# Resource Demand Prediction for Network Slices in 5G using ML Enhanced with Network Models

Luis A. Garrido, Anestis Dalgkitsis, Kostas Ramantas, Adlen Ksentini, and Christos Verikoukis

***Abstract*—The new technologies introduced by 5G, such as network slicing, will improve the capabilities of Vehicle-to-Vehicle (V2V) communications, enabling the introduction of a new range of services and new forms of Vehicle-to-Everything (V2X) interactions. In order to deploy these V2X services and the network slices they are associated with over the 5G network while ensuring Quality of Service (QoS), intelligent and proactive network resource managers and orchestrators (RMOs) need to be developed. The ability to forecast the slice resource demand can significantly increase the proactivity of these RMOs.**

**ML-based resource demand predictors (RDPs) are commonly integrated with RMOs to provide *accurate* forecasts of the slice resource demands in V2X use cases. However, prediction errors are still common, causing the RMOs to reallocate resources to the slices sub-optimally. When an RDP underestimates the resource demand, i.e. predicts less demand than expected, the impact is much more severe for the infrastructure providers (InPs) and service providers (SPs) than when it overestimates the demand. Also, the impact of this misprediction is also different for each InP/SP, for which it is necessary for RDPs to also consider this difference. In view of this, we introduce a new approach that makes ML-based RDPs aware of the asymmetry of misprediction and their dependence to a specific network model, making their forecasts more useful for RMOs. This approach enhances the design of RDPs by embedding within them knowledge of the underlying 5G network and of the relationship between resource demand, resource allocation and service/network performance. We refer to our approach as *Network-Aware Loss for Demand Prediction* (*NALDEP*), and it improves the prediction quality by 73.3% and 41.0% with respect to accuracy-based and other state-of-the-art predictors, respectively.**

***Index Terms*—5G, V2X, Beyond-5G, network slicing, resource prediction, deep neural network, loss function.**

## I. Introduction

The new 5G standard [1] has enabled new use cases for the communications industry as a result of the new technologies it introduces [2]. Amongst these, network slicing [3]–[5] adds a series of features to the communication infrastructure that enables the virtualization of network resources and abstract them into multiple virtual networks, i.e. network slices. Network slicing improves capabilities not just for Vehicle-to-Vehicle (V2V) communication services, but it does so as well for communications between Vehicle-to-Everything (V2X) [6]–[10], allowing for vehicles to communicate with more devices.

L. A. Garrido, A. Dalgkitsis, K. Ramantas are with the R&D Department of Iquadrat, S.L, Barcelona, Spain, e-mail: {l.garrido, a.dalgkitsis, k.ramantas}@iquadrat.com.

Adlen Ksentini is a is a professor in the Communication Systems Department of EURECOM in Sophia Antipolis, France, e-mail: adlen.ksentini@eurecom.fr.

C. Verikoukis is an Associate Professor at Department of Computer Engineering & Informatics of the University of Patras in Patras, Greece, e-mail: cveri@ceid.upatras.gr.

In this context, the orchestration and management of network resources amongst slices, while maintaining Quality of Service (QoS), is in itself a complex task [9], [10], [12]–[14]. Generally speaking, resource managers and orchestrators (RMOs) need to be proactive instead of reactive, which requires to anticipate the resource demand profiles of the network slices in order to reallocate the needed resources among them *before* their resource demand changes. Otherwise, if they remain reactive without any foresight, the QoS of the users will be compromised. Being proactive in this sense will guarantee that the QoS requirements, usually specified through Service Level Agreements (SLAs), are fulfilled.

For the RMOs to be proactive, they need to be aware of the resource demand profiles of the slices. Ideally, resource demand forecasts can be provided to the RMO, for which AI/ML-based resource demand predictors (RDPs) can be used. By improving RMOs' proactivity by using RDPs, it is possible to prevent SLA violations, i.e. QoS degradations, while increasing the amount of slices deployed by improving resource utilization efficiency. Preventing SLA violations avoids (monetary) costs for the InPs/SPs, and improvements on resource utilization efficiency reduces Operational Expenditure (OPEX) and Capital Expenditure (CAPEX) and also increases revenue by providing service to more users for the given resources.

SLA violations generate large costs both financial and technical for the infrastructure providers (InPs) and/or service providers (SPs), and the gravity and magnitude of these costs depend on the specifics of the network infrastructure, the slices and the services. However, when a slice gets resources beyond its demand (get over-provisioned), no SLA violations occur but the resource efficiency decreases instead, which doesn't impact the InPs/SPs as severely as SLA violations. This means that the *costs* associated to under- and over-provisioning of resources, although undesirable in both cases, is also asymmetrical.

When ML-based RDPs are integrated with RMOs [15], [16], RDPs are designed to provide *accurate* forecasts. These RDPs generate their predictions in a network and service agnostic way. These RDPs introduce very tangible risks associated to mispredictions for the InPs/SPs, since an RMO can make decisions based on this agnostic predictions, and then sub-optimally reallocate resources. Even if such conditions can be detected fast, it will take time before resource *reconfiguration* takes place, generating SLA violations in the meantime, and/or reducing revenue. These are situations that V2X verticals cannot afford to have. Hence, it is necessary to make network-aware RDPs that generate contextualized forecasts for RMOs.

In view of this, we propose a framework for ML-based RDPs that go beyond the accuracy objectives, and instead

generates resource demand predictions with an *awareness* of the cost asymmetry and of the specifics of the network infrastructure, services and slices. We refer to this framework as *Network-Aware Loss for Demand Prediction* (*NALDEP*). Our approach extends the design of ML-based RDPs by embedding knowledge of 5G network and resource provisioning models that impact network and service performance. This knowledge is embedded by modifying the loss functions used to train the predictors with regularization terms [17]–[19].

From a practical perspective, the deployment and application of NALDEP in real-world V2X services translates into: 1) reduction of SLA violations which improves QoS for end-users, because NALDEP prevents RMOs from under-provisioning resources to slices by avoiding underestimation of the resource demand, 2) increased resource utilization efficiency for the InP by preventing resource demand over-estimation which in turn prevents resource overprovisioning [20]–[23], and 3) ensuring QoS for users amongst slices by making NALDEP aware of the cost of resource reconfiguration that increases OPEX/CAPEX for InPs and SPs, a factor so far overlooked. In summary, our contributions are:

- A refined definition of network models synthesized as a form of knowledge embedded in the loss functions to train RDPs [24]. It extends [25] by introducing more sophisticated formulations of the relationship between resource allocation, prediction and SLA violations with *the purpose of steering the behavior of RDPs*.
- The *extended* NALDEP framework for resource prediction applied to 5G networks. The loss functions make the RDP aware of the misprediction costs associated to SLA violations, resource utilization efficiency and reconfiguration, the latter of which has been so far overlooked.
- An extensive exploration of the parameter space of the *novel* NALDEP-derived loss functions with different Deep Neural Network (DNN) architectures and comparison with other state-of-the-art predictors.

The rest of this paper is organized as follows. Section II provides the background on 5G/B5G networks and traffic prediction using ML. Section IV explains the system model considered and Section III explains the NALDEP formulation. Section V describes our experimental methodology. Section VI shows our results and Section VII concludes this work.

## II. Background and Related Work

### A. Overview of 5G and Beyond-5G Networks

5G brought about a leap forward in communication technologies with respect to its predecessors [1], [5]. The integration of AI/ML into the communications network, which has ushered in an improved version of 5G commonly referred to as Beyond-5G (B5G), is expected to push the threshold of innovation even further by enabling the automation of the monitoring, analytics, orchestration and management processes of the network substrate for many verticals, including V2X, all while increasing peak data rates and reducing latencies further.

Network slicing in 5G allows the creation of virtual networks over the same physical infrastructure, a feature attractive for V2X verticals. Network slicing separates the InPs from SPs (tenants), which deploy and *own* their slices. When a tenant deploys a slice, the InP requires an RMO to provision resources for the slices [26] depending on its characteristics and desired performance (QoS) constraints.

### B. Resource Allocation in V2X with Network Slicing

The tenant specifies the resources and QoS constraints of its network slices through SLAs. But the resources that the slices actually use may differ from those requested. If the network slice needs more resources, the slice ends up being *under-provisioned* [27]. This causes SLA violations, generating revenue losses for the InP and poor QoS for its users [28]. In the opposite situation, in which the network slice uses *less* resources than those allocated, the network slice becomes resource *over-provisioned* [27] causing the resources to sit idle while consuming energy and increasing OPEX. In addition, over-provisioning will potentially reduce the number of network slices that can be deployed and will reduce the number of users that get access to service. Moreover, network slices have variable resource demands [10], [11], yielding a situation in which the network slice can be over- and under-provisioned at different points in time. Thus, it is necessary to dynamically re-adjust the resources assigned to the slices [26].

### C. Resource Demand Prediction in Network Slices for V2X

For the RMOs to proactively reallocate slice resources, they need forecasts [4] of slice resource demand profiles [29]. One instance of this problem in 5G communication networks occurs at the Radio Access Network (RAN) domain, in which the resource demand profile translates into bandwidth demand. In this work, we will use NALDEP to design RDPs for bandwidth demand forecasts (Section IV). The ability to predict traffic becomes a good approach for bandwidth management mechanisms at the BS level [4], [30], [32].

There are many approaches for traffic prediction within the context of 5G/B5G [20], [28], [33]–[35] and V2X [11], [16], [36]–[38], with the latter being relatively limited. Predictors such as those using Autoregressive Integrated Moving Average (ARIMA) [40], Holt-Winters algorithms [4], [33], [35] and Least Minimum Mean Square Error predictors [11] have been extensively studied in the literature. ML has also been used for traffic prediction, in particular Linear Regression, Polynomial Regression, Gaussian Processes, Feed-Forward Neural Networks (FFNNs) [20], [37], Long Short-Term Memories (LSTMs) [38], [39] (some of which use attention-based LSTM layers) and Convolutional Neural Networks (CNNs) [28], [36]. Whereas FNNs and ARIMA methods have similar performance [41], ARIMA performs significantly worst than DNN-based predictors designed with LSTMs [42]. Given this prior work, NALDEP employs DNN-based predictors since these are known to better extract features from the traffic profiles.

### D. Knowledge Embedding in DNN models

Making ML-based RDPs aware of the network details and the assymetry of the misprediction errors can be achieved with multiple methods [43], [44]. These include, for example, preprocessing and/or transformation of the data's feature space

(sometimes resulting in the engineering of new features), deploying specific DNN architectures, or even knowledge distillation [45]–[47]. Another way is to use regularization to embed knowledge into the RDP by translating the desired insight into a mathematical expression and insert it as part of the loss function, which is the approach in this paper.

Solutions such as those proposed by Sliwa [48] and Manalastas [49] are able to successfully predict end-to-end data rates in vehicular 5G and handover failure, respectively, for which they exploit additional context knowledge from the network status. They make available this knowledge to their predictors through their feature space. However, their approaches do not use knowledge embedding as in AI/ML, which can generate higher potential benefits, and they don't target resource demand forecasts, both of which NALDEP does.

