Trustworthy artificial intelligence classification-based equivalent bandwidth control

Journal Pre-proof

Trustworthy artificial intelligence classification-based equivalent bandwidth control

Sara Narteni, Marco Muselli, Fabrizio Dabbene, Maurizio Mongelli

PII: S0140-3664(23)00232-3
DOI: https://doi.org/10.1016/j.comcom.2023.07.005
Reference: COMCOM 7542

To appear in: *Computer Communications*

Received date : 11 November 2022
Revised date : 12 June 2023
Accepted date : 4 July 2023

Please cite this article as: S. Narteni, M. Muselli, F. Dabbene et al., Trustworthy artificial intelligence classification-based equivalent bandwidth control, *Computer Communications* (2023), doi: https://doi.org/10.1016/j.comcom.2023.07.005.

Highlights

Trustworthy Artificial Intelligence Classification-based Equivalent Bandwidth Control

Sara Narteni, Marco Muselli, Fabrizio Dabbene, Maurizio Mongelli

- Data-driven equivalent bandwidth allocation based on the combination of trustworthy artificial intelligence and control
- Model selection guided through Clopper-Pearson generalization bounds helps identifying the minimum training duration required
- Robustness analysis via out-of-distribution detection based on rules satisfaction rates

Trustworthy Artificial Intelligence Classification-based Equivalent Bandwidth Control

Sara Narteni[a,b,*], Marco Muselli[a,c], Fabrizio Dabbene[a], Maurizio Mongelli[a]

[a] *CNR-IEIIT, Corso F.M. Perrone 24, Genoa, 16152, Italy*
[b] *Politecnico di Torino - DAUIN Department, Italy*
[c] *Rulex Innovations Labs, Via Felice Romani 9, Genoa, 16122, Italy*

Abstract

Nowadays, machine learning (ML) is a viable solution for the allocation of equivalent bandwidth (EqB) in telecommunication networks, i.e. the minimum service rate required by a traffic buffer to guarantee a satisfactory Quality of Service (QoS). Moreover, trustworthy artificial intelligence (AI) is gaining importance in regulating the implementation of ML models, requiring explainable AI (XAI) and uncertainty handling.
The paper extends prior works on the combined usage of control and rule-based classification for the EqB allocation, by adding the perspective of trustworthy AI. Simulation-based data collection is performed under a large setting of traffic conditions. Clopper-Pearson generalization bound is used as an efficient tool to select a rule-based model with adequate performance, also determining the minimum amount of data required for model training, which resulted in 3000 samples (≈ 3.3 hours of simulation). Also, robustness in terms of the model's capability to recognize out-of-distribution samples is studied, by comparing the different rates of satisfaction of rules in presence of training or operational data, which is quantified via mutual information, ℓ_1 and ℓ_2 norms. Results show that, while norms are more likely to capture the difference between training and operational data distribution, regardless its entity, mutual information seems sensitive to the entity of the separation between the training and the operational domains.

Keywords: Equivalent bandwidth, machine learning, measurement-based control, rule generation, trustworthy AI.

[*] *Corresponding Author*
Email Address: sara.narteni@ieiit.cnr.it (Sara Narteni)

1. Introduction

Machine learning (ML) classification methods are widely used to evaluate the impact of system variables when mathematical models can be barely derived. The consequent knowledge acquisition can guide the adaptive control of the system performance metrics.

In this context, bandwidth allocation constitutes an important scenario where ML and control can intersect very well. Indeed, it is a widely treated subject in telecommunications research since the end of the 1980s/beginning of the 1990s, with the scientific works about the Asynchronous Transfer Mode (ATM) technology [25]. Most of the schemes for bandwidth allocation are based on the concept of *Equivalent Bandwidth (EqB)*, which is defined as the minimum service rate to be provided to a traffic buffer to guarantee a certain degree of Quality of Service (QoS), in terms of objective parameters (e.g., packet loss, delay, jitter). Traditional model-based solutions have been proposed in the past for the bandwidth allocation problem, being based on closed-form expressions of the QoS metric of interest, thus assuming some a-priori knowledge of the system (of traffic sources, in particular) and mapping that knowledge into the parameters of the model. The complexity and the heterogeneity of the overall input flow process to a traffic buffer make these strategies hardly applicable.

In contrast, the variability of the bandwidth requirements and the complexity of the problem recommend automatic and dynamic bandwidth management through proper control schemes, which do not use a-priori information about traffic flows and buffers, nor closed-form expressions of the involved quantities. In this perspective, machine learning can prove a powerful data-driven solution, as the only hypothesis is the presence of a significant amount of data about the behaviour of the system. In particular, new generation 5G/6G communication systems can benefit of the more and more advanced Artificial Intelligence (AI) technologies, such as those coming from the deep learning world [47]. Nevertheless, another fundamental paradigm in today's AI research is referred to as *Trustworthy AI* [48] and aims at regulating the deployment of AI models in real-world contexts. For example, trustworthy AI validates if the amount of data required for ML model training is "good" enough to come up with an accurate model and if the model is able to infer out-of-distribution (OoD) conditions. These two points are of

interest if one wants to deploy the machine learning model at run time with at least acceptable performance. The problem is even more critical when the QoS should not degrade below the thresholds specified by contracts between service providers and their customers.

In this paper, our prior work on the topic [32] is extended by integrating trustworthy AI in a rule-based ML solution for EqB allocation, which combines machine learning and control. In particular, the effort is towards addressing one of the main pillars of trustworthy AI [48], i.e., technical robustness and safety. This involves choosing a model that is able to generalize well on new data and is robust to out-of-distribution data. Generalization helps to understand the minimum amount of packets that should be extracted from the network in order to feed machine learning properly. Robustness helps to measure the quality of the model when working on the network at run time and to understand if the ML model should be re-trained in front of new data.

The rest of the paper is organized as follows. Section 2 presents and compares previous literature works on the topic. Section 3 introduces the problem of EqB allocation and defines the inherent information vectors, while Section 4 describes the adopted allocation control strategies. Section 5 is the core of the paper, explaining the methodologies to integrate trustworthy AI in the mentioned control schemes. Section 6 reports the experiments carried out and discusses the obtained results. Finally, Section 7 concludes the paper and discusses future works on the topic.

2. Related Work

Equivalent bandwidth allocation is deeply studied in scientific literature. Two broad categories of approaches can be identified: more traditional model-based techniques, which leverage on mathematical modelling and closed-form expressions of the traffic features, and more recent data-driven methodologies, that only rely on the measurements that can be collected from the system.

Focusing on the model-based perspective, Georgoulas et al. [4] optimized the Call Admission Control when aggregating QoS-heterogeneous flows in a DiffServ network. Kumwilaisak et al. [26] outlined a cross-layer architecture for multiplexing video flows over a wireless link; it derived closed-form transmission rate requirements of aggregated streams under loss constraints when a priority-based scheduler is applied at the wireless link. Both the techniques

presented in [4] and [26] exploited traditional EqB. In Cheng and Zhuang's work [27], they addressed an original scheduling technique for multiplexing voice and video services over UMTS (Universal Mobile Telephone System). Kim and Krunz [28] dealt with a control scheme of the blocking probability of voice and data calls, multiplexed over a broadband wireless trunk. Another work [29] derived closed-form expressions of loss and delay when multiplexing WiMAX (Worldwide Interoperability for Microwave Access) services at a subscriber station; the inherent model was suitable for the offline planning of the network. Park [30] developed PID regulation of bandwidth requests for WiMAX service classes; the control target was the average queue length, thus implicitly influencing delay and jitter performance.

However, traditional EqB closed-form expressions can only be applied under statistically homogeneous traffic trunks and some QoS constraints (e.g., loss) [4]. Approaches combining model-based and data-driven techniques were then developed. For example, adaptive bandwidth allocation method based on the Gaussian process regression was studied in [39], where known versus unknown network parameter settings were considered. Traffic prediction was again the focus in [40] for cellular networks, with emphasis on the impact of error in trajectory and traffic load estimators in closed-forms and by a deep neural network. Even though hybrid solutions, the two latter works still privileged the model-based component.

When dealing with heterogeneous traffic conditions, data-driven techniques that rely on numerical approximation are required. Some interesting examples of this kind are the approaches driven by *Perturbation Analysis* (PA). In online gradient descent driven by PA [3], an accurate tuning of the gradient step-size is needed as a trade-off between fast reactions to traffic changes and convergence [6]. Other offline functional optimization or reinforcement learning (RL) approaches [16, 14] may overcome this drawback, but they may suffer of numerical instability in virtue of the corresponding regression problems they have to solve.

In this perspective, the main advantage of the Incremental Control (IC) approach used in our paper relies on the adoption of a classification problem for determining the correct action to be performed at any instant. In fact, the amount of information needed to solve a classification problem is considerably lower than that required in a regression problem [7]. Since the early works in neuro-dynamic programming for resource allocation in telecommunication networks (see, e.g., [49]), considerations emerged in favor of linear or quadratic approximating structures, in place of (shallow) neural

networks, as they guarantee both easier model tuning and suboptimal performance. Though neural networks can approximate arbitrary functions, their nonlinear input-output dependence typically leads to time consuming and unreliable performance (subsection 3.1 of [49]). When model-based, neural approximation of equivalent bandwidth may reveal to be more reliable than via pure data-driven approaches [50, 21].

The recent development of 5G technologies has also given rise to a strong push towards the application of machine learning methods [24] for the data-driven allocation of resources on the network, whose main focus lies precisely on EqB [23]. In this perspective, the research proposed in [22] introduced a Deep Reinforcement Learning (DRL) method to execute power control between primary and secondary user in the underlay access mode for a cognitive radio network. Another work [21] introduced neural network service level PID (NN-SPID) algorithm, specifically designed for the online tuning of a PID controller to ensure the QoS bandwidth requirements through the integration of a neural network, which proved to be more robust compared to genetic algorithms-based approach. Even though all the mentioned research made use of the abstractions (inflow/outflow processes, buffer) and mathematical quantities used in this paper, they did not explicitly address the trustworthy AI paradigm.

Though still beyond the trustworthy AI scope, other recent approaches are relevant to data-driven bandwidth adaptation. In human-to-machine communications, Ruan et al. [36] dealt with prediction of on-off bursts arriving at optical/radio access points through neural networks. Another work by Ruan at al. [37] gave an overview of machine learning models (kNN, SVM, logistic regression, and neural network) to the same problem, leaving the door open for further improvements in intelligent bandwidth allocation decisions. A deep learning solution was posed in [38] for Optical Access Networks to predict the user demands independent of the traffic arrival distribution. Deep reinforcement learning of [41] addressed resource allocation for network virtualization environments. Network traffic load for SDN-based peer-to-peer (p2p) networks was addressed by [42] for p2p categorization through random forest. Particle swarm optimization for network connectivity along with probabilistic neural network for bandwidth adaptation were studied in [43]. Moreover, K-means clustering for network connectivity and classification and regression tree (CART) for bandwidth allocation were studied in [44]; the performance evaluation included a skived and biased dataset. At the best of

the authors' knowledge, this is the only work including both XAI and model sensitivity to distorted datasets. In the robustness perspective, the present paper extends the sensitivity analysis in order to infer out-of-distribution data.

