RESEARCH-ARTICLE

A Framework for Resource Overprovisioning with Machine Learning to Maximise Revenue from 5G Core Network Slices

MAMUSHIANE LUSANI, University of Cape Town, Cape Town, Western Cape, South Africa

MWANGAMA JOYCE, University of Cape Town, Cape Town, Western Cape, South Africa

LYSKO ALBERT, The Council for Scientific and Industrial Research, Pretoria, Gauteng, South Africa

KOBO HLABISHI, The Council for Scientific and Industrial Research, Pretoria, Gauteng, South Africa

Open Access Support provided by:

University of Cape Town

The Council for Scientific and Industrial Research

Published: 09 December 2025

icARTi 2025: International Conference on Artificial Intelligence and its Applications
December 9 - 10, 2025
Port Louis, Mauritius

https://doi.org/10.1145/3774791.3774806
ISBN: 9798400721588

A Framework for Resource Overprovisioning with Machine Learning to Maximise Revenue from 5G Core Network Slices

Mamushiane Lusani
University of Cape Town
Cape Town, South Africa
RVHLUS001@myuct.ac.za

Mwangama Joyce
University of Cape Town
Cape Town, South Africa
joyce.mwangama@uct.ac.za

Lysko Albert
Council for Scientific and Industrial Research (CSIR)
Pretoria, South Africa
alysko@csir.ac.za

Kobo Hlabishi
Council for Scientific and Industrial Research (CSIR)
Pretoria, South Africa
hkobo@csir.ac.za

Abstract

Network slicing has emerged as a key enabler for delivering diverse services in 5G and beyond networks, where each slice must be provisioned with sufficient resources while maintaining operator profitability. Traditional admission control strategies are often conservative, leading to underutilization, while aggressive allocation risks violating service-level agreements (SLAs). This paper proposes a machine learning-based forecasting and admission control framework that leverages overprovisioning to balance utilization, revenue, and reliability. The framework combines CNN-LSTM forecasting for predicting slice resource demand with a policy that admits slice requests based on forecasted aggregate utilization scaled by an overprovisioning factor. To evaluate the approach, we use over 500 VM workload traces from the Materna dataset, grouping VMs into slices of ten. Results demonstrate that CNN-LSTM forecasting achieves consistent accuracy across diverse workloads, and that overprovisioning-based admission control improves net profit while reducing SLA violations compared to conservative baselines. These findings validate the feasibility of ML-driven overprovisioning for efficient and reliable slice admission in 5G networks.

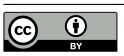
CCS Concepts

• **Networks** → **Network management**; • **Computing methodologies** → **Neural networks**.

Keywords

5G, Admission control, LSTM, CNN, Machine learning, Network slicing, Q-learning, Resource over-provisioning, Reinforcement learning.

ACM Reference Format:
Mamushiane Lusani, Mwangama Joyce, Lysko Albert, and Kobo Hlabishi. 2025. A Framework for Resource Overprovisioning with Machine Learning to Maximise Revenue from 5G Core Network Slices. In *2025 International Conference on Artificial Intelligence and its Applications (ICARTI 2025), December 09–10, 2025, Port Louis, Mauritius.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3774791.3774806



ICARTI 2025, Port Louis, Mauritius

ACM ISBN 979-8-4007-2158-8/25/12
https://doi.org/10.1145/3774791.3774806

1 Introduction

The concept of network slicing is aimed at enabling mobile network operators (MNOs) to tailor services according to the requirements of different network users. Network slicing is a fundamental component of the 5G architecture that allows for the division of a single physical network into multiple virtual networks, each tailored to meet specific service requirements. Facilitated by 5G networks, MNOs aim to deliver a diverse range of services with varying requirements, including enhanced mobile broadband (eMBB), ultra-reliable low latency communication (URLLC), and massive machine-type communication (mMTC). This capability not only supports a wide array of use cases, but also optimizes the use of network infrastructure. By optimizing resource allocation across various slices based on demand, operators can achieve improved profitability, greater operational efficiency, enhanced service customization, and improved cost-effectiveness. This concept is an innovative approach that unlocks new business opportunities for infrastructure providers (InPs), service providers (SPs), and other stakeholders within the telecommunication industry.

In spite of this, resource management remains a great challenge due to unpredictable demand. Due to variations in user activity, service usage, and application demands, the resource needs of the network slice change significantly over time. This may result in service-level agreement (SLA) violations with network slice tenants. In order to avoid violating the stringent SLA terms with tenants, MNOs often allocate resources based on peak potential traffic, which rarely materializes. This is known as resource over-provisioning, which is the allocation of more network resources, such as bandwidth, computing power, storage, than actually needed by the network slice to meet the SLA requirements. This often results in resource underutilization and wastage due to unpredictable user demands. This leads to a trade-off where MNOs must balance between the need for quality of service (QoS) and the cost of overprovisioning as well as the risk of resource underutilization. Therefore, the motivation of this article is to explore the transformative potential of integrating the fragment of the core network with machine learning (ML) to optimize resource overprovisioning. The main focus is to improve network slice resource management to maximize revenue gains for InPs while satisfying the QoS requirements of their tenants.