One of the few approaches we know of that use knowledge embedding for ML was proposed by Liang et al [50]. Their solution leverages a knowledge embedding mechanisms previously described, and it tackles the problem of power allocation for multiple receive-transmitter pairs in a fading multi-user interference channel. The authors provide a solution to determine a power allocation profile to maximize the sum of data rates of the receivers, for which they use a loss function that expresses this maximization objective based on the physical transmission model. On the other hand, NALDEP uses supervised learning for resources demand forecasts, which is a linear regression problem that naturally maps to this type of learning.

In general, there's very limited literature regarding the use of knowledge embedding (much less using regularization) into RDPs for forecasting problems in 5G/B5G networks despite the benefits reported by ML-based solutions for communication networks. NALDEP seeks to fill in this gap by providing a knowledge embedding framework through regularization for RDPs, as explained in Section III. This will extend their capabilities to generate network- and resource-aware predictions, by giving them knowledge about the misprediction costs associated to SLA violations, resource utilization inefficiency and costly resource reconfigurations, the latter so far overlooked.

## III. ML-BASED RDPS ENHANCED WITH KNOWLEDGE EMBEDDING WITH NALDEP

The NALDEP framework consists of an ML-based RDP trained with a knowledge-embedding loss function that extends it with problem domain awareness. This loss function includes service- and infrastructure-oriented models that define the misprediction costs for given V2X service and infrastructure instances. The loss functions are paired with different ML-based RDPs, generating a different predictor specific to a V2X service or infrastructure element. The way in which these loss functions are defined and the way they are paired with ML-based RDPs constitute the core of the NALDEP methodology. ML-based RDPs consist of a DNN designed to generate resource utilization forecasts from time series data describing the amount of used resources. From the perspective of ML, this is a linear regression problem in which the future value of a variable is predicted based on its historical values.

For regression cases, loss functions such as Mean-Squared Error (MSE) and Mean Absolute Error (MAE) are the most commonly used. However, MSE is less robust due to its susceptibility to noise in the data and statistical outliers [51]. Huber Loss functions improve this by linearly penalizing high variances, but quadratically penalizing smaller ones. Log Hyperbolic Cosine (Log-Cosh) [52] has similar properties to Huber Loss, but it can be differentiated more times.

All of these functions assume a symmetry in the misprediction cost, meaning that they make this cost independent of the direction of the error. This is so because this loss function gears the predictor towards improved accuracy, making it agnostic of the problem domain. In order to create RDPs that are useful in the context of resource manamagement and orchestration in 5G networks, we extend the formulation of a loss function used to train DNNs for regression problems.

Our formulation starts with the MAE loss function, shown in (1), in which $B$ is the batch of values used in the current iteration, $y_i$ is the ground-truth values of the variable (i.e. real resource demand in the context of RDPs) and $y_i^p$ is the predicted value (predicted resource demand). In an RMO setting, the predicted resource demand $y_i^p$ can be used as part of the feature space of the RMOs' policies in order to make it more proactive to changes in resource demand.

$$MAE = \frac{\sum_{i=1}^{B} |y_i - y_i^p|}{B} \tag{1}$$

Using MAE as the basis, we can define a marginal loss (per data sample) for the MAE function according to (2). We start with this marginal loss because it penalizes equally regions of small and high variance. This allows for more flexibility in the formulation of NALDEP losses, with the additional consideration that it has no quadratic nor transcendental terms.

$$MAE_{loss} = |y_i - y_i^p| \tag{2}$$

The capability of a predictor based on (2) can be enhanced by embedding knowledge about the 5G network models as regularization terms, resulting in a new model-enhanced marginal loss expressed in (3).

$$C_{model} = |y_i - y_i^p| + (\lambda_h) H_{reg}(y_i, y_i^p) \tag{3}$$

In ML, it is very common to use regularization terms to prevent over-fitting of the DNN model. In the case of NALDEP, the constraint $H_{reg}()$ embeds 5G network models to make the predictor aware of the cost of mispredictions as they relate to resource orchestration and management in 5G/B5G networks. In (3), $\lambda_h$ is a weighting factor that determines the relative importance an RDP will give to the knowledge embedded into it in compare to the accuracy factor. As $\lambda_h$ becomes larger, the RDP becomes biased into satisfying the constraints given by $H_{reg}$. But as $\lambda_h$ becomes smaller, then the loss function prioritizes prediction accuracy. Thus, $\lambda_h$ can be used to calibrate the behavior of the RDP.

In the context of RDPs for 5G/B5G, the variables $y_i$ and $y_i^p$ of (1)–(3) represent the real and the predicted resource demand, respectively. In this case, the predicted resource demand is a proxy of the resources allocated to a slice when an RMO follows the prediction with high fidelity. By considering this, the function $H_{reg}()$ in (3) can be defined as shown in (4).

$$H_{reg}(y_i, y_i^p) = \begin{cases} H_{u-p}(y_i, y_i^p) & \text{if } CD_{u-p}(y_i, y_i^p) = \text{True} \\ H_I(y_i, y_i^p) & \text{if } CD_I(y_i, y_i^p) = \text{True} \\ H_{o-p}(y_i, y_i^p) & \text{if } CD_{o-p}(y_i, y_i^p) = \text{True} \end{cases} \tag{4}$$

Equation 4 defines a piece-wise function $H_{reg}(y_i, y_i^p)$ as the regularization term. In (4) (and here on forward), the subscripts *u-p*, *I* and *o-p* refer to *under-provisioning*, *ideal* and *over-provisioning*, respectively, and $CD$ stands for *conditionals*. If the conditionals in this function are dependent on the domain of $y_i$ and $y_i^p$, then $H_{reg}(y_i, y_i^p)$ becomes an approximation constrained (regularization) function [24].

For example, $CD_{u-p}(y_i, y_i^p)$ in (4) represents the case when the RDP under-estimates the resource demand, condition in which the error direction of the RDP is negative. As previously explained, an RMO could decide to under-provision slice resources based on this prediction, and generate a lot of severe costs for the InP and SPs. On the other hand, $CD_I(y_i, y_i^p)$ represents a region of ideal prediction, in which no costs are incurred for the InP and/or the SPs, and thus neither are *penalties* generated for the RPD. Above this ideal region, there will be no SLA violations and resource over-provisioning will be tolerated as long as it is bounded. The reason why over-provisioning is tolerated above this region relates to the fluctation-prone behavior of resource demand profiles, which has small variations within the control cycle intervals in which sampling takes place. By allowing a zero-penalty region of over-provisioning, the resulting RDP can absorb these short-term fluctuations without excessive resource reconfigurations, which also imply costs for the InPs and/or SPs.

Finally, $CD_{o-p}(y_i, y_i^p)$ corresponds to the case where the RDP over-estimates the resource demand, prompting RMOs to over-provision resources for slices, which makes inefficient utilization of network resources, generating costs for the InPs/SPs, but less severe than those generated by $CD_{u-p}(y_i, y_i^p)$.

### A. Network Model Knowledge as Regularization Constraints

From the perspective of InPs/SP in 5G/B5G networks, the costs of under- and over-provisioning resources as a response to the RDPs inputs is dependent on multiple factors. In this paper, we differentiate between the terms "costs" and "penalty", the former referring to the quantitative/qualititave impact that mispredictions have on InPs' and SPs' operations, while the latter refers to the "punishment" measured by an RDP when it generates misprediction. These RDP penalties are used to control and tune its prediction capability.

*1) Penalty for Under-Estimating Resource Demand:* When a slice is under-provisioned of resources as a result of resource demand under-estimation, an RDP experiences the following penalties:

- A penalty proportional to the difference between the resources allocated to the slice (in response to the predicted load $y_i^p$) and its real demand. Under-provisioning based on $y_i^p$ will increase the probability of SLA violations.
- A penalty associated to resource reconfiguration that happens in response to the rearrangement of resources for the under-provisioned slice. This requires hardware and software support, which increases the total cost of ownership for the InP [53]. This process is complex and it is not instantaenous, even if it is automated.

When all of these considerations are accounted for, it is possible to define $H_{u-p}(y_i, y_i^p)$ as in (5). Here, $TD(y_i, y_i^p)$ represents the penalty of service degradation as it relates to resource demand under-estimation, while the function $RR(y_i, y_i^p)$ represents the penalty of *resource reconfiguration*.

$$H_{u-p}(y_i, y_i^p) = TD(y_i, y_i^p) + RR(y_i, y_i^p) \tag{5}$$

Following a similar formulation to the one in [53], we define $RR(y_i, y_i^p)$ according to (6), where the parameter $C_r$ represents the sensitivity of $H_{u-p}(y_i, y_i^p)$ to the cost of resource re-configuration. In this equation, the difference $y_i - y_i^p$ is larger than zero, and the *r* in $C_r$ stands for *reconfiguration*.

$$RR(y_i, y_i^p) = C_r(y_i - y_i^p) \tag{6}$$

In the case of $TD(y_i, y_i^p)$, there are three variations we can consider: 1) linear, 2) quadratic, and 3) square-root definitions. The linear case is presented in (7), in which the parameter $C_d$ (the $d$ meaning *degradation*) represents the sensitivity of $H_{u-p}(y_i, y_i^p)$ to the cost of service degradation. In most situations, it is expected that $C_d \gg C_r$ because the cost of service degradation is expected to be more severe (negative impact on QoS, on the business model of the InP).

$$TD(y_i, y_i^p) = C_d(y_i - y_i^p) \tag{7}$$

Given the severe impact of under-provisioning and resource demand under-estimation, the other alternatives for $TD(y_i, y_i^p)$ (quadratic and square-root) are considered as well. Defining $TD(y_i, y_i^p)$ as a second-degree polynomial yields (8). When $y_i - y_i^p > 1.0$, the value of $TD(y_i, y_i^p)$ in (8) is larger than in (7), generating a larger penalty for the RDP. But when $y_i - y_i^p < 1.0$, (8) generates smaller penalties with respect to (7).

$$TD(y_i, y_i^p) = C_d(y_i - y_i^p)^2 \tag{8}$$

The square-root definition of $TD(y_i, y_i^p)$, shown in (9), will generate a larger penalty than (7) and (8) when $y_i - y_i^p < 1.0$, but will generate smaller penalties than both when $y_i - y_i^p > 1.0$. In many real systems, the penalty costs related to under-provisioning grow really fast in the proximity of the predicted-to-real demand value, reach a maximum and do not grow indefinitely. Thus, different representations of $TD(y_i, y_i^p)$ allows to model different service degradation characteristics for the network and the slices.

$$TD(y_i, y_i^p) = C_d\sqrt{y_i - y_i^p} \tag{9}$$

Depending on the properties of the system, $H_{u-p}(y_i, y_i^p)$ can be presented in either of the three following variations shown in (10), (11) and (12).

$$H_{u-p}(y_i, y_i^p) = (C_d + C_r)(y_i - y_i^p) \tag{10}$$

$$H_{u-p}(y_i, y_i^p) = C_d(y_i - y_i^p)^2 + C_r(y_i - y_i^p) \quad (11)$$

$$H_{u-p}(y_i, y_i^p) = C_d\sqrt{y_i - y_i^p} + C_r(y_i - y_i^p) \quad (12)$$

*2) Penalty for Over-estimation of Resource Demand:* When an RDP over-estimates the resource demand of a network slice, causing an RMO to over-provision resources, other penalties are considered:

- A penalty related to the idle resources, which reduces the revenue perceived (a type of cost) for the InP due to OPEX and CAPEX increase.
- A penalty due to resource reconfiguration, as in the case of under-estimation.
- Assuming full resource allocation and resource *overbooking*, a network slice with idle resources increases the chances of under-provisioning of other network slices, and prevents more slices to be admitted for execution.