Table 1: Summary of related works, with respect to key topics. Lower-case 'x' means that the topic is partially matched, capital 'X' denotes full matching.

	model-based	**data-driven**	**deep structure**	**explainability**	**robustness**
Georgoulas et al. [4]	X				
Kumwilaisak et al. [26]	X				
Cheng and Zhuang [27]	X				
Kim and Krunz [28]	X				
Grossglauser and Tse [29]	X				
Park [30]		X			
Tang et al. [24]		X	X		
Mei et al. [23]	x	x			
Zhang et al. [22]		X	X		
Merayo et al. [21]		X			
Bonati et al. [16]		X	X		
Bruschi et al. [14]		X			
Ruan et al. [36]		X			
Ruan et al. [37]		X			
Hatem et al. [38]		X	X		
Kim and Hwang [39]	X	x			
Guo and Yang [40]	X	x	X		
Zhang et al. [41]		X	X		
Aldabbas [42]		X			
Navin Dhinnhesh et al. [43]	x	X			
Kori and Kakkasageri [44]				X	x
This paper		X		X	X

Table 1 summarizes the literature review presented in this Section and the positioning of our paper with respect to five key topics (model-based, data-driven, deep structure, explainability, and robustness). The first two terms refer to the main categorization of the equivalent bandwidth allocation problem, between traditional model-based techniques and more recent data-driven methodologies. The latter often involve the adoption of black-box deep learning models, instead of explainable AI methods shading light on the logic of the model. For this reason, we decided to add "deep structure" and "explainability" as other two key comparison criteria. Finally, "robustness" was chosen as it is another important pillar of this paper, dealing with

an evaluation of the model performance at run time (i.e., "in operation"), when fed with data different from those used during model training. The following relevant considerations emerged from the analysis of the table. Before the widespread adoption of AI, the largest part of the algorithms were based on closed-form expressions of the QoS. The intrinsic flexibility of data-driven approaches has recently attracted interest on machine learning. The management of 5G networks pushes pressure in this direction as the technologies for network monitoring and computing are now mature to offer support to machine learning infrastructures. Deep learning is one of the most adopted solutions in virtue of its precision and automatic feature extraction from raw data. However, due to their black-box complexity, deep models may be hardly explainable. Extracted features, such as, for example, filters in the hidden layers of a convolutional neural network, may not be interpretable by system developers and users. Studies on verifiable performance are of much interest as well (see, e.g.,[45, 46]). Trustworthy AI is thus becoming crucial to let the working conditions compliant with safe and reliable network autonomy. For this reason, contrary to all the mentioned literature, our approach tackles the data-driven computation of the exact EqB in heterogeneous conditions, with no a-priori information about traffic and closed-form expressions, but rather relying on explainability, generalization bounds and robustness of the machine learning solution, thus complying with the trustworthy AI paradigm.

3. The problem

Let us consider the problem of service rate allocation for a finite traffic buffer. The buffer is characterized by a stochastic input process, $\alpha(t)$ (with no specific assumptions for it), and a service rate $\theta(t)$. The overall optimization objective is to find $\theta^*(t)$ that minimizes the following functional cost:

$$\int_0^T \Delta(l(\theta), l^*(\theta))\, dt, \tag{1}$$

where $l(\theta) = l(\alpha(t), \theta(t))$ is the chosen performance index measured at time t; $l^*(\theta) = l(\alpha(t), \theta^*(t))$ is the performance target at the optimal service rate $\theta^*(t)$ and $\Delta(\cdot)$ is a distance function between $l(\theta)$ and the target $l^*(\theta)$, e.g., $\Delta(l(\theta), l^*(\theta^*)) = (l(\theta) - l^*(\theta))^2$. The explicit dependence of l and l^* from t and $\alpha(t)$ is removed from the notation for the sake of conciseness.

The exact solution $\theta^*(t)$, in continuous time, is unknown and it is approximated in discrete time in the rest of the paper.

Hence, discrete reallocations $\theta(k), k = 1, 2, ...$ are considered, defined on the basis of the feedback acquired during the system evolution. The feedback law $f(\cdot)$ (will be introduced in next Section 4), decides the reallocation at time step k, $\theta(k) = f(\theta(k-1), I(k))$, as a function of an information vector $I(k)$ collecting observations of some parameters of interest, acquired during the system evolution up to instant k. In detail, the components of $I(k)$ are related to the performance of the buffer. In our setting, $I(k)$ assumes the following form:

$$I(k) = [l(k), N(k), Bp(k), \overline{\tau}(k), \overline{\phi}(k), m(k), \sigma(k), B(k), B_{Max}(k)] \quad (2)$$

where l is the measured performance (averaged over the $[k-1, k]$ horizon), N is the number of active traffic sources giving origin to α; Bp, $\overline{\tau}$ and $\overline{\phi}$ are the peak bandwidth, the average burst size and the average silence duration of the sources, respectively; m and σ are the average and standard deviation of α; finally, B and B_{Max} are the current and maximum buffer size, respectively. The presence of $\overline{\tau}$ and $\overline{\phi}$ constitutes a certainty equivalent assumption concerning the presence of on-off traffic sources. The problem is thus to find the optimal sequence $\theta^*(k)$ of bandwidth reallocations over consecutive discrete time instants $k = 1, 2, ...$

The solution to this problem will be searched either using the above-introduced information vector as a whole and also in a *reduced* form, where only measurements available at runtime are kept. Therefore, two possible settings will be considered:

1. $I_{all}(k) = I(k)$

2. $I_{reduced}(k) = [l(k), m(k), \sigma(k), B(k), B_{Max}(k)]$

In next Sections 4 and 5, we will refer to a generic information vector $I(k)$, since the development of the proposed methodologies is independent of the specific kind of features considered (either $I_{all}(k)$ or $I_{reduced}(k)$).

4. Classification and Control

The problem described in Section 3 follows a feedback law expressed by the following Equation, which defines a control strategy called *incremental*

control (IC)[1]:

$$\theta(k) = f(\theta(k-1), I(k)), \quad k = 1, 2, ... \tag{3}$$

$$f(\theta(k-1), I(k)) \doteq (1 + \beta \cdot r(I(k))) \cdot \theta(k-1) \tag{4}$$

β is a pre-determined quantity (e.g., of 5%), $I(k)$ is the information vector, and $r(I(k))$ is a function, called r-mapping, assuming values in the set $\mathcal{R} \doteq \{-1, 0, +1\}$. This function defines the best allocation strategy to be performed at each step k: more precisely, if $r(I(k)) = -1$ the value of $\theta(k)$ will be decreased of $\beta \cdot \theta(k-1)$ with respect to $\theta(k-1)$; if $r(I(k)) = +1$, θ will be increased of the same quantity, whereas it will remain unchanged when $r(I(k)) = 0$.

In our approach, the r-mapping is determined as the solution of a machine learning classification task. A training set $D = \{(I^\kappa, r^\kappa), \kappa = 1, ..., \mathcal{K}\}$ is built as a set of labelled realizations I^κ of the information vector, where labels r^κ are chosen as for providing the best change in the considered performance θ^*. The classification tasks are addressed here under the Trustworthy AI paradigm, by relying on an eXplainable classification model and investigating its generalization and robustness, as detailed in next Section 5.

For performance comparison with IC, another approach is considered, i.e., the Reference Chaser Bandwidth Controller (RCBC, see Section 4.1 for further details).

4.1. Reference Chaser Bandwidth Controller

The technique called *Reference Chaser Bandwidth Controller* (RCBC) [3] consists of a gradient descent over θ, whose gradient is approximated through perturbation analysis (PA) [52]. Formally, it is defined through the following Equation 5:

$$\theta(k) = \theta(k-1) - \eta_k \cdot \left. \frac{\partial \Delta(l(\theta), l^*(\theta))}{\partial \theta} \right|_{\theta(k-1)}, \quad k = 1, 2, \ldots, \tag{5}$$

where $\theta(k)$ is the service rate at current time step k, $\theta(k-1)$ the service rate at the previous time step $k-1$, $\eta_k > 0$ is the gradient descent step-size to be found empirically to optimize the performance and $\Delta(\cdot)$ is the same function defined in Section 3. PA gives analytical instruments to derive an estimator

[1] As previously argued in Sec. 3, $I_{all}(k)$ and $I_{reduced}(k)$ settings are devised: accordingly, the IC strategy will lead to IC_{all} and $\text{IC}_{reduced}$ scenarios.

of the gradient as function of the lengths of the measured congestion periods of a traffic buffer. The sign of the estimated gradient then determines the direction of the bandwidth allocation, i.e., an increase if the gradient is negative and a decrease if it is positive. The estimator is obtained by analyzing the system as a Stochastic Fluid Model, but the associated values are based on real data. For this reason the technique is taken as a reference for performance comparison. It maintains the data-driven structure to the solution of the problem, though performing better than traditional measurement-based equivalent bandwidth [3] or proportional–integral–derivative (PID) controller [6]. The setting of η_k here follows [6]. RCBC$_\nu$ defines the adoption of the Vogl method (whose tunable parameter is ν) to let the step size be adaptive to the current value of the functional cost: $\eta_k = \nu \cdot |l(k) - l^*(k)|$.

5. Trustworthy AI

Trustworthy Artificial Intelligence is a recent concept [48] that can be used as an umbrella-term to indicate the requirements of modern AI systems for their real-world deployment. Among its components, explainability (or transparency) and uncertainty handling have an essential role. The intrinsic statistical error introduced by any machine learning algorithm may lead to criticism from safety and security engineers. In this respect, the learning assurance guidance developed in [15] places specific emphasis on the notion of generalization guarantees for ML models, as a fundamental path that requires further investigation, which is also the subject of statistical learning theory. Another essential aspect is the capability of a ML model to be sufficiently robust to perform as intended on unseen operational data.

In this paper, explainability is addressed via the adoption of a rule-based eXplainable AI model (the Logic Learning Machine, whose fundamentals are described in Section 5.1) and uncertainty handling via evaluation of its generalization (Section 5.2) and robustness capabilities (Section 5.3).

5.1. Logic Learning Machine

Logic Learning Machine [2], or LLM for short, is an efficient and more accurate evolution of the Switching Neural Networks (SNN) [9]. The LLM classifier comes in form of sets of intelligible rules of the *if-then* type, that are

[2] Available through the Rulex platform: `www.rulex.ai`

used here to understand the conditions involved in a specific decision about the control action to be performed. Hereafter, a brief introduction to the LLM model foundations is provided.

The LLM rules generation process starts by discretizing the feature space and binarizing it using the inverse-only-one coding criterion. The resulting binary strings are then concatenated into a single large string representing the considered sample. Consequently, shadow clustering is adopted to build logical structures, called implicants, which are finally transformed into simple conditions and combined into a collection of intelligible rules [10]. The peculiarity of rules constructed in this way is the fact that they are not disjoint, so more than one of them may describe the same input point, i.e. resulting in overlapped rules. As confirmed by previous studies [11], the LLM thus shows a better generalization ability with respect to that provided by other rule generation approaches, such as decision trees (DT).
Mathematically, consider an example space $\mathcal{X} \times \mathcal{Y} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{D}$, with $\mathbf{x}_i \in \mathbb{R}^{N_f}$, $y_i \in \{0, \ldots, C-1\}$, where N_f is the number of features and C the number of output classes. Using the example space, the LLM model learns a function $g : \mathcal{X} \longrightarrow \mathcal{Y}$, which can be described by a set of intelligible rules $\mathcal{R}_S = \{r_k\}_{k=1}^{N_r}$, each expressed in the **if** <*premise*> **then** <*consequence*> form. The <*premise*> constitutes the antecedent of the rule and is a logical conjunction (AND) of conditions on the input features in $\mathbf{x}$. The <*consequence*> states the output class label $\hat{y} \in \{0, \ldots, C-1\}$ predicted by the rule.

Given a generic ordinal feature X_j composing $\mathbf{x}$, a condition can be expressed as one of the following intervals: $X_j > s, \quad X_j \leq t, \quad s < X_j \leq t$, with $s,\ t$ being proper numerical thresholds. The premise of any rule r_k is the logical product of d_k conditions $c_{l_k}, l_k = 1_k, \ldots, d_k$, described by one of such forms. As for any classifier, a confusion matrix can be built for evaluating each single rule. The matrix is made up of four indices: $TP(r_k)$ and $FP(r_k)$, defined as the number of instances $(\mathbf{x}_i, y_i)$ that correctly and wrongly satisfy rule r_k, being $\hat{y} = y_i$ and $\hat{y} \neq y_i$ respectively; $TN(r_k)$ and $FN(r_k)$, defined as the number of samples $(\mathbf{x}_i, y_i)$ which do not meet at least one condition in rule r_k, with $\hat{y} \neq y_i$ and $\hat{y} = y_i$, respectively. Starting from these definitions, covering $C(r_k)$ and error $E(r_k)$ can be adopted to evaluate each single rule generated by the LLM:

$$C(r_k) = \frac{TP(r_k)}{TP(r_k) + FN(r_k)} \tag{6}$$

$$E(r_k) = \frac{FP(r_k)}{TN(r_k) + FP(r_k)} \tag{7}$$

The covering can also be interpreted as a measure of relevance for rule r_k; indeed the larger is the covering, the higher is the generality of the corresponding rule. The error $E(r_k)$ reflects how many points wrongly satisfy the rule, and its maximum value can be set before training, as a model parameter.

