Received 27 July 2023; revised 20 November 2023 and 2 January 2024; accepted 2 January 2024.
Date of publication 10 January 2024; date of current version 19 January 2024.

The associate editor coordinating the review of this article and approving it for publication was M. Mozaffari.

Digital Object Identifier 10.1109/TMLCN.2024.3352569

Energy-Efficient On-Board Radio Resource Management for Satellite Communications via Neuromorphic Computing

FLOR ORTIZ[1] (Member, IEEE), NICOLAS SKATCHKOVSKY[2] (Member, IEEE), EVA LAGUNAS[1] (Senior Member, IEEE), WALLACE A. MARTINS[1,3] (Senior Member, IEEE), GEOFFREY EAPPEN[1] (Member, IEEE), SAED DAOUD[1] (Member, IEEE), OSVALDO SIMEONE[4] (Fellow, IEEE), BIPIN RAJENDRAN[4] (Senior Member, IEEE), AND SYMEON CHATZINOTAS[1] (Fellow, IEEE)

[1] Interdisciplinary Centre for Security, Reliability, and Trust (SnT), 1855 Luxembourg City, Luxembourg
[2] Francis Crick Institute, NW1 1AT London, U.K.
[3] ISAE-SUPAERO, Université de Toulouse, 31055 Toulouse, France
[4] Department of Engineering, King's College London, WC2R 2LS London, U.K.

(Flor Ortiz and Nicolas Skatchkovsky contributed equally to this work.)

CORRESPONDING AUTHOR: F. ORTIZ (flor.ortiz@uni.lu)

This work was supported in part by the European Space Agency (ESA)-The Application of Neuromorphic Processors to Satcom Applications under Grant 4000137378/22/UK/ND; and in part by the Luxembourg National Research Fund (FNR) through the project SmartSpace under Grant C21/IS/16193290. The work of Osvaldo Simeone was supported in part by the European Union's Horizon Europe Project CENTRIC under Grant 101096379; in part by the Open Fellowship of the EPSRC under Grant EP/W024101/1; and in part by the Project REASON, a U.K. Government funded project through the Future Open Networks Research Challenge (FONRC) sponsored by the Department of Science Innovation and Technology (DSIT). The work of Bipin Rajendran was supported in part by the European Union's Horizon Europe Project CENTRIC under Grant 101096379 and in part by the Open Fellowship of the EPSRC under Grant EP/X011356/1.

ABSTRACT The latest Satellite Communication (SatCom) missions are characterized by a fully reconfigurable on-board software-defined payload, capable of adapting radio resources to the temporal and spatial variations of the system traffic. As pure optimization-based solutions have shown to be computationally tedious and to lack flexibility, Machine Learning (ML)-based methods have emerged as promising alternatives. We investigate the application of energy-efficient brain-inspired ML models for on-board radio resource management. Apart from software simulation, we report extensive experimental results leveraging the recently released Intel Loihi 2 chip. To benchmark the performance of the proposed model, we implement conventional Convolutional Neural Networks (CNN) on a Xilinx Versal VCK5000, and provide a detailed comparison of accuracy, precision, recall, and energy efficiency for different traffic demands. Most notably, for relevant workloads, Spiking Neural Networks (SNNs) implemented on Loihi 2 yield higher accuracy, while reducing power consumption by more than 100× as compared to the CNN-based reference platform. Our findings point to the significant potential of neuromorphic computing and SNNs in supporting on-board SatCom operations, paving the way for enhanced efficiency and sustainability in future SatCom systems.

INDEX TERMS Energy-efficient, neuromorphic computing, radio resource management, satellite communication, spiking neural networks.

I. INTRODUCTION

A. CONTEXT AND MOTIVATION

SATELLITE Communications (SatCom) have become increasingly important in recent years due to the surge in global connectivity demands. With the integration of terrestrial systems like 6G (Sixth-Generation) and the pressing need to reduce the digital divide and obtain ubiquitous coverage, SatCom plays a vital role in bridging the communication gap worldwide [1]. However, the growing traffic demand in SatCom systems presents significant challenges in effectively managing the allocation of radio resources to meet Quality-of-Service (QoS) requirements, while minimizing resource utilization [2].

Conventional SatCom systems typically employ static multi-beam configurations with fixed bandwidth and power allocations. These systems are incapable of adapting to the dynamic nature of traffic demands. As a result, resources may be wasted, while user demands may be left unfulfilled. Recognizing the temporal and spatial variations in demand, software-defined payloads have emerged, offering unprecedented flexibility and adaptability in Radio Resource Management (RRM) for SatCom [3].

Software-defined payloads have revolutionized the SatCom landscape by providing fully reconfigurable systems capable of dynamically allocating power and bandwidth resources. The advent of these payloads has driven the need for effective RRM techniques to optimize resource allocation and ensure efficient utilization. While traditional optimization-based solutions have been explored for RRM, they are often computationally cumbersome and lack the required flexibility to address the diverse and dynamic traffic patterns encountered in SatCom systems [4].

In recent years, Machine Learning (ML) algorithms have emerged as a promising alternative to conventional optimization approaches for RRM in SatCom [5], [6]. ML-based solutions not only offer the potential to adaptively learn and predict traffic patterns, but also contribute to accelerate complex RRM algorithmics and bring adaptation and flexibility to static optimization solutions. However, implementing ML algorithms on board may be problematic due to the potentially high energy budgets incompatible with satellites' available resources.

B. CONTRIBUTIONS

To address this problem, this paper investigates neuromorphic computing as an alternative to conventional neural network-based computing platforms to enhance the efficiency and sustainability of on-board SatCom operations [7] (see Fig. 1). Neuromorphic Processors (NPs) represent a new class of computing devices inspired by the human brain architecture and computational principles [8]. They offer unique advantages in low terms of power consumption, high parallelism, and real-time processing capabilities [9], [10]. Leveraging these benefits, NPs hold significant promise for enabling energy-efficient RRM on-board satellites.

To provide empirical evidence and benchmark the performance of NPs, we conduct experiments using Intel's cutting-edge neuromorphic processor, Loihi 2. Benchmarking is carried out by comparing the accuracy, precision, recall, and energy efficiency of NPs with conventional platforms such as Xilinx Versal VCK5000 under varying traffic demands for the problem of RRM. Particularly noteworthy are the results obtained using Spiking Neural Networks (SNNs) implemented on the Loihi processor, demonstrating superior accuracy while reducing power consumption by more than a factor of 100 compared to the reference platform based on Convolutional Neural Networks (CNNs).

The principal contributions of this paper are summarized as follows:

- We investigate the use of neuromorphic computing in the context of on-board satellite operations. Our investigation is among the first to systematically assess the impact of employing NPs in SatCom for efficient RRM, addressing the key limitations of conventional neural network-based platforms in terms of power and computational resource constraints.
- Through rigorous benchmarking, we demonstrate that SNNs implemented on Intel's advanced neuromorphic processor, Loihi 2, deliver a remarkable reduction in power consumptionexceeding a factor of 100 while maintaining superior accuracy, precision, and recall metrics when compared to the conventional CNNs implemented on Xilinx Versal VCK5000 platform.
- We provide empirical data and analysis, which were previously scarce, showcasing the tangible benefits of neuromorphic computing in a real-world satellite communications setting. This includes detailed comparisons under varying traffic demands, offering insights into the scalability and adaptability of NPs for RRM problems.
- The results of our experiments serve as a proof-of-concept benchmark for the deployment of neuromorphic computing in space applications, potentially revolutionizing RRM in SatCom by significantly enhancing the efficiency and sustainability of on-board operations.
- Lastly, our study aids in bridging the gap between the burgeoning field of neuromorphic computing and its practical application in the space sector.

C. RELATED WORK

1) MACHINE LEARNING FOR RRM IN SatCom

The efficient implementation of on-board RRM is crucial for optimizing performance and ensuring seamless connectivity in SatCom systems. In recent years, ML techniques have gained significant attention for resource management in various SatCom scenarios [11], [12]. A study by [13] focused on ML-based resource management in multi-beam Geostationary Earth Orbit (GEO) satellite systems. The authors analyze different ML techniques applied to systems with power, bandwidth, and/or beamwidth flexibility and systems with beam hopping capabilities. Furthermore, reference [14] proposes a combined learning and optimization approach to address a Mixed-Integer Convex Programming (MICP) problem in satellite RRM. The problem is decomposed into classification-like tasks and power control optimization, respectively solved by dual Deep Neural Networks (DNNs) and convex optimization.

Another notable work by Deng et al. [15] focuses on resource management in next-generation Heterogeneous Satellite Networks (HSNs), and introduces an innovative framework that encourages cooperation among independent satellite systems to maximize resource utilization. Deep Reinforcement Learning (DRL) optimizes resource allocation and supports intercommunication between different satellite systems. In a related study, Ferreira et al. [16] proposed a feasible solution for real-time, single-channel resource allocation problems using DRL. Their study discretized resources before allocation, which may not be optimal for continuous resources like satellite power. Luis et al. [17] addressed this issue by exploring a DRL architecture with constant, stateful action spaces for energy allocation, avoiding the need for discretization. Liu et al. [18], [19] presented a DRL-based Dynamic Channel Allocation (DRL-DCA) algorithm for multi-beam satellite systems, achieving lower blocking probabilities than traditional algorithms. Finally, Liao et al. [6] also introduced a game model and a DRL-based bandwidth allocation framework for satellite communication scenarios, dynamically allocating bandwidth in each beam. The proposed method effectively handles time-varying traffic and large-scale communication, albeit limited to managing a single resource on the satellite.

2) NEUROMORPHIC COMPUTING FOR COMMUNICATIONS

The application of neuromorphic processors to communications has gained significant interest in recent years as a low-power alternative to traditional systems. The work [20] explores the application of SNNs for learning and inference in battery-powered devices connected over bandwidth-constrained channels. It summarizes activity on federated learning for distributed training of SNNs [21] and the integration of neuromorphic sensing, SNNs, and pulse radio technologies for low-power remote inference [22]. The recent references [23], [24] generalized wireless neuromorphic communications to multi-access channels and introduced the concept of Neuromorphic Integrated Sensing And Communications (N-ISAC). N-ISAC utilizes a common impulse radio waveform for transmitting digital information and detecting radar targets, employing an SNN for decoding and target detection. The optimization of SNN operation balances data communications and radar sensing performance metrics, showcasing the synergies and trade-offs between these applications.

The recently presented work in [25] addresses the need for innovative spectrum monitoring techniques due to the growing Internet of Things (IoT) and spectrum congestion. It presents a novel approach using SNNs centered on image segmentation. This method outperforms traditional energy detection methods in detection efficiency and performance, and future work will aim to improve it with ML for better feature extraction and segmentation.

3) NEUROMORPHIC COMPUTING FOR SatCom

While the field of neuromorphic learning for communications is still in its infancy, there is a growing interest in its applications within the space sector due to its high energy efficiency and promising performance [26]. Both academia and industry have made great strides in developing non-space applications for neuromorphic computing, which is a milestone that the space industry can leverage as a "spin-off" technology.

The work presented in [27] discusses neuromorphic systems, which mimic biological neural networks and are ideal for low-power computing. These systems are particularly well suited for space applications because of their low power requirements and resilience to space challenges, such as resource scarcity and radiation. The article highlights the potential of neuromorphic devices to improve the efficiency and effectiveness of space missions, with previous studies demonstrating their capability in tasks such as optical flow sensing for spacecraft landing.

Recent research initiatives further highlight the increasing interest in exploring the potential benefits of neuromorphic computing. U.S National Aeronautics and Space Administration (NASA) has launched of TechEdSat-13, equipped with Intel Loihi neuromorphic processor, into Low Earth Orbit (LEO) in 2022 is a testament to their commitment to testing new capabilities for future Artificial Intelligence (AI) science and engineering applications in space [28]. Additionally, the European Space Agency (ESA) initiated an Advanced Research in Telecommunications Systems (ARTES) Future Preparation (FP) activity in 2021 [29], focusing on exploring the use of neuromorphic computing for SatCom systems, underscoring the significance of this technology within the space sector.

Our previous work, published in 2022, highlights the potential use cases and applications of neuromorphic processors for SatCom [7]. Most recently, the publication [30] highlights the energy limitations in satellite networks due to the dependence on solar energy as the only power source, as well as the lack of efficient and high-performance processors to run AI algorithms, which restrict the use of AI techniques in orbit. The authors note the need for new AI algorithms such as SNNs and specialized hardware to achieve resilient LEO satellite constellations for IoT. More specifically, the authors have discussed what is involved in highly dynamic channel prediction, spectrum sensing and classification, signal detection and demodulation, satellite and inter-satellite access network optimization, and network security, while outlining future paradigms for these mechanisms in practical networks. In this sense, Neuromorphic processors, such as Loihi [31], are, in fact, particularly well-suited for processing sparse time series data since their energy requirements are proportional to the number of emitted spikes, which occur only in the presence of relevant events.

D. ORGANIZATION

The document is structured as follows. Section II provides a detailed overview of the system model, covering the problem definition and the traffic demand model. Section III discusses our benchmark approach based on CNNs. In Section IV, we describe the approach based on neuromorphic computing. Section V discusses the implementation of both models on the respective hardware. Section VI presents experimental results, showing the superiority of the neuromorphic computing approach for practical workloads. Finally, Section VII

TABLE 1. **On-board RRM for flexible payload scenario.**

Scenario	On-board RRM for flexible payload
System architecture	SDR payload
Air interface	Any air interface supporting multicarrier
AI-based technique(s)	Supervised learning: Classification
Input / Output	Input: Geographical traffic demand distribution Output: Power and bandwidth configuration

offers a detailed analysis of the results and draws some conclusions.

II. SYSTEM MODEL

As illustrated in Fig. 1, the system architecture comprises a flexible, Software-Defined Radio (SDR) satellite payload that allows for adjustable bandwidth and power allocation to each beam. The management of communication resources is dynamically adapted in response to changes in traffic demand. Specifically, adopting a data-driven ML-based solution, we focus on designing ML models that take the traffic demand over the service area as input, and output an optimized resource configuration. After offline training, the ML payload controller can be deployed on board the satellite for real-time inference. This data-driven architecture offers the advantage of reduced processing times [32]. Table 1 provides an overview of the main characteristics of the setting under study.