If a slice has idle resources and the InP decides to re-allocate them to a different slice, then *resource overbooking* is taking place. If overbooking was not enabled, a slice would be taken out of execution once the InP reclaims its resources below the ones the slice has initially reserved [4], [54]. If a slice is over-provisioned in this scenario, these idle resources might be needed by another slice. However, the latter will not be able to get the resources it needs instantaneously, resulting in service degradation during this transient. As a result, there's a chance of other slices being under-provisioned.

In view of of this, it is possible to define $H_{o-p}(y_i, y_i^p)$ as shown in (13), in which $RS(y_i, y_i^p)$ represents the penalty of having idle resources in the network slice, $RR(y_i, y_i^p)$ is the penalty for resource reconfiguration, and $OB(y_i, y_i^p)$ represents the penalty of overbooking.

$$H_{o-p}(y_i, y_i^p) = RS(y_i, y_i^p) + RR(y_i, y_i^p) + OB(y_i, y_i^p) \quad (13)$$

In this case, $RR(y_i)$ has an identical form to that of (6). Similarly, $RS(y_i, y_i^p)$ can also be defined linearly as shown in (14). The parameter $C_w$ (the *w* stands for *waste*) is the sensitivity of $H_{o-p}(y_i, y_i^p)$ to resource idleness.

$$RS(y_i, y_i^p) = C_w(y_i^p - y_i) \quad (14)$$

The term $OB(y_i, y_i^p)$ represents a particular type of under-provisioning penalty because it is both *indirect* and *probabilistic* (because *other* slices are the ones that will suffer under-provisioning). As a result, it can be modeled in a similar way to (7), (8) and (9), as shown in (15), (16) and (17). In these equations, the parameter $C_d$ has an identical meaning as in Section III-A1, and *Pr* represents the sensitivity of the RDP to the condition of indirect under-provisioning of other slices.

$$OB(y_i, y_i^p) = P_r C_d(y_i^p - y_i) \quad (15)$$

$$OB(y_i, y_i^p) = P_r C_d(y_i^p - y_i)^2 \quad (16)$$

$$OB(y_i, y_i^p) = P_r C_d\sqrt{y_i^p - y_i} \quad (17)$$

Thus, $H_{o-p}(y_i, y_i^p)$ can be defined using either of its three possible forms, given by (18), (19) and (20).

$$H_{o-p}(y_i, y_i^p) = (C_w + C_r + P_r C_d)(y_i^p - y_i) \quad (18)$$

$$H_{o-p}(y_i, y_i^p) = (C_w + C_r)(y_i^p - y_i) + P_r C_d(y_i^p - y_i)^2 \quad (19)$$

$$H_{o-p}(y_i, y_i^p) = (C_w + C_r)(y_i^p - y_i) + P_r C_d\sqrt{y_i^p - y_i} \quad (20)$$

For all definitions of $H_{u-p(y_i, y_i^p)}$ and $H_{o-p}(y_i, y_i^p)$, the domain of the sensitivity parameters is $C_w,\ C_d,\ C_r \in \mathbb{R}^+$ (i.e. real positive values). The values they take are instrumental in tuning the RDP's knowledge of the network model to the specifics of the infrastructure and the slices. Also, their values need to be chosen in order to make training convergence feasible within reasonable time frames.

*3) Conditionals in $H_{reg}(y_i, y_i^p)$:* In Eq. 4, the conditionals $CD_{u-p}$, $CD_I$ and $CD_{o-p}$ were defined as corresponding to $H_{o-p}(y_i, y_i^p)$, $H_I(y_i, y_i^p)$, and $H_{u-p}(y_i, y_i^p)$.

For $CD_{u-p}$, it is straightforward to define it as $y_i^p - y_i < 0$, where the predicted resource demand $y_i^p$ for the current sample $i$ is smaller than the real demand $y_i$. In the same manner, it is possible to define $CD_{o-p}$ as $y_i^p - y_i \geq y_s$, where the predicted demand is larger than the real one. In the latter expression, the term on the right is introduced as a *"safety gap"* [20] or *"slack"* (the *s* in $y_s$) between the predicted and real resource demands. This value can be adjusted according to the constraints of the InP. These are cases in which a small degree of over-provisioning is desired in order to prevent negative effects related to short-term fluctuations of the resource demand profiles. Thus, $CD_I$ can be defined as $0 < y_i^p - y_i < y_s$, which provides a specific definition for the *ideal allocation region*. Hence, (4) can be re-written as (21).

$$H_{reg}(y_i, y_i^p) = \begin{cases} H_{u-p}(y_i, y_i^p) & y_i^p - y_i < 0 \\ H_I(y_i, y_i^p) & 0 \leq y_i^p - y_i \leq y_s \\ H_{o-p}(y_i, y_i^p) & y_s < y_i^p - y_i \end{cases} \quad (21)$$

The $y_s$ parameter can be shifted to move the ideal allocation region, satisfying different InP constraints, if the services tolerate a degree of under-provisioning without degrading QoS. For example, (22) considers this since it defines $y_s$ around the difference between $y_i^p$ and $y_i$, and reformulates (21) modifying $CD_{u-p}$, $CD_I$ and $CD_{o-p}$.

$$H_{reg}(y_i, y_i^p) = \begin{cases} H_{u-p}(y_i, y_i^p) & y_i^p - y_i < -\frac{y_s}{2} \\ H_I(y_i, y_i^p) & -\frac{y_s}{2} \leq y_i^p - y_i \leq \frac{y_s}{2} \\ H_{o-p}(y_i, y_i^p) & \frac{y_s}{2} < y_i^p - y_i \end{cases} \quad (22)$$

### *B. Defining Loss Functions for NALDEP*

After defining the misprediction penalties and how they relate to resource provisioning for network slices, we can formulate a series of NALDEP loss functions. We introduce the variable $\Delta x_i = y_i^p - y_i$ for readability and safe space.

*1) Linear Loss with a Positive Region of Ideal Estimation:* Similar to [25], we will use the linear variation of $H_{reg}(y_i, y_i^p)$ shown in (23), with the conditionals in (21).

$$H_{reg}(y_i, y_i^p) = \begin{cases} (C_d + C_r)(-\Delta x_i) & \Delta x_i < 0 \\ 0 & 0 \leq \Delta x_i \leq y_s \\ (C_w + C_r + P_r C_d)(\Delta x_i - y_s) & y_s < \Delta x_i \end{cases} \quad (23)$$

Given this definition for $H_{reg}(y_i, y_i^p)$, we can define the first marginal loss from NALDEP, labelled $C_{01}$ shown in (24).

$$C_{01} = |\Delta x_i| + \lambda_h * H_{reg}(y_i, y_i^p) = \begin{cases} \lambda_h(C_d + C_r + \frac{1}{\lambda_h})(-\Delta x_i) & \Delta x_i < 0 \\ 0 & 0 \leq \Delta x_i \leq y_s \\ \lambda_h(C_w + C_r + & \\ \quad P_r C_d + \frac{1}{\lambda_h})(\Delta x_i - y_s) & y_s < \Delta x_i \end{cases} \quad (24)$$

In (24), the marginal loss generates zero penalty within the ideal allocation region. We set $\lambda_h = 1$, integrating the differential of the MAE marginal loss into the knowledge constraints represented by the regularization term. The resulting marginal loss is shown in (25), considering that $C_r + 1 \approx C_r$.

$$C_{01}(y_i, y_i^p) = \begin{cases} (C_d + C_r)(-\Delta x_i) & \Delta x_i < 0 \\ 0 & 0 \leq \Delta x_i \leq y_s \\ (C_w + C_r + P_r C_d)(\Delta x_i - y_s) & y_s < \Delta x_i \end{cases} \quad (25)$$

In (25), $y_s$ is included both in the inequality $y_s < \Delta x$ and the function corresponding to this conditional, in order to make the marginal loss continuous in its domain. This is needed for it to be differentiable, a condition necessary for the training process [55]. The resulting loss function for $C_{01}(y_i, y_i^p)$ is shown in (26), referred to as $L_{01}$, where $B$ is the batch size.

$$L_{01}(y_i, y_i^p) = (\frac{1}{B}) \sum_{i=1}^{B} C_{01}(y_i, y_i^p) \quad (26)$$

*2) Linear Loss Shifting the Ideal Estimation Region:* $C_{01}$ can be modified to get a new linear marginal loss $C_{02}$ on $RS(y_i, y_i^p)$, $RR(y_i, y_i^p)$, $OB(y_i, y_i^p)$ and $TD(y_i, y_i^p)$ using the conditionals in (22), resulting in (27), and a loss function $L_{02}$ similar to (26), but with $C_{01}$ replaced by $C_{02}$.

$$C_{02}(y_i, y_i^p) = \begin{cases} (C_d + C_r)(\frac{y_s}{2} - \Delta x_i) & \Delta x_i < -\frac{y_s}{2} \\ 0 & -\frac{y_s}{2} \leq \Delta x_i \leq \frac{y_s}{2} \\ (C_w + C_r + P_r C_d)(\Delta x_i - \frac{y_s}{2}) & \frac{y_s}{2} < \Delta x_i \end{cases} \quad (27)$$

*3) Second-Degree Loss in $TD$:* In this case, we define a different marginal loss $C_{03}$ using the conditionals in (22), together with the definitions of $H_{u-p}(y_i, y_i^p)$ and $H_{o-p}(y_i, y_i^p)$ used in (11) and (19). Starting the formulation from the first term of (24) and applying a similar procedure, it yields (28). The resulting loss function is $L_{03}$, which is similar to (26) but with $C_{01}$ replaced by $C_{03}$.

$$C_{03}(y_i, y_i^p) = \begin{cases} C_r(\frac{y_s}{2} - \Delta x_i) + C_d(\frac{y_s}{2} - \Delta x_i)^2 & \Delta x_i < -\frac{y_s}{2} \\ 0 & -\frac{y_s}{2} \leq \Delta x \leq \frac{y_s}{2} \\ (C_w + C_r)(\Delta x_i - \frac{y_s}{2}) + & \\ \quad P_r C_d(\Delta x_i - \frac{y_s}{2})^2 & \frac{y_s}{2} < \Delta x_i \end{cases} \quad (28)$$

*4) Square-Root Loss on $TD$:* By using (22), with $H_{u-p}(y_i, y_i^p)$ and $H_{o-p}(y_i, y_i^p)$ used in (12) and (20), respectively, it is possible to derive a fourth marginal loss $C_{04}$. In this instance, new approximation constraints are obtained by changing the functions of $C_{04}$ in the ranges specified by conditionals in (22). The formulation of this marginal loss is presented in (29), which results in a loss function $L_{04}$ similar to (26), but with $C_{01}$ replaced by $C_{04}$.

$$C_{04}(y_i, y_i^p) = \begin{cases} C_r(\frac{y_s}{2} - \Delta x_i) + C_d\sqrt{-\Delta x_i} & \Delta x_i < -\frac{y_s}{2} \\ 0 & -\frac{y_s}{2} \leq \Delta x_i \leq \frac{y_s}{2} \\ (C_w + C_r)(\Delta x_i - \frac{y_s}{2}) + & \\ \quad P_r C_d\sqrt{\Delta x_i - \frac{y_s}{2}} & y_{s/2} < \Delta x_i \end{cases} \quad (29)$$

## IV. System Model

In order to make our RDPs relevant for 5G and V2X verticals, we will consider the resource demand forecasting problem in a 5G/B5G communication infrastructure supporting V2X services. We will assume that this infrastructure supports different network slices, each associated with a specific mobile service. This communication infrastructure possesses all the technological domains proper of a 5G network: 1) the 5G Mobile Core, 2) a 5G transport, 3) a Multi-Access Edge Computing Domain (MEC), and 4) a RAN Domain. In order to be more specific, NALDEP will consider the resource demand at the RAN, focusing mainly on the bandwidth demand at the Base Stations (BSs) of the RAN, for which we provided extensive related work in Section II-C. We assume also that the MEC nodes close to the BSs run the NALDEP solutions, in order to keep the RPDs close to the data.

We consider a RAN domain with a set $G$ of BSs. In each BS $g \in G$, there is a set of slices $S_g$ belonging to different V2X mobile services. We assume that the number of slices $|S_g|$ and the resources that all slices requires fit within the capacity of $g$. A slice $s_g^x \in S_g$, in which the variable $x$ refers to a specific slice in BS $g$, generates a traffic load (resource demand) of a specific service at different points in time $t$, and this load is measured periodically with a period

of $T_p$, which is the frequency at which the data is sampled. We don't consider the traffic of a specific user equipment (UE), but only the total traffic of the collective of the UEs. These UEs could be mobile phones, vehicles or any system in the V2X infrastructure. Fig. 1 illustrates the system model considered. We deploy different RDPs to forecast the traffic load (equivalent to resource demand) generated by three slices $\{s_g^1, s_g^2, s_g^3\} \subset S_g$. Every slice $s_g^x$ is associated with an RDP of its own. The RDPs are DNNs that are trained with the NALDEP-based loss functions.

*Forecasting and Orchestration*: Based on the time-series data, the predictors are trained to generate a traffic forecast for time $t+T_p$, in real time. The forecasts can be used by an RMO that re-allocates resources among the slices $s_g^x$, improving the latter's proactivity to reduce the probability of SLA violations and increase resource utilization efficiency. Based on the difference between the real and predicted traffic demand, the probability of SLA violations for each slice $s_g^x$ is calculated as well as the physical total costs associated to resource utilization inefficiency. Figure 2 illustrates these processes as described. The orange arrows represent the data flow for the training process that occurs offline and periodically (once every 12 or 24 hours), while the blue arrows represent the data flow for performing inferences at every timestep, generating predictions for future ones.

## V. Experimental Framework

Our evaluation consisted of training different RDPs with NALDEP-based loss functions enhanced with 5G network knowledge and insights on resource management and orchestration. The implementation of this framework was done in Python 3.8 and Tensorflow 2.1 [56].

### A. DNN Architectures for RDPs

Table I shows the DNN architectures used to implement the RDPs and the parameters used for each. The chosen hyperparameters for the STN ("Spatio-Temporal Neural Network") [34] and the 3D-CNN [28] architectures were based on the recommendations of their respective authors. STN uses an encoder-decoder paradigm, combining a stack of Convo-
l

In the case of DNNs based on LSTM and Conv-LSTM2D networks, the hyper-parameters were chosen experimentally.

### B. NALDEP-derived Loss Functions

Different loss functions were tested with each DNN. Our baseline for comparison consists of three loss functions: MAE, MSE and the loss function in [28] presented in (30). The latter has two parameters: $\alpha$ and $\epsilon$, evaluated with the values $[0.01, 0.1, 0.5, 1.0, 2.0, 5.0, 8.0, 10.0, 15.0, 20.0]$ and $[0.01, 0.1, 0.05, 0.25]$, respectively.

$$l'(x) = \begin{cases} \alpha - \epsilon(y_i - y_i^p) & \text{if } (y_i - y_i^p) \leq 0 \\ \alpha - (\frac{1}{\epsilon})(y_i - y_i^p) & \text{if } 0 < (y_i - y_i^p) \leq \epsilon\alpha \\ (y_i - y_i^p) - \alpha\epsilon & \text{if } (y_i - y_i^p) > \epsilon\alpha \end{cases} \tag{30}$$

We implemented the four loss functions $L_{01}$, $L_{02}$, $L_{03}$ and $L_{04}$ developed in Section III-B. Every DNN architecture was paired with these loss functions, generating 16 different (DNN, Loss Function) pairs. For every pair, a large set of experiments were run with different combinations of the parameters $C_w$, $C_d$ and $C_r$, resulting in a large number ($\sim 10^2$) of experiments.

### C. Dataset Used For Experimentation

To train our NALDEP-based RDPs, we used mobile traffic data from the city of Milano [57], assuming a communication infrastructure as described in Section IV. This dataset was generated from a pre-5G generation of technology. However our RDPs are agnostic to this and they can be trained using any time-series data that has resource demand information. Every BS in this system processes traffic from three different *service* types (calls, SMSs and Internet), each associated with a network slice for which our NALDEP-based RDPs perform slice-level prediction. For these data traces to be used in our framework, they need to be processed as follows: 1) do min-max normalization, 2) shift data to have rollback entries since the first prediction requires a specific number of prior samples, and 3) generate the appropriate data structure (a vector of specific dimensions) in order to be used by the DNNs.

### D. Measuring Compliance with RMO Constraints

We compare the predicted traffic values (predicted resource demand), used as proxy for resource allocation of RMOs, to



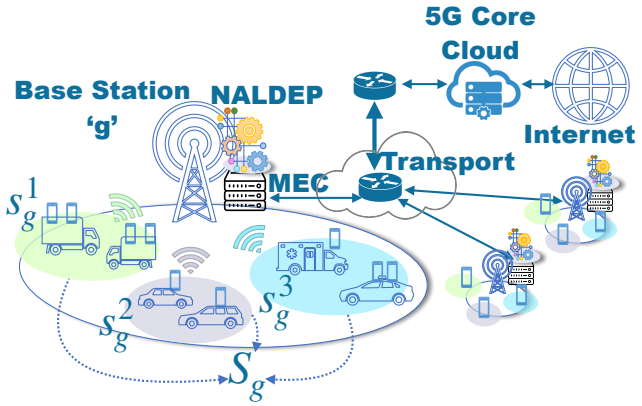



Fig. 1: 5G Communication Infrastructure for V2X Verticals.

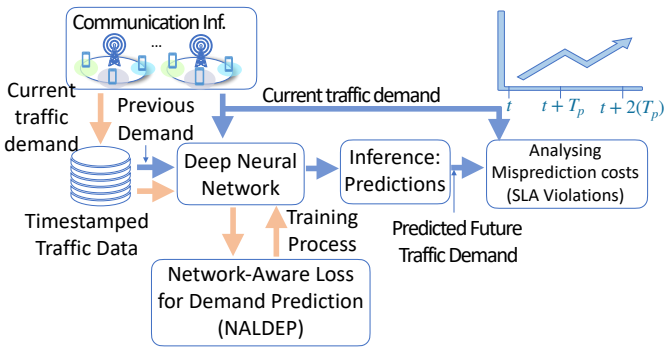



Fig. 2: Framework of NALDEP.

TABLE I: DNNs evaluated with NALDEP and their parameters. For all: learning rate=0.001, Adam Optimizer, Epochs=100.

| DNN | Parameters | | | | | | Description |
|---|---|---|---|---|---|---|---|
| | Hidden Layers | Hidden Layer/ Filter Size | Act. Func. | Dropout Prob. | Batch Size | Unrolling | |
| LSTM | 4 | 50 | tanh | 0.2 | 42 | 24 | Widely used for time-series prediction. |
| 3Dcnn | 4 | 32-16-64-32 | ReLu | 0.3 | 128 | 24 | DNN architecture used in [28]. |
| STN | 9 | 3-3-3-6-6-6-6-6-4 | ReLu | - | 128 | 6 | DNN architecture used in [34]. |
| Conv-LSTM2D | 4 | 4 | tanh | - | 128 | 6 | Using Conv-LSTM2D (CL2D) as hidden layers. |

the real traffic values of each slice (real resource demand), and calculate the physical cost $PC_{(o-p,i)}$ associated to over-provisioning of sample $i$ using (31).

$$PC_{(o-p,i)}(y_i, y_i^p) = (PI_{o-p})(y_i^p - y_i) \tag{31}$$

In (31), $PI_{o-p} \in \mathbb{R}$ quantifies a factor of *physical* impact of over-provisioning. It can, for example, represent the price (in currency) an InP pays for wasted resources. This is different to the sensitivity parameters used for $H_{reg}$ defined in Section III, since those are defined to tune the behavior of the RDP. Likewise, the *physical* cost of under-provisioning $PC_{(u-p,i)}$ for sample $i$ is calculated with (32).

$$PC_{(u-p,i)}(y_i, y_i^p) = (PI_{u-p})(y_i - y_i^p) \tag{32}$$

In (32), $PI_{u-p} \in \mathbb{R}$ is similar to $PI_{o-p}$, but in this instance it quantifies the *physical* impact of SLA violations i.e. of under-provisioning. It is possible to combine (31) and (32) into a single equation, as demonstrated by (33).

$$PC_i(y_i, y_i^p) = \begin{cases} (PI_{o-p})(y_i^p - y_i) & \text{if } \; y_i^p \geq y_i \\ (PI_{u-p})(y_i - y_i^p) & \text{if } \; y_i > y_i^p \end{cases} \tag{33}$$

In order to simplify (33), we normalize the quantified physical cost with respect to $PC_{u-p}$, which yields (34), for which we introduce the variable $CR_{\frac{UP}{OP}}$ as the ratio between the $PC_{o-p}$ and $PC_{u-p}$ in (35).

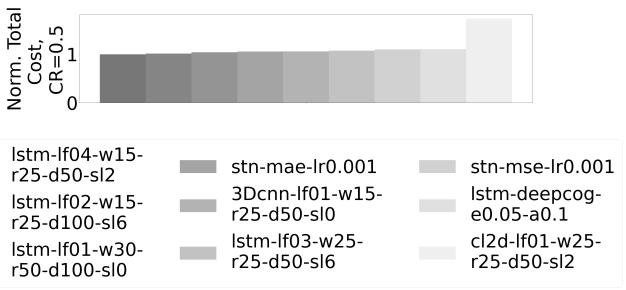
$$CR_{\frac{OP}{UP}} = \frac{PC_{o-p}}{PC_{u-p}} \tag{34}$$

$$PC_i(y_i, y_i^p) = \begin{cases} (CR_{\frac{OP}{UP}})(y_i^p - y_i) & \text{if } \; y_i^p \geq y_i \\ (1.0)(y_i - y_i^p) & \text{if } \; y_i > y_i^p \end{cases} \tag{35}$$

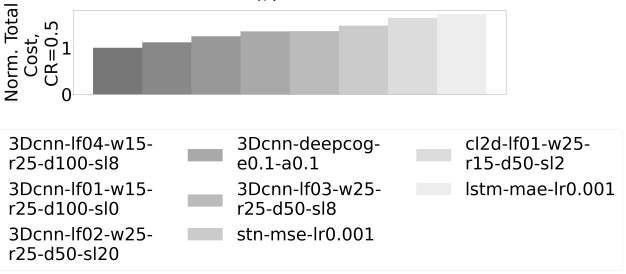
This ratio allows to understand $PI_{o-p}$ as a function of $PI_{u-p}$. For example, for $CR_{\frac{OP}{UP}} = 0.2$, the resulting physical cost $PC_t(y_i, y_i^p)$ corresponds to the case where $PC_{o-p}$ is 20% that of $PI_{u-p}$. In our evaluation, we use two different values for $CR_{\frac{UP}{OP}}$: 0.5, and 0.25.

Equation 36 is used to calculate the physical total cost $TC$ for each sample across a series of them. It is feasible to obtain $TC$ by a sum of the costs of each sample during a time period $P$ (usually towards the end) in which each experiment runs, as in (36). The smaller the value of $TC$, the better performing the (DNN, loss function) pair is.

$$TC = \sum_{i=1}^{i=P} PC_i(y_i, y_i^p) \tag{36}$$




(a) Normalized $TC$ for $CR_{\frac{OP}{UP}} = 0.5$ (less is better) in BS A.




(b) Normalized $TC$ for $CR_{\frac{OP}{UP}} = 0.5$ (less is better) in BS B.

Fig. 3: Normalized $TC$ for two BSs for $CR_{\frac{OP}{UP}} = 0.5$

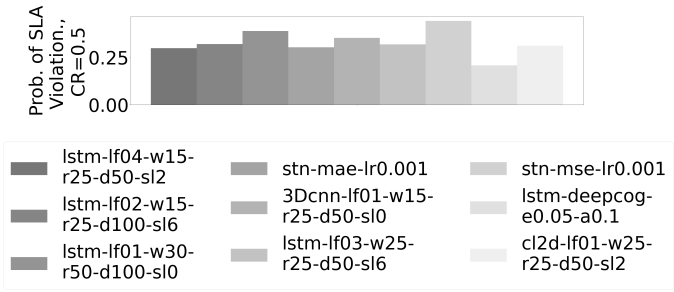
## VI. Results and Evaluation

First, we evaluate NALDEP considering a *network slice* of the same type in two different BSs, labeled $A$ and $B$, respectively. Second, we evaluate NALDEP considering the aggregate of the same slice type across the BSs. In order to ease readability of the result, the legend in the figures has been formatted as follows: {DNN}-{Loss Function}-{Parameters}, in which "$lf0x \rightarrow L_{0x}$" represent the NALDEP loss functions. The parameters are: $w \rightarrow C_w$, $r \rightarrow C_r$, $d \rightarrow C_d$, $sl \rightarrow y_s$, $lr \rightarrow$ *Learning Rate*, $e \rightarrow \epsilon$ and $a \rightarrow \alpha$. Different combinations of these yield an RDP with different sensitivity to slice resource over- and under-estimation, and to reconfigurability operations. Here, we want to evaluate what is the knowledge-enhanced RDP, with the best sensitivity tuning, with the smallest $TC$.

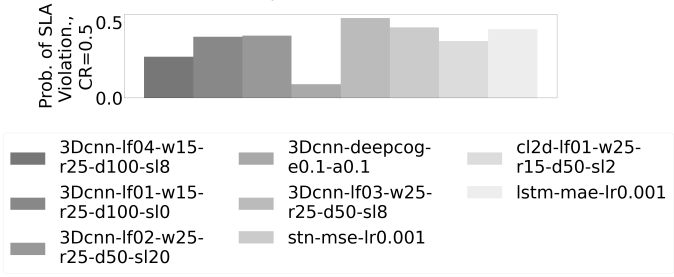
### A. Evaluating Predictions of the Slices in Two Base Stations

In this section, we show the results of BSs $A$ and $B$ chosen at random from the set of BSs present in the dataset. Showing all BSs is not feasible due to space constraints, but other BSs were examined, and they all present a consistent behavior.

*1) Results for $CR_{\frac{OP}{UP}} = 0.5$:* Figure 3a shows the normalized $TC$ for $CR_{\frac{OP}{UP}} = 0.5$ in $A$. In this case, the (LSTM, $L_{04}$) pair generates the smallest $TC$ (best performance). The pairs (STN, MSE) and (LSTM, $L_{deepcog}$) perform 10.0% and




(a) Probability of SLA violations for $A$.




(b) Probability of SLA violations for $B$.

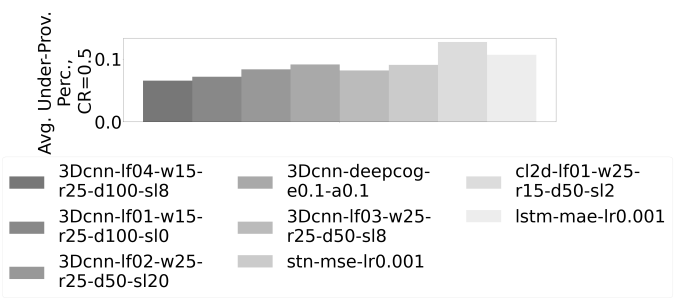
Fig. 4: SLA violation probability for $CR_{\frac{OP}{UP}} = 0.5$

10.6% worse, respectively. Figure 3b shows the normalized $TC$ for $CR_{\frac{OP}{UP}} = 0.5$ in $B$. In this case, the (3Dcnn, $L_{ecatp04}$) pair performs the best. The pair (STN, MSE) performs 47.6% worse, and the pair (3Dcnn, $L_{deepcog}$) has a $TC$ 35.3% worse.

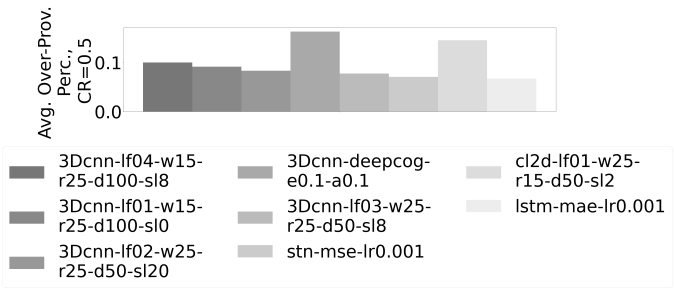
Notice that different (DNN, Loss Function) pairs perform the best for different BSs. For $A$, the LSTM-based DNNs perform better, while in $B$, it is the 3Dcnn-based DNNs. This is because different DNNs are better at extracting different features from the data, implying that BSs have traffic profiles with different properties. In all cases, NALDEP-derived loss functions generate less SLA violation costs. Moreover, there is a very small difference in performance among the (DNN, Loss Function) pairs that use NALDEP-based loss functions, and they perform significantly better than $L_{deepcog}$ and MSE.

Fig. 4 shows the SLA violation probability for both BSs, showing that the (3Dcnn, $L_{deepcog}$) pair has the smallest one for $B$, despite being the (3Dcnn, $L_{ecatp04}$) pair with the smallest penalty in Fig. 3b. In order to understand this, we need to look at the avg. magnitude and probability of under- and over-estimation, shown in Fig. 5. Notice that the (3Dcnn, $L_{ecatp04}$) pair has a larger magnitude (Fig. 5b) and probability (Fig. 5c) of over-provisioning, but the smallest under-provisioning magnitude (Fig. 5a). We make two observations regarding these results: 1) the over-provisioning cost of (3Dcnn, $L_{deepcog}$) is large enough to overcome the fact that $CR_{\frac{OP}{UP}} = 0.5$, 2) the (3Dcnn, $L_{ecatp04}$) pair performs the best due to its low SLA violation probability and its low magnitude of over-provisioning. These observations demonstrate the need to consider the costs of resource over-provisioning, not only for SLA violations, when evaluating the quality of an RDP.

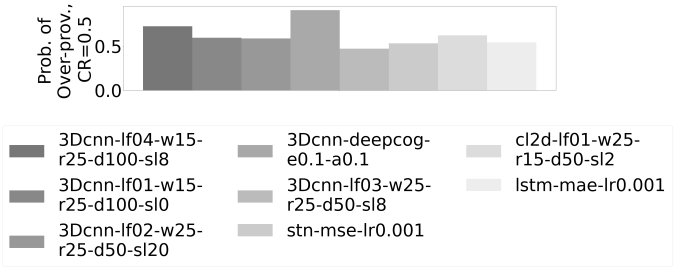
*2) Results for $CR_{\frac{OP}{UP}} = 0.25$ for Two BSs:* Figure 6a shows the normalized penalty for $CR_{\frac{OP}{UP}} = 0.25$ in $A$, meaning that the cost of resource under-estimation is 4x the cost of over-estimation. In this case, the (LSTM, $L_{04}$) pair incurs the smallest misprediction penalty. The pair (STN, MSE) performs




(a) Average magnitude of under-provisioning for BS B.

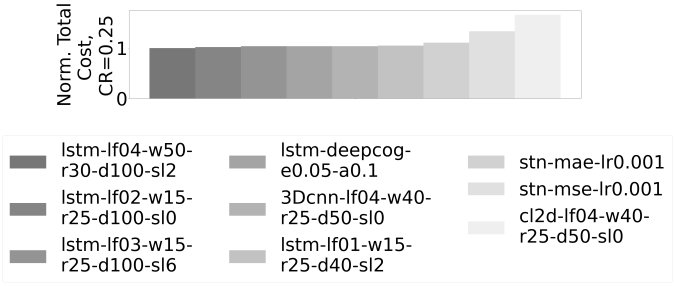



(b) Average magnitude of over-provisioning for BS B.

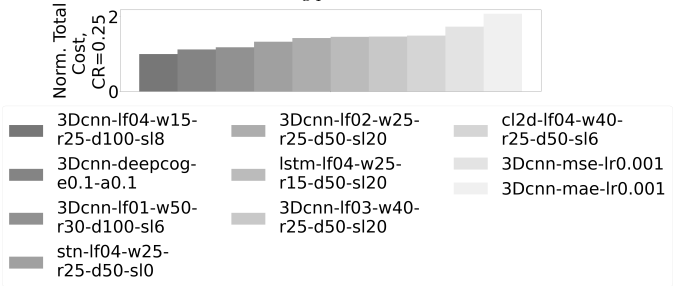



(c) Probability of over-provisioning for BS B.

Fig. 5: Avg. magnitude of under- and over-provisioning, and probability of over-provisioning for $B$, $CR_{\frac{OP}{UP}} = 0.5$.

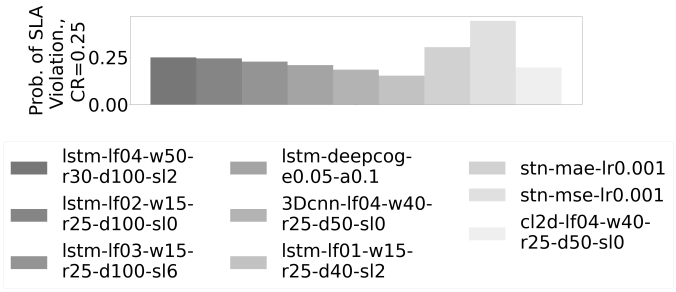



(a) Normalized $TC$ for $CR_{\frac{OP}{UP}} = 0.25$ (less is better) in BS A.

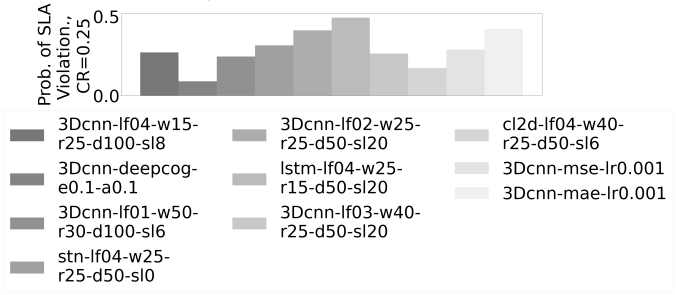



(b) Normalized $TC$ for $CR_{\frac{OP}{UP}} = 0.25$ (less is better) in BS B.

Fig. 6: Normalized $TC$ for two BSs for $CR_{\frac{OP}{UP}} = 0.25$




(a) Probability of SLA violation in A (less is better).




(b) Probability of SLA violation in B.

Fig. 7: SLA violation probability for $CR_{\frac{OP}{UP}} = 0.25$

33.2% worse, and the pair (LSTM, $L_{deepcog}$) has a penalty 3.7% worse. Figure 6b shows the normalized penalty for $B$. In this case, the (3Dcnn, $L_{04}$) pair performs the best. The (3Dcnn, $L_{deepcog}$) pair has a penalty 12.3% larger (worse), and the pair (3Dcnn, MSE) performs 73.3% worse.

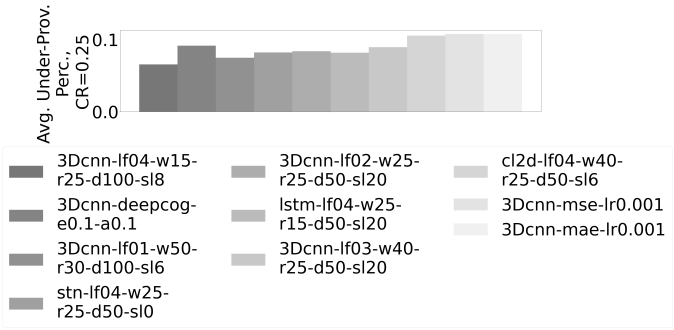
Fig. 7 shows the probability of SLA violations for both BSs, showing that the (3Dcnn, $L_{deepcog}$) pair has the smallest probability for $B$, despite being the (3Dcnn, $L_{04}$) pair the one that generates the smallest $TC$. In the case of $A$, the (LSTM, $L_{ecatp01}$) pair shows the smallest SLA violation probability. Following a similar reasoning to the case of $CR_{\frac{OP}{UP}} = 0.5$, we need to look at the avg. magnitude and probabilities of under- and over-provisioning for $B$, shown in Fig. 8.

In Fig. 8, the (3Dcnn, $L_{deepcog}$) pair presents the largest magnitude of under- and over-provisioning, with the larges probability of over-provisioning. The large magnitude of the (3Dcnn, $L_{deepcog}$) pair is what increases its penalty cost. Even considering that $CR_{\frac{OP}{UP}} = 0.25$, it still achieves a larger $TC$ due to the large amount of over-provisioning it generates. A similar thing happens to the (LSTM, $L_{ecatp01}$) pair in A, but we omitted those figures for lack of space.

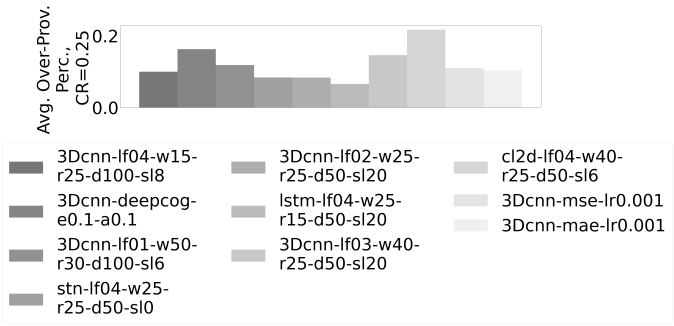
### B. Evaluating Predictions Across Multiple Base Stations

*1) Results for $CR_{\frac{OP}{UP}} = 0.5$:* Figure 9a shows the aggregated normalized $TC$ for $CR_{\frac{OP}{UP}} = 0.5$ across BSs. In this case, the (STN, $L_{01}$) pair performs the best. The (LSTM, $L_{deepcog}$) pair performs the worse with a $TC$ 41.0% higher.

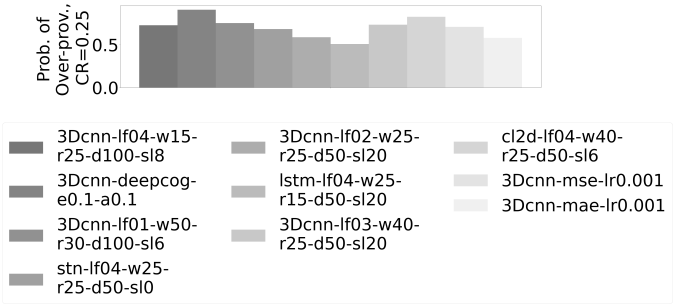
Figure 9b shows that the probability of SLA violation correlates with the $TC$ of each pair, except for (STN, $L_{04}$) that shows a lower probability than (STN, $L_{01}$). In order to understand this, we look at Figs. 9c, 9d, 9e which show that the first and second pair have a similar under-provisioning avg. magnitude, with (STN, $L_{04}$) having a larger one enough




(a) Average Magnitude of Under-provisioning.




(b) Average Magnitude of Over-provisioning




(c) Probability of over-provisioning.

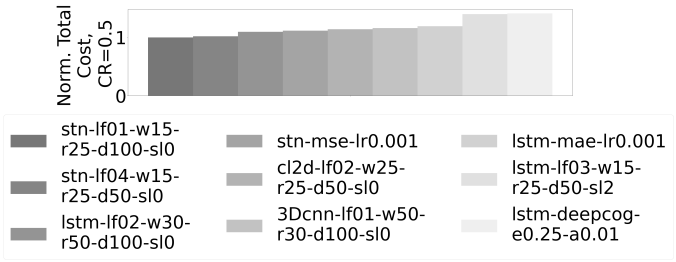
Fig. 8: Metrics for base station B for $CR_{\frac{OP}{UP}} = 0.25$

to push upwards its $TC$. This demonstrates further that the penalty for over-provisioning needs to be considered.

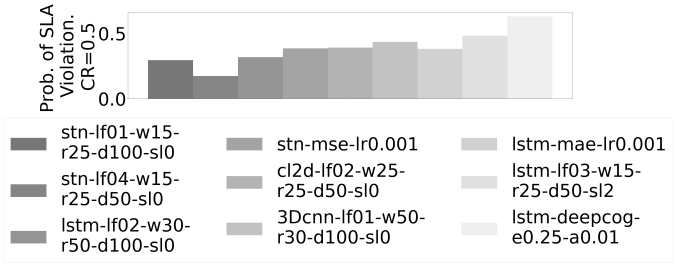
*2) Results for $CR_{\frac{OP}{UP}} = 0.25$:* Figure 10a shows the aggregated normalized penalty for $CR_{\frac{OP}{UP}} = 0.25$ across BSs, with the (STN, $L_{04}$) pair having the smallest $TC$, followed by (STN, $L_{01}$) with a $TC$ 9.6% larger (worse). The (3Dcnn, $L_{deepcog}$) pair with $\epsilon = 0.1$, $\alpha = 0.1$ has a $TC$ 27.3% higher, while (3Dcnn, MSE) performs 29.2% worse.

Figures 10b and 10a show that a reduction in the probability of SLA violations corresponds to a reduction in the $TC$. However, the (3Dcnn, $L_{deepcog}$) pair shows a lower probability of SLA violations than (STN, $L_{ecatp04}$). Fig. 11 shows that the (3Dcnn, $L_{deepcog}$) pair has a slightly larger probability and magnitude of over-provisioning, and a very comparable magnitude of under-provisioning. These three factors push upwards the $TC$ of the (3Dcnn, $L_{deepcog}$) pair.

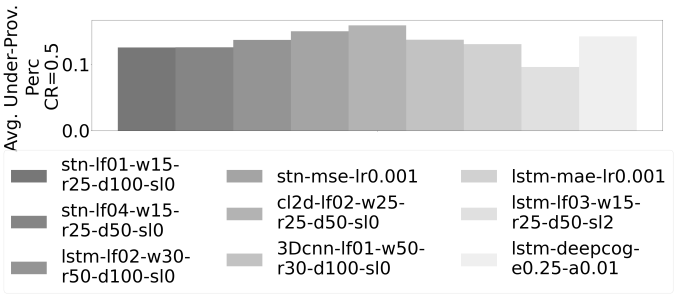
*3) Discussion:* The NARLEP-based loss $L_{04}$ generates the best results for the DNNs in $A$ and $B$ when considering $CR_{\frac{OP}{UP}} = 0.5$ and $0.25$, while the accuracy-oriented loss functions used as baselines, namely MSE and MAE, consistently generate the worst quality of prediction (high $TC$).



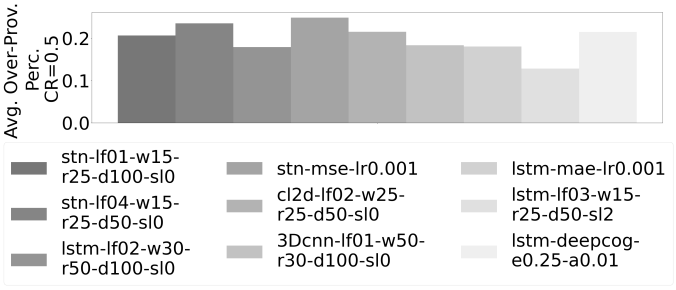

(a) Aggregated Normalized Total Penalty.



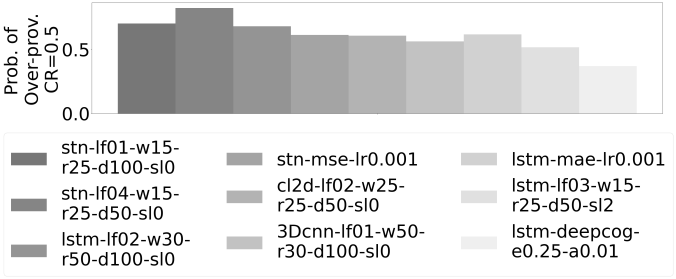

(b) Probability of SLA Violations.




(c) Avg. Magnitude of Under-provisioning.



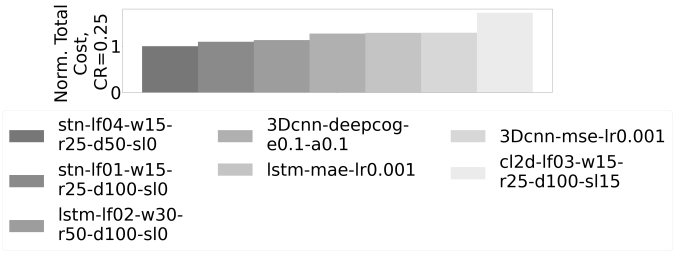

(d) Avg. Magnitude of Over-provisioning.



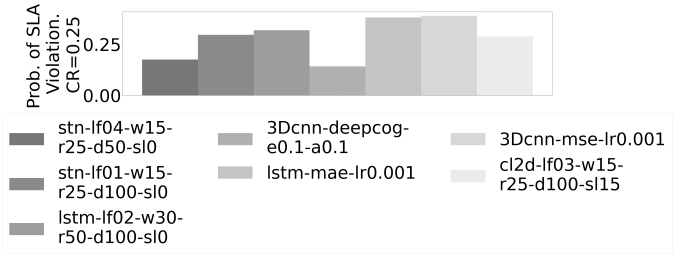

(e) Probability of Over-provisioning.

Fig. 9: Aggregate metrics for $CR_{\frac{OP}{UP}} = 0.5$ (less is better)

We also noted that the (3DCNN, $L_{deepcog}$) pair in $B$ for both values of $CR_{\frac{OP}{UP}}$ is less prone to under-estimate the demand (Figs. 4b and 7b, ), but very prone to over-estimate it. For all the parameters explored, it generated minimal costs for SLA violations at the expense of over-estimating demand

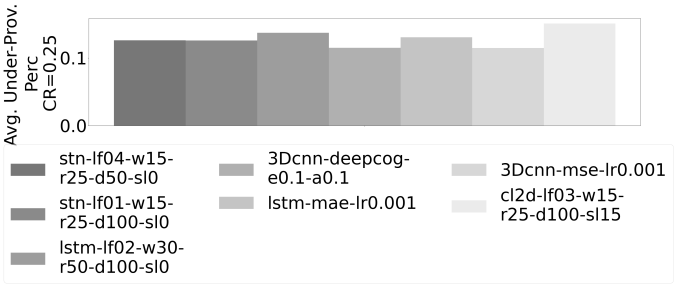



(a) Aggregated Normalized Total Penalty.

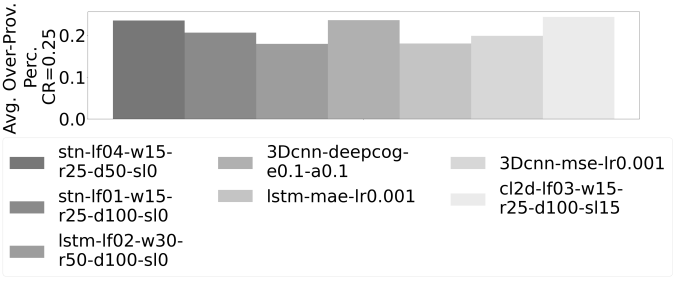



(b) Probability of SLA Violations.

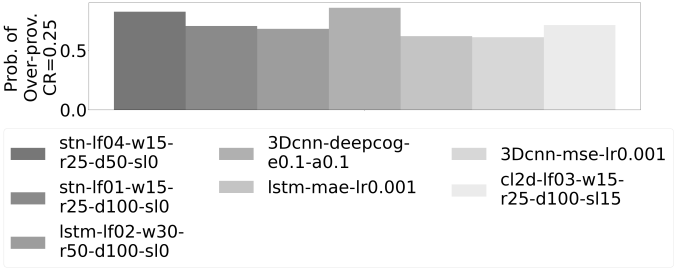
Fig. 10: Penalty and Probability of SLA violation for $CR_{\frac{OP}{UP}} = 0.25$ (less is better)




(a) Average under-provisioning magnitude.




(b) Average over-provisioning magnitude.




(c) Probability of over-provisioning.

Fig. 11: Metrics across BSs for $CR_{\frac{OP}{UP}} = 0.25$ (less is better)

i.e. causing inefficient resource utilization. Similar behavior regarding over-stimation is presented by $L_{01}$ in Fig. 7a.

In addition, we note also that for each individual BS,

different DNN architectures trained with NALDEP-based loss functions generated the least $TC$, being LSTM and CNN architectures for $A$ and $B$, respectively. This result shows that, for time-series forecasting, different DNN architectures to implement RDPs with knowledge embedding extract better the features present in the data. This capability of the DNNs is also affected by the sensitivity parameters $C_d$, $C_w$, $C_r$ and $y_s$. These data set features in question still need to be determined.

When aggregating inputs from multiple BSs, STNs with a NALDEP-based loss function generates the smallest $TC$. This result can be partly explained due to the fact that STN architectures exploit spatial correlation present in the data. In this case, the NARLEP-based $L_{01}$ loss generated the best prediction quality. Similarly to the individual cases of $A$ and $B$, the sensitivity to service degradation $C_d$ of the RDP is also 2 to 4 times higher than $C_r$ and $C_w$.

## VII. Conclusions

Our results for NALDEP show that embedding knowledge about network models and resource prediction and management increases the quality of the predictions, improving the proactivity of RMOs. This decreases the probability of SLA violations and increases resource utilization efficiency.

We showed that different (DNN, Loss Function) pairs using NALDEP-based loss functions generate the best prediction performance for the same slice in different BSs. The effectiveness of the (DNN, Loss Function) pairs is related to how well the DNN extracts features from the traffic profiles and the parameters of the NALDEP functions. When aggregating costs across BSs, we could observe that the STN architecture with NALDEP loss functions offered the best performance.

For future work, further refinement of the network and resource provisioning models will be required. A deeper exploration of the relation between individual BSs and aggregated TC using NALDEP is also of importance, since this will provide deeper analysis of the traffic profiles. Integration of NALDEP with an RMO is also a necessary future step.

## References


[1] 3GPP TS 23.501 v15.2.0 Release 15, "5G; System Architecture for the 5G System" (2018-06)

[2] A. Mpatziakas et. al., "AI-Based mechanism for the Predictive Resource Allocation of V2X related Network Services", in 18th Intl. Conf. on Network and Service Management, Greece, 2022, pp. 282-288.

[3] X. Foukas et al., "Network Slicing in 5G: Survey and Challenges", in IEEE Communications Magazine, vol. 55, no. 5, pp. 94-100, May 2017.

[4] J. X. Salvat et al. "Overbooking network slices through yield-driven end-to-end orchestration." In Proc. of the 14th Intl. Conf. on Emerging Networking EXperiments and Technologies, ACM. NY, USA, 353–365.

[5] D. Loghin et al., "The Disruptions of 5G on Data-Driven Technologies and Applications," in IEEE Trans. on Knowledge and Data Engineering, vol. 32, no. 6, pp. 1179-1198, 1 June 2020.

[6] C. Campolo et al., "5G Network Slicing for Vehicle-to-Everything Services," in IEEE Wireless Communications, vol. 24, no. 6, pp. 38-45, 2017.

[7] J. Deng, D. Adelberger and L. del Re, "Hybrid power train control with dynamic traffic prediction based on real-world V2X information," 2021 American Control Conf., New Orleans, USA, 2021, pp. 1644-1649.

[8] H. Qiu, M. Qiu and R. Lu, "Secure V2X Communication Network based on Intelligent PKI and Edge Computing," in IEEE Network, vol. 34, no. 2, pp. 172-178, March/April 2020.

[9] B. Fan et. al., "Deep Learning Empowered Traffic Offloading in Intelligent Software Defined Cellular V2X Networks," in IEEE Trans. on Vehicular Technology, vol. 69, no. 11, pp. 13328-13340, Nov. 2020.

[10] Guleng, Siri et al. "Edge-Based V2X Communications With Big Data Intelligence." IEEE Access 8 (2020): 8603-8613.

[11] Chu, Ping et al. "Semi-Persistent V2X Resource Allocation with Traffic Prediction in Two-Tier Cellular Networks." 2019 IEEE 89th Vehicular Technology Conf. (VTC2019-Spring) (2019): 1-6.

[12] V. S. Varanasi and S. Chilukuri, "Adaptive Differentiated Edge Caching with Machine Learning for V2X Communication," in 11th Intl. Conf. on Communication Systems & Networks (COMSNETS), Bengaluru, India, 2019, pp. 481-484.

[13] A. Okic et. al., "$\pi$-ROAD: a Learn-as-You-Go Framework for On-Demand Emergency Slices in V2X Scenarios," in IEEE Conf. on Computer Communications (INFOCOM), Vancouver, Canada, 2021, pp. 1-10.

[14] Nakao, A. et al. End-to-end Network Slicing for 5G Mobile Networks. J. Inf. Process., 2017, 25, 153-163.

[15] M. A. Khan et al., "Robust, Resilient and Reliable Architecture for V2X Communications," in IEEE Trans. on Intelligent Transportation Systems, vol. 22, no. 7, pp. 4414-4430, July 2021, doi: 10.1109/TITS.2021.3084519.

[16] W. Tong et. al., "Artificial Intelligence for Vehicle-to-Everything: A Survey," in IEEE Access, vol. 7, pp. 10823-10843, 2019.

[17] A. Borghesi, B. Federico and M. Milano. "Improving Deep Learning Models via Constraint-Based Domain Knowledge: a Brief Survey." ArXiv abs/2005.10691 (2020).

[18] L. Li et al. "Domain Knowledge Embedding Regularization Neural Networks for Workload Prediction and Analysis in Cloud Computing". J. Inf. Tech. Res. 11, 4 (October 2018), 137–154.

[19] J. Xu et al. "A Semantic Loss Function for Deep Learning with Symbolic Knowledge", in Proc. of the 35th Intl. Conf. on Machine Learning, 2018.

[20] L. Le et al. "Applying Big Data, Machine Learning, and SDN/NFV to 5G Traffic Clustering, Forecasting, and Management," in 4th IEEE Conf. on Network Softwarization and Workshops (NetSoft), Montreal, QC, 2018, pp. 168-176.

[21] A. Sang, and S. Li. "A predictability analysis of network traffic". In Proc. of the 19th Annual Joint Conf. of the IEEE Computer and Communications Societies, 1, 342-351 vol.1. 2000.

[22] D. Hisano et al. "Predictive Bandwidth Allocation Scheme With Traffic Pattern and Fluctuation Tracking for TDM-PON-Based Mobile Fronthaul," in IEEE Journal on Selected Areas in Communications, vol. 36, no. 11, pp. 2508-2517, Nov. 2018.

[23] T. Nunome and K. Furukawa, "The effect of bandwidth allocation methods on QoE of multi-view video and audio IP transmission," 2017 IEEE 22nd Intl. Workshop on Computer Aided Modeling and Design of Communication Links and Networks, Lund, 2017, pp. 1-6.

[24] N. Muralidhar et al. "Incorporating Prior Domain Knowledge into Deep Neural Networks," In IEEE Intl. Conf. on Big Data, Seattle, WA, USA, 2018, pp. 36-45.

[25] L.A. Garrido et al. "Context-Aware Traffic Prediction: Loss Function Formulation for Predicting Traffic in 5G Networks", in IEEE Intl. Conf. on Communications, Montreal, Canada, June 2021.

[26] A. Fendt et al. "A Network Slice Resource Allocation and Optimization Model for End-to-End Mobile Networks," 2018 IEEE 5G World Forum (5GWF), Silicon Valley, CA, 2018, pp. 262-267.

[27] K. Rao etal. "SmartSlice: Dynamic, self-optimization of application's QoS requests to 5G networks," 2021 8th Intl. Conf. on Software Defined Systems (SDS), Gandia, Spain, 2021, pp. 1-7.

[28] D. Bega et al. "DeepCog: Cognitive Network Management in Sliced 5G Networks with Deep Learning," IEEE Conf. on Computer Communications, Paris, France, 2019, pp. 280-288.

[29] Z. Wang et al. "Spatial-Temporal Cellular Traffic Prediction for 5G and Beyond: A Graph Neural Networks-Based Approach," in IEEE Trans. on Industrial Informatics, vol. 19, no. 4, pp. 5722-5731, April 2023.

[30] U. Paul et al., "Traffic-profile and machine learning based regional data center design and operation for 5G network," in J. of Communications and Networks, vol. 21, no. 6, pp. 569-583, Dec. 2019.

[31] C. Gutterman et al. RAN Resource Usage Prediction for a 5G Slice Broker. In Proc. of the 20th ACM Intl. Symp. on Mobile Ad Hoc Networking and Computing (Mobihoc '19). Association for Computing Machinery, New York, USA, 231–240.

[32] V. Sciancalepore, X. Costa-Perez and A. Banchs, "RL-NSB: Reinforcement Learning-Based 5G Network Slice Broker," in IEEE/ACM Trans. on Networking, vol. 27, no. 4, pp. 1543-1557, Aug. 2019.

[33] Clemente, D. et al. (2019). "Traffic Forecast in Mobile Networks: Classification System Using Machine Learning," 2019 IEEE 90th Vehicular Technology Conf. (VTC2019-Fall), Honolulu, HI, USA, pp. 1-5.

[34] Zhang, Chaoyun and Paul Patras. "Long-Term Mobile Traffic Forecasting Using Deep Spatio-Temporal Neural Networks." In Proc. of the 18th ACM Intl. Symp. on Mobile Ad Hoc Networking and Computing (2018).

[35] L. Chen et al. “Deep mobile traffic forecast and complementary base station clustering for C-RAN optimization.” J. Netw. Comput. Appl. 121 (2018): 59-69.
[36] W. Zhang et. al., ”Latency Prediction for Delay-sensitive V2X Applications in Mobile Cloud/Edge Computing Systems,” 2020 IEEE Global Communications Conf., Taipei, Taiwan, 2020, pp. 1-6.
[37] S. Barmpounakis et. al, ”AI-driven, QoS prediction for V2X communications in beyond 5G systems,” Computer Networks, Volume 217, 2022.
[38] F. Kavehmadavani et. al., ”Intelligent Traffic Steering in Beyond 5G Open RAN based on LSTM Traffic Prediction,” in IEEE Trans. on Wireless Communications, March 2023.
[39] M. Chen et. al, ”Intelligent Traffic Adaptive Resource Allocation for Edge Computing-Based 5G Networks,” in IEEE Trans. on Cognitive Communications and Networking, vol. 6, no. 2, pp. 499-508, June 2020.
[40] H.W. Kim et al., ”Dynamic bandwidth provisioning using ARIMA-based traffic forecasting for Mobile WiMAX.” Comput. Commun. 34, 1 (January, 2011), 99–106.
[41] C. Nichiforov, et al. ”Energy consumption forecasting using ARIMA and neural network models,” 2017 5th Intl. Symp. on Electrical and Electronics Engineering (ISEEE), Galati, 2017, pp. 1-4.
[42] S. Siami-Namini, N. Tavakoli and A. Siami Namin, ”A Comparison of ARIMA and LSTM in Forecasting Time Series,” In the 17th IEEE Intl. Conf. on Machine Learning and Applications (ICMLA), Orlando, FL, 2018, pp. 1394-1401.
[43] A. Borghesi, F. Baldo and M. Milano, ”Improving deep learning models via constraint-based domain knowledge: a brief survey”, arXiv:2005.10691, 2020.
[44] J. Yang and S. Ren, “Informed Learning by Wide Neural Networks: Convergence, Generalization and Sampling Complexity.” Intl. Conf. on Machine Learning (2022).
[45] J. Gou, et al. ”Knowledge Distillation: A Survey”. Int J Comput Vis 129, 1789–1819 (2021).
[46] G. Hinton, O. Vinyals, J. Dean. ”Distilling the Knowledge in a Neural Network”, NIPS Deep Learning and Representation Learning Workshop, 2015.
[47] M. Phuong and C. Lampert, ”Towards Understanding Knowledge Distillation”, in Proc. of the 36th Intl. Conf. on Machine Learning, vol. 97, 5142-5151, 2019.
[48] B. Sliwa, H. Schippers and C. Wietfeld, ”Machine Learning-Enabled Data Rate Prediction for 5G NSA Vehicle-to-Cloud Communications,” in IEEE 4th 5G World Forum, Montreal, Canada, 2021, pp. 299-304.
[49] M. Manalastas et. al., ”Machine Learning-Based Handover Failure Prediction Model for Handover Success Rate Improvement in 5G,” in IEEE 20th Consumer Communications& Networking Conf. (CCNC), Las Vegas, USA, 2023, pp. 684-685.
[50] F. Liang et. al., ”Towards Optimal Power Control via Ensembling Deep Neural Networks,” in IEEE Trans. on Communications, vol. 68, no. 3, pp. 1760-1776, March 2020.
[51] H. Tong, ”Functional linear regression with Huber loss”, Journal of Complexity, Vol. 74, 2023.
[52] P. Chen, G. Chen, S. Zhang, ”Log Hyperbolic Cosine Loss Improves Variational Auto-Encoder”, in Proc. of the 6th Intl. Conf. on Learning Representations; Vancouver, Canada, April 2018.
[53] P. Martinez-Julia, V. P. Kafle and H. Asaeda, ”Using the Total Cost of Ownership to Decide Resource Adjustment in Virtual Networks,” 22nd Conf. on Innovation in Clouds, Internet and Networks and Workshops (ICIN), Paris, France, 2019, pp. 329-335.
[54] L. Zanzi et al. ”OVNES: Demonstrating 5G network slicing overbooking on real deployments,” In IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, 2018, pp. 1-2.
[55] L. Bottou. ”Stochastic Gradient Learning in Neural Networks”. In Proceedings of Neuro-Nîmes 91, EC2, Nimes, France, 1991.
[56] M. Abadi et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
[57] Telecom Italia, 2015, ”Telecommunications - SMS, Call, Internet - MI”, Harvard Dataverse, V1.

**Luis A. Garrido** is a Senior Researcher at Iquadrat Informatica S.L since March 2020. He received his PhD degree from the Department of Computer Architecture of the Polytechnic University of Catalonia (UPC), Spain, in 2019. He holds an Electronics Engineering and Telecommunications degree (2011) and a M.Sc. in Electrical Engineering and Computer Science (2013) from Technological University of Panama and National Chiao Tung University (NCTU), Taiwan, respectively. Currently, he is actively involved in EU-funded research projects, while his main research interests include Low-Level Virtualization, Cloud Computing, Machine Learning, and 5G/6G Communication Networks.

**Anestis Dalgkitsis** received the Diploma degree (5 years) in Informatics & Telecommunications Engineering from University of Western Macedonia (UoWM), Kozani, Greece in 2018. Since October 2018, he works as a Research Engineer at Iquadrat and is involved in EU-funded research projects, while pursuing his Ph.D. degree at the Technical University of Catalonia (UPC), Barcelona, Spain. His main research interest are Network Function Virtualization Management and Orchestration, with Machine Learning algorithms and techniques.

**Dr. Kostas Ramantas** has received the Diploma of Computer Engineering, the MSc degree in Computer Science and Engineering and the PhD degree from the University of Patras, Greece, in 2006, 2008 and 2012 respectively. Up to now, he has been the recipient of two national scholarships and has been actively involved in multiple E.C. funded projects, such as SEMIoTICS, 5GMediaHub, performing joint research with many European research groups. His research interests are in modelling and simulation of network protocols, and scheduling algorithms for QoS provisioning. In June 2013, he joined IQUADRAT as a senior researcher, and has supervised 1 PhD student and many ESRs in the framework of Marie Curie RISE and ITN programmes (e.g., Spotlight, 5GAura). His work has been published in more than 30 conferences and Journals.

**Adlen Ksentini** is a professor in the Communication Systems Department of EURECOM. He is leading the Network softwarization group. He is involved in several EU projects related to Network Slicing and 5G, such as 5G!Drones and MonB5G. He is leading the Network softwarization group activities related to Network Slicing and Edge Computing. He has been involved in several H2020 EU projects on 5G, such as 5G!Pagoda, 5GTransformer, 5G!Drones and MonB5G. Adlen Ksentini research interests are on Network Sofwerization and Network Cloudification focusing on topics related to: network virtualization, Software Defined Networking (SDN), Edge Computing, Network slicing for 5G and beyond networks. He is interested on both system and architectural issues, but also on algorithms problems related to those topics, using Markov Chains, Optimization algorithms and Machine Learning (ML). Adlen Ksentini has received the best paper award from IEEE IWCMC 2016, IEEE ICC 2012, and ACM MSWiM 2005 conferences, and has been awarded the 2017 IEEE Comsoc Fred W. Ellersick (best IEEE communications Magazine’s paper).

**Christos Verikoukis** (Senior Member, IEEE) received the Ph.D. degree from UPC, Barcelona, Spain, in 2000. He is currently an Associate Professor (Tenure Track) with the University of Patras. He has authored more than 145 journal articles, 220 conference articles, three books, 14 book chapters, and five patents. He has participated in more than 30 competitive research projects, while he has supervised 19 Ph.D. students and 10 Postdoctoral researchers. He is a Member-at-Large and Vice-Chair of IEEE ComSoc GITC. He was the recipient of the the Best Paper Award at the IEEE ICC 2011 and 2020, the IEEE GLOBECOM 2014 and 2015, and the EuCNC 2016, and the EURASIP 2013 Best Paper Award of the Journal on Advances in Signal Processing.