Feature ranking (FR) refers to the increasing ordering of the feature attributes by their relevance in determining a given class label, thus providing insights on which of them contributed the most to the rule generation. The relevance computing for a generic attribute X_j for a class $\hat{y}$ starts by finding the relevance of single conditions c_{l_k} referring to that variable. This value is obtained by considering rule r_k with and without the occurrence of condition c_{l_k} (let us denote as r'_k the rule obtained in the latter case). Since the premise of r'_k becomes less stringent, it holds that $E(r'_k) \geq E(r_k)$; hence, the relevance of the condition can be measured as the difference in the two errors, i.e., $R(c_{l_k}) = (E(r'_k) - E(r_k))C(r_k)$. As anticipated, each condition c_{l_k} refers to a specific attribute X_j, and it is satisfied by some specific values ν_j. An overall measure of relevance $R_j^{\hat{y}}$ for feature X_j is then derived by the following equation 8:

$$R_j^{\hat{y}} = 1 - \prod_k \left(1 - R\left(c_{l_k}\right)\right), \tag{8}$$

where the product is computed on every rule r_k that includes a condition c_{l_k} on the attribute of interest X_j.

5.2. Generalization

Generalization refers to obtaining 'generalization bounds' or measuring the 'generalization gap', that is the difference between the performance observed during training and the ideal one, which is related to the perfect knowledge of the underlying probability distribution of the data [12]. Given a measurable input space $\mathcal{I}$, a label space $\mathcal{R}$, a function h (known as *hypothesis* in learning theory) predicting $r \in \mathcal{R}$ for arbitrary information vector $I \in \mathcal{I}$ (especially for those not contained in the training data), a loss function $\mathcal{L}$ that measures how far $h(I)$ is from the respective r and a probability measure $\mathcal{P}$ over $\mathcal{I} \times \mathcal{R}$, then the risk $R(h)$ on h (also known as out-of-sample

error or generalization error) is[3]:

$$R(h) = \int_{\mathcal{I}\times\mathcal{R}} \mathcal{L}(r, h(I)) d\mathcal{P}(I, r). \tag{9}$$

The seminal work by Vapnik and Chervonenkis [13] established a relationship between the generalization capability of a learning algorithm and its hypothesis space complexity. Various forms of generalization bounds have been derived since then. It appears that, in the current state of knowledge, the values of the generalization upper bounds obtained for large models (such as neural networks) are often too high, unless an unreasonable amount of training data is available. Provided to consider the 0-1 loss function for classifiers, the Clopper-Pearson bound is used here as it does not need a measure of the hypothesis space complexity, which is not available for a rule-based system like the one adopted in this work. Though considered very conservative, it stills gives crucial information about the generalization bound, as soon as training and test sets can be built with appropriate sizes. This is exactly the way followed in this paper, in which the Clopper-Pearson bound drives the search for the model with smallest complexity (lowest number of rules or with less conditions) and with low bias (as measured by the bound itself).

Having a test set T with m samples different from the training set, with probability at least $1 - \delta$ (with $\delta \in [0, 1]$) over an i.i.d. draw of a test set $T \in (\mathcal{I} \times \mathcal{R})^m$, the Clopper-Pearson bound is:

$$R(h) \leq \hat{R}_T(h) + \sqrt{\frac{\ln \frac{1}{\delta}}{2m}}, \tag{10}$$

with $\hat{R}_T(h)$ being the empirical test error:

$$\hat{R}_T(h) = \frac{1}{m} \sum_{(I,r)\in T} (\mathcal{L}(r, h(I))). \tag{11}$$

5.3. Robustness

5.3.1. Why robustness

Despite generalization tests may give a clear indication about the reliability of the model over a parameters domain, the model could be still

[3]Due to space limitation, the reader is referred to [12] for details on notation

susceptible to errors when working outside of that domain. And this is when robustness comes into play. Robustness test refers to a validation set, different from the training set. Such a set corresponds to the production stage (i.e., once the machine learning model is deployed at run time on the "production line", without further re-training), under the assumption that the (unknown) probability distribution generating the data is the same at training and production stages. The hypothesis may be reasonable or not, depending on the specific application scenario. For this reason, model robustness refers to perturbations in the design phase due to fluctuations in the training dataset (learning algorithm stability) and perturbations in the operational phase due to fluctuations in the data input and prediction output or in the model itself (model stability). The definition of abnormal range tests in operation is a necessary means to reach this objective. Operational conditions on traffic sources are of main interest in the present work.

5.3.2. How robustness

More specifically, the model is tested over larger ranges of the features in $I(\cdot)$ than the ones used for training. This stresses the operational conditions and one wants to know in advance if specific perturbations may lead to critical conditions (bandwidth underestimation in this case). Testing the inherent error is however not the final goal to pursue. It is even more crucial to understand *when* the operational conditions are different from the ones used in training. This would lead to understand if the samples are *in-distribution* or *out-of-distribution*, namely, if the system is working (at the operational level) under the same probability distribution used in training [15, 20]. Since such probability distribution may be hardly derived from data even at design time, the in- versus out-of- distribution question constitutes one of the most challenging issues in machine learning today.

5.3.3. Robustness via XAI

The problem of out-of-distribution detection is approached by leveraging XAI, proposing a solution that exploits the different satisfaction of the rules when data points are in- or out-of distribution. Indeed, since each sample of the information vector (at both training and production stages) may verify, or not, any rule of the model, just counting how many times each rule is satisfied by the samples may give an insight into the underlying structure of the data (as seen by the model itself).

The approach can be formalized as follows. Suppose that the LLM model has been trained on the available data, resulting in a ruleset made up of N_r rules, each predicting an r value as explained in Sec. 4; therefore, such rules can be adopted to perform several tests for the inference of the proper bandwidth allocation given some input samples.

In particular, let N_{tr} and N_{op} be the number of test repetitions using data samples drawn from the training domain and the operational domain, respectively. Also, let $N_w = N_{tr} + N_{op}$ be the total number of test iterations. For each test iteration, each rule may (or not) be satisfied a certain number of times by the samples [4]. This number is referred to as the *number of hits* for that rule. With these premises, N_w vectors can be defined as follows:

$$\mathbf{w}^j = \{w_i^j\}, \quad i = (1, \ldots, N_r),\ j = (1, \ldots, N_w) = (1, \ldots, N_{tr}, N_{tr}+1, \ldots, N_w) \tag{12}$$

Each vector $\mathbf{w}^j$ can be thought as a histogram representing the number of hits for each rule in a fixed test iteration j. For the sake of clarity, consider $j = (1, \ldots, N_{tr})$ as the index of the N_{tr} tests with training domain data and $j = (N_{tr}+1, \ldots, N_w)$, as the index of the N_{op} ones with operational data.

When training and operational data distributions are different, it is possible to suppose that the number of hits, for each rule, also varies, and the training/operational histograms shape as well. In order to confirm this hypothesis, the adoption of quantitative metrics able to capture such variations is proposed, thus allowing to understand if an out-of-distribution occurs or not.

The following simple statistics are first considered for each $\mathbf{w}^j$: the mean $m(\mathbf{w}^j)$, the variance $\sigma^2(\mathbf{w}^j)$, the skewness $s(\mathbf{w}^j)$ and the kurtosis $\kappa(\mathbf{w}^j)$. Several couples of histograms are then compared, by computing for each one three metrics: the mutual information $\boldsymbol{\mu I}$ between the two histograms, and the ℓ_1 and ℓ_2 vector norms of the difference between the two histograms. Two kinds of comparisons can be made, depending on the interest in finding possible changes in the rules adoption when remaining in the training domain (*baseline*) or when going in operational. In the first case, couples $(\mathbf{w}^{j_1}, \mathbf{w}^{j_2})$ are considered, with $j_1, j_2 \in \{1, \ldots, N_{tr}\}$, $j_1 \neq j_2$. As the chosen metrics are symmetric, results for couple $(\mathbf{w}^{j_1}, \mathbf{w}^{j_2})$ are the same as for $(\mathbf{w}^{j_2}, \mathbf{w}^{j_1})$.

[4] Samples can satisfy multiple rules and there may be samples satisfying none of the rules.

Hence, only the first couple is considered for the computations. The number of the couples from training domain is then:

$$C_{tr} = \binom{N_{tr}}{2} \tag{13}$$

In the operational case, two directions are possible. On the one hand (*approach 1* from now on), "mixed" couples can be considered, by comparing one histogram from training with another one from the operational domain: $(\mathbf{w}^{j_1}, \mathbf{w}^{j_2})$, with $j_1 \in \{1, \ldots, N_{tr}\}$, $j_2 \in \{N_{tr}+1, \ldots, N_w\}$. In this case, the number of couples is:

$$C_{op} = N_{tr} \cdot N_{op} \qquad (approach\ 1) \tag{14}$$

On the other hand (*approach 2* in the following) couples can be formed by selecting histograms within the operational domain only. Similarly to what holds for the training domain, it is possible to individuate:

$$C_{op} = \binom{N_{op}}{2} \qquad (approach\ 2) \tag{15}$$

operational couples $(\mathbf{w}^{j_1}, \mathbf{w}^{j_2})$, with $j_1, j_2 \in \{N_{tr}+1, \ldots, N_w\}$, $j_1 \neq j_2$.
After computing the metrics for all the in-training couples, three vectors are built, of length C_{tr}, denoted with $\boldsymbol{\mu}I_{tr}, \mathbf{l1}_{tr}, \mathbf{l2}_{tr}$, containing the obtained values for the mutual information and ℓ_1, ℓ_2 norms, respectively; thus, the following baseline ranges can be individuated, by taking minimum and maximum values of these vectors:

$$\boldsymbol{\mu}I_{tr}^{base} \doteq [min(\boldsymbol{\mu}I_{tr}), max(\boldsymbol{\mu}I_{tr})] \tag{16}$$

$$\mathbf{l1}_{tr}^{base} \doteq [min(\mathbf{l1}_{tr}), max(\mathbf{l1}_{tr})] \tag{17}$$

$$\mathbf{l2}_{tr}^{base} \doteq [min(\mathbf{l2}_{tr}), max(\mathbf{l2}_{tr})] \tag{18}$$

The final objective is to assess if and how much far are the same metrics when computed in the operational setting, with respect to the training baseline ranges. To this aim, we build three vectors $\boldsymbol{\mu}I_{op}, \mathbf{l1}_{op}, \mathbf{l2}_{op}$, all of length C_{op}, containing the values of the metrics for the operational couples. Similarly to the training domain, operational ranges can be defined as follows:

$$\boldsymbol{\mu}I_{op}^{range} \doteq [min(\boldsymbol{\mu}I_{op}), max(\boldsymbol{\mu}I_{op})] \tag{19}$$

$$\mathbf{l1}_{op}^{range} \doteq [min(\mathbf{l1}_{op}), max(\mathbf{l1}_{op})] \tag{20}$$

$$\mathbf{l2}_{op}^{range} \doteq [min(\mathbf{l2}_{op}), max(\mathbf{l2}_{op})] \tag{21}$$

When the operational data distribution is completely diverse from the training one, a full separation between training and operational ranges can be expected, i.e., $max(x_{op}) < min(x_{tr}) \vee min(x_{op}) < max(x_{tr})$, where x is any of the considered metrics. This occurs if all the operational couples are recognised as OoD by the metric x. However, reality is much more complex. There may be cases in which the operational data distribution partially resembles the training one and this reflects into an only partial separation between the baseline and the operational ranges. Moreover, discordant conclusions may also arise from the three adopted metrics. The final decision is made by following a conservative approach, in which operational data are considered OoD if at least one metric recognises them as such.