In more detail, we consider a GEO high-throughput satellite system consisting of a single multibeam satellite that provides coverage to a wide region of Earth through B spot-beams. We focus on the forward link, consisting of uplink feeder and downlink user links, and we assume a total of K single-antenna User Terminals (UTs) distributed across the satellite coverage area. The considered payload can flexibly manage power using a Traveling-Wave Tube Amplifier (TWTA) with adaptive Input Back-Off (IBO). Furthermore, it manages spectrum utilization via a channelizer on board that separates signals into frequency blocks and rearranges them to achieve flexible bandwidth allocations and to avoid co-channel beam interference, frequency reuse is assumed [13].

A. PROBLEM STATEMENT

The heterogeneous distribution of traffic demands across the satellite beams and over the satellite's lifetime motivates the use of dynamic RRM [3]. The objective of RRM is to efficiently allocate available bandwidth and power resources to minimize the discrepancy between the offered capacity C_τ^b and the requested capacity R_τ^b on each b-th beam during any time slot τ.

The offered capacity C_τ^b [bps] in the b-th beam during time slot τ can be calculated as

$$C_\tau^b = \mathrm{W}_\tau^b \cdot \kappa_\tau^b, \tag{1}$$

where κ_τ^b [bps/Hz] is the spectral efficiency of the selected Modulation and Coding (ModCod) scheme, and W_τ^b [Hz] denotes the bandwidth allocated to b-th beam.

The spectral efficiency κ_τ^b depends on the Carrier-to-Interference-plus-Noise Ratio (CINR) γ_τ^b of the b-th beam at slot τ. Accordingly, we write it as a generic function $\kappa_b = f(\gamma_b)$, where $f(\cdot)$ denotes the mapping between CINR and the selected ModCod scheme, which can be found in standards such as Digital Video Broadcasting Satellite Second Generation (DVB-S2) [33].

The CINR γ_τ^b in turn depends on the power P_τ^b [W] and bandwidth W_τ^b [Hz] allocated to the b-th beam. In particular, we have

$$\gamma_{N,\tau}^b = \frac{P_\tau^b |h^b|^2}{I_\tau^b + N_0 \mathrm{W}_\tau^b}, \tag{2}$$

where N_0 is the power spectral density of noise, and $|h^b|^2$ represents the channel gain for beam b. For the latter, we use the standard model

$$|h_\tau^b|^2 = \frac{G_{\mathrm{SAT}}\left(\theta_\tau^b\right) G_{\mathrm{RX,max}}}{\left(4\pi D_\tau^b/\lambda\right)^2 L_\tau^b}, \tag{3}$$

where D_τ^b is the distance between the satellite and b-th beam center on the ground; λ indicates the wavelength; L_τ^b denotes the shadowing and atmospheric gas losses; and the satellite off-boresight transmit angle to the different beams is θ_τ^b. The variables $G_{\mathrm{SAT}}(\theta)$ and $G_{\mathrm{RX,max}}$ denote the satellite antenna gain towards a specific off-boresight angle θ and the user terminal receive antenna gain, respectively. The user terminal antenna is assumed to point towards the GEO satellite, and therefore, the received antenna gain is fixed and equal to the maximum supported by the receiver's antenna.

Inspired by [34], our goal is to use minimal power and spectral bandwidth to match the offered capacity given in (1) to the aggregated traffic demand R_τ^b [bps] of each beam b at each particular time instant τ. In particular, focusing on a specific time slot τ, the objective function can be formulated as

$$U(P_\tau, W_\tau) = \beta_0 \sum_{b=1}^{B} |C_\tau^b - R_\tau^b| - \beta_1 \sum_{b=1}^{B} P_\tau^b - \beta_2 \sum_{b=1}^{B} \mathrm{W}_\tau^b, \tag{4}$$

where P_τ and W_τ denote the set of power and bandwidth allocation variables $\{P_\tau^b\}$ and $\{\mathrm{W}_\tau^b\}$, respectively. The first term in (4) denotes the mismatch between the offered capacity C_τ^b and the requested capacity R_τ^b. The second term facilitates the minimization of the total transmit power of the satellite system, i.e. $\sum_{b=1}^{B} P_\tau^b$. Finally, the third term prioritizes solutions in which the total bandwidth $\sum_{b=1}^{B} \mathrm{W}_\tau^b$ allocated to the beams is minimized.

The hyperparameters $\beta_0 \geq 0$, $\beta_1 \geq 0$ and $\beta_2 \geq 0$ determine the relative weights of three terms in (4). The relative importance of the terms should reflect the priorities of the resource management system. For example, if the main concern is to minimize the discrepancy between offered and

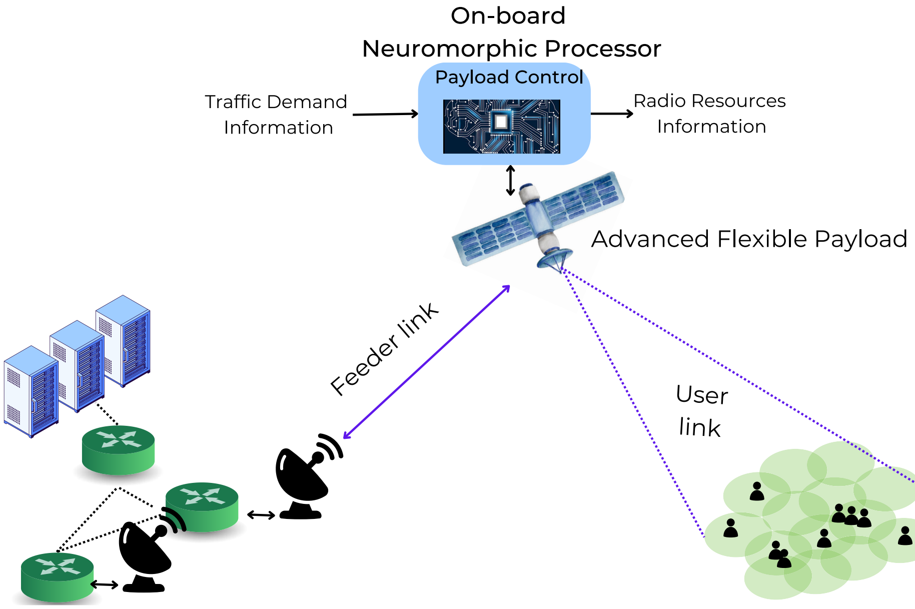


FIGURE 1. On-board resource management for flexible payload in a multibeam satellite.

requested capacity, hyperparameter β_0 should be larger than β_1 and β_3. Alternatively, if it is crucial to reduce power and bandwidth, the respective weights should be increased. These choices depend largely on the specific operational objectives of the satellite communication system.

The objective function (4) is minimized under total bandwidth constraints and total system power constraints as

$$\begin{aligned} &\underset{P_\tau, W_\tau}{\text{minimize}}\ U(P_\tau, W_\tau) \\ &\text{s.t.}\ \ \sum\nolimits_{b=1}^{B} P_\tau^b \leq P_{\max} \\ &\qquad \sum\nolimits_{b=1}^{B} \mathrm{W}_\tau^b \leq \mathrm{W}_{\max}. \end{aligned} \tag{5}$$

The objective (4) includes terms supporting power and bandwidth minimization so as to potentially reduce the power and bandwidth consumption beyond the upper bounds imposed by the constraints in (5).

We assume that power and bandwidth variables P_τ^b and W_τ^b are to be selected within discrete sets $\mathcal{P}_\tau = \{P_1, P_2, \ldots, P_N\}$ and $\mathcal{W}_\tau = \{W_1, W_2, \ldots, W_M\}$ of feasible solutions, imposing the additional constraints $P_\tau^b \in \mathcal{P}_\tau$ and $\mathrm{W}_\tau^b \in \mathcal{W}_\tau$. As a result, problem (5) is of combinatorial nature, as it involves the selection of variables from a finite set of options, and its complexity scales exponentially with the number of beams. Since the aggregated beam demands R_τ^b generally change over time, problem (5) needs to be solved anytime that there is a relevant change in the traffic profile, resulting in an update on the power and bandwidth assignment.

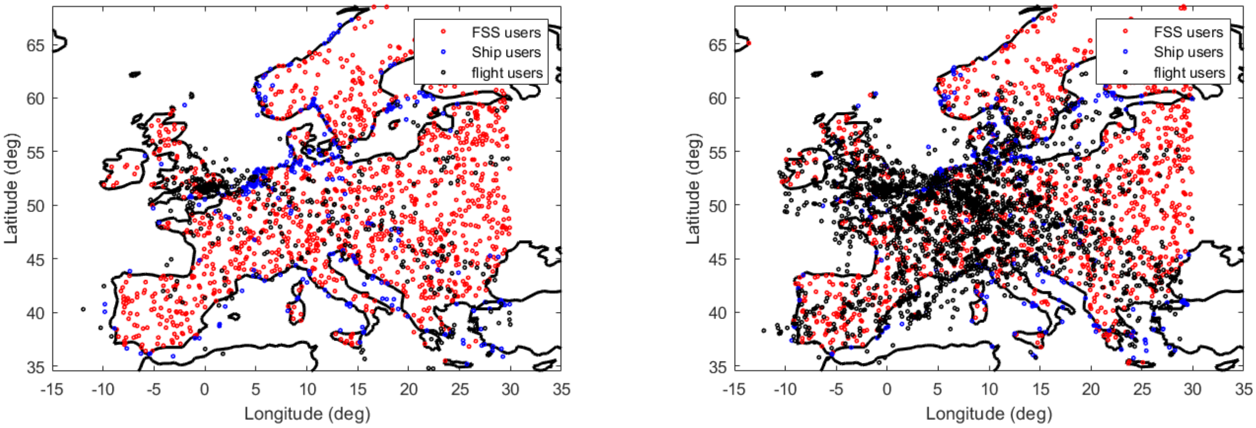
B. TRAFFIC MODEL

In order to model the requested capacity R_τ^b, we make use of the traffic simulator developed by University of Luxembourg and presented in [35], which integrates population, aeronautical, and maritime data to accurately model traffic demand and distribution within a specific service area. As depicted in Fig. 2, first, the population dataset considers the distribution of broadband Fixed Satellite Service (FSS) terminals, encompassing the fundamental spatial patterns of FSS traffic. The population data has been obtained from the NASA Socioeconomic Data and Applications and Data Center (SEDAC) population density database [36]. Secondly, the simulator incorporates real variations in aeronautical traffic by utilizing data extracted from an anonymous and unfiltered flight tracking source [35], [37]. This inclusion enables the examination of the impact of flight volume on the geographical density of traffic at different time intervals, thereby ensuring an accurate representation of the spatio-temporal distribution of aeronautical traffic.

Lastly, the maritime dataset accounts for the potential demand for satellite connectivity through ship communications, which exhibit significant changes over time and location. To capture this, the simulator employs a dataset obtained from Vessel Traffic Services (VTS), comprising vessel positions and maritime traffic detected by the global Automatic Identification System (AIS) [35], [38].

To prepare the collected datasets for analysis, a preprocessing unit within the traffic simulator handles tasks such as eliminating redundant and conflicting traffic records, resolving missing information, and extracting user positions. UTs are categorized and assigned to their respective service beams based on the geographic longitudes and latitudes. The simulator also considers the limited use of FSS in large urban areas, recognizing the prevalence of alternative broadband technologies in such regions.

For modeling daily hourly traffic demands, the aeronautical data traces are pre-processed by collecting and gathering



(a) Traffic demand at 2 am

(b) Traffic demand at 10 am

FIGURE 2. **The traffic demand model in Europe depends on population density, aerial, and maritime density.**

flight data for one hour. Similarly, in capturing the temporal maritime traffic demand, the maritime data is analyzed, taking into account the position of each ship's first appearance within the covered area during each hour. This approach allows for a reasonable estimation of the current and anticipated demand within an hour [35].

C. TRAINING DATA

The per-beam demand requests one formatted in a matrix form $\boldsymbol{R}_\tau$, which represents the traffic demand at each geographic location in the service area as

$$\boldsymbol{R}_\tau = \begin{bmatrix} r_{1,1} & \cdots & r_{1,n} \\ \vdots & \ddots & \vdots \\ r_{m,1} & \cdots & r_{m,n} \end{bmatrix}, \tag{6}$$

where m and n denote a grid of latitude and longitude points, respectively, within the service area. In (6), the $r_{i,j}$ denotes the required traffic in Mbps at a location indexed by i and j.

To generate training data, we collect labels for several realizations of the traffic matrix in (6) by running an exhaustive search method on the problem (5). Although data generation is a data-intensive task, it is performed offline. Each label represents the optimal choices $\mathcal{P}_\tau$ and $\mathcal{W}_\tau$. Once trained, ML models can quickly determine the appropriate power and bandwidth allocation for each beam based on the demand pattern while minimizing payload resource consumption.

In principle, the generated labels should encompass all potential payload configurations, which consist of feasible combinations of power and bandwidth pairs per-beam derived from the sets $(\mathcal{P}_\tau, \mathcal{W}_\tau)$. In fact, a payload configuration can be represented as $S_\tau = \left[(P_\tau^1, \mathrm{W}_\tau^1), (P_\tau^2, \mathrm{W}_\tau^2), \ldots, (P_\tau^B, \mathrm{W}_\tau^B)\right]$, with each power or bandwidth variable selected from the corresponding set $\mathcal{P}_\tau$ or $\mathcal{W}_\tau$. However, due to the total power and bandwidth constraints defined in equation (5), numerous configurations fail to satisfy these constraints and are, therefore, not viable outputs. Moreover, many different configurations will be discarded due to the traffic pattern. However, this will be known after many iterations of exhaustive searches during training data generation. We refer to Section V for further details.

The time it takes to generate each sample using an exhaustive search to allocate resources optimally depends on the number of possible configurations on the satellite and the characteristics of the computer on which the data is generated. We used the High Performance Computing (HPC) facilities of the University of Luxembourg [39] to generate a large amount of data, and on average, it took more than 8 hours per 500 samples generated for the parameters defined in Section V.

