

A multiscale CNN has been shown to provide better predictions for other applications in engineering systems compared to the CNN [37]. Therefore, in future work, it is suggested to implement this algorithm for DRM and make a comparison with the CNN performance.

## APPENDIX

The CNN architecture can be divided into the input layer, the hidden convolution layers, the full connection hidden layers, and the output layer, as shown in Figs 3–6.

In the convolutional layers, the features are extracted from the kernels obtaining at the output  $Y_j$  as represented in (19). In this sense, the activation function  $g_1(\cdot)$  we have used in the convolutional layers is the ReLu function defined in [34].

MaxPooling was used to pool the convolutional layers. MaxPooling works to position a  $2 \times 2$  matrix on the feature map and choose the largest value in that matrix. The  $2 \times 2$  matrix moves from left to right across the feature map by choosing the largest value on each pass. These values form a new matrix called the pooled features map. MaxPooling works to preserve the main features while reducing the size of the image.

After the features map is obtained, the next step is to flatten it. The flatten layer involves the transformation of the entire feature map matrix into a single column that is fed into the neural network for processing.

In the neural networks area, perceptron refers to the artificial neuron or basic unit of inference, from which an algorithm is developed.

In the full connection layers, the features obtained in the convolutional layers are analyzed in order to allocate the resources that minimize the DRM cost function (12). The multilayer perceptron are obtained with (20) where the activation function used  $g_2(\cdot)$  in the full connection layers is defined by the hyperbolic tangent (tanh) function in [34].

The softmax function was used in the output layer of a CNN classifier. Softmax generates a vector that represents the probability distributions of a list of potential outcomes.

Softmax takes a vector  $z$  of  $L$  possible configurations as input and normalizes it into a probability distribution consisting of  $L$  probabilities proportional to the exponentials of the input numbers. That is, before applying softmax, some vector elements could be negative or greater than one depending on the cost function; and it could not add up to 1; but after applying softmax, each element will be in the range of 0–1, and the elements will add up to 1, so that they can be interpreted as probabilities.

Since the CNN is a supervised learning algorithm, the cost function of the DRM (12) is used to assign in a supervised way

the labels ( $Y$ ) corresponding to the training data generated with the traffic models (9) and (10).

The CNN cost function (21) aims to minimize the error that exists between the predicted values ( $\tilde{y}$ ) and the labels ( $Y$ ) in the output layer. In this article, the CNN output layer is a classification layer (see Fig. 5) where each class represents a possible configuration of satellite resources. In that sense, the  $Q(\cdot)$  function in (21) is defined by the logistic regression loss function [34]

$$Q = -\frac{1}{L} \sum_{l=1}^L [Y_{o,l} \log \tilde{y}_{o,l} + (1 - Y_{o,l}) \log (1 - \tilde{y}_{o,l})] \quad (27)$$

subject to

$$\tilde{y}_{o,l} = f_6(\Omega, x_o) \quad (28)$$

where  $L$  is the number of possible satellite resource configurations (classes),  $Y_{o,l}$  is a binary value (0 or 1) indicating if class label  $l$  is the correct classification for observation  $o$ , and  $\tilde{y}_{o,l}$  is the predicted probability observation  $o$  of class  $l$  and is a function of CNN configuration ( $\Omega$ ) and the input observation  $o$ .

The method implemented for learning all the supervised DL algorithms is known as "backpropagation." The error outputs are propagated backward from the output layer to all the neurons in the hidden layers that contribute directly to the output. This process is repeated, layer by layer, until all the neurons in the network have received an error signal describing their relative contribution to the total error. For more detail on how the backpropagation algorithm works, see El-Amir and Hamdy [33].