5.3.4. On the adoption of basic statistical measures

In previous Section 5.3.3, the adoption of four simple statistics on single histograms $\mathbf{w}^j$ are mentioned. Despite these statistics may seem useless as they tend to compress information too much, thus loosing representativeness of the probability distribution change, they are considered for performance comparison. A more subtle reason also suggests to include them in the analysis. The motivation relies on the lack of information about data variation at operational level.

An example to understand why the mean could recognize a variation between training and operation deals with the case of zero rule hits in operation. If some non zero hits have been registered in training, the mean would experience some change, in proportion to the number of hits. This is actually an extreme case, but it may happen when a strong out-of-distribution arises. The same concept applies to the case when small portions of operational data satisfy the ruleset and large hits have been registered in training. In that case, the mean would change in proportion to the rule hits differences (e.g., with 3 rules, number of hits in training: 10, 20, 30, number of hits in operation: 1, 2, 3). It is however also true that, in less extreme cases, small and large variations of the hits (in training and operation) may induce similar mean values. This is what apparently happens in the results of the paper (for the mean).

6. Performance Evaluation and Discussion

In this Section, the experimentation carried out in this work is described and discussed. The Section starts by describing the system setting and the simulation process that leads to data collection (Sec. 6.1 and 6.2), then it describes the model training with generalization provided by the Clopper-Pearson bound, which is useful to understand when the training process can be stopped and a good model is reached (Sec. 6.3). Results of the control schemes application and some analyses of the inherent rule-based models are presented in Sections 6.4 and 6.5. Finally, robustness tests are presented in Section 6.6: these are of paramount importance, since a QoS violation may take place in case out-of-distribution. In this eventuality, a fallback may be applied, such as immediate bandwidth over-provisioning while re-starting a new training process (with new data collected at run time).

Before going into details, Fig. 1 shows a flowchart of the main experimental steps, thus capturing the overall logic of the approach.

Data and code to reproduce the experiments of this Section are made available at the following link: `https://github.com/saranrt95/TrustEqB`.

6.1. System setting

In order to generate a very large set of working conditions, N on-off traffic sources with variable parameters are considered. The QoS metric of interest is the *packet loss probability* (PLP), whose target value is $l^* = 10^{-2}$.

Just to give an example, as to ITU-T P.59, Voice over IP (VoIP) traffic follows "on" (speech) periods under exponentially distributed durations with average of $\overline{\tau} = 1.008$s; the "off" (silent) periods are of the same type with average $\overline{\phi} = 1.587$s. VoIP peak bandwidth lies in $[5.25, 64]$ kbps as to codec requirements.

The "on" period $\overline{\tau}$ is here set to 1.0s and $\overline{\phi}$ is taken variable in [0, 5]s. The other parameters of the system change as follows: peak bandwidth $Bp \in [5, 50]$ kbps, number of connections $N \in [70, 120]$, buffer size $B_{Max} \in [5, 500]$.

The considered system is implemented via simulations through an ad-hoc C++ simulator, implementing RCBC and IC, to set the service rate of a Digital Bandwidth Broadcasting (DVB) buffer over which IP packets of size 80 bytes are encapsulated. The technology setting is the Broadband Satellite Multimedia (BSM) architecture by the European Telecommunication Standardization Institute (ETSI) BSM standardization group. The architecture

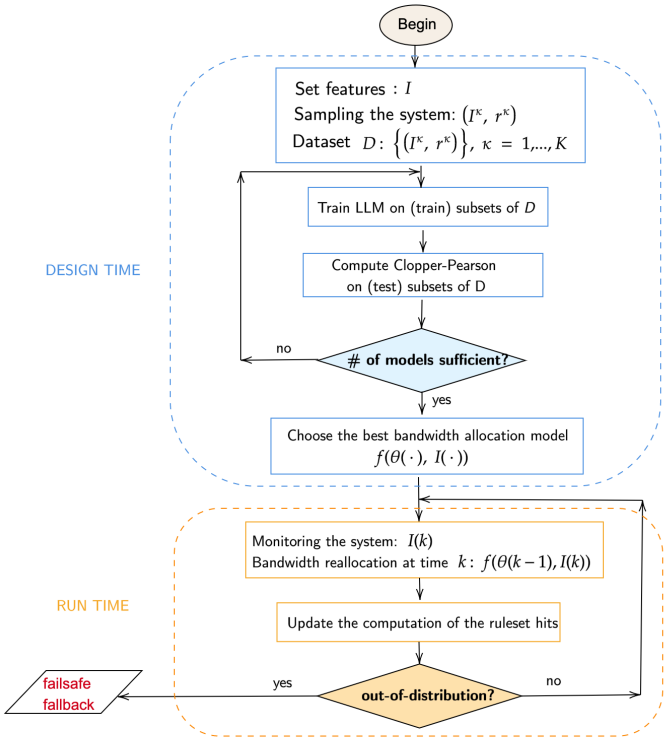


Figure 1: Trustworthy AI-based EqB control flowchart

deals with the interaction between the Satellite Independent (SI) and Satellite Dependent (SD) technology layers. The SD stack should offer a QoS service to the upper layers independent of a Service Level Agreement fixed at the SI layer. That "independence" opens the door to a QoS mapping problem between SI and SD with inherent bandwidth control implications [51]. The simulator used here considers IP protocol for SI and DVB for SD. The first version of the simulator was developed in [51], in which a first version of the RCBC (used in the paper for performance comparison with the new machine learning approach) has been studied.

Packets loss is monitored and PLP is computed over the reference time period. The simulation lasts 15 hours and, every 15 minutes, a new system parameters setting is picked according to a uniform distribution into the

mentioned ranges[5]. Unless otherwise declared, a service rate re-allocation is set every minute according to the control paradigm at hand.

6.2. Data collection

Based on the simulation process and the setting previously described in Section 6.1, a dataset of $\mathcal{K} = 10^4$ samples is built. The Sobol distribution is used to randomly extract a set of working conditions. One training point, i.e. (I^κ, r^κ), corresponds to 4 minutes of simulation. The first minute is the transient period, the subsequent minute drives the information vector collection and the last two minutes collect the losses. Variations under β are fixed and amount to 15%. The *burstiness* $b = (\overline{\tau} + \overline{\phi})/\overline{\tau}$ represents a measure of the regularity of the traffic sources and drives the allocation with respect to $r^\kappa = 0$ through $\theta(k) = \bar{\bar{\theta}} = N \cdot B_p \cdot b$. The collection of PLP

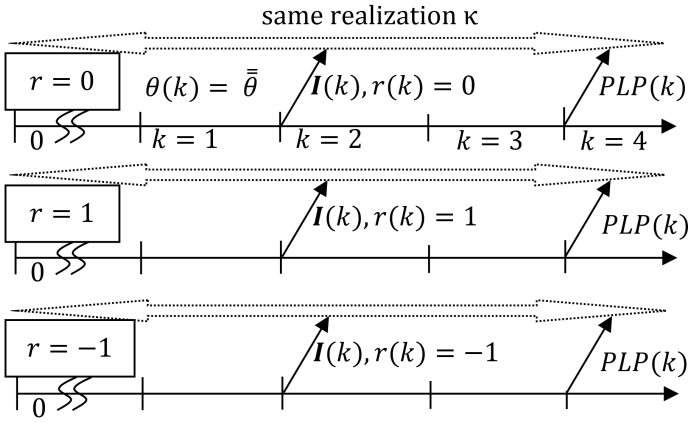


Figure 2: Simulation-based collection scheme for one sample (I^κ, r^κ) of the dataset.

under $r^\kappa = 0$ defines the baseline for database collection (for each sample κ). Bandwidth θ under $r^\kappa = 1$ and $r^\kappa = -1$ then defines two independent data collections under the same realizations of stochastic processes involved in previous simulation under $r^\kappa = 0$. The r^κ corresponding to the simulation run closest to the PLP target is then registered in the database. The overall database collection lasted 25 hours with an Intel Q6600 2.4Ghz CPU. The database collection may be posed in parallel over different computing units

[5]The same realization for each variable is applied under each different usage of control stategies (RCBC or IC).

as only the three simulations above ($r^{\kappa} = \{0, 1, -1\}$) should run together; so, different runs may work over several computing units. Figure 2 provides a visual representation of the simulation runs for $r^{\kappa} = \{0, 1, -1\}$, which helps understand the process of data collection.

The dataset is later posed to the LLM model[6] and control rules are found, as better detailed in next Section.

6.3. LLM model selection via generalization bound

In order to choose an adequate model to perform the bandwidth reallocation task, the LLM generalization capability to unseen test sets was evaluated through the Clopper-Pearson bound [12]. Several training sets were extracted from the dataset (of $\mathcal{K} = 10^4$ samples), with the following sizes: 9000, 5000, 3000, 2500, 2000, 1500, 1000 and 500 samples. Accordingly, the corresponding test sets were defined, with sizes 1000, 5000, 7000, 7500, 8000, 8500, 9000 and 9500, respectively. Then, the LLM was trained on each of the 8 training sets, thus generating 8 sets of rules, by setting the maximum rule error to 5%. In the following, each model is referred to as 'train$S$', where S indicates the training size.

Given a training size, the corresponding test set was considered as a whole and also in its portions of 80%, 60%, 40% and 20%. On each of such test sets, Clopper-Pearson bound was derived through Eq. 10 by setting different values for δ, i.e., $\delta = 0.1$, $\delta = 0.05$, $\delta = 0.02$ and $\delta = 0.01$.

Figure 3 shows the Clopper-Pearson bounds for each ruleset at the different δ values. For training sizes below 5000, the obtained values indicate a good overall generalization of LLM models, being the Clopper-Pearson bound at most 0.09 regardless of the training sets sizes. Conversely, the 'train9000' model led to the worst (highest) values of the bound, which may denote overfitting. With this exception, as expected, the other models better generalize when trained on larger datasets and when setting higher δ values.