One important consideration is that the larger the search space, the more complex the generation of training data based on exhaustive search becomes to the point that it may become infeasible. Although it may seem an alternative to generate the training database using traditional optimization techniques, it has been shown that most of these techniques only achieve sub-optimal performance due to the non-convex and non-linear structure of the RRM problems [3]. In that sense, despite the drawbacks and limitations of the exhaustive search, we have decided to use it for the generation of the database in order to achieve the absolute optimum resource allocation and thus improve the reliability of the data used to train the ML models. On the other hand, the whole training process is assumed offline to mitigate the inconvenience of the processing time for the training data generation.

Additionally, we have made our generated database available to the general public. To access it and obtain more information about the structure of this database, please refer to [40].

III. CONVENTIONAL ML BENCHMARK

In this section, we establish a CNN-based conventional ML benchmark for resource management tasks within the scope of on-board RRM systems using the Xilinx Versal VCK5000 platform. The CNN architecture, illustrated in

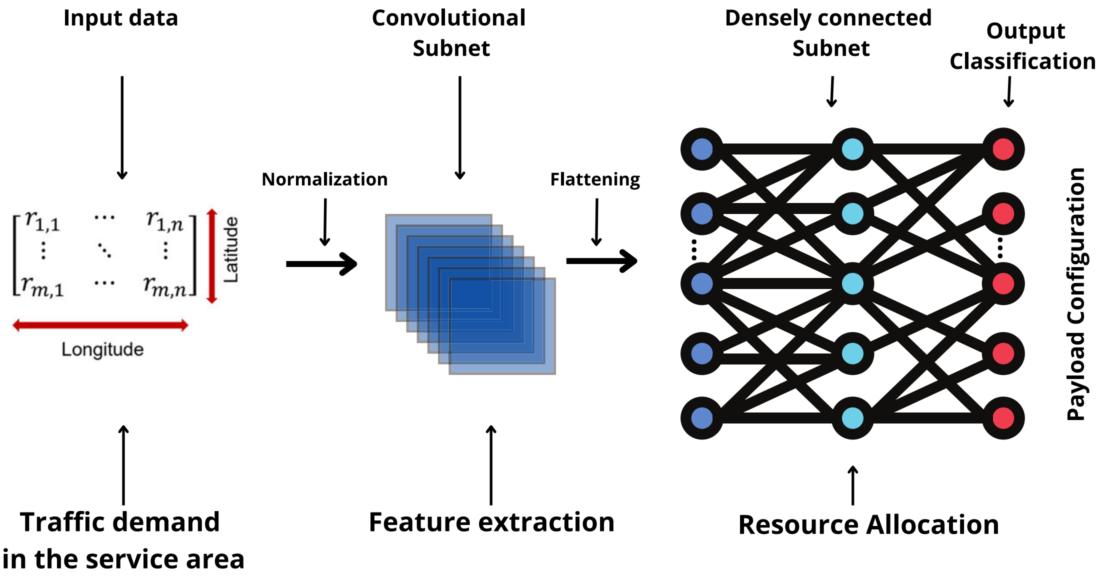


FIGURE 3. Benchmark approach based on convolutional neural networks (CNNs) for on-board RRM.

Figure 3, is tailored for the classification of potential payload configurations.

A. ARCHITECTURE

The proposed CNN takes input matrices $\boldsymbol{R}_\tau$, which encapsulate the demand requests across various beams and outputs a series of viable payload configurations S_τ. These configurations, represented as $S_\tau = \left[(P_\tau^1, \mathrm{W}_\tau^1), \ldots, (P_\tau^B, \mathrm{W}_\tau^B)\right]$, comply with the constraints imposed by the sets $\mathcal{P}_\tau$ and $\mathcal{W}_\tau$.

Our CNN comprises of convolutional layers for feature extraction, Rectified Linear Unit (ReLU) for nonlinearity, pooling layers for dimensionality reduction, and a fully-connected layer for classification. By constraining the output layer in alignment with $\mathcal{P}_\tau$ and $\mathcal{W}_\tau$, we effectively compress the solution space, as described in Section II-C and Section V alongside training data synthesis.

The architecture is optimized for performance, capitalizing on the ability of the convolutional layer to reduce parameter count and facilitate weight sharing. This efficiency is crucial in interpreting the complex input data and enabling fast, reliable classification:

- Convolutional layers conduct feature mapping via the convolution operation, maintaining spatial demand relationships to highlight input characteristics.
- ReLU layers implement nonlinearity, setting negative values to zero within the feature maps.
- Pooling layers condense feature dimensions while preserving salient information, thus summarizing the features extracted.
- The fully-connected layer harnesses these features to discern among the classes defined by the training data.

B. CNN TRAINING

For the CNN, we refine the network parameters $\boldsymbol{\mu}$ using a training dataset $\mathcal{D}$. Each element in $\mathcal{D}$ is a tuple $(\boldsymbol{R}_\tau, S_\tau)$, where $\boldsymbol{R}_\tau$ signifies the input matrix of traffic demands and S_τ represents the corresponding optimal payload configurations. The conversion of raw inputs into the format suitable for CNN processing will be elaborated upon in a forthcoming subsection.

We introduce a loss function $\mathcal{L}_{\boldsymbol{R}_\tau, S_\tau}(\boldsymbol{\mu})$ that measures the performance of the CNN. This function captures the difference between the network predictions and the actual configurations across a sequence of time-stamped inputs. Formally, the loss for a single data instance is

$$\mathcal{L}_{\boldsymbol{R}_\tau, S_\tau}(\boldsymbol{\mu}) = \sum_{b=1}^{B} \mathscr{L}\left(S_\tau^b, \mathfrak{F}_b(\boldsymbol{\mu}, \boldsymbol{R}_\tau)\right), \tag{7}$$

where $\mathscr{L}$ is a suitable loss function that compares the true payload configuration S_τ^b for each beam b with the CNN's prediction $\mathfrak{F}_b(\boldsymbol{\mu}, \boldsymbol{R}_\tau)$.

Aggregating these individual losses, the cumulative training loss $\mathcal{L}_\mathcal{D}$ over the entire dataset $\mathcal{D}$ is given by

$$\mathcal{L}_\mathcal{D}(\boldsymbol{\mu}) = \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{R}_\tau, S_\tau) \in \mathcal{D}} \mathcal{L}_{\boldsymbol{R}_\tau, S_\tau}(\boldsymbol{\mu}), \tag{8}$$

and the training goal is to find the optimal parameters $\boldsymbol{\mu}^*$ that minimize this loss, i.e.

$$\boldsymbol{\mu}^* = \underset{\boldsymbol{\mu}}{\operatorname{argmin}}\ \mathcal{L}_\mathcal{D}(\boldsymbol{\mu}). \tag{9}$$

For the classification of multi-dimensional input data, we utilize a loss function that enables probabilistic interpretation of class predictions, such as categorical cross-entropy. The loss for a given data instance is then

$$\mathcal{L}_{\boldsymbol{R}_\tau, S_\tau}(\boldsymbol{\mu}) = -\sum_{b=1}^{B} S_\tau^b \log(\mathfrak{F}_b(\boldsymbol{\mu}, \boldsymbol{R}_\tau)). \tag{10}$$

The CNN leverages gradient-based optimization techniques like Stochastic Gradient Descent (SGD) or Adam, which adaptively estimates lower-order moments to steer the network weights $\boldsymbol{\mu}$ towards the optimal values. Such optimization is facilitated by the backpropagation algorithm, which computes gradients efficiently through the network layers. We refer the reader to the references [41] for more details on this approach.

IV. NEUROMORPHIC COMPUTING FOR SatCom RRM

ML-based algorithms have gained popularity due to their performance and flexibility, but their practical application is hindered by the substantial computational power required for training and inference. This limitation becomes particularly evident when considering scenarios such as deploying these models on board of satellites, where the use of one or several Graphics Processing Units (GPUs), which is common for modern ML architectures, is impractical due to power restrictions. To address this challenge, one promising approach involves developing more energy-efficient versions of standard ML algorithms, e.g., by employing quantization of a model's weights. In this work, we investigate a potentially more efficient alternative, SNNs, which draw inspiration from the low-power operation of biological brains. In the following section, we provide an introduction to SNNs, including training techniques and data encoding methods. We further overview the Loihi 2 chip developed by Intel, and discuss the hardware deployment of SNNs.

A. SPIKING NEURAL NETWORKS

In general, an SNN is a directed, possibly cyclic, network of spiking neurons. Each spiking neuron is a dynamic system with inputs and outputs given by sequences of *spikes*, or binary {0, 1}, signals. SNNs comprise *read-out*, or *visible*, neurons, forming the network's outputs, as well as *exogeneous* input neurons. It also consists of *hidden* neurons, whose role is to facilitate the output of a desired spiking sequence [42], given a sequence of exogeneous inputs.

For the purpose of this study, we consider SNNs with fully connected layered topologies, i.e., each neuron in one layer is connected to all of the neurons in the next. Denoting as L the number of layers in the network, each layer $\ell \in \{1, \ldots, L\}$ consists of N_ℓ spiking neurons; we denote by $\boldsymbol{w}_\ell$ the $N_{\ell+1} \times N_\ell$ weight matrix between layer ℓ and layer $\ell + 1$ and by $\boldsymbol{w}$ the vector of all parameters. Each neuron k in layer $\ell + 1$ receives inputs from the set $\mathcal{N}_\ell$ of neurons in layer ℓ, i.e., we do not consider recurrent connections. The last layer comprises Z output neurons, corresponding to the number of classes (reduced number of payload configurations).

We consider neurons following the standard Spike Response Model (SRM) [43]. At every time-step $t = 1, \ldots, T$, where T denotes the temporal horizon of the task, each spiking neuron k outputs a binary signal $s_{k,t} \in \{0, 1\}$, with "1" representing the firing of a spike and "0" an idle neuron.

Following the SRM, each neuron k maintains at every time step t an internal analog state variable $u_{k,t}$, known as the *membrane potential*. Mathematically, the membrane potential $u_{k,t}$ is defined by the sum of filtered contributions from incoming spikes and from the neuron's own past outputs, i.e.,

$$u_{k,t} = \sum_{j \in \mathcal{N}_\ell} w_{k,j,\ell} \cdot (\alpha_t * b_{j,t}) + \beta_t * s_{k,t}, \tag{11}$$

where $w_{k,j,\ell}$ is the element (k, j) of matrix $\boldsymbol{w}_\ell$, which corresponds to the synaptic weight between neuron $j \in \mathcal{N}_\ell$ and neuron k in layer $\ell + 1$; α_t represents the synaptic response to a spike from the presynaptic neurons $j \in \mathcal{N}_\ell$ to a postsynaptic neuron k; β_t describes the synaptic response to the spike emitted by the neuron itself; and $*$ is the convolution operator. Neuron k outputs a spike at time step t when its membrane potential $u_{k,t}$ passes some fixed threshold ϑ, i.e.,

$$s_{k,t} = \Theta(u_{k,t} - \vartheta). \tag{12}$$

We refer to the review [20] for more details.

B. SNN TRAINING

The training loss over the parameter vector $\boldsymbol{w}$ is defined using the training dataset $\mathcal{D} = \{(\boldsymbol{r}, \boldsymbol{y})\}$ composed of the encoded traffic requirement signals $\boldsymbol{r} = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_T)$ and corresponding targets $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T)$. How to obtain the encoded signals $\boldsymbol{r}$ and $\boldsymbol{y}$ from input $\mathbf{R}$ in (6) and targets $S_\tau = \left[(P_\tau^1, \mathrm{W}_\tau^1), (P_\tau^2, \mathrm{W}_\tau^2), \ldots, (P_\tau^B, \mathrm{W}_\tau^B)\right]$ will be discussed in the next subsection.

We define the loss $\mathcal{L}_{\boldsymbol{r},\boldsymbol{y}}(\boldsymbol{w})$ measured with respect to a data $(\boldsymbol{r}, \boldsymbol{y}) \in \mathcal{D}$ as the error between the reference signals $\boldsymbol{y}$ and the output spiking signals produced by the SNN with parameters $\boldsymbol{w}$, given the input $\boldsymbol{r}$. Accordingly, the loss is written as a sum over time instants $t \in \{1, \ldots, T\}$ and over the Z read-out neurons as

$$\begin{aligned} \mathcal{L}_{\boldsymbol{r},\boldsymbol{y}}(\boldsymbol{w}) &= \sum_{t=1}^{T} \mathcal{L}_{\boldsymbol{r}^t,\boldsymbol{y}_t}(\boldsymbol{w}) \\ &= \sum_{t=1}^{T} \sum_{k=1}^{Z} \mathscr{L}\big(y_{k,t}, f_k(\boldsymbol{w}, \boldsymbol{r}^t)\big), \end{aligned} \tag{13}$$

where function $\mathscr{L}\big(y_{k,t}, f_k(\boldsymbol{w}, \boldsymbol{r}^t)\big)$ is a local loss measure comparing the target output $y_{k,t}$ of neuron k at time t and the actual output $f_k(\boldsymbol{w}, \boldsymbol{r}^t)$ of the same neuron, given the inputs $\boldsymbol{r}^t = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_t)$ up to time t. The notations $f_k(\boldsymbol{w}, \boldsymbol{r}^t)$ and $\mathcal{L}_{\boldsymbol{r}^t,\boldsymbol{y}_t}(\boldsymbol{w})$ are used as a reminder that the output of the SNN and the corresponding loss at time t generally depend on the input $\boldsymbol{r}^t$ up to time t, and on the target output $\boldsymbol{y}_t$ at time t. Specifically, the notation $f_k(\boldsymbol{w}, \boldsymbol{r}^t)$ makes it clear that the output of neuron $k \in N_\ell$ is produced with the model parameters $\boldsymbol{w}$ from exogeneous input $\boldsymbol{r}^t$, consisting of all input samples up to time t, using the SRM (11)-(12).

The training loss $\mathcal{L}_{\mathcal{D}}(\boldsymbol{w})$ is given as

$$\mathcal{L}_{\mathcal{D}}(\boldsymbol{w}) = \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{r},\boldsymbol{y}) \in \mathcal{D}} \mathcal{L}_{\boldsymbol{r},\boldsymbol{y}}(\boldsymbol{w}), \tag{14}$$

and training is done by minimizing the loss as

$$\min_{\boldsymbol{w}} \; \mathcal{L}_{\mathcal{D}}(\boldsymbol{w}). \tag{15}$$

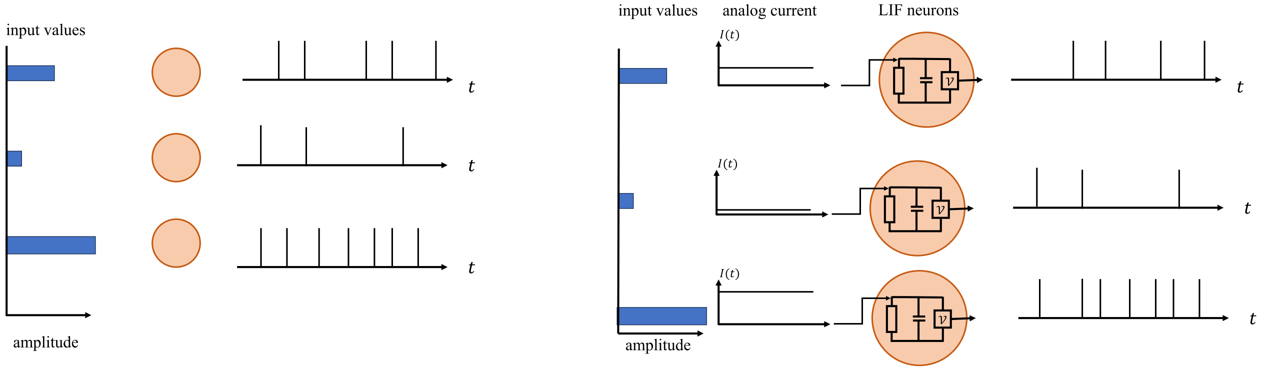


FIGURE 4. **Illustration of the encoding techniques employed for this study. Left: rate coding. Right: encoding with a time encoding machine composed of LIF neurons.**

We consider a *spike-rate* loss, whereby the SNN is given a rate ρ to dictate the desired rate of the correct output neuron for each example, and a spike rate ρ_F to dictate the desired spike rate of the other output neurons. For a classification problem with C classes, if c is the correct class for the given input, the spike-rate loss is defined as

$$\mathcal{L}_{\boldsymbol{r},\boldsymbol{y}}(\boldsymbol{w}) = \frac{1}{2}\sum_{t=1}^{\mathrm{T}}\sum_{k\neq c}(f_k(\boldsymbol{w},\boldsymbol{r}^t)-\rho_F)+(f_c(\boldsymbol{w},\boldsymbol{r}^t)-\rho). \tag{16}$$

Throughout, we set $\rho_F = 0.01$.

Problem (15) cannot be directly solved using standard gradient-based methods since the spiking mechanism (12) is not differentiable in $\boldsymbol{w}$ due to the presence of the threshold function $\Theta(\cdot)$. To tackle the former problem, surrogate gradients (SG) methods replace the derivative of the threshold function $\Theta(\cdot)$ in (12) with a suitable differentiable approximation. We direct the reader to the reviews [20], [44] for more details on this approach.

C. SPIKE ENCODING

Although spiking neuron models can in principle receive data in the form of an analog input current, neuromorphic processors can typically handle data only in the form of binary inputs. Consequently, the natural signals $(r_{i,j})$ representing the required traffic to be encoded into binary spikes for processing using the neuromorphic chip. Encoding into spiking signals is performed as follows and as illustrated in Fig. 4.

Considering the feature matrix $\boldsymbol{R} \in \mathbb{R}^{n\times m}$ in (6), we first perform max-pooling with pool size (ds, ds) to obtain an $(n/ds)\times(m/ds)$ matrix. This dimensionality reduction was key to reduce the transmission time of data to the neuromorphic chip. The resulting matrix is flattened into $(n/ds)(m/ds)\times 1$ column vector $[r_{1,1}\ldots r_{n/ds,m/ds}]^{\mathrm{T}}$. We finally perform the encoding of this column vector into a collection of spiking signals $\boldsymbol{r} \in \{0,1\}^{(n/ds)(m/ds)\times T}$, with T being the number of encoding time-steps. Although a number of encoding techniques are used in the literature, we propose to compare in this study results obtained using rate encoding [42] and a Time Encoding Machine (TEM). We detail both techniques below. Note that time encoding may potentially increase the sparsity of the signals, hence reducing power consumption, but we were not able to train models successfully using this method.

(i) Rate encoding: As seen in Fig. 4, each input value, as shown using the horizontal blue bars on the left, is encoded in the spike rate of the corresponding encoding neuron: a larger input generates a large number of spikes within a fixed encoding window time.

(ii) Time encoding machine: A TEM is a system that receives as input a (bounded) natural signal $x(t)$, and outputs binary spikes. We consider a TEM model based on Leaky Integrate-and-Fire (LIF) neurons, whereby a spike is emitted when the voltage of the neuron crosses a pre-defined threshold. More specifically, the TEM operates using the following set of recursive equations

$$\begin{aligned} u_t &= (1-\alpha_u)\cdot u_{t-1} + x_t \\ v_t &= (1-\alpha_v)\cdot v_{t-1} + u_t \\ s_t &= \Theta(v_t - \vartheta) \\ v_t &= v_t \cdot (1 - s_t), \end{aligned} \tag{17}$$

where u_t and v_t denote the current and voltage of the neuron at time instant t; $x_t := x(t)$; $1 > \alpha_u > 0$ and $1 > \alpha_v > 0$; ϑ is the threshold of the neuron; $s_t \in \{0,1\}$ is the spike output at instant t; and $\Theta(\cdot)$ is the Heaviside step function. In practice, a single signal x can be encoded through N TEMs with varying decays α_u and α_t.

V. CHIPSET IMPLEMENTATION

In this section, we present the implementation of the CNN model and the SNN model in hardware chipsets, namely Xilinx Versal VCK5000 and Intel Loihi2, respectively.

In selecting the Xilinx Versal VCK5000 for benchmarking, we aimed to align with the cutting-edge of satellite processing technology. The ESA SPAICE project adoption of the Versal family underscores its suitability for space applications [45], offering advanced on-board data processing and AI capabil-

TABLE 2. **Simulation parameters for generating training data.**

Frequency	19 GHz
Satellite position	13 E
Satellite altitude	35786 km
$\theta_{3\,\mathrm{dB}}$	1 deg.
Number of beams	8

Beam centres	lat =[39.3 42 44.7 47.4 51 53.7 56.4 39.5]
	long =[-5.3 0 5.3 10.6 -0.5 6 12.3 14.4]

G/T_{RX}	17 dB/K
P_t^b	10, 12, or 14 dBW
W_t^b	250 or 500 MHz

ities [32]. With the first space-grade Versal AI Core devices already in use, these processors represent the next generation of satellite hardware, offering significant enhancements in signal processing and AI inferencing. Our benchmark choice thus reflects an anticipation of future industry standards and the evolving requirements of satellite technology.

First, let us define the simulation parameters. The focus of this study is on downlink analysis for the forward link. The parameters used during the simulations are listed in Table 2. We consider two different values of bandwidth, $\mathcal{W}_\tau = \{250, 500\}$ MHz, and three values of power, $\mathcal{P}_\tau = \{10, 12, 14\}$ dBW. This results in six possible values of capacity that can be offered in each beam as listed in Table 3.

We focus on eight beams and six configurations per beam, and hence the number of possible payload configurations is more than 40,000 options. However, after setting $P_{\max,T} = 115$ W, all configurations that do not meet the power constraint are eliminated, resulting in the reduction of the number of feasible configurations to less than 1%. Finally, many configurations will be discarded again due to the traffic pattern as explained in Section III, resulting in $Z = 6$ distinct possible classes (number of configurations that are actually deployed on the satellite).

30,000 samples were generated, The data was divided into two sets, with 80% used for training and 20% used for validation.

A. CNN MODEL

The CNN model is implemented on the VCK5000 AI accelerator, a high-performance platform based on the Xilinx 7nm Versal ACAP architecture. It utilizes matrices as input and applies convolutional layers for feature extraction, followed by fully connected layers for classification. The CNN architecture comprises Conv2D, Maxpooling2D, and Dense layers.

The specific CNN architecture employed consists of Conv2D layers with 8 filters and a kernel size of (3,3), followed by Maxpooling2D layers with a pool size of (2,2). This is followed by additional Conv2D and Maxpooling2D layers with 4 filters and a kernel size of (3,3) and (2,2), respectively. The subsequent layers include a Flatten layer, Dense layers with 512 and 256 units and ReLU activation, and a final Dense layer with 6 units and softmax activation, representing the different payload configurations. The trainable parameters sum up to a total of 3,192,058.

To implement the CNN model, we consider using the VCK5000 Versal development card, designed to provide high-throughput AI inference and signal processing compute performance. It supports popular ML frameworks such as TensorFlow, PyTorch, and Caffe, using Python or C++ APIs. The Vitis AI framework facilitates the deployment of TensorFlow/PyTorch trained models on the VCK5000 for inference. In Table 4 we summarize the main VCK5000 card features.

B. NEUROMORPHIC MODEL FOR RMM

As seen in Fig. 5, we consider a layered SNN with $L = 4$ layers, where the hidden layers comprise $512, 256$ and 512 neurons respectively, and $Z = 6$. We train the system via the SG-based method SLAYER [46]. SGD is carried out using the Adam optimizer. Models are trained using Intel's Lava library [47] with Loihi bit-accurate precision, on a single A100 GPU. On-chip training was not yet available on Intel's Loihi 2 at the time of writing. Decisions are obtained via rate decoding, i.e., by selecting the output neuron with the largest spiking rate.

Following the approach proposed in reference [34], training is completed using a dataset $\mathcal{D}$, composed of measurements of the required capacity in each geographical zone. Each example $\boldsymbol{R} \in \mathbb{R}^{m\times n}$ in dataset $\mathcal{D}$ consists of $n \times m$ resource requirements, in Mbps, for each geographical position, as detailed in Section II. We preprocess each example independently as follows. First, we set all the outlier values over a given percentile p to the value r_p, which is the value such that p% of the entries in $\boldsymbol{R}$ are smaller than r_p. Throughout, we set $p = 0.98$. We then normalize the examples to the range [0, 1], and perform max-pooling with stride ds to reduce the input size to $(n/ds) \times (m/ds)$ before encoding into binary spiking signals, as described in Section IV-C.

We perform inference using SNNs as described in the previous sections on Intel's Loihi 2 chips [47]. Loihi 2 is a research neuromorphic chip that uses asynchronous spiking neurons to implement fine-grained, event-driven, adaptive, self-modifying, parallel computations. Loihi's first iteration was fabricated on Intel's 14 nm process and houses 128 clusters of 1,024 artificial neurons each, for a total of 131,072 simulated neurons, which is about 130 million synapses, which is still far below the 800 trillion synapses in the human brain. As members of the Intel Neuromorphic Research Community (INRC), we were given access to Loihi 1 under the Kapoho Bay form factor (see Table 5), as well as the second iteration of the chip via Intel's cloud services. Experimental results were obtained on Loihi 2.

The Lava software library gives access to a number of metrics of interest, of which we show a summary in Fig. 6. We can hence measure the evolution of the execution time per algorithmic step, and the total power consumed by them,

TABLE 3. Possible resource allocation configurations in a beam.

W_τ^b [MHz]	P_τ^b [dBW]	$\mathrm{EIRP}_{\max}^b$ [dBW]	γ_τ^b [dB]	κ_τ^b [bps/Hz]	C_τ^b [Mbps]
250	10	54.94	6.4615	1.8865	471.6312
500	10	54.94	3.4817	1.3350	667.4827
250	12	56.94	8.4261	2.2502	562.5421
500	12	56.94	5.4638	1.7019	850.9288
250	14	58.94	10.3705	2.6101	652.5215
500	14	58.94	7.4357	2.0668	1033.400

TABLE 4. Summary of VCK5000 card features.

Feature	Description
ACAP Device	XCVC1902-2MSEVSVD1760
Configuration Options	2 GB OSPI memory MT35XU01GBBA2G12-0SIT
Memory	16 GB - four 4 GB modules, part MT40A512M16TB-062E:J
DDR Maximum Data Rate	3200 MT/s
DDR Memory Bandwidth	102.4GBs
JTAG and UART Debug Interface	JTAG and UART access through USB and Maintenance connector
Edge Connector Interface	PCIe Gen. 3 x16 / 2x Gen. 4 x 8 / Gen. 4 x 8 /CCIX
Network Interface	2x QSFP28
ACAP VCCINT VRM	6 phases. PMBUS control for VOUT, temperature, and current
Satellite Controller (SC)	MSP432P4111IPZR
Temperature Monitoring	ACAP (internal monitoring), QSFP, inlet and outlet on-board thermal sensors
Power Monitoring	PCIe 12V, AUX0, AUX1, and VRM VCCINT & 1.2V
Status LEDs	Activity LED
Form Factor	Passive configuration: 3/4-length, full height, double slot, x16 PCIe form factor. Active configuration: full length, full height, double slot, x16 PCIe form factor
Motherboard	PCIe 3.0-compatible with one dual-width x16 slot.
System Power Supply	Minimum 225W available via PCIe slot connection and 6-pin and 8-pin PCIe auxiliary power cables
Operating System	Linux, Windows
Thermal Options	Active or Passive

TABLE 5. Summary of Kapoho Bay USB Flash Drive incorporating 2 Loihi chips.

Feature	Description
Form factor	USB stick
Host interface	USB 3.0
DVS interface	For neuromorphic sensors such as a camera
Loihi chips	2
Neuromorphic cores	256
Artificial neurons	262,144
Synapses	260 million
Process	14 nm
Transistors	4.14 billion

as well as the activity over the various cores on the chip. Activity measurements comprise the number of synaptic operations, neuron updates, and output and input spikes per core. As can be seen, the execution time per step decreases after the first few steps. In our experiments, we exploit this by initializing the network by sending all-zeros inputs up to $T = 8$ time-steps, and start recording predictions from that point onwards, with the black dotted line marking the beginning of the recording. During inference for the task at hand, the power consumption is seen to vary around 0.12 W, which comprises the power expenditure for input and output of spikes to the chip. Finally, the activity of the network is balanced over several cores, which is directly optimized by the Lava library.

C. EVALUATION METRICS

We evaluate the performance of the proposed algorithm in terms of *average capacity gap*. This is a measure of the capacity gap between the predicted configuration, and the resource requirements. Formally, it is defined as

$$G = \frac{1}{B|\mathcal{D}|}\sum_{\boldsymbol{R}\in\mathcal{D}}\sum_{b=1}^{B}\left|C_{\boldsymbol{R}}^{b} - Y_{\boldsymbol{R}}^{b}\right|, \tag{18}$$

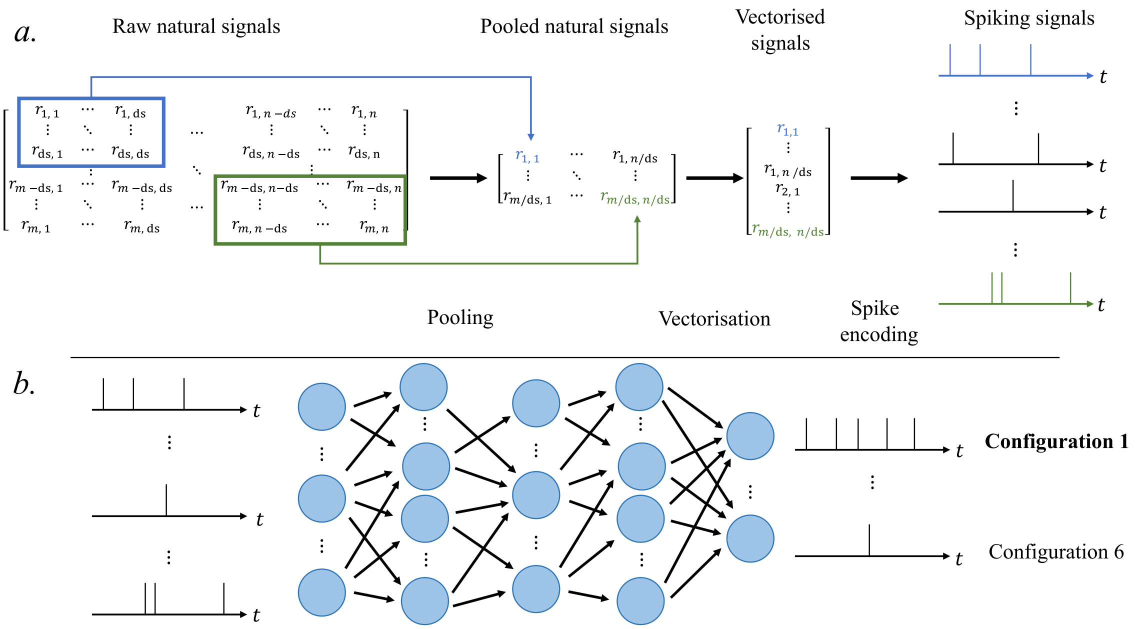


FIGURE 5. Proposed neuromorphic solution. (a) Encoding of natural signals into spikes. (b) Spiking neural networks with a layered architecture comprising three hidden layers. The prediction is given by rate decoding, that is, by selecting the index of the readout neuron producing the most spikes.

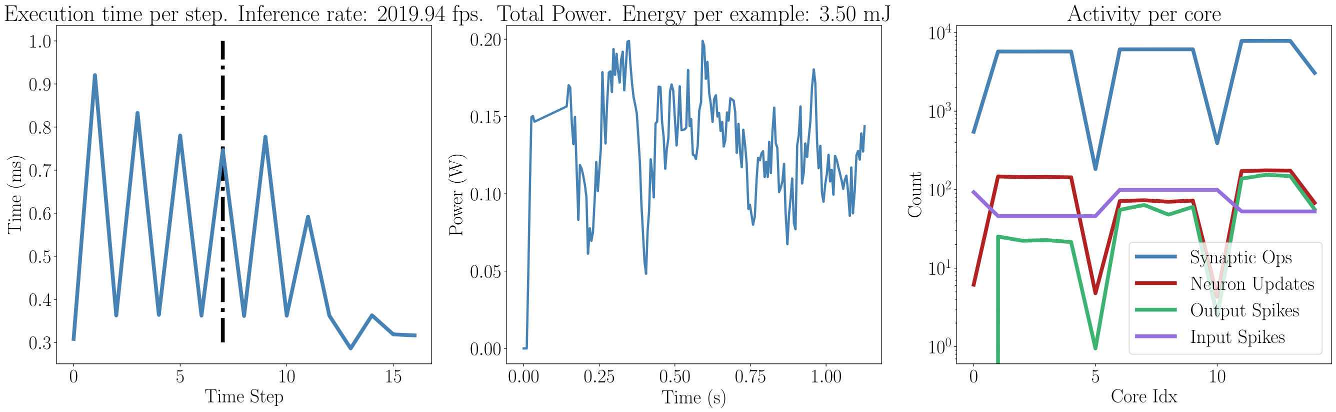


FIGURE 6. Summary of the metrics measured on Loihi 2. Left: Execution time per algorithmic time-step. Middle: Evolution of the total power consumption. Right: Activity per core.

where we have defined $C_{\boldsymbol{R}}^{b}$, the required capacity for beam b in example $\boldsymbol{R}$, and the corresponding prediction $Y_{\boldsymbol{R}}^{b}$.

Further metrics can be employed when comparing a neuromorphic to a conventional approach. These metrics include accuracy, but also latency and energy consumption, as we detail now.

- *Accuracy*: Accuracy measures the ability of the RRM algorithm to choose the most appropriate configuration given the current traffic requirements. It is obtained by comparing the configuration predicted by the model to the ground truth. Comparing the accuracy achieved by both approaches allows to determine which approach performs better in effectively utilizing the available resources.
- *Latency*: Latency is a measure of the computational efficiency and responsiveness of the algorithm under study, and a key criterion in many SatCom applications. In the case of RRM, it measures how quickly the proposed algorithm can measure a change in the traffic requirement conditions, and propose an alternative configuration.
- *Energy Consumption*: Energy consumption is a critical metric, especially in satellite systems, wherein power resources are often limited. Evaluating the energy consumed by RRM approaches allows us to assess their suitability for deployment on board of satellites. The energy consumption of the neuromorphic solution includes all the energy spent by the chip during

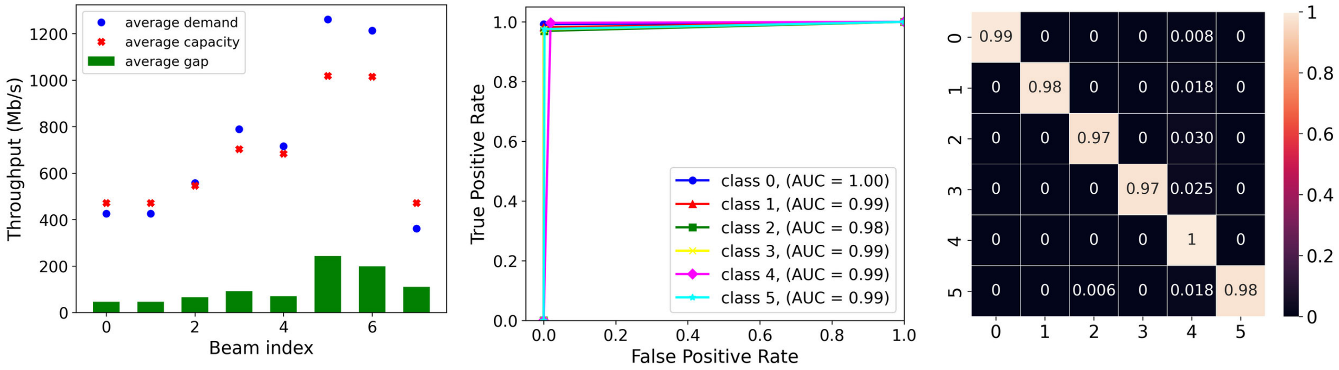


FIGURE 7. Results obtained for a CNN model with max-pooling stride *ds* = 3. Left: Capacity gap per beam. Middle: ROC curves. Right: Confusion matrix.

inference, including I/O interfaces. For the conventional approach, energy consumption relates to the power consumed by the VCK chip during inference. Comparing the energy consumed by both approaches allows to inform choices pertaining to the deployment of algorithms on board of satellite systems.

VI. EXPERIMENTAL RESULTS

In this section, we present the most outstanding numerical results.

A. BENCHMARKING

The CNN model was trained for 25 epochs with a batch size of 128. The training for the conventional models was carried out using the HPC facilities of the University of Luxembourg [39] (see hpc.uni.lu). The training configuration used the SGD optimizer with a learning rate of 0.01, momentum of 0.9, and Nesterov acceleration. The loss function used was the categorical cross-entropy and the model's accuracy was used as the metric.

We determined experimentally that the set of hyperparameters providing the maximal accuracy. The final accuracy of the model on the training data corresponded to max-pooling stride $ds = 3$ was 98.93% and the validation accuracy was 98.87%. In Fig. 7, we show the capacity gap, Receiver Operating Characteristic (ROC) curves and confusion matrix for the predictions given by this model. The capacity gaps stem from the power constraint of the system, which does not allow to serve all the users. Concerning the system performance, Fig. 8 shows the average capacity gap in the validation data in which we observe that in all cases the capacity follows the traffic demand and in the cases where the gap is larger, it is due to the lower and upper limit on the satellite resources. In this regard, the confusion matrix shows that for all configurations we obtain a ratio higher than 96%, reaching a ratio of 100% in class 4, which is the class that has the highest probability of occurrence.

The ROC plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at different classification thresholds. It provides information about the trade-off between TPR and FPR, and it is useful for selecting the optimal classification threshold. The Area Under the Curve (AUC) is the area under the ROC curve. AUC ranges between 0 and 1, where a higher value indicates a better model. Accordingly, we can observe that the CNN results for the flexible payload maintain an area close to 1 for all classes.

TABLE 6. F1-score for test data.

Classes	F1 Score
Configuration 0	0.996
Configuration 1	0.990
Configuration 2	0.972
Configuration 3	0.987
Configuration 4	0.991
Configuration 5	0.985

As for the F1-score, it measures the model's balance between precision and recall. A high F1-score indicates a good balance between the two. Table 6 shows the obtained F1-scores. The results indicate that the CNN architecture and the training configuration used were able to effectively manage the power and bandwidth of a multibeam satellite as a function of traffic demand.

Finally, we vary the size of the traffic matrix, $\boldsymbol{R}_\tau$, by considering several max-pooling strides ds. As seen in Table 7, by increasing ds (i.e., reducing the matrix size), the execution time can be reduced. However, this comes at the cost of accuracy. Hence, we use $ds = 3$ for the purpose of comparison with the neuromorphic model in the following sections.

B. NEUROMORPHIC MODEL

We determined experimentally that the set of hyperparameters providing the maximal accuracy whilst minimizing energy expenditure and execution time corresponded to max-pooling stride $ds = 32$; number of encoding time-steps $T = 8$; and target spike rate $\rho = 0.5$, when using TEM encoding. This set of hyperparameters provides a reference

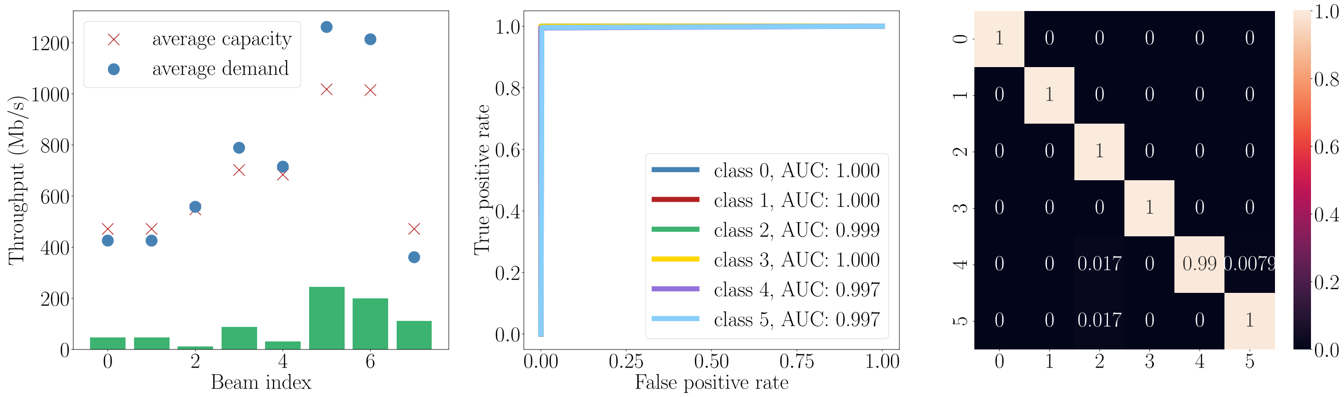


FIGURE 8. **Results obtained for an SNN model with max-pooling stride $ds = 32$; number of encoding time-steps $T = 8$; target spike rate $r = 0.5$, with TEM encoding. Left: Capacity gap per beam. Middle: ROC curves. Right: Confusion matrix.**

TABLE 7. **Accuracy and time per example for different size of the traffic matrix using maxpooling.**

Max-pooling strides ds	Accuracy	time per example (ms)
No max-pooling	0.988	126
$ds = 3$	0.988	114
$ds = 28$	0.962	68
$ds = 30$	0.957	67
$ds = 32$	0.951	67
$ds = 34$	0.945	64
$ds = 36$	0.931	64

model, which reaches 99.6% of test accuracy. In Fig. 8, we show the capacity gap, ROC curves and confusion matrix for the predictions given by this model. As in the results observed with the CNN model, the capacity gaps stem from the power constraint of the system, which does not allow to serve all the users. The predictions made by the SNN provide better AUC than the CNN for all classes, only predicting the wrong classes for labels 4 and 5, contrary to the conventional solution, which seldom reach perfect accuracy for a single class. Overall, the SNN is seen to obtain better accuracy than the CNN.

In Fig. 9, we compare results in terms of average execution time and energy expenditure per example between the neuromorphic processor and Intel's *lava-dl* simulator running on an A100 GPU. Measurements on the GPU are carried using the PyJoules library [48]. It can be seen that the simulator running on GPU is slower than the execution on Loihi 2, but also six orders of magnitude more costly in terms of energy. It is to be noted that execution on GPU also allows one to perform inference over a minibatch $\mathcal{B}$ of size larger than one, which is not possible on the neuromorphic hardware.

In the following, we extensively study how the different hyperparameters of the system affect the metrics of interest – that is, accuracy, execution time, energy and the number of synaptic operations. In all the following figures, the red data point corresponds to the model with the hyperparameters of reference described above.

We start by comparing the two encoding techniques described in Section IV-C, i.e., rate encoding and time encoding machines. As can be seen in Fig. 10, the TEM generally provides much higher accuracy, at the cost of a slightly larger execution time and energy expenditure compared with rate encoding with $T = 8$. By increasing the number of encoding time-steps, one can improve the accuracy of the model for rate encoding, although this proportionally increases execution time, energy expenditure and number of synaptic operations. Even with a 4× longer encoding of $T = 32$, rate encoding only reaches 85.6% test accuracy, which is 14 points below the results obtained by encoding the signal with a TEM.

Next, we vary the size of the exogenous inputs by considering several max-pooling strides ds. As seen in Fig. 11, by increasing the value of ds (that is, by reducing the exogenous signal size), one can reduce the execution time, energy and synaptic operations per example. However, this comes at the cost of accuracy, By reducing ds from $ds = 28$ to $ds = 36$, which results in decreasing the size of the inputs from $(n/22) \times (m/22) = 299$ to $(n/30) \times (m/30) = 180$, the test accuracy only diminishes by less than 1%.

In Fig. 12, we demonstrate how using smaller target rates ρ during training affects the metrics of interest. As one can see, using a rate smaller than $\rho = 0.5$ generally increases the processing time, energy per example, and the number of synaptic operations. This can be explained by the fact that forcing a smaller output rate actually pushes the hidden neurons to spike more to encode the signals of interest. Considering the fact that these are more numerous than the output neurons, this results in a generally larger number of synaptic operations, which also increases the processing time and energy expenditure. We determined experimentally that $\rho = 0.5$ provided the best trade-off between sparsity of the read-out neurons, sparse activation of the hidden neurons, and test accuracy.

We now examine how the number of encoding time-steps T affects the model when encoding the input signals

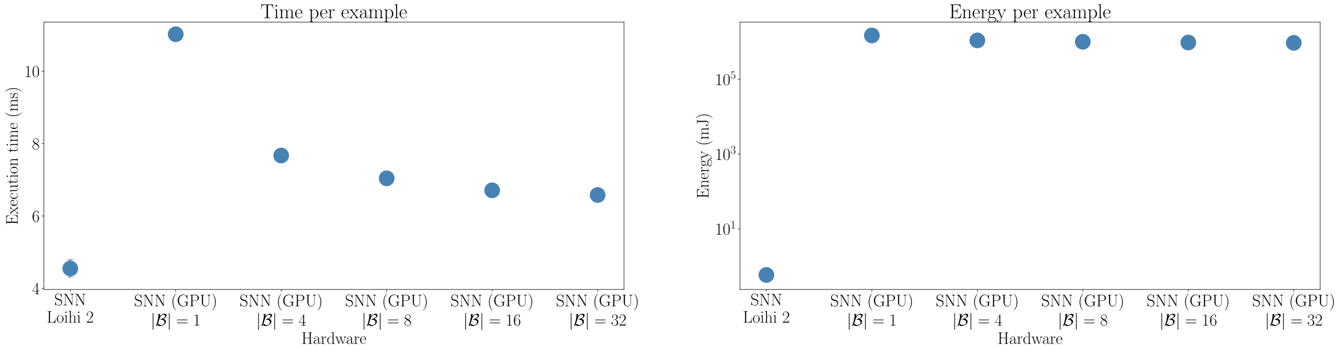


FIGURE 9. **Comparison between execution of an SNN on Loihi 2 and simulation on an NVIDIA A100 GPU with various batch sizes. Left: Average execution time per example. Right: Energy expenditure.**

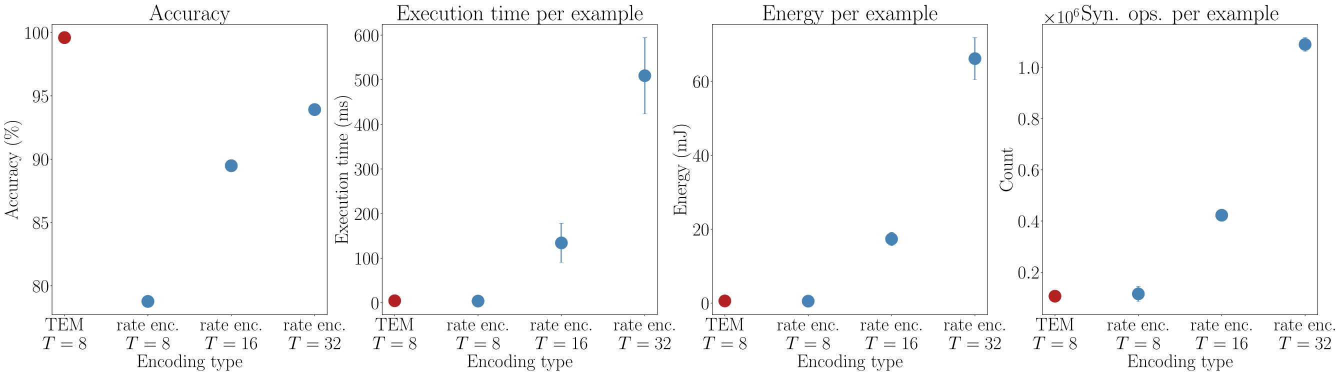


FIGURE 10. **Comparison between various types of encoding techniques and number of encoding time-steps. From left to right are shown: Test accuracy, average execution time per example, average energy expenditure per example, average number of synaptic operations per example. In red are shown the parameters selected for comparison with the conventional approach.**

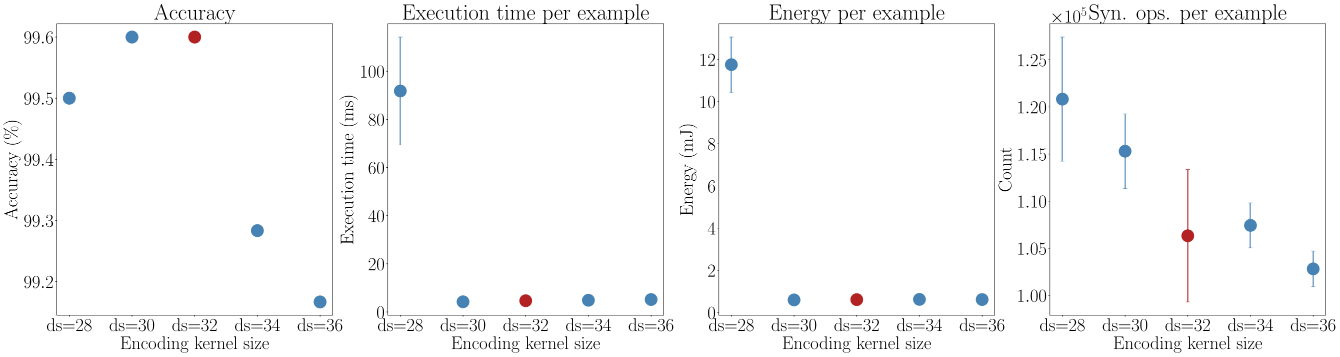


FIGURE 11. **Comparison between various encoding max-pooling strides *ds*. From left to right are shown: Test accuracy, average execution time per example, average energy expenditure per example, and average number of synaptic operations per example. In red are shown the parameters selected for comparison with the conventional approach.**

with a TEM. As can be seen, decreasing T below the reference value $T = 8$ provides worse-than-chance accuracy. This is due to the fact that with $T = 4$, the encoder does not produce enough spikes for the model to carry out the classification task. Increasing the number of time-steps does not provide an improvement in terms of accuracy.

Lastly, varying the threshold ϑ of the TEM in (17) allows to control the sparsity of the input signal. We observe that increasing ϑ generally decreases the accuracy of the system, although the decrease is almost negligible for $\vartheta < 5$. As one can expect, sparser rates in the exogenous inputs in turn cause the network to spike less, which results in a smaller energy footprint. However, this does not bring benefits in terms of execution time per example.

Overall, we explored how a variety of hyperparameters impact the operation of the system. We found a reference set of parameters that allowed us to miminize the execution

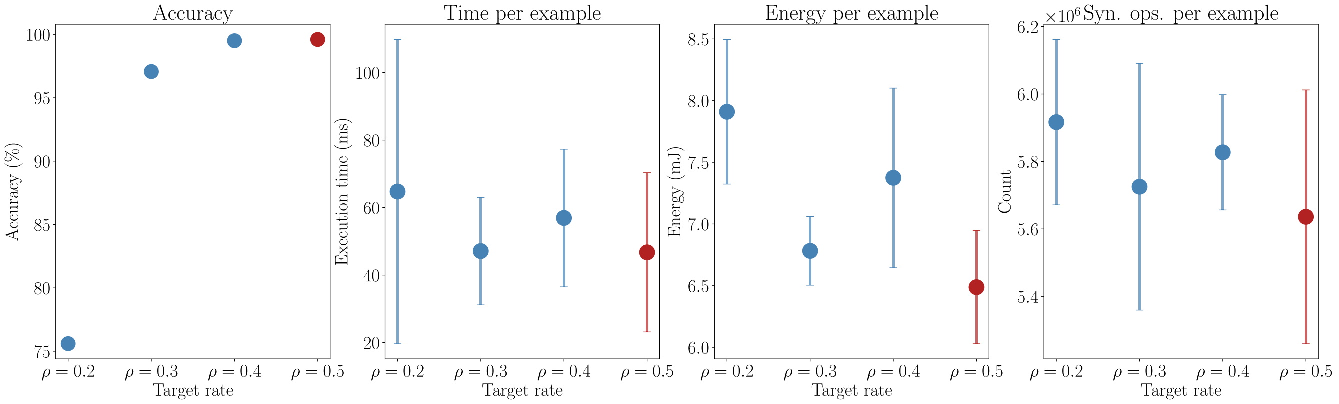


FIGURE 12. Comparison between various target rates ρ. From left to right are shown: Test accuracy, average execution time per example, average energy expenditure per example, and average number of synaptic operations per example. In red are shown the parameters selected for comparison with the conventional approach.

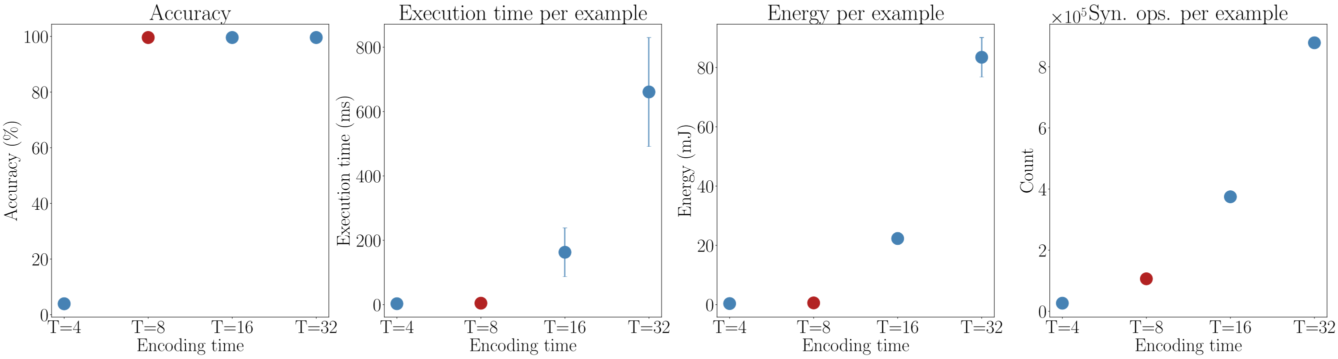


FIGURE 13. Comparison between the number of encoding time-steps T. From left to right are shown: Test accuracy, average execution time per example, average energy expenditure per example, and average number of synaptic operations per example. In red are shown the parameters selected for comparison with the conventional approach.

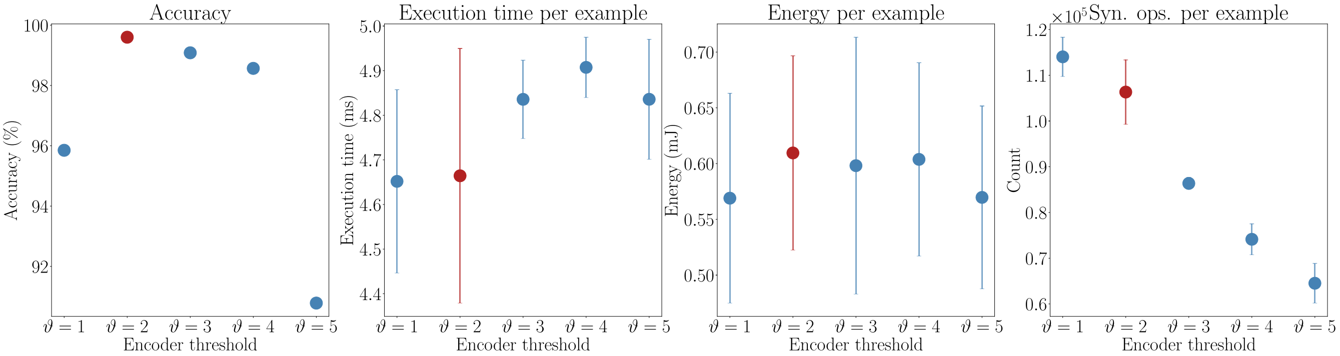


FIGURE 14. Comparison between the number of TEM threshold ϑ. From left to right are shown: Test accuracy, average execution time per example, average energy expenditure per example, and average number of synaptic operations per example. In red are shown the parameters selected for comparison with the conventional approach.

time and energy expenditure while maintaining the highest accuracy we were able to obtain. We note that we were able to further reduce the processing time, even by reducing more the input size or sparsity of the signal. We suspect this is because, at this level of sparsity, the execution time is lower-bounded by the general operation time of the chip.

C. COMPARISON BETWEEN NON-NEUROMORPHIC AND NEUROMORPHIC MODELS

We present a comparative analysis between the results obtained from inference using SNN on Loihi 2 and the VCK5000 chip for the CNN model. Fig. 15 presents a comprehensive comparison of accuracy and power consumption for both combinations. The results clearly demonstrate

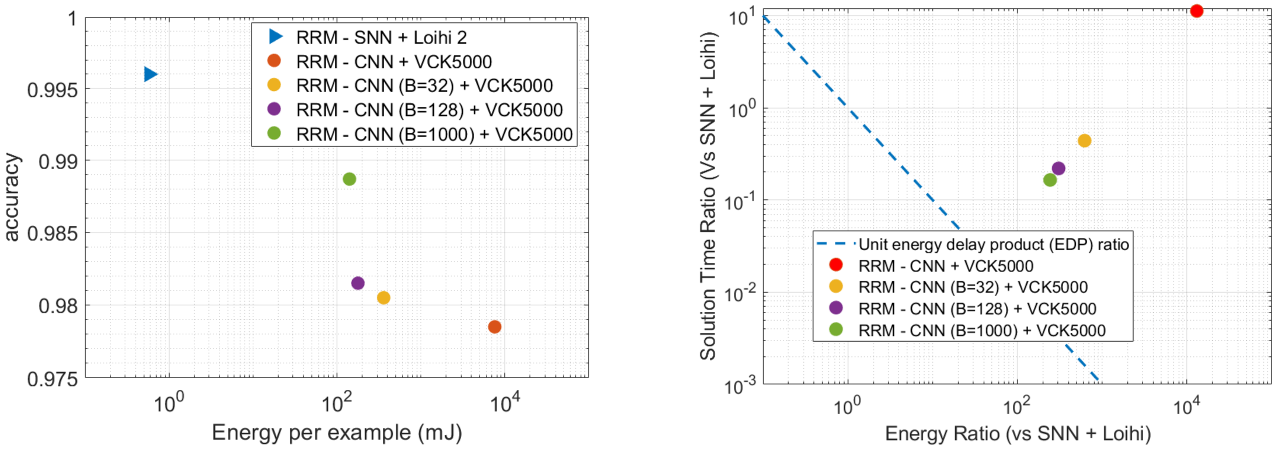


FIGURE 15. Comparison between the execution of an SNN on Loihi 2 and the execution of a CNN on the CPU of the VCK 5000 chip. Left: Accuracy vs. Energy Consumption. Right: Solution Time Ratio.

that the neuromorphic model consistently outperforms the non-neuromorphic approach in terms of accuracy while drastically reducing energy consumption. In fact, the energy consumption is up to two orders of magnitude lower than the models running on the VCK5000 chip.

One of the factors contributing to the superior performance of the neuromorphic approach is the deployment of the CNN model on the VCK5000 chip. To facilitate this deployment, the CNN model must undergo quantization, which involves reducing its accuracy. This quantization process inevitably leads to a loss of accuracy compared to the original values obtained in Section VI-A. Consequently, the lower accuracy of the CNN model quantized on the VCK5000 chip is one of the reasons for the difference in performance between the two approaches.

However, the primary and more significant reason behind the higher accuracy achieved through inference using SNN on Loihi 2 can be attributed to the characteristics of the traffic matrix $\boldsymbol{R}_\tau$. We have observed that the traffic matrix exhibits a large spatial sparsity, where a substantial portion of the values are zeros. The neuromorphic approach, particularly the SNN on Loihi 2, benefits significantly from this spatial sparsity, making it inherently well-suited to leverage such data characteristics.

The ability of the neuromorphic approach to process sparse data efficiently is a key advantage that enables it to outperform the non-neuromorphic approach. The unique design of the Loihi 2 chip allows for specialized processing of sparse data patterns, leading to optimized and accurate inferences. As a result, the neuromorphic models excel in extracting valuable information from the sparse traffic matrix $\boldsymbol{R}_\tau$, enabling them to achieve higher accuracy levels compared to the non-neuromorphic approach. Conversely, conventional deep neural networks are known to require dense data to learn informative features, which further motivates the use of neuromorphic learning for such use cases.

In Fig. 15, we present the results of the comparative benchmarks for latency and energy in a unified view. This two-dimensional plot highlights the key advantages offered by neuromorphic hardware, such as the Loihi chip, as compared to commercially available programmable architectures. The dashed diagonal line represents the energy-delay ratio parity line [49], with benchmark points located below and to the left of this line indicating architectures that outperform Loihi. In contrast, points located above and to the right indicate superior performance for Loihi. In all the scenarios presented here, the conventional models lie above the parity line, indicating that neuromorphic algorithms running on dedicated hardware provide clear benefits to perform RRM on board of satellites.

D. COMPLEXITY ANALYSIS

We now describe the computational capabilities and constraints of CNN and SNN models, with direct implications for on-board satellite systems. The CNN model comprises of 3,192,058 trainable parameters, while the SNN model incorporates 395,312 for $ds = 32$, reflecting the computational intensity required for operations like feature extraction and classification.

Comparing the memory complexity used for ML models with traditional optimization methods such as exhaustive search (which was used for the generation of training data), we have that for ML models the memory required involves the storage of parameters and intermediate data for the CNN and the retention of states and synaptic weights for each of the 1,280 neurons of the SNN model on the Intel Loihi2 chip. In comparison, exhaustive search requiring considerable memory for combinatorial exploration of configurations that for our scenario making to over 40,000 options.

Power consumption between both ML-based approaches contrasts the power consumption of the Versal VCK5000 board with the average of 21 W during CNN inference with the remarkable efficiency of SNN with an average of 0.12 W on Intel Loihi2. On the other hand, although not directly measured, the power overhead of exhaustive search is presumed significant due to its intensive computation.

Latency is evaluated by the time efficiency of the CNN and SNN in responding to data inputs, with a comparative illustration in Figure 15. The exhaustive search latency is inherently higher, as it encompasses the duration to traverse all configurations to produce viable training data sets.

VII. CONCLUSION

SNNs excel in processing sparse temporal data due to their reliance on spike patterns. However, they face limitations with dense datasets common across domains, hindering their universal adoption over CNNs. SNNs energy efficiency benefits rely on specialized neuromorphic hardware, yet its limited availability poses a barrier to widespread implementation. Furthermore, the learning curve associated with neuromorphic computing, its evolving frameworks, and tools contrasts with the mature ecosystem surrounding CNNs. Despite these challenges, ongoing research and technological advancements offer promise for neuromorphic computing to complement traditional methods, paving the way for more energy-efficient processing in the future.

This article presents an extensive investigation into the benefits of incorporating neuromorphic computing and SNNs for on-board radio resource management in SatCom systems. By leveraging innovative approaches, we addressed the challenge of implementing on-board RRM, comparing the performance of the proposed neuromorphic computing approach with a traditional CNN model. Our experiments demonstrate that SNNs, enabled by dedicated hardware, offer higher accuracy and significantly reduce energy consumption and latency. These remarkable results underscore the potential of neuromorphic computing and SNNs in improving RRM for SatCom, leading to better efficiency and sustainability for future SatCom systems.

To advance this research further, several avenues of investigation remain open. An important aspect is the implementation of the proposed approach in a real system, taking into account factors such as radiation tolerance, which holds great significance in the space environment. Moreover, future research could focus on optimizing the SNN architecture to achieve better performance and energy efficiency, considering the specific requirements and constraints of SatCom systems.

Although our current model is suitable for GEO satellite systems [50], we recognize the dynamic nature of LEO/MEO systems [51], [52], where the Doppler effect, rapid elevation angle changes, and other factors significantly influence the analysis. In future work we intend to extend our model to address these challenges, recognizing that channel and traffic conditions vary much more rapidly due to the higher relative velocity of LEO/MEO satellites. This will require a more complex and robust model to accommodate the highly dynamic environment.

We also wish to emphasize the generalization performance of the proposed models. Rigorous testing through cross-validation on diverse datasets simulating various operational scenarios has shown that the SNN model, in particular, exhibits strong generalization capabilities. This is evidenced by its ability to maintain high accuracy and low energy consumption when exposed to unseen data, indicating its robustness in real-world deployments. We acknowledge, however, that substantial changes in the operational environment, such as a transition from GEO to LEO/MEO systems, will necessitate the adaptation of the model. To this end, our future work will focus on enhancing the model complexity to cope with the increased dynamics of LEO/MEO systems, ensuring that the generalization capabilities extend across different orbital conditions. The planned integration of continual learning mechanisms is anticipated to bolster the model adaptability further, allowing it to update its parameters in response to evolving traffic patterns and channel conditions, thereby sustaining high performance without frequent retraining. These enhancements will be pivotal for deploying SNN-based RRM in the highly variable and demanding environment of space communications.

The findings of this study lay a solid foundation for the application of neuromorphic computing and SNNs in the field of SatCom RRM. Future investigations can build upon this work to further advance the state-of-the-art in SatCom systems, leveraging the benefits and insights gained from this comprehensive study.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of Intel Labs through the Intel Neuromorphic Research Community (INRC) and Tomas Navarro as ESA Officer. Please note that the views of the authors of this article do not necessarily reflect the views of ESA.

REFERENCES

[1] D. Wang, A. Traspadini, M. Giordani, M.-S. Alouini, and M. Zorzi, "On the performance of non-terrestrial networks to support the Internet of Things," in *Proc. 56th Asilomar Conf. Signals, Syst., Comput.*, Mar. 2023, pp. 881–887.

[2] N. Kato et al., "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 140–147, Aug. 2019.

[3] S. Kisseleff, E. Lagunas, T. S. Abdu, S. Chatzinotas, and B. Ottersten, "Radio resource management techniques for multibeam satellite systems," *IEEE Commun. Lett.*, vol. 25, no. 8, pp. 2448–2452, Aug. 2021.

[4] F. G. Ortíz-Gómez, R. M. Rodríguez-Osorio, M. Salas-Natera, and S. Landeros-Ayala, "Adaptive resources allocation for flexible payload enabling VHTS systems: Methodology and architecture," in *Proc. 36th Int. Commun. Satell. Syst. Conf. (ICSSC)*, Oct. 2018, pp. 1–8.

[5] M. Á. Vázquez et al., "Machine learning for satellite communications operations," *IEEE Commun. Mag.*, vol. 59, no. 2, pp. 22–27, Feb. 2021.

[6] X. Liao et al., "Distributed intelligence: A verification for multi-agent DRL-based multibeam satellite resource allocation," *IEEE Commun. Lett.*, vol. 24, no. 12, pp. 2785–2789, Dec. 2020.

[7] F. Ortiz et al., "Towards the application of neuromorphic computing to satellite communications," in *Proc. 39th Int. Commun. Satell. Syst. Conf. (ICSSC)*, Oct. 2022, pp. 91–97.

[8] A. Shrestha, H. Fang, Z. Mei, D. P. Rider, Q. Wu, and Q. Qiu, "A survey on neuromorphic computing: Models and hardware," *IEEE Circuits Syst. Mag.*, vol. 22, no. 2, pp. 6–35, 2nd Quart., 2022.

[9] J. Park, “Neuromorphic computing using emerging synaptic devices: A retrospective summary and an outlook,” *Electronics*, vol. 9, no. 9, p. 1414, Sep. 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/9/1414/htm

[10] G. Indiveri and T. K. Horiuchi, “Frontiers in neuromorphic engineering,” *Frontiers Neurosci.*, vol. 5, p. 118, Jan. 2011.

[11] M. Lin and Y. Zhao, “Artificial intelligence-empowered resource management for future wireless communications: A survey,” *China Commun.*, vol. 17, no. 3, pp. 58–77, Mar. 2020.

[12] Y. Sun and M. Peng, “Edge intelligence assisted resource management for satellite communication,” *China Commun.*, vol. 19, no. 8, pp. 31–40, Aug. 2022.

[13] F. G. Ortiz-Gomez et al., “Machine learning for radio resource management in multibeam GEO satellite systems,” *Electronics*, vol. 11, no. 7, p. 992, Mar. 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/7/992/htm

[14] A. Wang, L. Lei, E. Lagunas, S. Chatzinotas, and B. Ottersten, “Dual-DNN assisted optimization for efficient resource scheduling in NOMA-enabled satellite systems,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Madrid, Spain, Dec. 2021, pp. 1–6.

[15] B. Deng, C. Jiang, H. Yao, S. Guo, and S. Zhao, “The next generation heterogeneous satellite communication networks: Integration of resource management and deep reinforcement learning,” *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 105–111, Apr. 2020.

[16] P. V. R. Ferreira et al., “Multiobjective 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.

[17] J. J. G. Luis, M. Guerster, I. del Portillo, E. Crawley, and B. Cameron, “Deep reinforcement learning for continuous power allocation in flexible high throughput satellites,” in *Proc. IEEE Cognit. Commun. Aerosp. Appl. Workshop (CCAAW)*, Jun. 2019, pp. 1–4.

[18] 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. 2020, doi: 10.1002/sat.1310.

[19] 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.

[20] N. Skatchkovsky, H. Jang, and O. Simeone, “Spiking neural networks—Part III: Neuromorphic communications,” *IEEE Commun. Lett.*, vol. 25, no. 6, pp. 1746–1750, Jun. 2021.

[21] N. Skatchkovsky, H. Jang, and O. Simeone, “Federated neuromorphic learning of spiking neural networks for low-power edge intelligence,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 8524–8528.

[22] N. Skatchkovsky, H. Jang, and O. Simeone, “End-to-end learning of neuromorphic wireless systems for low-power edge artificial intelligence,” in *Proc. 54th Asilomar Conf. Signals, Syst., Comput.* Monterey, CA, USA: IEEE Computer Society, Nov. 2020, pp. 166–173.

[23] J. Chen, N. Skatchkovsky, and O. Simeone, “Neuromorphic integrated sensing and communications,” *IEEE Wireless Commun. Lett.*, vol. 12, no. 3, pp. 476–480, Mar. 2023.

[24] J. Chen, N. Skatchkovsky, and O. Simeone, “Neuromorphic wireless cognition: Event-driven semantic communications for remote inference,” *IEEE Trans. Cognit. Commun. Netw.*, vol. 9, no. 2, pp. 252–265, Apr. 2023.

[25] K. Dakic, B. A. Homssi, and A. Al-Hourani. (Oct. 2023). *Spiking-UNet: Spiking Neural Networks for Spectrum Occupancy Monitoring*. [Online]. Available: https://www.techrxiv.org/articles/preprint/Spiking-UNet_Spiking_Neural_Networks_for_Spectrum_Occupancy_Monitoring/24426955

[26] A. S. Kucik and G. Meoni, “Investigating spiking neural networks for energy-efficient on-board AI Applications. A case study in land cover and land use classification,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 2020–2030.

[27] D. Izzo, A. Hadjiivanov, D. Dold, G. Meoni, and E. Blazquez, “Neuromorphic computing and sensing in space,” 2022, *arXiv:2212.05236*.

[28] *TechEdSat-13: The First Flight of a Neuromorphic Processor*. Accessed: May 4, 2023. [Online]. Available: https://ntrs.nasa.gov/citations/20220005780

[29] *NeuroSat | ESA TIA*. Accessed: May 4, 2023. [Online]. Available: https://artes.esa.int/projects/neurosat

[30] B. A. Homssi et al., “Artificial intelligence techniques for next-generation massive satellite networks,” *IEEE Commun. Mag.*, pp. 1–7, Oct. 2023.

[31] M. Davies et al., “Loihi: A neuromorphic manycore processor with on-chip learning,” *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018.

[32] F. Ortiz et al., “Onboard processing in satellite communications using AI accelerators,” *Aerospace*, vol. 10, no. 2, p. 101, Jan. 2023. https://www.mdpi.com/2226-4310/10/2/101

[33] *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) | Engineering360*, Standard ETSI-EN 302 307-2, 2005. [Online]. Available: https://standards.globalspec.com/std/14417001/EN%20302%20307-2

[34] F. G. Ortiz-Gomez, D. Tarchi, R. M. Rodriguez-Osorio, A. Vanelli-Coralli, M. A. Salas-Natera, and S. Landeros-Ayala, “Supervised machine learning for power and bandwidth management in VHTS systems,” in *Proc. 10th Adv. Satell. Multimedia Syst. Conf. 16th Signal Process. Space Commun. Workshop (ASMS/SPSC)*, Oct. 2020, pp. 1–7.

[35] H. Al-Hraishawi, E. Lagunas, and S. Chatzinotas, “Traffic simulator for multibeam satellite communication systems,” in *Proc. 10th Adv. Satell. Multimedia Syst. Conf. 16th Signal Process. Space Commun. Workshop (ASMS/SPSC)*, Oct. 2020, pp. 1–8.

[36] *NASA, Socioeconomic Data and Applications Center (SEDAC)*. Accessed: Mar. 20, 2022. [Online]. Available: https://sedac.ciesin.columbia.edu/

[37] *ADS-B Exchange World’s Largest Co-Op of Unfiltered Flight Data*. Accessed: Mar. 20, 2022. [Online]. Available: https://www.adsbexchange.com/data/

[38] *AIS-Ship Tracking of Marine Traffic*. Accessed: Mar. 20, 2022. [Online]. Available: https://www.vesselfinder.com/

[39] S. Varrette, P. Bouvry, H. Cartiaux, and F. Georgatos, “Management of an academic HPC cluster: The UL experience,” in *Proc. Int. Conf. High Perform. Comput. Simul. (HPCS)*. Bologna, Italy, Jul. 2014, pp. 959–967.

[40] *SmartSpace Dataset*. Accessed: Jun. 7, 2023. [Online]. Available: https://fnr-smartspace-project.uni.lu/datasets/

[41] F. G. Ortiz-Gomez, D. Tarchi, R. Martinez, A. Vanelli-Coralli, M. A. Salas-Natera, and S. Landeros-Ayala, “Convolutional neural networks for flexible payload management in VHTS systems,” *IEEE Syst. J.*, vol. 15, no. 3, pp. 4675–4686, Sep. 2021.

[42] N. Skatchkovsky, H. Jang, and O. Simeone, “Spiking neural networks—Part II: Detecting spatio-temporal patterns,” *IEEE Commun. Lett.*, vol. 25, no. 6, pp. 1741–1745, Jun. 2021.

[43] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[44] E. O. Neftci, H. Mostafa, and F. Zenke, “Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks,” *IEEE Signal Process. Mag.*, vol. 36, no. 6, pp. 51–63, Nov. 2019.

[45] *SPAICE Satellite Signal Processing Techniques using a Commercial Off-the-Shelf AI Chipset*. Accessed: May 27, 2023. [Online]. Available: https://connectivity.esa.int/projects/spaice

[46] S. B. Shrestha and G. Orchard, “SLAYER: Spike layer error reassignment in time,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, pp. 1–10, 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/82f2b308c3b01637c607ce05f52a2fed-Paper.pdf

[47] M. Davies. (Sep. 2021). *Intel Advances Neuromorphic With Loihi 2, New Lava Software Framework and New Partners*. Accessed: Jul. 6, 2022. [Online]. Available: https://www.intel.com/content/www/us/en/newsroom/news/intel-unveils-neuromorphic-loihi-2-lava-software.html

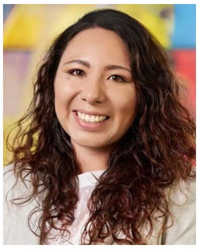
[48] C. Belgaid, A. d’Azmar, and R. Rouvoy. (2023). *pyJoules*. Accessed: Feb. 6, 2023. [Online]. Available: https://github.com/powerapi-ng/pyJoules

[49] M. Davies et al., “Advancing neuromorphic computing with Loihi: A survey of results and outlook,” *Proc. IEEE*, vol. 109, no. 5, pp. 911–934, May 2021.

[50] M. Milojevic, M. Haardt, E. Eberlein, and A. Heuberger, “Channel modeling for multiple satellite broadcasting systems,” *IEEE Trans. Broadcast.*, vol. 55, no. 4, pp. 705–718, Dec. 2009.

[51] F. Vatalaro, G. E. Corazza, C. Caini, and C. Ferrarelli, “Analysis of LEO, MEO, and GEO global mobile satellite systems in the presence of interference and fading,” *IEEE J. Sel. Areas Commun.*, vol. 13, no. 2, pp. 291–300, 1995.

[52] G. Zeng, Y. Zhan, and H. Xie, “Channel allocation for mega LEO satellite constellations in the MEO–LEO networked telemetry system,” *IEEE Internet Things J.*, vol. 10, no. 3, pp. 2545–2556, Feb. 2023.



FLOR ORTIZ (Member, IEEE) received the B.S. degree in telecommunications engineering and the M.S. degree in electrical engineering-telecommunications from Universidad Nacional Autónoma de México (UNAM), Mexico City, Mexico, in 2015 and 2016, respectively, and the Ph.D. degree in telecommunication engineering from Universidad Politecnica de Madrid (UPM), Madrid, Spain, in September 2021. During the Ph.D. studies, she performed a research period with the University of Bologna, Bologna, Italy. She started a close collaboration between UPM and the University of Bologna, opening a new research line for both groups on applying machine learning for radio resource management. She joined as a Research Associate with the Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg. Her research interests include implementing cutting-edge machine learning techniques, including continual learning and neuromorphic computing for operations in satellite communications systems.

NICOLAS SKATCHKOVSKY (Member, IEEE) received the B.Eng. and M.Eng. (Hons.) degrees in electrical engineering from CentraleSupélec, Gif-sur-Yvette, France, and the Ph.D. degree in electrical engineering from King's College London, U.K., in 2022, under the supervision of Prof. Osvaldo Simeone. From 2022 to 2023, he was a Post-Doctoral Research Associate with King's College London, where he worked on Bayesian learning and applications of neuromorphic learning to space communications. He is currently a Post-Doctoral Research Associate with the Francis Crick Institute, working on the modeling of attention in mammalian brains using machine learning techniques. His current research interests include applications of AI to neurosciences under an information-theoretic perspective.

EVA LAGUNAS (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in telecommunications engineering from the Polytechnic University of Catalonia (UPC), Barcelona, Spain, in 2010 and 2014, respectively. From 2009 to 2013, she was a Research Assistant with the Department of Signal Theory and Communications, UPC. In 2009, she was a Guest Research Assistant with the Department of Information Engineering, University of Pisa, Pisa, Italy. From November 2011 to May 2012, she was a Visiting Researcher with the Center for Advanced Communications, Villanova University, PA, USA. In 2014, she joined the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, where she is currently a Research Scientist. Her research interests include radio resource management and general wireless network optimization.

WALLACE A. MARTINS (Senior Member, IEEE) received the degree in electronics engineering and the M.Sc. and D.Sc. degrees in electrical engineering from the Federal University of Rio de Janeiro (UFRJ), Brazil, in 2007, 2009, and 2011, respectively. From 2013 to 2022, he was an Associate Professor with UFRJ. From 2019 to 2023, he was a Researcher with the Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg, Luxembourg. He is currently a Full Professor with the Institut Supérieur de l'Aéronautique et de lEspace (ISAE-SUPAERO), Université de Toulouse, France. His research interests include digital signal processing and telecommunications, focusing on future wireless and satellite networks. He was a recipient of several awards, such as the Best Student Paper Award from EURASIP at EUSIPCO-2009 and the 2011 Best Brazilian D.Sc. Dissertation Award from Capes. He is an Associate Editor of the editorial boards of the IEEE Signal Processing Letters and the *EURASIP Journal on Advances in Signal Processing*.

GEOFFREY EAPPEN (Member, IEEE) received the Ph.D. degree from the Vellore Institute of Technology (VIT), Vellore, and Brunel University London. He was a Senior Research Fellow with the Council of Scientific and Industrial Research India, Department of Wireless Communication (SENSE), VIT. He is currently a Research Associate with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg. His research interests include cognitive radio networks, spectrum sensing methodologies, metaheuristic optimization schemes, artificial neural networks, 5G beamforming, and satellite communication. He was a recipient of the U.K. Commonwealth Fellowship.

SAED DAOUD (Member, IEEE) received the M.Sc. degree in electrical engineering with a specialization in wireless communication from the Jordan University of Science and Technology, Jordan, in 2010, and the Ph.D. degree in electrical engineering with a specialization in wireless communication from Concordia University, Montreal, QC, Canada, in 2015. He was a Post-Doctoral Researcher with École Polytechnique de Montréal. He is currently a Research Associate with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg. His research interests include physical-layer transceiver design and performance evaluation for underwater acoustic (UWA), and terrestrial and satellite communication systems, for technologies, such as OFDM, SC-FDE, cognitive radio, spread spectrum, MIMO, and precoding.

OSVALDO SIMEONE (Fellow, IEEE) received the M.Sc. (Hons.) and Ph.D. degrees in information engineering from Politecnico di Milano, Milan, Italy, in 2001 and 2005, respectively. From 2006 to 2017, he was a Faculty Member with the Electrical and Computer Engineering (ECE) Department, New Jersey Institute of Technology (NJIT), where he was affiliated with the Center for Wireless Information Processing (CWiP). He is currently a Professor of information engineering with the Centre for Telecommunications Research, Department of Engineering, King's College London, where he directs the King's Communications, Learning and Information Processing Laboratory. His research has been supported by the U.S. NSF, the ERC, the Vienna Science and Technology Fund, as well as by a number of industrial collaborations. He is the coauthor of two monographs, two edited books published by Cambridge University Press, and more than 100 research journal articles. His research interests include information theory, machine learning, wireless communications, and neuromorphic computing. He is a fellow of the IET. He was a co-recipient of the 2019 IEEE Communication Society Best Tutorial Paper Award, the 2018 IEEE Signal Processing Best Paper Award, the 2017 JCN Best Paper Award, the 2015 IEEE Communication Society Best Tutorial Paper Award, and the Best Paper Awards of IEEE SPAWC 2007 and IEEE WRECOM 2007. He was awarded a Consolidator Grant by the European Research Council (ERC) in 2016. He currently serves on the editorial board of the *IEEE Signal Processing Magazine*. He is the Vice-Chair of the Signal Processing for Communications and Networking Technical Committee of the IEEE Signal Processing Society. He was a Distinguished Lecturer of the IEEE Information Theory Society in 2017 and 2018. He is currently a Distinguished Lecturer of the IEEE Communications Society.

BIPIN RAJENDRAN received the B.Tech. degree from IIT Kharagpur, India, in 2000, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, U.K., in 2003 and 2006, respectively. From 2006 to 2012, he was a Master Inventor and a Research Staff Member with the IBM Thomas J. Watson Research Center, USA. He held faculty positions in India and the USA. He is currently a Professor of intelligent computing systems and an EPSRC Fellow with Kings College London (KCL). His research interests include building algorithms, devices, and systems for brain-inspired computing. He has coauthored more than 90 papers in peer-reviewed journals and conferences, one monograph, one edited book, and 59 issued U.S. patents.

SYMEON CHATZINOTAS (Fellow, IEEE) received the M.Eng. degree in telecommunications from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2003, and the M.Sc. and Ph.D. degrees in electronic engineering from the University of Surrey, Guildford, U.K., in 2006 and 2009, respectively.

He is currently a Full Professor/Chief Scientist I and the Head of the SIGCOM Research Group, Interdisciplinary Centre for Security, Reliability, and Trust, University of Luxembourg. He is also an Adjunct Professor with the Department of Electronic Systems, Norwegian University of Science and Technology, and a Collaborating Scholar with the Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos." In the past, he was a Visiting Professor with the University of Parma, Italy, and contributed in numerous research and development projects for the Institute of Telematics and Informatics, Center of Research and Technology Hellas and Mobile Communications Research Group, Center of Communication Systems Research, University of Surrey. He has authored more than 700 technical papers in refereed international journals, conferences, and scientific books.

Dr. Chatzinotas received numerous awards and recognitions, including the IEEE Fellowship and an IEEE Distinguished Contributions Award. He is currently on the editorial board of IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE OPEN JOURNAL OF VEHICULAR TECHNOLOGY, and the *International Journal of Satellite Communications and Networking*.