## REFERENCES

- [1] G. Giambene, S. Kota, and P. Pillai, "Satellite-5G integration: A network perspective," *IEEE Netw.*, vol. 32, no. 5, pp. 25–31, Sep./Oct. 2018.
- [2] A. Agarwal and P. Kumar, "Analysis of variable bit rate SOFDM based integrated satellite-terrestrial broadcast system in presence of CFO and phase noise," *IEEE Syst. J.*, vol. 13, no. 4, pp. 3827–3835, Dec. 2019.
- [3] F. Ortiz-Gomez, R. Martínez, M. Salas-Natera, A. Cornejo, and S. Landeros-Ayala, "Forward link optimization for the design of VHTS satellite networks," *Electronics*, vol. 9, no. 3, 2020, Art. no. 473.
- [4] F. Li, K.-Y. Lam, M. Jia, K. Zhao, X. Li, and L. Wang, "Spectrum optimization for satellite communication systems with heterogeneous user preferences," *IEEE Syst. J.*, vol. 14, no. 2, pp. 2187–2191, Jun. 2020.
- [5] European Commission, "Connectivity for a competitive digital single market-towards a European Gigabit Society," COM(2016)587 and Staff Working Document-SWD(2016)300, Sep. 2016, [Online] Available: <https://ec.europa.eu/digital-single-market/en/news/communication-connectivity-competitive-digital-single-market-towards-european-gigabit-society>
- [6] N. Pachler, J. J. G. Luis, M. Guerster, E. Crawley, and B. Cameron, "Allocating power and bandwidth in multibeam satellite systems using particle swarm optimization," in *Proc. IEEE Aerosp. Conf.*, Mar. 2020, pp. 1–11.
- [7] F. G. Ortiz-Gomez, R. M. Rodriguez-Osorio, M. A. Salas-Natera, and S. Landeros-Ayala, "Adaptive resources allocation for flexible payload enabling VHTS systems: Methodology and architecture," in *Proc. 36th AIAA Int. Commun. Satell. Syst. Conf.*, Oct. 2018, pp. 1–8.
- [8] Y. Abe, M. Ogura, H. Tsuji, A. Miura, and S. Adachi, "Resource and network management framework for a large-scale satellite communications system," *IEICE Trans. Fundam. Electron., Commun., Comput. Sci.*, vol. E103-A, no. 2, pp. 492–501, Feb. 2020.
- [9] M. Guerster, J. Grotz, P. Belobaba, E. Crawley, and B. Cameron, "Revenue management for communication satellite operators—Opportunities and challenges," in *Proc. IEEE Aerosp. Conf.*, Mar. 2020, pp. 1–15.
- [10] G. Cocco, T. de Cola, M. Angelone, Z. Katona, and S. Erl, "Radio resource management optimization of flexible satellite payload for DVB-S2 systems," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 266–280, Jun. 2018.
- [11] A. Paris, I. Del Portillo, B. Cameron, and E. Crawley, "A genetic algorithm for joint power and bandwidth allocation in multibeam satellite systems," in *Proc. IEEE Aerosp. Conf.*, Mar. 2019, pp. 1–15.
- [12] Y. Kawamoto, T. Kamei, M. Takahashi, N. Kato, A. Miura, and M. Toyoshima, "Flexible resource allocation with inter-beam interference in satellite communication systems with a digital channelizer," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 2934–2945, May 2020.
- [13] 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," *Int. J. Satell. Commun. Netw.*, vol. 38, no. 1, pp. 74–83, Jan./Feb. 2020.
- [14] P. V. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Bilen, and J. Mortensen, "Multi-objective reinforcement learning for cognitive satellite communications using deep neural network ensembles," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 5, pp. 1030–1041, May 2018.
- [15] P. V. Ferreira *et al.*, "Multi-objective reinforcement learning-based deep neural networks for cognitive space communications," in *Proc. Cogn. Commun. Aerosp. Appl. Workshop*, Jun. 2017, pp. 1–8.
- [16] S. Liu, X. Hu, and W. Wang, "Deep reinforcement learning based dynamic channel allocation algorithm in multibeam satellite systems," *IEEE Access*, vol. 6, pp. 15733–15742, 2018.
- [17] D. Bayrd, "NASA explores artificial intelligence for space communications," Dec. 2017, [Online]. Available: <https://www.nasa.gov/feature/goddard/2017/nasa-explores-artificial-intelligence-for-space-communications>
- [18] W. Xiang *et al.*, "Verification for machine learning, autonomy, and neural networks survey," Oct. 2018, *arXiv: 1810.01989*.
- [19] AIKO S.R.L., "AIKO: Our deep learning is in orbit," Jan. 2020, [Online]. Available: <https://www.aikospace.com/>
- [20] H. Kim, D. Seo, J. Jung, D. Cha, and D. Lee, "Blind motion deblurring for satellite image using convolutional neural network," in *Proc. Digit. Image Comput.: Techn. Appl.*, 2019, pp. 1–8.
- [21] M. Zhao, Y. Wang, S. Du, and T. Yin, "The study of rapid localization algorithm from beam scanning by single satellite based on neural network," in *Proc. 1st IEEE Int. Conf. Comput. Commun. Internet*, 2016, pp. 186–190.
- [22] P. V. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Bilen, and J. Mortensen, "Reinforcement learning for satellite communications: From LEO to deep space operations," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 70–75, May 2019.
- [23] F. G. Ortiz-Gomez, R. Martínez, M. A. Salas-Natera, S. Landeros-Ayala, D. Tarchi, and A. Vanelli-Coralli, "On the use of neural networks for flexible payload management in VHTS systems," in *Proc. 25th Ka Broadband Commun. Conf.*, Oct. 2019, pp. 1–10.
- [24] F. Ortiz-Gomez, R. Martínez, M. Salas-Natera, and S. Landeros-Ayala, "On the use of machine learning for flexible payload management in VHTS systems," in *Proc. 70th Int. Astronautical Congr.*, Oct. 2019, pp. 1–6.
- [25] J. J. G. Luis, M. Guerster, I. De Portillo, E. Crawley, and B. Cameron, "Deep reinforcement learning for continuous power allocation in flexible high throughput satellites," in *Proc. IEEE Cogn. Commun. Aerosp. Appl. Workshop*, Jun. 2019, pp. 1–4.
- [26] "Generic flexible payload study," ESA ARTES 3/4 Final Report, Nov 2015, [Online]. Available: <https://artes.esa.int/projects/generic-flexible-payload-solution>
- [27] N. Komiya, A. Miura, T. Orikasa, and Y. Fujino, "Development of resource allocation re-construction technology (digital beam former and digital channelizer)," *J. Nat. Inst. Inf. Commun. Technol.*, vol. 62, no. 1, pp. 151–160, 2015.
- [28] Digital Video Broadcasting (DVB), "Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 2: DVB-S2 Extensions (DVB-S2X)," Geneva, Switzerland, ETSI EN 302 307-2, rev. 1.1.1, Oct. 2014.
- [29] A. J. Roumeliotis, C. Kourgiorgas, and A. Panagopoulos, "Optimal capacity allocation strategies in smart gateway satellite systems," *IEEE Commun. Lett.*, vol. 23, no. 1, pp. 56–59, Jan. 2019.
- [30] A. J. Roumeliotis, C. Kourgiorgas, and A. Panagopoulos, "Dynamic capacity allocation in smart gateway high throughput satellite systems using matching theory," *IEEE Syst. J.*, vol. 13, no. 2, pp. 2001–2009, Jun. 2019.
- [31] *Propagation Data and Prediction Methods Required for the Design of Earth-Space Telecommunication Systems*, ITU-R Rec. P.618-13, International Telecommunications Union, Geneva, Switzerland, Dec. 2017.