6.3.1. Optimal training set size

The generalization analysis gives a preliminary indication about the minimum training set size needed to achieve stable and reliable bounds: specifically, both 'train3000' and 'train5000' resulted in similar values of the bound, lower than the other available rulesets. The choice between the two is based

[6]The Rulex platform is used `www.rulex.ai`

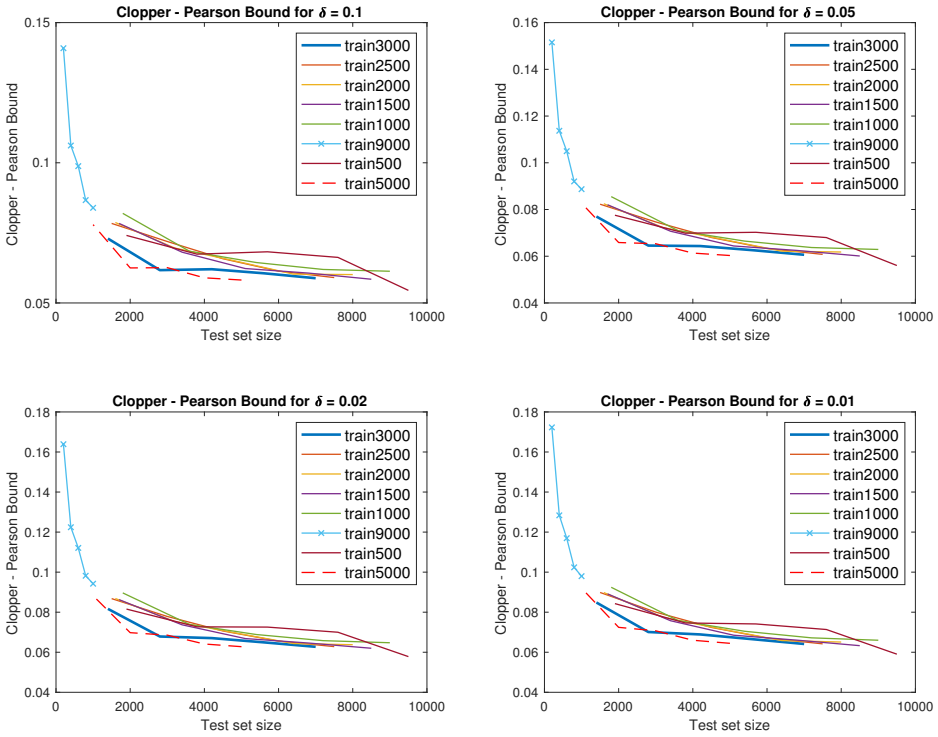


Figure 3: Plots of Clopper-Pearson Bounds obtained for each tested level of probability 1-δ on different test set portions. Fixing a δ value, the different curves correspond to the Clopper-Pearson bound for rulesets generated by the LLM with a different number of training samples (9000, 5000, 3000, 2500, 2000, 1500, 1000 and 500).

on the covering of the rules composing them: 'train3000' is formed by higher-covering rules with respect to 'train5000', thus denoting a better quality of the model. In light of this result, the 'train3000' model is chosen as the reference ruleset for feeding IC and RCBC controllers. Moreover, 3000 samples represents the minimum training set size mentioned before: below that threshold, the curves (shown in Fig. 3) of the generalization gaps show a significant increase. This analysis also constitutes a practical guidance for the design of the algorithm in case of data scarcity: if no significant performance improvements arise from the generalization gap, in correspondence of an increasing training set size, the system may be considered under-sampled

and the usage of any machine learning at risk.

6.3.2. Practical considerations

As 3000 samples correspond to 4*3000 minutes of (virtual) simulation (see subsection 6.2), this represents the minimum monitoring period necessary to stabilize the model. The same concept applies to any monitoring situation of this kind. Typically, the network engineer asks the data analyst which is the minimum monitoring time horizon to assess a stable model, which becomes applicable in subsequent time periods. Generalization analysis is an answer to that question. However, generalization is applicable within the ranges of the parameters considered for the training set. What happens outside those ranges remains questionable and this is where robustness comes into play.

6.4. Bandwidth allocation performance comparison: IC vs RCBC

Following the results of Clopper-Pearson analysis (Sec. 6.3), the most performing LLM model was chosen for the EqB allocation. In this respect, Incremental Control (IC) scheme was compared to another technique, i.e, the *Reference Chaser Bandwidth Controller* (RCBC) described in Section 4.1.

As anticipated, two possible settings of IC were considered: i) IC_{all}, using the complete information vector $I_{all}(k)$ and ii) $\text{IC}_{reduced}$, dealing with the reduced information vector $I_{reduced}(k)$. Besides the basic speed of 1 allocation per minute, two accelerated bandwidth allocations were considered: 2 allocations per minute (denoted with "...*2..." notation) and 5 allocations per minute (referred to as"...*5..."). Regarding RCBC, the notation RCBC_{fs} is used to indicate the RCBC with fixed stepsize s (i.e., $\eta_k = s, \forall k$).

Table 2 deals with all the relevant results for the considered techniques and scenarios. It reports the averages and variances of all the performance metrics involved in the problem. More precisely, $\bar{l}$ and σ_l^2 are the mean and variance of the PLP; $\overline{\theta}$ and σ_θ^2 are the mean and variance of the bandwidth; l^*_{over} expresses the percentage of the periods in which the PLP is larger than the target; finally, the quantity $|\bar{l} - l^*|$ represents the average difference between measured and target loss (which is 10^{-2}). It is clear from the table that RCBC performance is not optimal: in particular, $\bar{l}$ is far from the target and the l^*_{over} metric is higher than for IC. Unlike RCBC, IC_{all} is very satisfactory as the over target losses are very low (3%). The same statement applies to the corresponding bandwidth allocation, i.e., 3.63 Mpbs on average, so that it may be considered close to the (unknown) optimal one. The corresponding variance of 1.06 denotes that the allocation is variable

Table 2: Obtained performance with Incremental Control (IC) and Reference Chaser Bandwidth Controller (RCBC) techniques. In bold, the best $\bar{l}$ (corresponding to the target $l^* = 10^2$) and the best value for l^*_{over} are pointed out. Regarding $\bar{\theta}$ and σ^2_θ, the obtained minimum and maximum values are evidenced, which derive from RCBC technique; however, in these cases, the corresponding $\bar{l}$ values are far from the target.

	$\bar{l}$	σ^2_l	l^*_{over} [%]	$\lvert\bar{l} - l^*\rvert$	$\bar{\theta}$ [Mbps]	σ^2_θ
RCBC_{f1}	$2.56{\cdot}10^{-2}$	$5.80{\cdot}10^{-2}$	56	$1.86{\cdot}10^{-2}$	**1.92**	**0.96**
RCBC_{f5}	$2.00{\cdot}10^{-2}$	$5.70{\cdot}10^{-2}$	28	$1.61{\cdot}10^{-2}$	2.22	1.43
$\text{RCBC}_{f0.5}$	$1.66{\cdot}10^{-2}$	$5.53{\cdot}10^{-2}$	27	$1.29{\cdot}10^{-2}$	**4.10**	**6.83**
IC_{all}	$5.60{\cdot}10^{-3}$	$3.85{\cdot}10^{-2}$	**3**	$5.28{\cdot}10^{-3}$	3.63	1.06
$\text{IC}_{reduced}$	$1.76{\cdot}10^{-2}$	$6.92{\cdot}10^{-2}$	14	$1.50{\cdot}10^{-2}$	2.13	1.11
$\text{IC*2}_{reduced}$	$1.10{\cdot}10^{-2}$	$5.32{\cdot}10^{-2}$	12	$8.84{\cdot}10^{-3}$	2.36	1.52
$\text{IC*5}_{reduced}$	$\mathbf{1.00{\cdot}10^{-2}}$	$4.13{\cdot}10^{-2}$	19	$6.89{\cdot}10^{-3}$	3.40	3.88
$\text{IC}_{all}(1000)$	$1.76\ {\cdot}10^{-2}$	$6.34{\cdot}10^{-2}$	18	$1.54\ {\cdot}10^{-2}$	2.90	1.68
$\text{IC}_{reduced}(1000)$	$3.45\ {\cdot}10^{-2}$	$7.63{\cdot}10^{-2}$	39	$2.94\ {\cdot}10^{-2}$	1.92	0.92

over time. $\text{IC}_{reduced}$, with the same time granularity of IC_{all} (one reallocation every minute), is unsatisfactory, although still better than RCBC: since the bandwidth is underestimated (2.13 Mbps, on average, with similar variance to IC_{all}), the reduced information vector is not able to properly drive the necessary bandwidth increments. However, speeding it up significantly improves the performance, especially when it is applied 5 times faster ($\text{IC*5}_{reduced}$). Despite the losses over target are still appreciable (19%), the average loss is very close to the target and the average bandwidth is close to the optimal one (as approximated by IC_{all}), albeit with larger variance. The latter, especially with $\text{IC*5}_{reduced}$ (σ^2_θ=3.88), denotes some level of bandwidth instability, but, on the other hand, the ability to react adequately to system changes. Overall, $\text{IC*5}_{reduced}$ turns out to be a very practical solution, as it performs sufficiently well despite the lack of direct information on the traffic source characteristics. The use of the models corresponding to the worst Clopper-Pearson (i.e., 'train1000' model in Fig. 3) leads to worse performance too. They are denoted in Table 2 with $\text{IC}_{all}(1000)$ and $\text{IC}_{reduced}(1000)$. The former severely degrades the target (18% of over target losses) and underestimates the bandwidth, as it leads to 2.9 Mbps on average versus 3.63 Mbps with IC_{all}, with a significant increase of bandwidth instability: variance is 1.68 versus 1.06. The same is true to $\text{IC}_{reduced}(1000)$,

with even more degraded performance with respect to the loss target and bandwidth requirement. Furthermore, small granularity of $\text{IC}_{reduced}(1000)$ does not provide any performance recovery. The bandwidth underestimation is quite evident, both in terms of average loss and average bandwidth. The situation is opposite with the models chosen after generalization tests, in which $\text{IC*5}_{reduced}$ approximates the performance of IC_{all}: $1{\cdot}10^{-2}$ of average loss versus $5.6{\cdot}10^{-3}$ and 3.4 Mpbs of average bandwidth (despite the much larger variance) versus 3.63 Mpbs.

6.5. Rule analysis for Incremental control

In terms of the performance metrics considered in Table 2, IC_{all} ensures staying below the target, but also its faster variant, $\text{IC}_{reduced}$, achieves good results. Furthermore, the underlying rule-based models can allow knowledge extraction from the system. To this aim, the LLM rules scoring a covering larger than 15% are reported below, for the IC_{all} setting.

if $((l > 2 \cdot 10^{-2}) \wedge (\overline{\phi} \leq 4.44))$ $r = \mathbf{1}$;
if $((m > 0.57) \wedge (l > 2 \cdot 10^{-2}) \wedge (B \leq 138))$ $r = \mathbf{1}$;
if $((m > 0.57) \wedge (l > 2 \cdot 10^{-2}) \wedge (B_{Max} > 251) \wedge (Bp \leq 44.4))$ $r = \mathbf{1}$;
if $((N > 84) \wedge (\sigma \leq 0.13) \wedge (l > 2 \cdot 10^{-2}))$ $r = \mathbf{1}$;
if $((N \leq 90) \wedge (l > 2 \cdot 10^{-2}) \wedge (B > 77))$ $r = \mathbf{1}$;
if $((\sigma \leq 0.06) \wedge (l > 2 \cdot 10^{-2}))$ $r = \mathbf{1}$;
. . . . . .
if $((l \leq 5 \cdot 10^{-4}) \wedge (\theta > 2.7) \wedge (B \leq 266) \wedge (\overline{\phi} > 3.4))$ $r = \mathbf{-1}$;
if $((N > 87) \wedge (l \leq 10^{-3}) \wedge (\theta > 1.7) \wedge (B_{Max} > 69) \wedge (\overline{\phi} > 3.1) \wedge (Bp \leq 40.1))$ $r = \mathbf{-1}$;

Remarkably, without any knowledge of the target (1%), IC_{all} understands the $l > 2\%$ criterion to drive the bandwidth increase (first rule). Again from the first rule, the threshold of 4.44 s on average silence periods is something difficult to understand from model-based approaches. The same applies to $m > 0.57$ in the second rule. The relevance of the two rules comes from their covering: 94% and 62%, respectively. Conversely, the reasoning involved in bandwidth decrease decision seems more complex, as highlighted by the high number of conditions in the last rule. Feature ranking can also be derived, and individuates l, θ, $\overline{\phi}$ and Bp as the most influential features. This means that N or σ have less significance in the problem.