Due to the finite nature of core network resources, the intuition is to approve a slice request solely when it is projected that there will be no reduction in service quality for both the new slice and the

existing slices. This flexibility is crucial to serve diverse and evolving digital ecosystems and to foster innovation in areas such as smart cities, industrial automation, and the Internet of Things (IoT). The integration of network slicing and ML techniques creates a powerful combination that drives the efficiency and effectiveness of both technologies. ML improves the adaptability of network slices by providing real-time, data-driven resource allocation (RA). Furthermore, the concept of resource over-provisioning, which refers to the strategic allocation of more resources than the physical capacity of the system under the assumption that not all resources will be used simultaneously, can be effectively managed with ML to further optimize network utilization without compromising service quality. Therefore, the aim of this paper is to present a comprehensive framework that harnesses ML to optimize resource overprovisioning in network slicing. The main objective is to maximize revenue while maintaining high standard of service delivery. Thus, the main contributions of this article are summarized as follows:

- We propose a forecasting- and admission-control framework for resource overprovisioning in 5G networks that integrates CNN-LSTM demand prediction with policy-driven slice admission.
- We conduct an experimental evaluation using more than 500 VM workload traces from the Materna dataset, aggregated into slices of ten VMs, thereby grounding the framework in realistic workload dynamics.
- We present a simulation-based analysis of admission control under different overprovisioning factors, highlighting the trade-offs between revenue, penalties, and SLA compliance.
- We provide empirical evidence that ML-driven overprovisioning improves operator profit and reliability, validating the framework as a practical baseline for future reinforcement learning extensions.

The remainder of this article is organized as follows: Section 2, introduces the overall resource overprovisioning framework. Section 3 defines a slice request. Section 4 proposes a neural network-based demand forecasting framework and some preliminary results. Section 5 proposes an overprovissioning framework. Section 6 discusses the admission control framework, related works and reports the evaluation results. Section 7 discusses the limitations of our results and future work. Section 8 concludes the paper.

2 Overall Resource Management Framework

In the context of 5G Networks, the 5G core network (5GC) architecture, as defined by 3GPP (specified in TS 23.501), constitutes at least 10 virtual network functions. The 5GC can be deployed on bare metal or using virtualisation technologies such as containers or virtual machines (VMs), across standalone servers or public, private, or hybrid cloud infrastructure. Cloud-based deployments are advantageous for supporting infrastructure sharing through network slicing, following multi-tenancy principles and enabling the provisioning of infrastructure as a service to different tenants. In CN slicing, dynamically allocating resources (CPUs, memory, and bandwidth) can lead to idle capacity due to tenant overestimation [13]. Redirecting this idle capacity through resource overprovisioning can maximise revenue and improve resource utilisation. Figure 1 illustrates the key components of a system designed to admit as many network slice requests as possible while adhering to the service level agreements (SLAs) negotiated with the tenants. These components include the forecast engine, the overprovisioning module, and the admission control engine. In the subsequent sections, we describe each component in detail and related works.

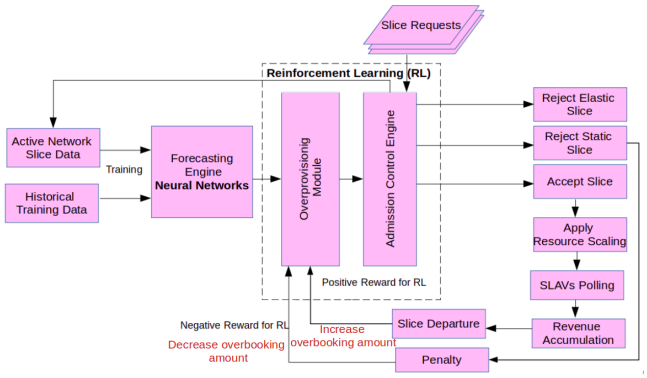


Figure 1: Proposed admission framework

3 Slice Request

Slice tenants lease an arbitrary virtual resources bundle, hereafter referred to as slice requests, denoted by $x \in X$, from the InP. Each accepted network slice $y \in Y$ is operated by the slice tenant to serve their customers. Two categories of slice requests are envisioned from tenants: Elastic and Fixed slices [8]. Fixed slice requests entail strict resource requirements with zero tolerance for SLA violations (SLAVs). Conversely, Elastic slices feature flexible SLAs safeguarded by a threshold value commonly referred to as a service degradation percentage. Every request for an Elastic slice is characterised by parameters such as the amount of resources needed (represented by U^r, where r is a component of computing (c, memory m and bandwidth resources b), the time duration that specifies its runtime operation (d), the cost per unit time for using the resource (p), the type of slice (either Elastic or Fixed) (z), and the percentage tolerance for SLAV (v). For Fixed slices, SLAV tolerance is zero.

Therefore, the InP defines and maintains a predetermined set of network slice classes (U^r, z, p) available to prospective tenants during resource bidding. At any point during the network window, the InP allocates resources to slice requests within the constraint of the maximum resource capacity of the underlying infrastructure. This is expressed in Equation 1, where C, M, and B denote the total available compute, memory, and bandwidth capacities of the infrastructure, respectively, and c_j, m_j, and b_j represent the corresponding amounts of compute, memory, and bandwidth resources allocated to slice j at a given time. Consequently, the available resource capacity at any given time during the network window is determined by Equation 2, where R_c, R_m, and R_b denote the remaining compute, memory, and bandwidth resources, respectively.

$$\sum_{j=1}^{n} c_j \leq C, \sum_{j=1}^{n} m_j \leq M, \sum_{j=1}^{n} b_j \leq B \tag{1}$$

$$R_c = C - \sum_{j=1}^{n} c_j, R_m = M - \sum_{j=1}^{n} M_j, R_b = B - \sum_{j=1}^{n} b_j \quad (2)$$

Slice requests from the tenant are modelled using a stochastic process known as the Poisson process. In this process, each slice request arrives at a rate denoted by λ_c, with the duration of each slice following an exponential random variable characterised by μ_c. Slice requests can be processed using one of three strategies: (i) the first-come-first-serve (FCFS) strategy [19][4], which guarantees fairness; (ii) the random strategy, which processes requests without following any specific order or priority; and (iii) a priority-based strategy, where Elastic slices are given precedence due to their flexibility, while Fixed slices are prioritised because of their higher revenue potential, as they typically involve premium services or applications. We advocate for the adoption of a priority-based strategy that prioritises the acceptance of Fixed slices. This not only enhances revenue gains but also simplifies the complexity and enhances the accuracy of forecasting slice resource demands, as the demands of Fixed slices remain constant throughout their lifecycle.

4 Online Demand Forecasting Framework

To determine the amount of resource overprovisioning possible in the 5GC, it is crucial to accurately forecast future resource demands of active and scheduled/reserved Elastic slices to minimise SLAVs beyond tolerated thresholds, which could result in penalties and revenue reduction for the InP. There has been a plethora of studies exploring the use of machine learning and probabilistic approaches to address dynamic resource management challenges based on workload forecasting in cloud environments for use cases such as energy consumption prediction, online workload profiling, thermal management, dynamic VM placement, SLA-based VM management, and QoS-aware resource provisioning. Given the vast volume and complexity of 5G data, alongside the non-linear variations in resource demand patterns, many studies advocate for the application of techniques based on neural networks, including deep learning neural networks (DNN) and recurrent neural networks (RNN). In this section, we discuss several noteworthy studies that have employed neural networks for demand forecasting in 5G networks. Additionally, we propose a forecasting framework for supporting the resource overprovisioning use case.

DNN is a feedforward neural network (FFNN) consisting of three primary layers: the input layer, one or more hidden layers, and the output layer. Data are fed into the network through the input layer, as an input vector of features $x = (x_1....x_n)$, processed through the hidden layers $h = (h_1....h_n)$, to generate the results from the output layer $y = (y_1....y_n)$, without a memory mechanism. Equations 3 and 4 capture this process, where W denotes weight matrices between consecutive layers and b_h and b_y represent the bias vector of the hidden and output layers, respectively.

$$h = f(Wx + b_h) \quad (3)$$

$$y = g(Wh + b_y) \quad (4)$$

RNNs are quite similar to DNN except that they incorporate loops within their architecture, allowing outputs from layers to cycle back into their network for better memory, making them particularly effective for network slice resource demand forecasting. In the RNN model, the network processes an input sequence $x = (x_1....x_T)$ to compute the corresponding sequences of hidden vectors $h = (h_1....h_T)$ and output vectors $y = (y_1....y_n)$. This computation involves iterating through Equations 5 and 6 from $t = 1$ to T.

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (5)$$

$$y_t = W_{hy}h_t + b_y \quad (6)$$

The problem with RNN is the vanishing gradient problem, which makes it difficult for the earlier layers of the network to learn meaningful representations of the input data, which hinders the overall training process. Specialised architectures such as long short-term memory (LSTM) [11] and gated recurrent units (GRUs) [7] have been developed to mitigate this issue, by incorporating memory units that enable the network to retain information over time.

In their study, the authors of [2] implement and benchmark both DNN and RNN, specifically LSTM, for mobile network traffic forecasting in a 5G context. They aim to proactively scale-in and scale-out AMF resources in a virtualised environment to optimise QoS. Their results demonstrate LSTM's superiority over DNN, with a performance advantage of 10%. However, the authors suggest DNN for scenarios with more neurones and more training data, which may not always be the case.

Another study [18] utilises data traces–GEANT and Abilene [1]–to explore RNN architectures, including standard RNN, GRU, and LSTM. These architectures are applied to predict network traffic volume and forecast future packet classifications and distributions. The results indicate that LSTM and GRU exhibit comparable effectiveness in volume prediction tasks, particularly when dealing with intricate and non-linear time series data. However, RNN shows a higher susceptibility to overfitting. Conversely, RNNs outperform other models in terms of packet classification accuracy, followed by GRU.

The authors of [10] developed and applied an augmented LSTM architecture (called CAT-LSTM), incorporating context embedding, aspect embedding, and attention to forecasting VNF resource demands, CPU usage in particular for VNFs in a service function chain (SFC). The prediction utilises service function chaining (SFC) data, comprising details about VNFs within the same chain. This data encompasses CPU, memory, and disk resources, as well as chain IDs, length, historical SFC information, and resource usage. Evaluating CAT-LSTM against the basic LSTM demonstrated that CAT-LSTM is better adapted to predict the CPU resource demands of VNFs. This effectiveness is attributed to the fact that the CPU usage in the VNFs tested showed frequent irregular fluctuations, which were not adequately addressed by the basic LSTM.

Different tools are available to evaluate the performance of forecasting models. These include the mean absolute percentage error (MAPE), mean squared error (MSE), root mean squared error (RMSE), and R-squared (R_2). The authors of [23] benchmark the performance of artificial neural networks (ANN), Long Short-Term Memory (LSTM), and statistical methods, namely the autoregressive integrated moving average (ARIMA) and the simple moving average (SMA) for a network traffic forecasting use case. Their

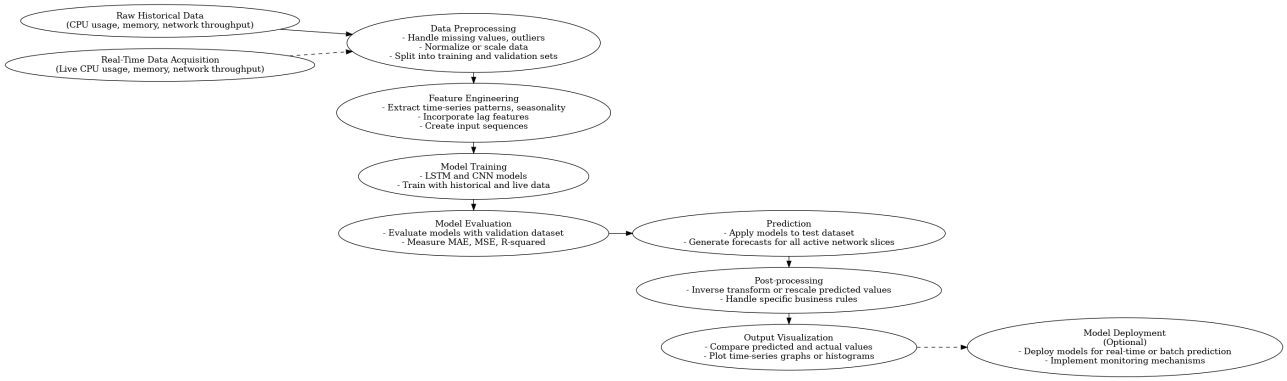


Figure 2: Proposed forecasting framework

analysis revealed that the ANN and LSTM models exhibited superior accuracy and reliability compared to statistical approaches. Interestingly, while both ANN and LSTM performed well, the ANN model showed the added advantage of faster forecasting, making it a suitable choice for applications requiring real-time predictions.

Finally, in the study [17], a hybrid approach is introduced, which combines convolutional neural networks (CNN) and LSTM networks to predict workload in a cloud environment. CNN was used for feature engineering, identifying patterns and relationships in the time series data, such as recurring spikes or trends in CPU usage. Subsequently, LSTM was employed to generate predictions by analysing the complex and nonlinear relationships between past and present observations of CPU usage over time. The proposed model demonstrates an improvement in accuracy ranging from approximately 3.8% to 10.9%, and a reduction in the percentage error rate of 7% to 8.5% compared to the vector autoregressive model with GRU and multi-layer perceptron (MLP).

Based on the above findings and reported performance results, we propose the use of LSTM and CNN for demand forecasting in 5GC network slicing, based on both historical and real-time metrics such as CPU, memory, and network throughput. Figure 2 summarises the proposed framework for forecasting the resource requirements of active and scheduled 5GC network slices. The interquartile range (IQR) and Z-score approaches are recommended for outlier removal during preprocessing. Furthermore, Min-Max scaling or standardisation is suggested to normalise the data within the range [0,1].

4.1 Evaluation

For the evaluation of our proposed forecasting models, we employed the Materna Grid Workloads Trace dataset [3], which captures real-world resource usage from 520 virtual machines in a distributed cloud environment. Each trace spans approximately one month of operation with more than 8,000 time-stamped entries, including CPU, memory, and disk I/O metrics, thus reflecting both regular utilization patterns and bursty workload spikes. In this study, we used the workload traces labeled "406.csv" and "520.csv" for detailed model evaluation, while conducting sensitivity analysis across 300 VM workload traces to assess the robustness and generalizability of the models.

Hyperparameter optimization was an essential step in improving forecasting accuracy. While grid search is generally considered the best option for exhaustive exploration of parameter combinations, its computational cost makes it less practical for rapid experimentation and operational deployment. Grid search is ideal for offline benchmarking because it thoroughly explores the parameter space, but in practice it is computationally expensive and does not scale well when retraining models or adapting them dynamically in production. Instead, we adopted randomized search, which provides a trade-off by sampling a subset of configurations to achieve near-optimal results with significantly lower runtime. This choice reflects the context of forecasting for network slice overprovisioning, where computational efficiency and timely retraining are more critical than absolute optimality. In practice, grid search may still be used in offline benchmarking, while randomized search offers a more realistic solution for adaptive operational environments.

All experiments were executed on a workstation running Ubuntu 22.04 LTS (64-bit) with an Intel Core i7-12800H CPU (14 cores, 20 threads, up to 4.8 GHz) and 64 GB RAM, ensuring sufficient computational capacity to evaluate both lightweight baselines and deep learning models.

4.2 Results

The evaluation of the CNN+LSTM model across different VM workload traces highlights the diversity of behaviors that directly influence slice-level resource forecasting. Figure 4 shows results for VM 406, which exhibits highly bursty and irregular CPU usage patterns. In this case, the model achieves an RMSE of 0.0119, MAPE of 8.61%, and R^2 of 0.9076, indicating that while overall variance is well captured, the model struggles to predict sharp bursts accurately. These bursts, when mapped into the slicing context, represent unpredictable service demands such as sudden traffic surges in ultra-reliable low-latency communication (uRLLC) slices. Accurately forecasting such behavior remains a challenge since anomalies disrupt temporal dependencies learned by the model.

By contrast, Figure 3 presents results for VM 520, which displays more regular and periodic workload patterns. The best parameter configuration was found to be filters=128, kernel_size=3, lstm_units=32, dropout=0.3, batch_size=32, epochs=50 under which the model achieved an RMSE of 0.0092, MAPE of 43.47%, and

R^2 of 0.9626. Despite a higher MAPE caused by underestimation of some peak values, the model achieves strong variance explanation and close alignment between predicted and actual workloads. This makes CNN+LSTM particularly well suited for slices such as enhanced mobile broadband (eMBB) or massive machine-type communication (mMTC), where traffic is more periodic and stable.

From a slicing perspective, these findings indicate that CNN+LSTM provides robust forecasts for slices with stable or periodic usage but has limitations in bursty environments. This does not weaken its suitability as a baseline model for slice resource forecasting.Rather, it highlights the need for hybrid strategies. Future work could involve combining CNN+LSTM with anomaly detection to handle rare bursts.

An important implication of these findings is the necessity of conducting sensitivity analysis across multiple VM workload traces. Since slices in operational 5G environments may carry diverse traffic types, ranging from highly predictable to highly irregular, evaluating models against a wide spectrum of workloads ensures that performance is not overstated for only one slice category. Sensitivity analysis therefore provides a more comprehensive assessment of generalizability, helping to identify the operational contexts in which CNN+LSTM excels, and where complementary mechanisms are needed.

Figure 5 illustrates the distribution of RMSE values obtained from this sensitivit analysis across over 500 VMs. The narrow interquartile range and low median RMSE confirm that the CNN+LSTM model maintains consistent performance for the majority of workload types. A limited number of outliers with higher error values reflect cases where the model struggles, likely due to abrupt or highly varible traffic. This variability reinforces the importance of designing slice-specific forecasting pipelines. Overall, the results validate CNN+LSTM as a strong baseline for forecasting in network slices with relatively stable behavior, while also highlighting the edge cases where forecasting confidence may require reinforcement through hybrid or adaptive methods. While the CNN+LSTM achieved lower prediction error compared to baseline models, even small underestimations of CPU or memory demand can lead to SLA violations once overprovisioning is applied. Conversely, conservative forecsts reduce admission ratios and system revenue. This underscores the fact that forecasting accuracy directly shapes admission control effectiveness in overprovisioned 5G systems.

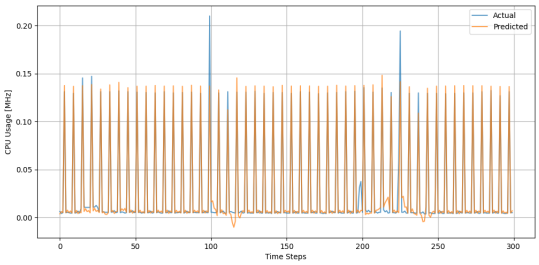


Figure 3: CPU prediction for VM "520.csv"

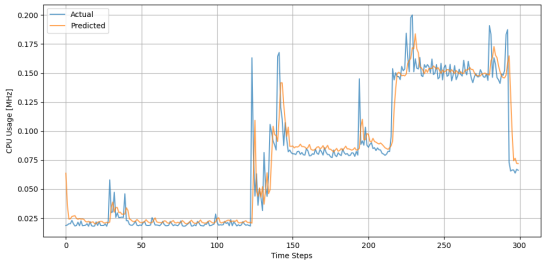


Figure 4: CPU prediction for VM "406.csv"

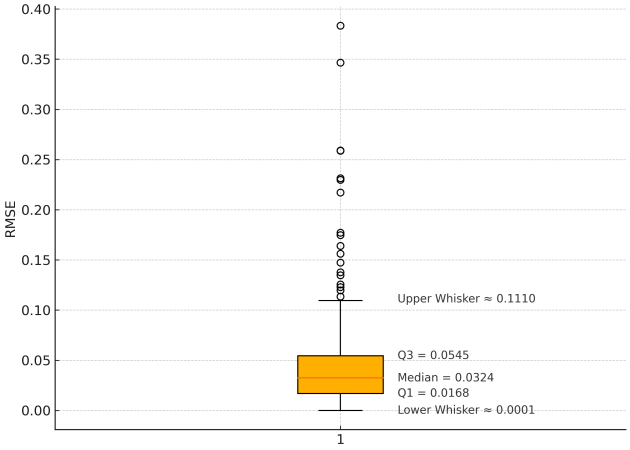


Figure 5: Box Plot of RMSE Across ~500 Virtual Machines

5 Resource Overprovisioning

In this section, we present the resource overprovisioning framework underpinned by a heuristic algorithm. The overall goal of overprovisioning is to maximise InP revenue, subject to constraints on resource availability and SLAVs. Before accepting new slice requests, the infrastructure manager, particularly the admission control engine, should be equipped with information on both current and forecasted resource demands (F_r) of all active Elastic network slices. This information is sourced from the online demand forecasting agent.

The online demand forecasting agent is not always perfect. It can either under-predict, over-predict, or accurately predict resource demands in future network windows. Therefore, in case of under-prediction, the InP allocates more resources than available, resulting in SLAVs and penalties. In case of over-prediction, there is a benefit, an opportunity to overprovision, by accepting more network slices that borrow resources from Elastic network slices.

A significant hurdle in overprovisioning is accurately identifying the amount of 5GC resources (CPU, memory, and network throughput, in our scenario) that can be borrowed. Techniques such as feedback control loops can be used to dynamically regulate resource overprovisioning. The authors in [21][20] propose the use of a reinforcement learning (RL) feedback control mechanism that automatically converges the overprovisioning process to the best overprovisioning amount. At any point in time, the InP must

refrain from implementing resource overprovisioning if it results in penalties. This is captured by Equation 7, where $opf_k^r(t-1)$ is the overprovisioning factor for active Elastic slice k in the previous network time window denoted by $t-1$, and p_k^r is the penalty associated with slice k and its resources.

$$\text{opf}_k^r(t) = \begin{cases} \text{opf}_k^r(t-1) + \xi, & p_k^r = 0 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

An important constraint to note is that the InP must never borrow resources from "Fixed" network slices. To determine the appropriate overprovisioning amount, the amount of resources allocated to Fixed slices at time t must first be subtracted from the total resource capacity, as captured by Equation 8, where $L_r(t)$ denotes the net available capacity, $T_r(t)$ denotes the total system capacity, and $C_k^r(t)$ denotes the total resources allocated to Fixed slices. The initial overprovisioning factor is then calculated as shown in Equation 9, where $O_r^i(t)$ denotes the overprovisioning factor at the beginning of a network window, and $F_r(t)$ is the forecasted resource demand for Elastic slices. If the value of $O_r^i(t)$ exceeds one(1), then overprovisioning can be applied; otherwise, it is not applied since doing so would incur penalties. To calculate the optimal overprovisioning factor $O_r^f(t)$, RL is recommended. The final overprovisioning factor is calculated by rewarding actions that minimise SLAVs and penalising those that result in SLAVs.

Based on the outcome of each action, whether it resulted in a SLAV or not, the regulating factor given by γ is recalculated and applied to $O_r^i(t)$ to determine the final overprovisioning factor, using Equation 10. For example, if an SLAV is encountered, a small predetermined quantity ξ is subtracted from the regulating factor (γ); otherwise, ξ is added to γ. The value of ξ should be as small as possible to gradually increase or decrease the overprovisioning factor, thereby avoiding excessive overprovisioning. The amount of resources pool allocated ($G_r^k(t)$) to active Elastic slice at time t is determined using Equation 11. Figure 6 shows the proposed framework for overprovisioning resources.

$$L_r(t) = T_r(t) - \sum_{k \in F} C_k^r(t) \tag{8}$$

$$O_r^i(t) = \frac{L_r(t)}{F_r(t)} \tag{9}$$

$$O_r^f(t) = (\gamma + \xi) \times O_r^i(t) \tag{10}$$

$$G_r(t) = L_r(t)(1 - O_r^f(t)) \tag{11}$$

6 Admission Control

An admission control engine makes decisions to accept or reject new slice requests based on the outcome of the overprovisioning and forecasting processes. Given that Fixed network slices generate higher revenue compared to Elastic slices, the admission control engine should prioritise the acceptance of Fixed network slices. Therefore, the goal is to maximise the InP's revenue by accepting as many Fixed network slices as possible while minimising SLAVs on active and scheduled Elastic slices. To address this, we formulate the problem using the RL, where the goal is to maximise reward (revenue in our case) and minimise penalties. The state space vector is defined to include the net available resource capacity (V_r),

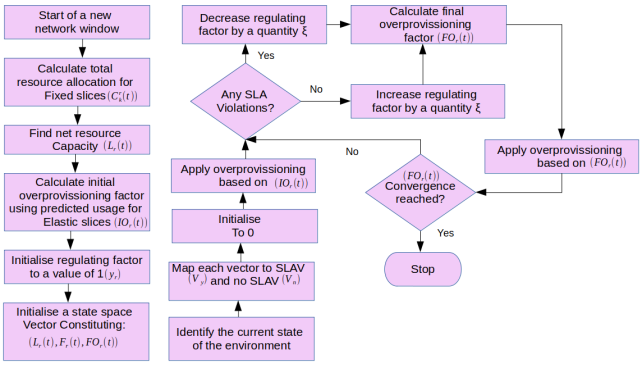


Figure 6: Proposed Overprovisioning Framework

slice type (Fixed or Elastic) denoted by (z), and the requested resources (U^r) as defined in Section 3. Significant efforts have been made in the literature to address the admission control problem in 5G network slicing, employing various approaches, including RL and probabilistic approaches. Some notable references in this area include [15], [19], [6], [9], [5], [8], [16] and [13].

The study [13], addresses the admission control problem by employing an overprovisioning policy based on the Benders decomposition algorithm and a sub-optimal heuristic that expedites decision-making. They leverage a real mobile operator dataset and demonstrate a revenue gain three times higher than the case without overprovisioning. Another study [22] designs an admission control system based on spatial distribution and RL to drive the system toward an optimal state. The system relies on forecasted user traffic and distribution to reconfigure the resource allocation for network slice requests, thus allowing the system to handle more slice requests. In [19], the authors propose a slice admission policy based on predictions of synthetic big data analytics (BDA) and the FCFS admission strategy. The simulation results show that using BDA predictions leads to a 50.7% increase in profit, compared to a slice admission policy without BDA.

The study [6] proposes an LSTM-based forecasting engine to aid admission decision-making and a Mondrian Random Forest algorithm for the reconfiguration of resource networks. This helps in dynamically reallocating resources from lower-priority slices (Elastic slices) to higher-priority slices (Fixed slices). The study [14] proposes DeepAR, a probabilistic forecasting method, to predict the resource requirements of the slice from a real dataset and admits the slices based on the availability of resources. Other studies, [9] and [8], present an algorithm for the admission and allocation of network slice requests, aiming to maximise revenue while ensuring SLA guarantees for tenants. Their contributions encompass a theoretical model that delineates the admissibility zone for a network-slicing-enabled 5G RAN, an examination of the system using a Semi-Markov Decision Process for revenue optimisation, and the creation of an adaptive algorithm based on deep Q-learning, which attains almost optimal results. In [5], a genetic algorithm is employed for intelligent slice admission control. The results demonstrated that the genetic algorithm outperformed Q-Learning. However, compared to Q-Learning, genetic algorithms have drawbacks:

they rely on quantised fitness values, converge more slowly, struggle with exploration in high-dimensional spaces, are less suitable for continuous spaces, and find it challenging to adapt to dynamic environments [12].

The study [24] introduces a mechanism for admission control that includes two approaches: one utilising reinforcement learning, named SARA, and another employing deep reinforcement learning, termed DSARA, to optimise revenue and resource utilisation in the 5GC domain. The studies [15], [16], [9] focus on the admission of network slices in the RAN domain. Meanwhile, [13] and [5] focus on end-to-end slice admission while [6] does not specify the domain scope.

We propose the use of Q-Learning, a model-free RL algorithm used to set a policy that instructs an agent on the appropriate actions to take in various situations. This algorithm does not require a model of the environment and can handle problems with stochastic transitions and rewards. Q-learning is based on the Bellman equation (shown in Equation 12), which expresses the value of a state as the sum of the immediate reward ($R(s, a)$) for taking action a in state s.The learning rate (α), controls the extent to which newly acquired information overrides old information, while γ acts as the discount factor, which assesses the significance of future rewards. Additionally, $\max(Q(s', a'))$ represents the highest Q-value achievable in the subsequent state across all potential actions.

$$Q(s, a) = (1 - \alpha)Q(s, a) + \alpha\{R(s, a) + \gamma max(Q(s', a'))\} \quad (12)$$

In the context of network slice overprovisioning, we propose Markov decision process (MDP) to model the admission control problem using Q-Learning. MDP requires definitions of parameters, namely, the environment, the state space, actions, and rewards. For the slice overprovisioning problem, we define the state space to consist of the resource demand of new slices, the slice category (whether Elastic or Fixed), and the available resource capacity. The environment is defined based on the total system capacity as given in Equation 1. There are two possible actions (a) exist: accepting a slice request or rejecting a slice request. Two events can trigger state transitions: acceptance of a network slice request, which generates a reward; or departure of a slice, which triggers state transitions with no reward. A network slice departure does not trigger state transitions. Two types of rewards are possible, a negative reward (penalty, given by Equation 14) resulting from under-predicting the future demands of active Elastic slices, resulting in SLAVs or positive reward (revenue gain, given by Equation 13) resulting from over-prediction of future demand. In Equations 13 and 14, I_i is the revenue associated with each incoming slice i, p_i is the unit price of each slice, d_i is the slice duration, L_k is the penalty associated with a violating active slice k, and Z_k, d_k represent the per-unit penalty price and the duration of service degradation, respectively.

$$I_i = p_i d_i \quad (13)$$

$$L_k = Z_k d_k \quad (14)$$

The overall objective of Q-Learning in this context is to optimise long-term revenue by increasing the acceptance of Fixed slices due to their higher profitability. This is achieved by applying a fake penalty, denoted as V, to the agent for each rejected Fixed slice. Figure 1 presents the proposed Q-learning framework for the overprovisioning use case. The optimal policy aims to maximise $Y(t)$ at any given time t, as determined by Equation 15.

$$Y[t] = Y[t-1] + \sum_{i \in A, t} I_i - \sum_{k \in E, t} L_k \quad (15)$$

6.1 Evaluation

The evaluation of the proposed overprovisioning-based admission control framework followed a structured methodology consisting of data preparation, system capacity definition, admission control, and performance evaluation:

- **Data preparation and forecasting:** Materna VM workload traces were used to build slice demand profiles. Slices were formed by grouping 10 VMs (inspired by [20], with per-slice demand computed as the aggregate CPU and memory of the group. All workloads were normalized to the range $[0, 1]$. For each slice, both true demand and a forecasted profile were generated using the CNN-LSTM hybrid predictor.
- **System capacity definition:** The operator's total capacity was derived from the cumulative demand of all slices. CPU and memory demands were summed across slices, and the 95^{th} percentile of the aggregate distribution was taken as the nominal system capacity. This follows network dimensioning practice, where resources are provisioned for typical peak loads while excluding rare extremes.
- **Admission control with overprovisioning:** Slice requests were generated following a Poisson arrival process with arrival rate λ. Each request was randomly assigned as elastic or inelastic, carrying both a forecasted demand (for admission decisioning) and true demand (for SLA validation). A request was admitted if the cumulative forecasted load of active slices plus the new request was within the system capacity scaled by an overprovisioning factor (OPF). Otherwise, it was rejected. Accepted slices were assigned random lifetimes, after which resources were released. SLA compliance was evaluated by comparing true demand against forecasted allocations.
- **Performance metrics:** The framework was evaluated in terms of (i) total revenue from admitted slices, (ii) penalties for rejecting inelastic requests, (iii) penalties from SLA violations, (iv) net profit (revenue minus penalties), and (v) the number of SLA violations. Metrics were reported for OPF values ranging from 1.0 to 1.3 to capture the trade-off between utilization and reliability.

6.2 Results

The results in Figures 7 and 8 illustrate the trade-off between revenue, penalties, and service reliability under different OPFs.

As expected, increasing the OPF led to a reduction in SLA violations. With OPF=1.0 (no overprovisioning), the system exhibited over 2,300 violations, while at OPF=1.3 violations decreased to below 1,900. This counterintuitive reduction occurs because higher OPFs allowed more slices to be admitted, smoothing aggregate utilization and preventing frequent rejections that would otherwise push admitted slices into SLA breaches.

From an economic perspective, revenue initially dipped at OPF=1.1 but rose steadily thereafter, surpassing the baseline at

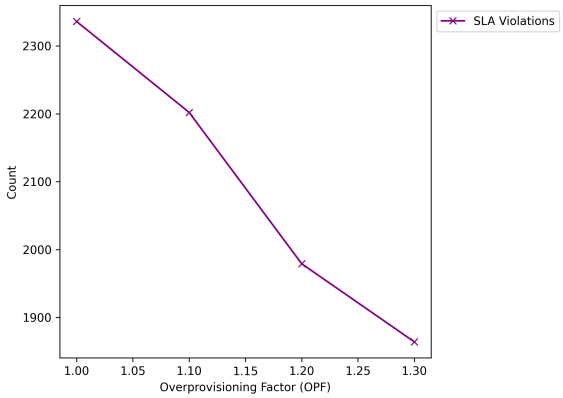


Figure 7: SLA violations under different OPFs.

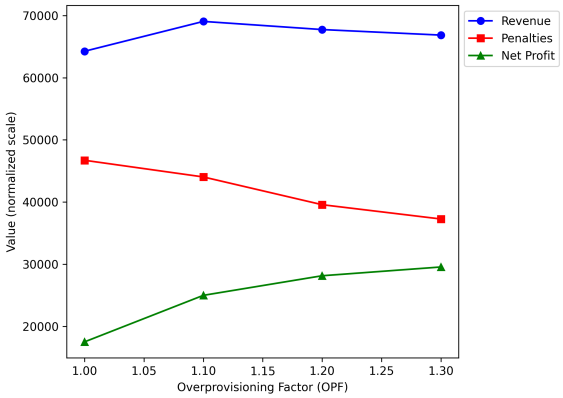


Figure 8: Revenue, penalties, and net profit as a function of OPFs.

OPF=1.2 and OPF=1.3. Penalties consistently declined with increasing OPF, driven both by reduced SLA violations and fewer forced rejections of inelastic slices. As a result, net profit improved significantly at higher OPFs, with the steepest increase between OPF=1.1 and OPF=1.2. These results suggest that conservative admission control (OVF=1.0) is economically suboptimal, while controlled overprovisioning provides measurable benefits in both profit and reliability

7 Discussion and Future Work

While the proposed CNN-LSTM and overprovisioning-based admission framework demonstrated measurable improvements in both forecasting accuracy and profitability, several limitations remain. Forecasting was based on CNN-LSTM predictions, and errors may have influenced admission decisions. Penalties were modeled as fixed constants, whereas in practice they depend on SLA terms and operator policies. In addition, slices were modeled as uniform groups of ten VMs and only CPU and memory resources were considered. These simplifications limit direct generalization to real networks.

Future work should address these limitations by incorporating adaptive or ensemble forecasting methods, modeling penalties in a SLA-driven manner, and extending the resource model to include bandwidth and storage. Reinforcement learning-based admission policies also present an opportunity to improve decision-making under uncertainty, while evaluating heterogeneous slice portfolios with variable sizes and priorities would bring the framework closer to operational 5G deployments.

8 Conclusion

This paper reviewed state-of-the-art techniques for slice admission control and proposed a machine learning-driven framework that combines CNN-LSTM forecasting with an overprovisioning-based admission policy. By grounding the evaluation in real-world Materna workload traces, we demonstrated that the proposed approach delivers both accuracy in forecasting and tangible improvements in economic performance. Results show that while conservative admission control leads to frequent SLA violations and lower revenue, controlled overprovisioning significantly increases net profit and reduces penalties by balancing admitted load against available capacity.

These findings confirm the potential of ML-driven overprovisioning to maximize infrastructure provider revenue while maintaining service quality in multi-tenant 5G networks. Future work will extend the model to include bandwidth and storage resources, refine penalty modeling based on SLA tiers, and investigate reinforcement learning policies capable of adapting admission decisions to heterogeneous slice portfolios and dynamic traffic conditions.

Acknowledgments

The authors acknowledge the support of the Council for Scientific and Industrial Research (CSIR) and the University of Cape Town. Portions of text were refined using ChatGPT (OpenAI; Accessed 24 Oct 2025) for grammar and clarity improvement. The authors reviewed and edited all AI-assisted text and take full responsibility for the content.

References

[1] 2020, Last accessed 17 April 2024. GEANT/Abilene. https://openresearch.surrey.ac.uk/esploro/outputs/99514653002346
[2] Imad Alawe, Adlen Ksentini, Yassine Hadjadj-Aoul, and Philippe Bertin. 2018. Improving traffic forecasting for 5G core network scalability: A machine learning approach. *IEEE Network* 32, 6 (2018), 42–49.
[3] AtLarge Research. 2025. GWA-T-13 Materna. https://atlarge-research.com/gwa-t-13/. Accessed: 2025-09-30.
[4] B. Han et al. 2019. A utility-driven multi-queue admission control solution for network slicing. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 55–63.
[5] B. Han et el. 2018. Admission and congestion control for 5G network slicing. In *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 1–6.
[6] Tulja Vamshi Kiran Buyakar, Harsh Agarwal, Bheemarjuna Reddy Tamma, and A Antony Franklin. 2020. Resource allocation with admission control for GBR and delay QoS in 5G network slices. In *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE, 213–220.
[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).
[8] D. Bega et al. 2017. Optimising 5G infrastructure markets: The business of network slicing. In *IEEE INFOCOM 2017-IEEE conference on computer communications*. IEEE, 1–9.
[9] D. Bega et al. 2019. A machine learning approach to 5G infrastructure market optimization. *IEEE Transactions on Mobile Computing* 19, 3 (2019), 498–512.

[10] H.-G. Kim et al. 2019. Machine Learning-Based Method for Prediction of Virtual Network Function Resource Demands. In *2019 IEEE Conference on Network Softwarization (NetSoft)*. 405–413. doi:10.1109/NETSOFT.2019.8806687

[11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[12] Savio D Immanuel and Udit Kr Chakraborty. 2019. Genetic algorithm: an approach on optimization. In *2019 international conference on communication and electronics systems (ICCES)*. IEEE, 701–708.

[13] J. X. Salvat et al. 2018. Overbooking network slices through yield-driven end-to-end orchestration. In *Proceedings of the 14th Int'l Conf. on Emerging Networking Experiments and Technologies*. 353–365.

[14] Weiwei Jiang, Yafeng Zhan, Guanming Zeng, and Jianhua Lu. 2022. Probabilistic-forecasting-based admission control for network slicing in software-defined networks. *IEEE Internet of Things Journal* 9, 15 (2022), 14030–14047.

[15] M. R. Raza et. al. 2019. Reinforcement learning for slicing in a 5G flexible RAN. *Journal of Lightwave Technology* 37, 20 (2019), 5161–5169.

[16] M. Sulaiman et al. 2022. Multi-agent deep reinforcement learning for slicing and admission control in 5G C-RAN. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 1–9.

[17] Soukaina Ouhame, Youssef Hadi, and Arif Ullah. 2021. An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model. *Neural Computing and Applications* 33, 16 (2021), 10043–10055.

[18] Nipun Ramakrishnan and Tarun Soni. 2018. Network traffic prediction using recurrent neural networks. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 187–193.

[19] Muhammad Rehan Raza, Ahmad Rostami, Lena Wosinska, and Paolo Monti. 2019. A slice admission policy based on big data analytics for multi-tenant 5G networks. *Journal of Lightwave Technology* 37, 7 (2019), 1690–1697.

[20] Shivani Saxena and Krishna M Sivalingam. 2022. DRL-based slice admission using overbooking in 5G networks. *IEEE Open Journal of the Communications Society* 4 (2022), 29–45.

[21] Shivani Saxena and Krishna M Sivalingam. 2022. Slice admission control using overbooking for enhancing provider revenue in 5G Networks. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 1–7.

[22] Vincenzo Sciancalepore, Xavier Costa-Perez, and Albert Banchs. 2019. RL-NSB: Reinforcement Learning-Based 5G Network Slice Broker. *IEEE/ACM Transactions on Networking* 27, 4 (2019), 1543–1557. doi:10.1109/TNET.2019.2924471

[23] Jainul Trivedi and Manan Shah. 2024. A Systematic and Comprehensive Study on Machine Learning and Deep Learning Models in Web Traffic Prediction. *Archives of Computational Methods in Engineering* (2024), 1–25.

[24] William F Villota-Jacome, Oscar Mauricio Caicedo Rendon, and Nelson LS da Fonseca. 2022. Admission control for 5G core network slicing based on deep reinforcement learning. *IEEE Systems Journal* 16, 3 (2022), 4686–4697.