- [32] J. J. G. Luis, N. Pachler, M. Guerster, I. del Portillo, E. Crawley, and B. Cameron, "Artificial intelligence algorithms for power allocation in high throughput satellites: A comparison," in *Proc. IEEE Aerosp. Conf.*, Mar. 2020, pp. 1–15.
- [33] H. El-Amir and M. Hamdy, "Convolutional neural network," in *Deep Learning Pipeline*. Berkeley, CA, USA: Apress, 2020.
- [34] W. H. Lopez-Pinaya, S. Vieira, R. Garcia-Dias, and A. Mechelli, "Chapter 10—Convolutional neural networks," in *Machine Learning*. Cambridge, MA, USA: Academic, 2020.
- [35] U. Michelucci, "Fundamentals of convolutional neural networks," in *Advanced Applied Deep Learning*. Berkeley, CA, USA: Apress, 2019.
- [36] "Eutelsat 9B KA-SAT," Feb. 2020, [Online]. Available: <https://www.eutelsat.com/en/satellites/eutelsat-9-east.html>
- [37] M. Kordestani, M. Saif, M. E. Orchard, R. Razavi-Far, and K. Khorasani, "Failure prognosis and applications—A survey of recent literature," *IEEE Trans. Rel.*, vol. 70, no. 2, pp. 728–748, Jun. 2021.



**Flor G. Ortiz-Gomez** received the B.S. degree in telecommunications engineering and the M.S. degree in electrical-telecommunications engineering from the Universidad Nacional Autónoma de México (UNAM), Mexico City, Mexico, in 2015 and 2016, respectively. She is currently working toward the Ph.D. degree with the Universidad Politécnica de Madrid, Madrid, Spain.

She is currently with the Radiation Group, Universidad Politécnica de Madrid.



**Daniele Tarchi** (Senior Member, IEEE) received the Ph.D. degree in informatics and telecommunications engineering from the University of Florence, Florence, Italy, in 2004.

Since 2019, he has been an Associate Professor with the University of Bologna, Bologna, Italy. He is an author of more than 100 papers, among which more than 30 on international journals. His main research interests include wireless communications, resource allocation techniques, edge computing and fog computing scenarios.

Prof. Tarchi is an Editorial Board Member for *IEEE WIRELESS COMMUNICATIONS LETTERS*, *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, and *IET Communications*. He has been a Symposium Co-Chair at the IEEE Wireless Communications & Networking Conference in 2011, IEEE Global Communications Conference in 2014 and 2018, and IEEE International Conference on Communications in 2020, and a Workshop Co-Chair at the IEEE International Conference on Communications in 2015.



**Ramón Martínez** received the Ph.D. degree in electrical engineering from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 2004.

In 2002, he joined the Telecommunication Engineering School, UPM (ETSIT-UPM), where he has been a Full Professor since 2019 and a member of the Information Processing and Telecommunications Center. He has also been a Lecturer in SatCom systems since 2004. In the past years, he has worked in the optimization of high throughput satellite systems, with the use and operation of flexible payloads

using machine learning techniques. He has led and participated in different international projects related to space technology, with collaborations with ESA and space communication industry in technology transfer programs. He led the design of antenna systems for intersatellite links in nanosatellite missions, and is currently the Principal Investigator of FUTURE-RADIO, a project to evaluate radio technologies for future communication systems including 5G. His research interests include wireless and satellite communications, including the application of antenna array processing systems.

Prof. Martínez was a recipient of awards for his Ph.D. dissertation focused on the application of smart antennas to cellular communications and the UPM Innovative Teaching Award.



**Alessandro Vanelli-Coralli** (Senior Member, IEEE) received the Dr. Ing. degree in electronics engineering and the Ph.D. degree in electronics and computer science from the University of Bologna, Bologna, Italy, in 1991 and 1996, respectively.

In 1996, he joined the University of Bologna, where he is currently an Associate Professor with the Department of Electrical, Electronic, and Information Engineering (Guglielmo Marconi). From 2013 to 2018, he chaired the Ph.D. Board, Electronics, Telecommunications and Information Technologies.

From 2003 to 2005, he was a Visiting Scientist with the Qualcomm, Inc., San Diego, CA, USA. He participates in national and international research projects on wireless and satellite communication systems. He has been a Project Coordinator and scientifically responsible for several European Space Agency and European Commission funded projects. His research interests include wireless communications, digital transmission techniques, and digital signal processing.

Dr. Vanelli-Coralli was co-recipient of several best paper awards. He was a General Chair and the Technical Chair for several scientific conferences. He is an appointed member of the Editorial Board of the *International Journal of Satellite Communications and Networking* (Wiley InterScience) and a Guest Co-Editor for several special issues of the international scientific journals.



**Miguel A. Salas-Natera** received the master's degree in space technologies in 2011 and the Ph.D. degree in technologies and communications systems from Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 2011.

In 2012, he was a Technical Director with Antenna System Solutions Company (spin-off of the UPM) for two years. He is currently an Assistant Professor and a Researcher with the Radiation Group of UPM. He is also an Electrical Engineer major in telecommunications. In addition, he is responsible for

the special session on satellite communications systems (SATCOM) since its beginnings in URSI2016 in Spain. He has led and participated in a number of international and local projects related to the development of new antenna technologies, antenna test and validation, and analysis of SATCOM Systems. He has also worked in the areas of uncertainty analyses and calibration methods for active antenna arrays. He is currently involved in the development and design of ultracompact reflector antennas including novel surface design and manufacturing technologies. He is actively participating in the development of algorithm for resource allocation of flexible-transparent transponder systems for classical GEO satellites systems as well as for novel massive low Earth orbit (LEO) satellites and high altitude platform (HAP) systems.



**Salvador Landeros-Ayala** received the bachelor's degree in mechanical and electrical engineering from the Universidad Nacional Autónoma de México (UNAM), Mexico City, Mexico, in 1977, the M.S. degree in electrical engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 1980, and the Ph.D. degree in electrical engineering from UNAM in 1999.

Since 1977, he has been with the National Autonomous University of Mexico, where he is in several academic positions, leading research, and industry

satellite communications projects.

Dr. Ayala has been a member of the Mexican Academy of Engineering and the Head of the Mexican Space Agency since 2019.