In the case of $\text{IC}_{reduced}$, the rules are as follows:

if $((l > 2 \cdot 10^{-2}) \wedge (\theta \leq 1.63))$ $r = \mathbf{1}$;
if $((\sigma \leq 0.17) \wedge (l > 2 \cdot 10^{-2}) \wedge (B_{Max} > 164))$ $r = \mathbf{1}$;
if $((l > 2 \cdot 10^{-2}) \wedge (B > 83) \wedge (B_{Max} \leq 455))$ $r = \mathbf{1}$;
if $((m \leq 2.0) \wedge (l > 2 \cdot 10^{-2}) \wedge (B_{Max} \leq 455))$ $r = \mathbf{1}$;

. . . . . .
if $((m \leq 2.0) \wedge (\sigma \leq 0.21) \wedge (l \leq 2 \cdot 10^{-4}) \wedge (\theta > 2.2) \wedge (B_{Max} > 164))$ $r =$ **-1**;
if $((m \leq 1.7) \wedge (l = 0) \wedge (\theta > 2.2))$ $r =$ **-1**;
if $((m \leq 2.4) \wedge (l = 0) \wedge (\theta > 2.75) \wedge (B > 30))$ $r =$ **-1**;

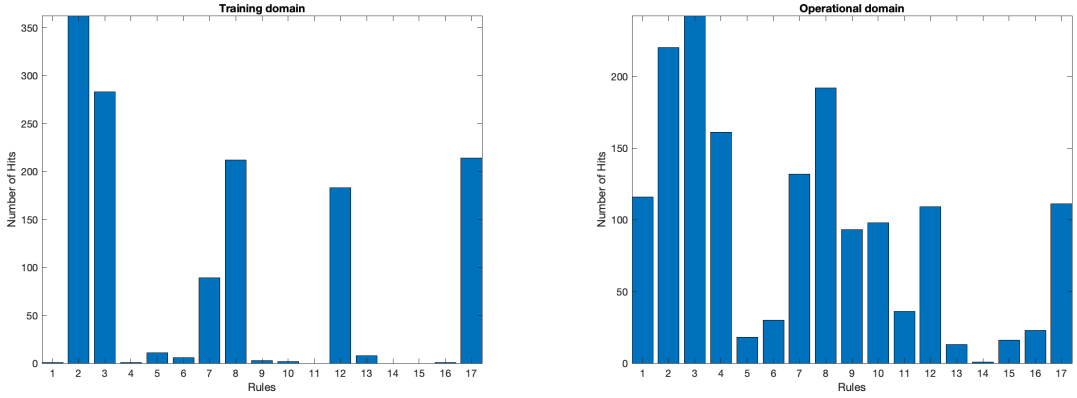
It is surprising to see that in the first rule, whose covering is 58%, the average silence period ϕ (of the first rule in IC_{all}) is replaced with the service rate θ. The rule states that, if the current θ is lower that 1.63, and the loss is larger than 2%, then the service rate should be increased (i.e., $r = 1$). $\sigma < 0.17$ in the second rule, whose covering is 57%, reveals that small variance of input traffic, together with loss larger than 2%, imply service rate increase. The 4^{th} rule states that if the maximum buffer size is small (lower than 455 packets), as well as loss larger than 2% again, the service rate should be increased despite an average input smaller than 2.0 Mpbs. As to the bandwidth decrease in the last three rules (whose covering is around 20% in all of them), the service rate threshold of 2.2/2.7 Mpbs represents the deduced upper bound inherent to buffer decongestion and consequent bandwidth release. It is worth noting that neither human reasoning nor model-based approaches would have hardly found the inferred dependencies.

The overall relationships between system variables, as inferred by the rule generation, look complex and are reflected in the feature ranking, which identifies m, l and B_{Max} as the most important variables for $\text{IC}_{reduced}$.

6.6. Robustness

The robustness test deals with training the same model as above, but under a very narrow range of $\overline{\phi}$ in training, i.e., $\overline{\phi} \in [3, 4]$ (training domain, $\overline{\phi}_{34}$ in the following) and testing the model with $\overline{\phi} \in [0, 5]$ (operational domain, $\overline{\phi}_{05}$ in the following), while the other parameter ranges are left untouched. As this puts a lot of stress on the working conditions, the $\text{IC*5}_{reduced}$ setting is used. With $\overline{\phi} \in [3, 4]$ and $\overline{\phi} \in [0, 5]$, the control scheme performance is as follows: $\bar{l} = 5.0 \cdot 10^{-3}$, $|\bar{l} - l^*| = 3.5 \cdot 10^{-3}$ and $\bar{l} = 9.0 \cdot 10^{-3}$, $|\bar{l} - l^*| = 7.0 \cdot 10^{-3}$, respectively, with sensible increase of the bandwidth allocation and its variance in the second case. Though robust, the model seems prone to underestimate the bandwidth if additional perturbations would be added in operation.

With the approach described in Sec. 5.3.3, $N_w = 10$ independent test iterations are performed, including $N_{tr} = 5$ tests on the training domain and $N_{op} = 5$ tests on the operational domain, giving rise to histograms $\mathbf{w}^j$, $j = (1, \ldots, N_w)$.



(a) Bar histogram with test samples from the training domain ($\overline{\phi}_{34}$)

(b) Bar histogram with test samples from the operational domain ($\overline{\phi}_{05}$)

Figure 4: Histograms representing the number of hits (in the y-axis) for each rule of the model (represented by the index in the x-axis), arising from the training (left) and the operational domain (right).

Figure 4 provides a visual comparison between two histograms obtained from testing the model inside the training domain and outside of it (operational domain). The bars show the number of hits, i.e., how many times each rule is covered by the test samples: it is straightforward to notice that this number has a significant variation between the two data domains. In other words, this difference in the number of hits distribution suggests that rules are 'used' differently if data are drawn from the training domain or from the operational.

The following analysis aims to quantify these differences through some statistical metrics, as mentioned in Sec. 5.3.3. First, four simple statistics over single histograms are computed: mean $m(\mathbf{w}^j)$, variance $\sigma^2(\mathbf{w}^j)$, skewness $s(\mathbf{w}^j)$ and kurtosis $\kappa(\mathbf{w}^j)$. Despite mean and kurtosis do not evidence any difference between training and operational, variance and skewness do. As shown in Fig. 5, variance and skewness average baselines (for $\overline{\phi}_{34}$) are around 15000 and 1.1, respectively, while operational variance and skewness are around 6000 and 0.6, on average. Then, the distance between couples of different histograms (as described in Sec. 5.3.3) is evaluated by measuring mutual information $\boldsymbol{\mu}I$, and the ℓ_1 and ℓ_2 norms. In detail, there are $C_{tr} = 10$ (from Eq. 13) couples for the in-training comparisons; for the operational setting, it is $C_{op} = 25$ "mixed" couples of histograms (*approach 1*, from Eq.

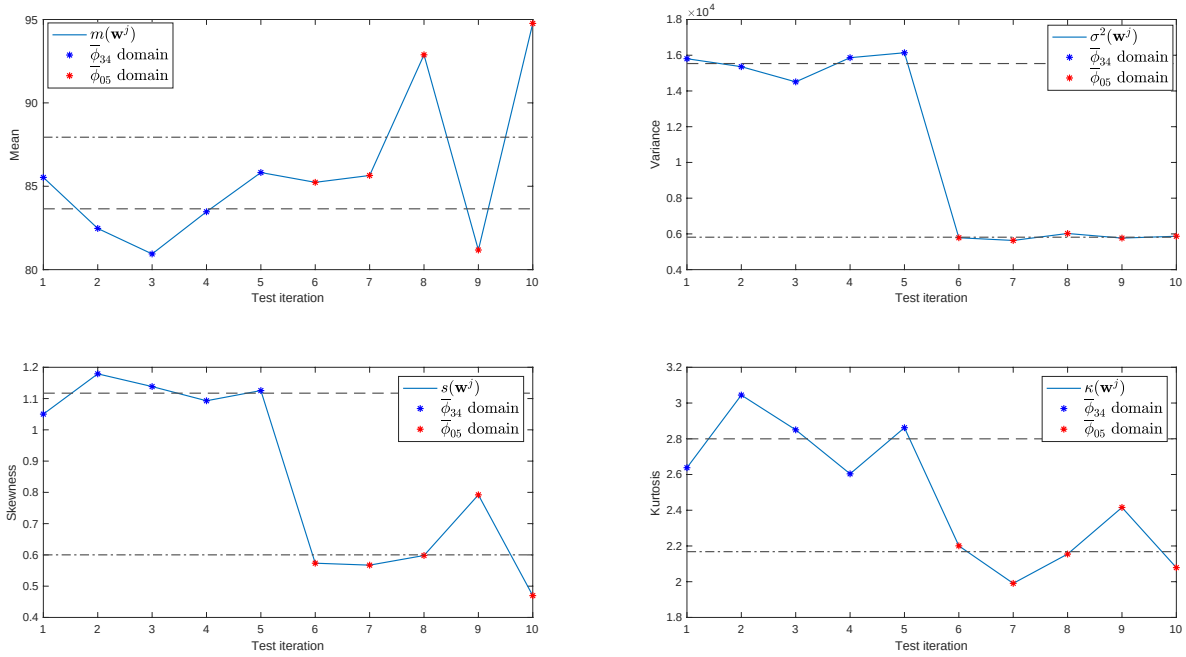


Figure 5: **Simple statistics (*approach 1*).**Plots of the obtained values for the considered statistics for each histogram $\mathbf{w}^j$: mean (top left), variance (top right), skewness (bottom left), kurtosis (bottom right). The x-axis refers to the different test iterations: 1 to 5 for the training domain ($\overline{\phi}_{34}$), 6 to 10 for the operational domain ($\overline{\phi}_{05}$). Black dashed lines denote the statistics averages over the iterations in $\overline{\phi}_{34}$, while black dot dash lines are the averages within $\overline{\phi}_{05}$ domain.

14), while $C_{op} = 10$ with in-operational couples (*approach 2*, from Eq. 15). The obtained results with *approach 1* are reported hereafter. A comparison with *approach 2* is carried out in Section 6.6.1 instead. From the mutual information, a baseline $\boldsymbol{\mu}I_{tr}^{base}$ =[0.30, 0.88] was obtained within the training domain, while the values at operational vary in a different range, being $\boldsymbol{\mu}I_{op}^{range} = [0.09, 0.52]$. Though, in this case, baseline and operational mutual information ranges are not perfectly separated, they achieve the desired role of reliable out-of-distribution indicators (Fig. 6)[7]. The difference in the usage of the rules between $\overline{\phi}_{34}$ and $\overline{\phi}_{05}$ is even more evident when evaluating the ℓ_1 and ℓ_2 norms for the same "mixed" couples of histograms used for

[7]Further investigation of this aspect follows in Section 6.6.1

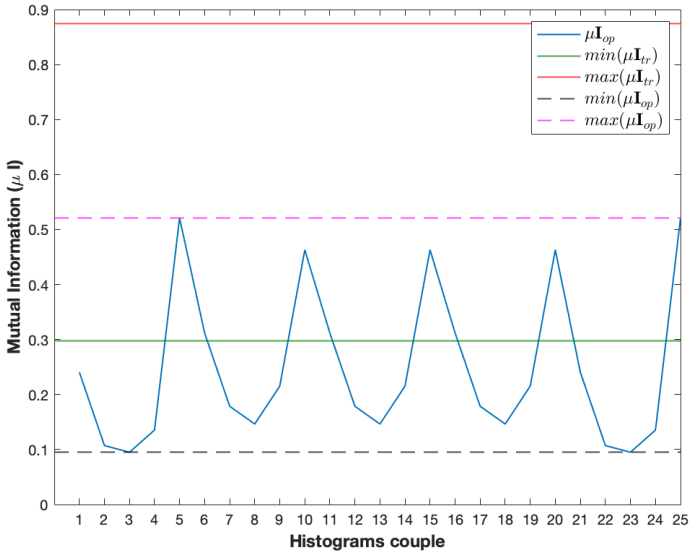


Figure 6: **Mutual Information (*approach 1*)**. Plot of the mutual information trend for the different considered couples of histograms. Each couple is formed by a histogram from $\overline{\phi}_{34}$ and the other from $\overline{\phi}_{05}$. The mutual information, computed for baseline histograms within $\overline{\phi}_{34}$, varies in [0.30, 0.88] (green and red lines), while its range for operational settings lowers to [0.09, 0.52] (pink and black dashed lines), showing out-of-distribution.

the mutual information. The obtained baselines for the training domain are $\mathbf{l1}_{tr}^{base} = [0.04, 0.14]$ and $\mathbf{l2}_{tr}^{base} = [0.02, 0.05]$. In contrast, the ranges between training and operational become sensitively higher than the baselines, being $\mathbf{l1}_{op}^{range} = [0.72, 0.93]$ for the ℓ_1 norm and $\mathbf{l2}_{op}^{range} = [0.23, 0.29]$ for the ℓ_2 norm. Figure 7 shows these ranges and the trend of the norms values obtained for all the considered "mixed" histograms couples. These results corroborate our hypothesis that it is possible to acknowledge the out-of-distribution condition by analyzing the firing (number of hits) of the available rules.

6.6.1. Comparison of $\boldsymbol{\mu I}$*:* approach 1 *vs* approach 2

The tests performed above concern the ϕ_{05} range as operational domain, which is larger than the training domain (ϕ_{34}). This is rather an unusual case, since the design phase should anticipate what happens in operation. It is also worth noting that the other parameters remained unchanged from the training to the operational domain. The intention was in fact to stress

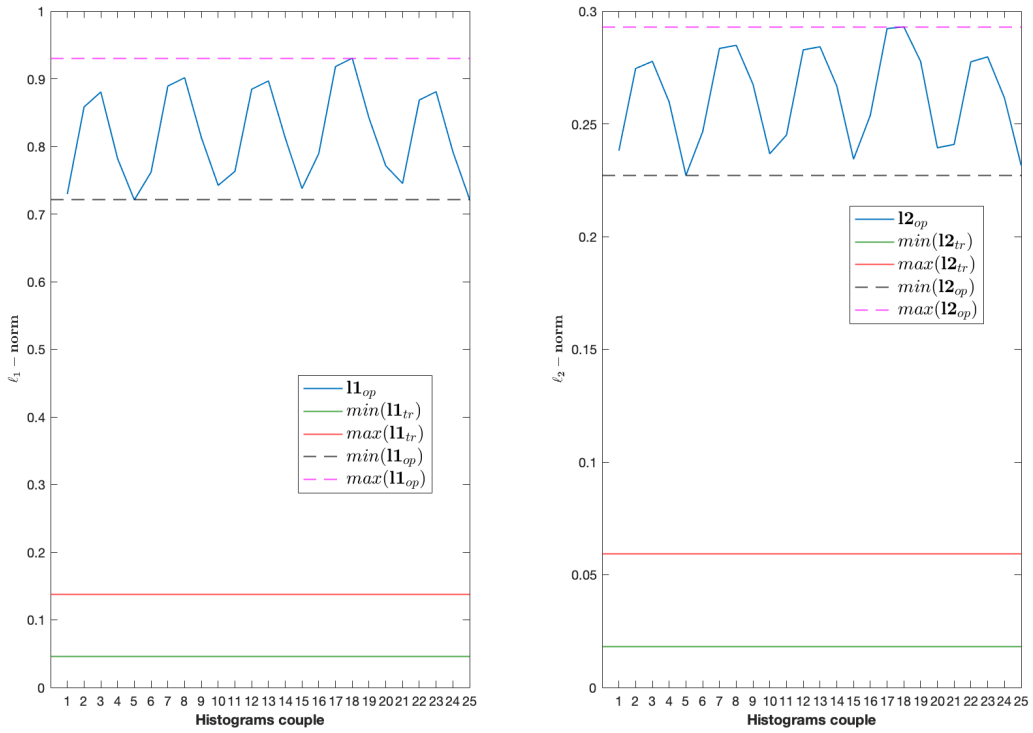


Figure 7: **Norms (*approach 1*).** Plot of the ℓ_1 and ℓ_2 norms trend for the different considered mixed couples of histograms. Each couple is formed by a histogram of the number of hits for $\overline{\phi}_{34}$ domain and the other from $\overline{\phi}_{05}$ domain. The baseline ranges (pink and red dashed lines) correspond to far lower values than the ranges for the comparison with the operational domain (black and green dashed lines), showing out-of-distribution.

the working conditions of our tests. For this reason, some similarity between training and operation was observed, with an incomplete separation between $\boldsymbol{\mu}I_{tr}^{base}$ and $\boldsymbol{\mu}I_{op}^{range}$. However, to further investigate how this behavior of the mutual information changes with respect to the designed ranges for training and operational domains, more extensive tests were performed, by comparing approaches 1 and 2 in multiple scenarios. Before discussing the settings and results, let us recall that *approach 1* (whose results were reported before) involves comparing $\boldsymbol{\mu}I_{tr}^{base}$ with $\boldsymbol{\mu}I_{op}^{range}$, where the latter is computed by comparing "mixed" couples of training-operational histograms. Conversely,

approach 2 calculated $\boldsymbol{\mu}I_{op}^{range}$ on the basis of couples of histograms defined within the operational domain.

Table 3 reports the obtained results of the additional tests. The first row corresponds to the scenario already defined at the beginning of Section 6.6, while the other rows define the additional experiments. Specifically, the same training domain baseline of the previous experiment (ϕ_{34}) was first retained, but tested against a new operational range with $\phi \in [0, 1]$ (or ϕ_{01}, second row in the table). Lastly, a new setting was devised, by generating a new training baseline with $\phi \in [1, 2]$ (ϕ_{12}), which was compared to three different operational settings, namely $\phi \in [2, 3]$, $\phi \in [3, 4]$ and $\phi \in [4, 5]$.

Table 3: Comparison of approaches for out-of-distribution detection via $\boldsymbol{\mu}I$.

			Approach 1		**Approach 2**	
Training	**Operational**	$\boldsymbol{\mu}I_{tr}^{base}$	$\boldsymbol{\mu}I_{op}^{range}$	**Result**	$\boldsymbol{\mu}I_{op}^{range}$	**Result**
ϕ_{34}	ϕ_{05}	[0.30, 0.88]	[0.09, 0.52]	in	[0.50, 1.00]	in
ϕ_{34}	ϕ_{01}	[0.30, 0.88]	[0.19, 0.36]	in	[1.45, 2.01]	out
ϕ_{12}	ϕ_{23}	[0.72, 0.92]	[0.72, 0.97]	in	[1.52, 2.26]	out
ϕ_{12}	ϕ_{34}	[0.72, 0.92]	[0.17, 0.17]	out	[0.97, 0.97]	out
ϕ_{12}	ϕ_{45}	[0.72, 0.92]	[0.17, 0.17]	out	[0.97, 0.97]	out

It is worth noting that, in the $\phi_{34} - \phi_{05}$ case (first row), both *approach 1* and 2 individuated some overlap between the operational ranges, that was expected in light of the considerations above. The second row ($\phi_{34} - \phi_{01}$) deals with an operational domain that is far away from the training one: while *approach 1* still defined some overlap between the $\boldsymbol{\mu}I$ ranges, *approach 2* managed to correctly individuate the out-of-distribution. The same result was found out with the new training baseline, i.e., ϕ_{12}, when tested against ϕ_{23} (third row) as operational domain. Finally, when the operational domain gets further away, i.e., with ϕ_{34} and ϕ_{45} (fourth and fifth rows), both approaches agreed on the same results (out-of-distribution).

6.6.2. Practical considerations

As soon as the robustness test raises the flag of out-of-distribution, the network engineer has to re-consider the applicability of the model and start a new training according to brand new monitoring data. Otherwise, there is the risk to work with an under-performing model and lose QoS control. As introduced at the beginning of Section 6.6, the fastest IC has been used (IC*$5_{reduced}$), thus limiting the performance degradation in operation. Decreasing the control time granularity would have led to much performance

decrease with IC*2$_{reduced}$: $\bar{l} = 1.31 \cdot 10^{-2}$, $|\bar{l} - l^*| = 8.8 \cdot 10^{-3}$. Such unpredictability of the performance degradation makes the understanding of out-of-distribution conditions one of the hottest topic for trustworthy network autonomy.

7. Conclusions and Future Works

This paper has investigated a new integration of trustworthy AI and control for autonomous equivalent bandwidth allocation. Given some simulation-based input measurements describing the behavior of the system, a rule-based classification model was adopted to learn a proper control mapping ($r-$mapping) to predict the increase ($r = 1$), decrease ($r = -1$) or maintenance ($r = 0$) of the bandwidth. Pursuing the need for a well generalizing model, Clopper-Pearson bound was adopted as an efficient way to perform model selection, also leading to discover the minimum of 3000 samples (i.e., about 3.3 hours of simulation) required to achieve a good solution. The explainability of the model has allowed to gain insights into the logic of the reallocation strategy, by individuating dependencies between system parameters and the direction of bandwidth allocation. Robustness was also investigated by exploiting XAI, through an innovative method to acknowledge the presence of out-of-distribution data. This method relies on the different rate of satisfaction of the rules at training and operational stages, measured through different metrics (i.e., mutual information, ℓ_1 and ℓ_2 norms). Results have shown that, while the norms are sensitive to changes with respect to training data distribution in all cases, mutual information ability also depends on the entity of the separation between the training and the operational domains. Practical considerations were also derived with respect to the detection of dangerous network variations in operation.

Despite the large ranges considered for the parameters of the problem at hand, other realistic network scenarios, involving next generation cellular networks [16] or cybersecurity [33, 34], may be considered to study more refined variations of the approach. First of all, the explainability of the model may be posed to the attention of network engineers via questionnaires, in order to quantitatively assess the AI logic and provide a synthesis of artificial and natural intelligence. The practical deployment of generalization and robustness over real network devices should be also considered to improve the applicability of the study. Mathematically speaking, other generalization tests, such as [31], may track the probability distribution better than

Clopper-Pearson, thus facilitating the use of as much less monitoring data as possible. Incremental statistical techniques, see, e.g., [34], may be also exploited to accelerate the data monitoring process in operation.

Acknowledgements

This work was supported in part by REXASI-PRO H-EU project, call HORIZON-CL4-2021-HUMAN-01-01, Grant agreement ID: 101070028. The work was also supported by Future Artificial Intelligence Research (FAIR) project, Recovery and Resilience Plan ("Piano Nazionale di Ripresa e Resilienza"), Spoke 3 - Resilient AI.

We would like to thank the anonymous Reviewers of the manuscript, who helped us improve its quality.

References

[1] L. Cruvinel, T. Vazao, "Profile-based Adaptive DiffServ Policing with Learning Techniques," 20th International Conference on Computer Communications and Networks (ICCCN) 2011, Maui, Hawaii, July 31 2011-Aug. 4 2011.

[2] A. Jayaraj, T. Venkatesh, C. S. R. Murthy, Loss Classification in Optical Burst Switching Networks using Machine Learning Techniques: Improving the Performance of TCP, IEEE Journal on Select. Ares in Commun., vol. 26, no. 6, Aug. 2008, pp. 45-54.

[3] M. Marchese, M. Mongelli, Measurement-based Computation of Generalized Equivalent Bandwidth for Loss Constraints, IEEE Comm. Letters, vol. 11, no. 12, Dec. 2007, pp. 1007-1009.

[4] S. Georgoulas, P. Trimintzios, G. Pavlou, K. Ho, An Integrated Bandwidth Allocation and Admission Control Framework for the Support of Heterogeneous Real-time Traffic in Class-based IP Networks, Computer Communications, vol. 31, no. 1, Jan. 2008, pp. 129-152.

[5] J. Aweya, M. Oullette, D. Y. Montuno, "A self-regulating TCP acknowledgment (ACK) pacing scheme," Int. Journal of Network Manag., vol. 12, no. 3, May/June 2002, pp. 145-163.

[6] M. Marchese, M. Mongelli, Performance Evaluation of Bandwidth Adaptation over DVB Satellite Channels, Proc. IEEE Global Communication Conference 2011 (Globecom 2010), Houston, Texas, USA, 5-9 Dec. 2011.

[7] L. Devroye, L. Györfi, G. Lugosi, A Probabilistic Theory of Pattern Recognition, New York: Springer-Verlag, 1997.

[8] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification (second ed.), New York: John Wiley and Sons, 2001.

[9] M. Muselli "Switching Neural Networks: A New Connectionist Model for Classification", in WIRN/NAIS 2005, vol. 3931 of Lecture Notes in Computer Science, 2006, Eds. B. Apolloni, M. Marinaro, G. Nicosia, R. Tagliaferri, Berlin: Springer-Verlag, pp. 23-30.

[10] M. Muselli, E. Ferrari, "Coupling Logical Analysis of Data and Shadow Clustering for partially defined positive Boolean function reconstruction", IEEE Transactions on Knowledge and Data Engineering, 2011, vol. 23, pp. 37-50.

[11] E. Ferrari, M. Muselli "Efficient constructive techniques for training Switching Neural Networks", in Constructive Neural Networks, vol. 258 of Studies in Computational Intelligence, 2009, Eds. L. Franco, D.A. Elizondo, J.M. Jerez, Berlin: Springer-Verlag, pp. 25-48.

[12] Michael M. Wolf, Mathematical Foundations of Supervised Learning, Die Technische Universität München, lecture notes, July 3, 2018, https://www-m5.ma.tum.de.

[13] V.N. Vapnik, A.Y. Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. In: Vovk, V., Papadopoulos, H., Gammerman, A. (eds) Measures of Complexity (1971). Springer, Cham, vol. 2, pp. 264–279 . https://doi.org/10.1007/978-3-319-21852-6_3

[14] R. Bruschi, F. Davoli, M. Mongelli, "Adaptive Frequency Control of Packet Processing Engines in Telecommunication Networks," IEEE Commun. Lett., July 2014, vol. 18, no. 7, pp. 1135-1138, doi: 10.1109/LCOMM.2014.2323244.

[15] "Easa concept paper: First usable guidance for level 1 machine learning applications, a deliverable of the easa ai roadmap," European Union Aviation Safety Agency, Daedalean, AG, Standard, Apr. 2021.

[16] L. Bonati, S. D'Oro, M. Polese, S. Basagni and T. Melodia, "Intelligence and Learning in O-RAN for Data-Driven NextG Cellular Networks," in IEEE Communications Magazine, vol. 59, no. 10, pp. 21-27, October 2021, doi: 10.1109/MCOM.101.2001120.

[17] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Elsevier, San Diego, CA, 2009.

[18] M. Aiello, M. Mongelli, and G. Papaleo, "Dns tunneling detection through statistical fingerprints of protocol messages and machine learning," International Journal of Communication Systems, vol. 28, no. 14, pp. 1987–2002, 2015.

[19] M. Mongelli, M. Aiello, E. Cambiaso and G. Papaleo, "Detection of DoS attacks through Fourier transform and mutual information," 2015 IEEE International Conference on Communications (ICC), 2015, pp. 7204-7209, doi: 10.1109/ICC.2015.7249476.

[20] Liang S, Li Y, Srikant R. "Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks," Internat. Conf. on Learning Repres., Vancouver CANADA, 30 Apr.03 May 3rd, 2018.

[21] N. Merayo et al., "PID controller based on a self-adaptive neural network to ensure qos bandwidth requirements in passive optical networks," in Journal of Optical Communications and Networking, vol. 9, no. 5, pp. 433-445, May 2017, doi: 10.1364/JOCN.9.000433.

[22] H. Zhang, N. Yang, W. Huangfu, K. Long and V. C. M. Leung, "Power Control Based on Deep Reinforcement Learning for Spectrum Sharing," in IEEE Transactions on Wireless Communications, vol. 19, no. 6, pp. 4209-4219, June 2020, doi: 10.1109/TWC.2020.2981320.

[23] J. Mei, X. Wang and K. Zheng, "An intelligent self-sustained RAN slicing framework for diverse service provisioning in 5G-beyond and 6G networks," in Intelligent and Converged Networks, vol. 1, no. 3, pp. 281-294, Dec. 2020, doi: 10.23919/ICN.2020.0019.

[24] F. Tang, B. Mao, Y. Kawamoto and N. Kato, "Survey on Machine Learning for Intelligent End-to-End Communication Toward 6G: From Network Access, Routing to Traffic Control and Streaming Adaption," in IEEE Communications Surveys & Tutorials, vol. 23, no. 3, pp. 1578-1598, thirdquarter 2021, doi: 10.1109/COMST.2021.3073009.

[25] R. Guerin, H. Ahmadi and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," in IEEE Journal on Selected Areas in Communications, vol. 9, no. 7, pp. 968-981, Sept. 1991, doi: 10.1109/49.103545.

[26] W. Kumwilaisak, Y. T. Hou, Qian Zhang, Wenwu Zhu, C. . -C. J. Kuo and Ya-Qin Zhang, "A cross-Layer quality-of-service mapping architecture for video delivery in wireless networks," in IEEE Journal on Selected Areas in Communications, vol. 21, no. 10, pp. 1685-1698, Dec. 2003, doi: 10.1109/JSAC.2003.816445.

[27] Yu Cheng and Weihua Zhuang, "Dynamic inter-SLA resource sharing in path-oriented differentiated services networks," in IEEE/ACM Transactions on Networking, vol. 14, no. 3, pp. 657-670, June 2006, doi: 10.1109/TNET.2006.876199.

[28] Jeong Geun Kim and M. M. Krunz, "Bandwidth allocation in wireless networks with guaranteed packet-loss performance," in IEEE/ACM Transactions on Networking, vol. 8, no. 3, pp. 337-349, June 2000, doi: 10.1109/90.851980.

[29] M. Grossglauser and D. N. C. Tse, "A time-scale decomposition approach to measurement-based admission control," IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No.99CH36320), 1999, pp. 1539-1547 vol.3, doi: 10.1109/INFCOM.1999.752176.

[30] E. Park, "Efficient Uplink Bandwidth Request with Delay Regulation for Real-Time Service in Mobile WiMAX Networks," in IEEE Transactions on Mobile Computing, vol. 8, no. 9, pp. 1235-1249, Sept. 2009, doi: 10.1109/TMC.2009.35.

[31] Mirasierra, V., Mammarella, M., Dabbene, F., & Alamo, T. (2021). Prediction error quantification through probabilistic scaling. IEEE Control Systems Letters, 6, 1118-1123.

[32] M. Mongelli, M. Muselli and M. Marchese, "A Unified View to Machine Learning and Control for Measurement-based Equivalent Bandwidth," 2020 16th International Conference on the Design of Reliable Communication Networks DRCN 2020, 2020, pp. 1-6, doi: 10.1109/DRCN48652.2020.1570609737.

[33] I. Vaccari, S. Narteni, M. Aiello, M. Mongelli and E. Cambiaso, "Exploiting Internet of Things Protocols for Malicious Data Exfiltration Activities," in IEEE Access, vol. 9, pp. 104261-104280, 2021, doi: 10.1109/ACCESS.2021.3099642.

[34] Aiello, M, Mongelli, M, Muselli, M, Verda, D. Unsupervised learning and rule extraction for Domain Name Server tunneling detection. Internet Technology Letters 2019; 2:e85. https://doi.org/10.1002/itl2.85.

[35] Vogl, T.P., Mangis, J.K., Rigler, A.K. et al. Accelerating the convergence of the back-propagation method. Biol. Cybern. 59, 257–263 (1988). https://doi.org/10.1007/BF00332914.

[36] L. Ruan, M. P. I. Dias and E. Wong, "Machine Learning-Based Bandwidth Prediction for Low-Latency H2M Applications," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 3743-3752, April 2019, doi: 10.1109/JIOT.2018.2890563.

[37] L. Ruan, M. P. I. Dias and E. Wong, "Enhancing latency performance through intelligent bandwidth allocation decisions: a survey and comparative study of machine learning techniques," in Journal of Optical Communications and Networking, vol. 12, no. 4, pp. B20-B32, April 2020, doi: 10.1364/JOCN.379715.

[38] J. A. Hatem, A. R. Dhaini and S. Elbassuoni, "Deep Learning-Based Dynamic Bandwidth Allocation for Future Optical Access Networks," in IEEE Access, vol. 7, pp. 97307-97318, 2019, doi: 10.1109/ACCESS.2019.2929480.

[39] J. Kim and G. Hwang, "Adaptive Bandwidth Allocation Based on Sample Path Prediction With Gaussian Process Regression," in IEEE Transactions on Wireless Communications, vol. 18, no. 10, pp. 4983-4996, Oct. 2019, doi: 10.1109/TWC.2019.2931570.

[40] J. Guo and C. Yang, "Impact of Prediction Errors on High Throughput Predictive Resource Allocation," in IEEE Transactions on Vehicular Technology, vol. 69, no. 9, pp. 9984-9999, Sept. 2020, doi: 10.1109/TVT.2020.3004552.

[41] S. Zhang, C. Wang, J. Zhang, Y. Duan, X. You and P. Zhang, "Network Resource Allocation Strategy Based on Deep Reinforcement Learning," in IEEE Open Journal of the Computer Society, vol. 1, pp. 86-94, 2020, doi: 10.1109/OJCS.2020.3000330.

[42] Aldabbas, H. Efficient bandwidth allocation in SDN-based peer-to-peer data streaming using machine learning algorithm. J Supercomput 79, 6802–6824 (2023). https://doi.org/10.1007/s11227-022-04929-y

[43] Navin Dhinnesh, A.D.C., Sabapathi, T. Probabilistic neural network based efficient bandwidth allocation in wireless sensor networks. J Ambient Intell Human Comput 13, 2001–2012 (2022). https://doi.org/10.1007/s12652-021-02961-z

[44] Kori, G. S., & Kakkasageri, M. S. (2023). Classification and Regression Tree (Cart) Based Resource Allocation Scheme for Wireless Sensor Networks. Computer Communications, 197, 242-254.

[45] Wang, S., Pei, K., Whitehouse, J., Yang, J., & Jana, S. (2018). Efficient formal safety analysis of neural networks. Advances in neural information processing systems, 31.

[46] Henriksen, P., & Lomuscio, A. (2020). Efficient neural network verification via adaptive refinement and adversarial search. In ECAI 2020 (pp. 2513-2520). IOS Press.

[47] Matin, M.A., Goudos, S.K., Wan, S. et al. Artificial intelligence (AI) and machine learning (ML) for beyond 5G/6G communications. J Wireless Com Network 2023, 22 (2023). https://doi.org/10.1186/s13638-023-02212-z

[48] European Commission, Directorate-General for Communications Networks, Content and Technology, (2019). Ethics guidelines for trustworthy AI, Publications Office. https://data.europa.eu/doi/10.2759/346720

[49] Marbach, P., Mihatsch, O., & Tsitsiklis, J. N. (2000). Call admission control and routing in integrated services networks using neuro-dynamic programming. IEEE Journal on selected areas in communications, 18(2), 197-208.

[50] Cello, M., Marchese, M., & Mongelli, M. (2016). On the qos estimation in an openflow network: The packet loss case. IEEE Communications Letters, 20(3), 554-557.

[51] M. Marchese, M. Mongelli, On-line bandwidth control for quality of service mapping over satellite independent service access points, Computer Networks, Volume 50, Issue 12, 2006, Pages 2088-2111, https://doi.org/10.1016/j.comnet.2005.10.006

[52] M. Baglietto, F. Davoli, M. Marchese and M. Mongelli, "Neural approximation of open-loop feedback rate control in satellite networks," in IEEE Transactions on Neural Networks, vol. 16, no. 5, pp. 1195-1211, Sept. 2005, doi: 10.1109/TNN.2005.853424.

Sara Narteni: Conceptualization, Data curation, Software, Investigation, Visualization, Writing; Marco Muselli: Formal analysis, Methodology, Validation, Writing - review and Funding acquisition (FAIR); Fabrizio Dabbene: Formal analysis, Methodology, Validation, Writing – review; Maurizio Mongelli: Conceptualization, Data curation, Software, Investigation, Visualization, Writing and Funding acquisition (REXASI-PRO).

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: