Accepted for presentation in: IEEE Future Networks World Forum (FNWF), Lisbon, Portugal, October 2026.

Scalable ML-Driven Network Slicing for Critical Infrastructures

Dennis Overbeck, Muhammad Waleed, Pascal Jörke and Christian Wietfeld
Chair of Communication Networks, TU Dortmund University, Dortmund, Germany
Email: {dennis.overbeck, muhammad.waleed, pascal.joerke, christian.wietfeld}@tu-dortmund.de

***Abstract*—Critical infrastructures, such as the future energy grid with its dense population of Intelligent Electronic Devices (IEDs) and Distributed Energy Resources (DERs), necessitate a robust yet flexible Information and Communications Technology (ICT) backbone. Since the introduction of 5G, public ICT infrastructure can provide isolated radio resources via network slicing. However, as e.g. Phasor Measurement Units (PMUs) must deliver their information within a few milliseconds, proactive allocation of radio resources becomes necessary. By integrating Machine Learning (ML), latency can be reduced to this range while maintaining overall spectral efficiency. In this paper, we examine the scalability of ML-based proactive resource allocation in public mobile radio infrastructure from both the network operator's and the user's perspective. We demonstrate the applicability of our solution in a baseline congested cell scenario using the simulation tool *OMNeT++*. Finally, we validate our scheduler in a realistic scenario based on a reference energy grid, representing a critical infrastructure use case.**

***Index Terms*—network slicing, proactive scheduling, machine learning, critical infrastructure, OMNeT++, Simu5G**

I. Introduction

Critical infrastructures and public Information and Communications Technology (ICT) networks are becoming increasingly intertwined. The modern energy grid exemplifies this development, as the shift towards renewable, decentralized generation lets Distributed Energy Resources (DERs) potentially increase communication traffic for control and measurement data in densely populated areas. To monitor and control these devices without building a dedicated communication network, 5G provides network slicing for isolated resource allocation. In contrast to Quality of Service (QoS) streams, where traffic is handled on a best effort basis, these resources are dedicated to the slice tenants only, remaining feasible even for crowded radio infrastructures. To reduce resource scheduling wait times for User Equipments (UEs), the standard defines two procedures for proactive allocation without prior scheduling requests. While the utilization of Configured Grants (CGs) concentrates on long-term allocation for strictly recurrent transmissions via Radio Resource Control (RRC) signaling, the so-called Proactive Grants are scheduled via Downlink Control Information (DCI) messages and thus offer a flexible resource allocation, changeable virtually every *millisecond*. If these resources match the actual demands of the UEs, scheduling requests within the prioritized slice are eliminated entirely. This allows leveraging Machine Learning (ML) to optimize allocation based on observed traffic, even in public networks (Fig. 1). Therefore, we concentrate our approach on the uplink resource allocation, assuming inference modules can be co-located with the scheduler, as enabled by emerging Open Radio Access Network (RAN) architectures.

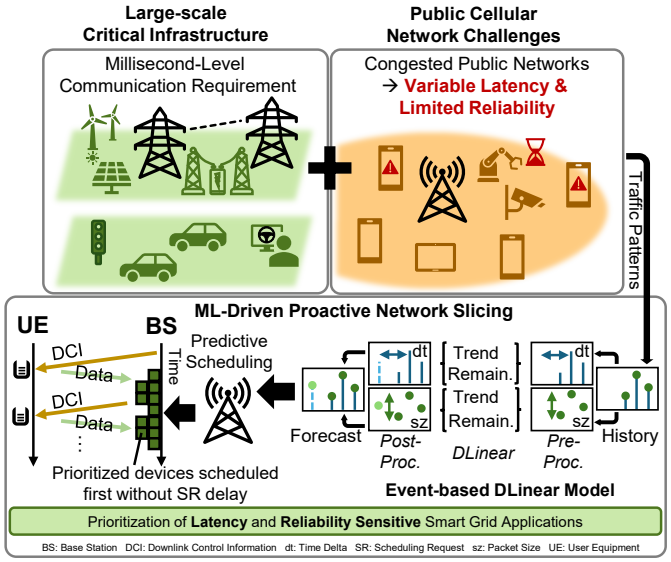

Fig. 1: By leveraging traffic prediction at the base station, proactive grants enable isolated, low-latency radio access for critical infrastructure communication even under congested network conditions.

In previous work, this approach was concentrated on experimental platforms using Long Short-Term Memory (LSTM)-based prediction for individual device traces with small numbers of devices [1]. This work scales the approach by utilizing a DLinear model trained on a heterogeneous synthetic dataset, assessing deployment without per-device retraining and providing insights on the impact for public network infrastructures. Furthermore, we demonstrate low latency even in congested networks and introduce a novel approach to model realistic ICT networks on top of low-to-medium voltage energy grids. While instantiated for the Smart Grid domain, the underlying event-based prediction and scheduling approach is application-agnostic and transfers to other critical infrastructure use cases with predictable traffic patterns, such as intelligent transportation systems. The remainder is structured as follows: Sec. II discusses related work, Sec. III the novel modeling approach for digitalized energy systems, and Sec. IV the design aspects of ML-based network slicing for proactive resource allocation. Finally, Sec. V discusses the results within two distinct simulation environments, before Sec. VI concludes the paper and gives an outlook to future work.

©2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, including reprinting/republishing this material for advertising or promotional purposes, collecting new collected works for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

II. Related Work

Within this work the focus is on the resource orchestration to enable precise network slices, adapting to different communication patterns while decreasing experienced latency per user. In this section we focus on the current works regarding proactive resource allocation prediction on millisecond timescales as well as in the context of critical infrastructures, with focus on energy use cases. For DERs, the usage of network slicing was researched within [2] with focus on the security aspects, highlighting the suitability for low latency control via the technology. Another work [3] focuses on the use of ML to classify Smart Grid data streams and perform a long-term prediction (in hours) using Bidirectional Long Short-Term Memory (BILSTM) based on the historic traffic. However, the results are based on the classification and prediction models only, thus were not tested in a simulation or real-world testbed.

The authors of [4] propose a distributed LSTM architecture for the problem of resource provisioning, however with focus on long-term prediction with a horizon of minutes. To lessen the impact of data transmitted to centralized predictors, simpler Deep Neural Networks (DNNs) are located at the edge, which perform well on half of the data, preserving Service Level Agreements (SLAs). The authors of [5] concentrate on the classification into network slice classes as well as their dimensioning in context to the packet delay budgets. The authors of [6] implement an Hierarchical Token Bucket (HTB)-based queueing to mimic network slicing behavior in OMNeT++, defining guaranteed and maximum data rates per stream within a small-scale scenario comprising three different applications. The focus is therefore to balance Quality of Experience (QoE) for prioritized services with fair bandwidth sharing across flows, whereas the focus of this work is on enabling low latency communications via the air interface on scheduling level. We generate and validate the dataset using packet-level simulators, a standard approach in research as demonstrated by works like [7], highlighting OMNeT++'s applicability in the research community. To the best of our knowledge, previous works concentrate either on long-term traffic prediction or small-scale scenarios using predictive, proactive resource management.

III. Smart Grid Scenario Generator

While reference energy grid topologies exist (e.g., IEEE 39-Bus), low-voltage grid data is typically synthesized and abstracted geospatially [8], with no corresponding ICT counterpart networks available. As part of this work, a scenario generation tool based on benchmark grids [9] was developed to close this gap, providing realistic ICT networks on top of pre-existing energy grids, as depicted in Fig. 2. Focusing on the distribution of base stations and the distance-maintaining overlay of fiber connections over feeder cables, the placement is performed leveraging OpenStreetMap (OSM) geospatial data and publicly available data of cellular locations provided by the Federal Network Agency of Germany. With this information, the electrical grid is first re-organized along actual roads based on user input for centered geodata. Besides, the number

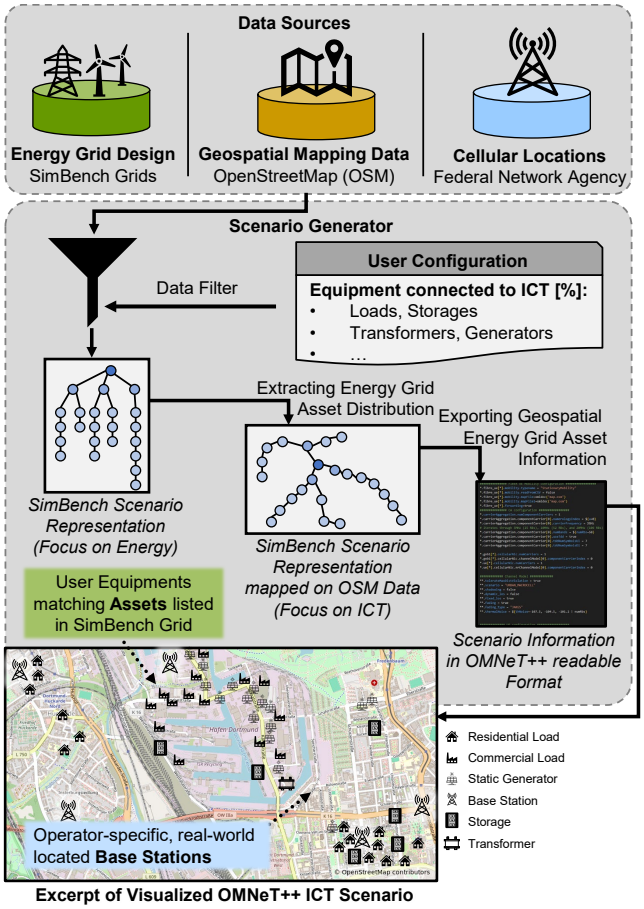

Fig. 2: In this work, public data sources were leveraged to provide reference ICT infrastructure for energy grids using a hybrid modeling approach. Thus, abstracted grids result in a realistic topology with corresponding real-world public mobile radio infrastructure.

of energy devices considered for wireless monitoring and control can be chosen and thus the communication scenario be fine-tuned by the user. Based on the location, base station data is acquired for this region and mapped provider-specific on the geospatial map. For the OMNeT++ representation, the created scenario is provided as input and the corresponding input file is generated for later use. Since grid operators are bound to pre-existing public radio infrastructure, this hybrid modeling approach reflects realistic deployment conditions. For the evaluation scenario, we choose a medium-to-low-voltage energy grid modeled within a mixed industrial urban area, as depicted in Fig. 2 at the bottom.

IV. ML-based Network Slicing for Critical Infrastructure

We concentrate on uplink resource allocation, using energy devices operating under International Electrotechnical Commission (IEC) 61850 as a representative critical use case, where jitter in transmissions prevents deterministic scheduling and the base station lacks advance knowledge of UE demand. We therefore propose a proactive resource scheduler based on pre-trained ML models to combine low latency with reliable transmission for critical applications.

A. Proactive Resource Management

As packet traces in the energy domain, especially regarding status information of the grid, are predictable, the latency for critical scenarios such as re-connecting distributed devices after a blackout can be reduced to a minimum as timely information is crucial here. While IEC 61850 Manufacturing Message Specification (MMS) reporting follows fixed integration periods at the application level, real transmissions are affected by network jitter, making exact arrival times uncertain. This motivates the event-based reformulation and multi-step prediction, forecasting the next H arrivals simultaneously to account for cumulative timing uncertainty.

Algorithm 1 Proactive Event-Based Scheduler

Require: Pre-trained ML model $\mathcal{M}$, look-back window L, prediction horizon H, tolerance τ, grant lead ℓ

procedure ONPACKETARRIVAL(msg)
  $\delta_{obs} \leftarrow offset + (currentTime - lastArrival)$
  if msg on non-proactive grant **and** buffer full **then**
    $\delta_{bsr} \leftarrow (currentTime - T_{BSR}) + \Delta_{proc}$
    $\delta_{corr} \leftarrow \max(1, \delta_{obs} - \delta_{bsr})$
    $offset \leftarrow \delta_{bsr}$ ▷ carried forward
  else
    $\delta_{corr} \leftarrow \delta_{obs}$; $offset \leftarrow 0$
  end if
  UPDATEBUFFERS($\delta_{corr}, msg.size$)
  $count \leftarrow count + 1$
  if $count \equiv 0 \pmod{H}$ **and** buffer full **and** state is new **then**
    $\mathbf{Y} \leftarrow$ RUNINFERENCE(buffer.tail(L)) ▷ Returns 1D array of sizes (0 if empty)
    UPDATETIMESERIESBUFFER($\mathbf{Y}$)
  end if
end procedure

procedure ONEACHTTI
  $prediction \leftarrow$ LOOKUP($currentTime + \ell \pm \tau$)
  if $prediction > 0$ **then**
    $P \leftarrow prediction$
  else
    $P \leftarrow 0$ ▷ Fall back to normal BSR/Best Effort
  end if
  ALLOCATERESOURCES($\max(P, BSR_{actual})$)
end procedure

To achieve this, the use of proactive allocation of resource blocks, as defined in [10], becomes a key enabler. Trained ML models detect traffic patterns automatically, without manual reconfiguration of the scheduling algorithm.

1) Event-Based Scheduler Design: The fundamental design principle of the improved Slice-Aware Machine Learning-based Ultra-Reliable Scheduling (SAMUS) [11], outlined in Algorithm 1, is that the deployed model operates on discrete packet arrival events rather than continuous time series. The key distinctions to [1] are the replacement of the LSTM model, which was trained on individual real-world traces, with a lightweight DLinear architecture trained on a large synthetic multi-trace dataset, and the extension to multi-cell deployments. The input sequence consists of events parameterized by Inter-Arrival Time (IAT) δ_k and packet size s_k.

2) Buffer Status Report (BSR) Delay Compensation: Packets served reactively carry the BSR signaling delay into the observed IAT, shifting the model input away from the application-level traffic pattern. This occurs, e.g., right after the look-back history has been filled, when predictions are already active but packets are still served reactively. Whenever a packet of a prioritized UE arrives on a non-proactive grant and the history is full, the scheduler therefore isolates the BSR-induced delay, δ_{bsr}, which accounts for the time elapsed since the BSR was received plus a fixed processing latency $\Delta_{\text{proc}} = 8\,\text{ms}$. The corrected IAT is calculated as $\delta_{\text{corr}} = \max(1, \delta_{\text{obs}} - \delta_{\text{bsr}})$. To preserve the global timing integrity of the sequence, the subtracted δ_{bsr} is carried forward as the initial offset for the subsequent IAT measurement.

3) Prediction Pipeline: Event data is maintained in a sliding window of L entries and Open Neural Network Exchange (ONNX) inference is triggered when the accumulated event count is a multiple of H and the buffer is full, at which point the most recent L corrected IAT and packet size values are passed to the model.

4) Model Inference and Scheduling Integration: After normalization, the values are structured into a single two-dimensional input tensor and passed to the ONNX runtime, and the outputs are subsequently transformed back to their original domain via inverse scaling. If a valid prediction is identified within the tolerance window of $\tau = \pm 4$ Transmission Time Intervals (TTIs) (i.e., $\pm 4\,\text{ms}$ at 15 kHz Subcarrier Spacing (SCS)), issued with a grant lead of $\ell = 2$ TTIs to offset uplink delay, the predicted packet size is compared with any actual BSR to set the allocation. This window allows for covering residual timing uncertainty from model mispredictions and jitter while keeping unnecessary pre-allocation bounded. To enforce slicing constraints, the Proportional Fair (PF) scheduler categorizes active connections into a priority queue for prioritized UEs and a Best Effort (BE) queue. Prioritized connections are served exhaustively before any remaining resources are allocated to the BE queue, ensuring isolation and timely delivery for prioritized devices even prior to the arrival of actual data. In the absence of a valid prediction within the tolerance window, the scheduler falls back to standard BSR-triggered reactive allocation, ensuring graceful degradation for falsely predicted traffic patterns.

5) Machine Learning Model: The model chosen for inference is derived from the lightweight DLinear [12], a linear forecaster decomposing its input into trend and remainder components, which proved to be suitable for real-world deployments with low inference times [13]. However, we adapt this model into a DLinear with two output heads for event-based predictions, i.e., separate heads for IAT and packet size, using a moving average kernel to control the smoothing

window for trend extraction. This model is deployed at each base station, enabling proactive allocation for a network-wide defined set of UEs. The predictor is intentionally lightweight and exchangeable, as the contribution of this work lies in the scheduler integration and its system-level scalability.

6) Loss Function: We leverage a custom loss function based on the Huber loss, which treats the IAT and packet size components separately. To mitigate drift from step-wise prediction errors, the loss includes the cumulative sum of IAT errors over the prediction horizon H, defined as:

$$\begin{aligned}\mathcal{L}_{\text{total}} &= \lambda_{\text{dt}} \cdot L_\delta(\hat{\mathbf{y}}_{\text{dt}}, \mathbf{y}_{\text{dt}}) \\ &+ \lambda_{\text{cum}} \cdot L_\delta\left(\left[\sum_{j=1}^{i} \hat{y}_{\text{dt},j}\right]_{i=1}^{H}, \left[\sum_{j=1}^{i} y_{\text{dt},j}\right]_{i=1}^{H}\right) \\ &+ \lambda_{\text{sz}} \cdot L_\delta(\hat{\mathbf{y}}_{\text{sz}}, \mathbf{y}_{\text{sz}}),\end{aligned} \tag{1}$$

where $\hat{\mathbf{y}}_{\text{dt}}, \mathbf{y}_{\text{dt}}$ are predicted and true IATs, while $\hat{\mathbf{y}}_{\text{sz}}, \mathbf{y}_{\text{sz}}$ are predicted and true packet sizes. L_δ denotes the element-wise Huber loss with threshold δ in normalized units. The factor λ_{cum} weights the cumulative term, which penalizes systematic timing drift over the prediction horizon. All loss weights are selected jointly via hyperparameter search (cf. Tab. I).

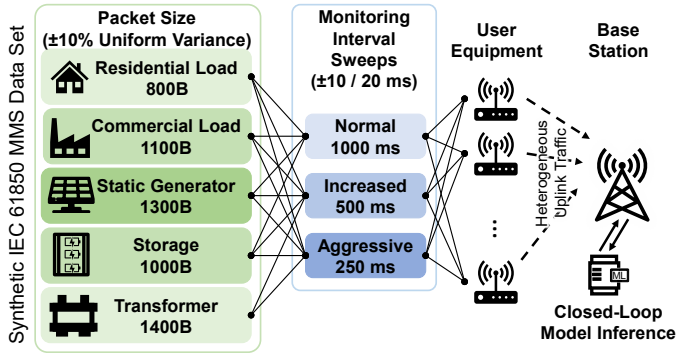

Fig. 3: The synthetic IEC 61850 dataset, consisting of 7,200 individual traces, was emulated using open-source tools [14] providing temporal variance in transmissions.

7) Model Training: The synthetic dataset (cf. Fig. 3) is split at the trace level into a 10 % held-out test set and a 90 % training and validation set. Training uses 5-fold cross-validation with Adaptive Moment Estimation (ADAM) and Optuna-based Tree-structured Parzen Estimator (TPE) search as well as an Asynchronous Successive Halving Algorithm (ASHA) scheduler. Selected values are provided in Tab. I.

Tab. II compares both models under an identical training and evaluation protocol. For comparison, a naive last-value forecast yields 0.77 ms IAT MAE and 99.99 % accuracy within ± 4 TTIs. As the scheduler is agnostic to the predictor, models for less regular traffic can be deployed without modification. DLinear's sub-byte size errors yield exact byte-level grants. The window size itself is motivated by the offline results, as tightening the tolerance to ± 2 or ± 1 TTIs reduces the share of usable predictions to 98.2 % and 74.1 %, respectively. For model input and output, a feasible combination providing

TABLE I: Candidate and Selected Hyperparameters

Hyperparameter	Candidate Values
Kernel Size	[3, 5, 7, 9, **11**]
Bias	[**True**, False]
Batch Size	[16, 32, **64**]
Learning Rate*	[1e-5, 5e-3] (**3.31e-4**)
Weight Decay*	[1e-6, 1e-2] (**1.01e-6**)
Huber Delta δ*	[1e-2, 1.0] (**2.70e-1**)
λ^*_{dt}	[5e-2, 2.0] (**9.51e-2**)
λ^*_{sz}	[5e-2, 2.0] (**1.15e-1**)
λ^*_{cum}	[5e-2, 2.0] (**1.32e-1**)

selected hyperparameters
*sampled log-uniformly over given interval

TABLE II: Predictor Comparison on the Held-Out Test Set. Inference: median latency of calls at fixed intervals, incl. pre-/post-processing, single-threaded ONNX Runtime, virtualized Intel Xeon Silver 4216.

Predictor	IAT MAE	Size MAE	Acc. ± 4 TTI	Inf. Time	Param.
DLinear (ours)	0.51 ms	0.09 B	100.0 %	198 µs	272
LSTM [1]	1.14 ms	2.39 B	98.7 %	508 µs	200,712

reliable performance was found for a look-back window of 16 values and a prediction horizon of the next four values. Each prediction issues one grant per TTI across the window, i.e., at most $1/(2\tau + 1)$ of them carry data. As timely delivery of packets is the key goal, this ± 4 TTI window acts as an over-provisioning safety margin, whose cost is included in the resource block usage reported in Sec. V.

B. Scenario Definition

The scenarios chosen for this work are two-fold. First, a validation scenario is defined to validate the scalability of the SAMUS scheduler in terms of slice isolation, when deployed at a single base station with varying numbers of total active devices connected. Second, a realistic scenario derived from a reference energy grid for a mixed industrial urban environment is investigated, deploying the inference model on multiple base stations simultaneously. The scenario is created using the previously described Smart Grid Scenario Generator (cf. Sec. III). Here, the prioritized UEs randomly pick a packet trace to transmit from the held-out test pool.

V. Evaluation

We compare standard PF scheduling with a proactive static approach as well as our enhanced SAMUS scheduler. The delay is defined as the one-way latency from packet creation at the UE until reception at the User Plane Function (UPF) of the core network. The simulation results are gathered in closed-loop simulation runs employing model inference at runtime, implemented in OMNeT++ 6.0.3 using Simu5G 1.2.3 [15] and INET 4.5.4. Starting with the validation scenario in Fig. 4, we deploy ten prioritized grid UEs around a single base station, where each is placed equally distant to put an emphasis on the scheduler itself. Heterogeneous link conditions, including cell-edge UEs requiring substantially more resource blocks per

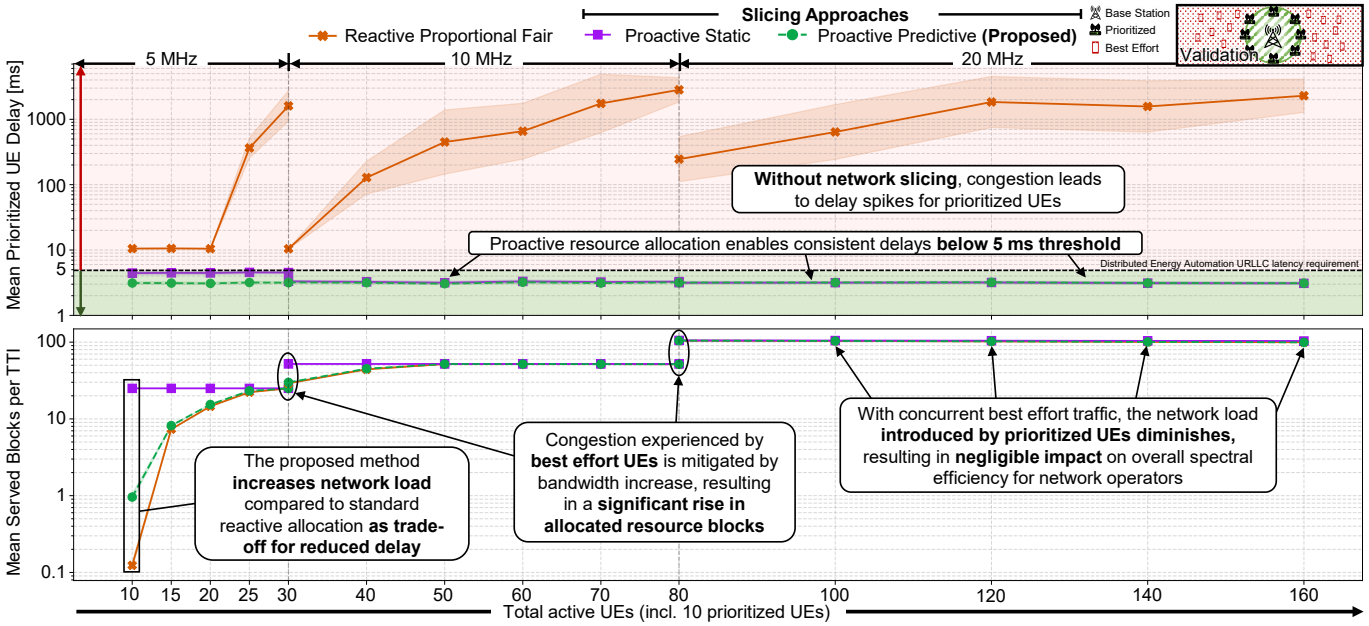

Fig. 4: Systematic validation: The SAMUS approach sustains low latency for prioritized energy devices. UE counts are comparable only within each bandwidth. Even under high utilization, resources stay isolated from congestion. Shaded bands: 95 % confidence intervals.

TABLE III: Simulation Configuration

Category	Configuration
Network Deployment	Validation: 1 BS / 10 prioritized UEs Realistic: 5 BS / up to 135 prioritized UEs
Radio Configuration	FDD, 800 MHz, 15 kHz SCS BS/UE Transmit Power: 46/23 dBm
Channel Model	Suburban Macrocell
Traffic Models	Prioritized: IEC 61850 MMS Traffic Best Effort: 1000 to 2000 kbps
Simulation Parameters	Simulated Time: 120 s (20 s Warm-Up, 15 s Cool-Down), 10 Replications per Config. ML Model: Event-based DLinear

packet, are covered by the realistic scenario. The amount of competing BE UEs is increased to induce congestion on the cell. The remaining simulation parameters are given in Tab. III.

Starting on the left-hand side of the delay plot, the lower bandwidth of 5 MHz becomes congested fairly early, since only 25 resource blocks are available per TTI. Thus, proactive static allocation induces heavy network load, increasing the overall latency in comparison to our proposed proactive predictive allocation. This is highlighted by the mean served resource blocks on the bottom of Fig. 4, showing that even with only prioritized devices being served proactively in a static manner, the constant load is not feasibly manageable for the cell. However, in comparison to standard reactive scheduling, the delay is substantially decreased for both proactive resource management approaches, enabling low latency below the threshold of 5 ms for distributed energy automation [16]. Beyond the mean, the predictive approach keeps the 99th-percentile prioritized delay below 4.1 ms across all configurations. From a network operator perspective, the predictive approach reduces resource block usage by 96.2 % versus static allocation. As prioritized devices are typically a fraction of total users, the spectral efficiency impact diminishes at scale. At 160 total devices, the added load is only 0.3 % above reactive scheduling, compared to 3.8 % for static proactive allocation. The unused grants of the tolerance window cost 0.8 % to 3.4 % of the cell capacity at the highest load per bandwidth, while fallbacks to reactive allocation after warm-up are virtually non-existent. Overall, the predictive approach thus leaves headroom for further delay components, e.g., backbone or processing, within the latency threshold.

Fig. 5 examines the scalability of prioritized devices within a global network slice deployed across five base stations, where diverse UE-to-base-station distances necessitate dynamic resource block allocation [17]. We randomize the device placement within pre-defined locations from the scenario generation tool and add varying numbers of BE UEs randomly placed over the map to simulate real-world variability and competition for resources. Resource usage is reported as a network-wide average across all base stations. As in the validation scenario, prioritized devices consistently achieve low latency (green line). Static allocation, however, increases the overall mean latency because specific cells become congested early on. In contrast, predictive proactive resource allocation allows for constant low delays for prioritized devices connected to different cells, while reducing resource block usage by 47 % on average compared to static allocation. The 99th-percentile prioritized delay stays below 4.5 ms across all configurations. Nevertheless, cells reaching their capacity may suffer intra-slice congestion and require tailored provisioning, although real-world deployments typically host fewer prioritized devices than modeled here.

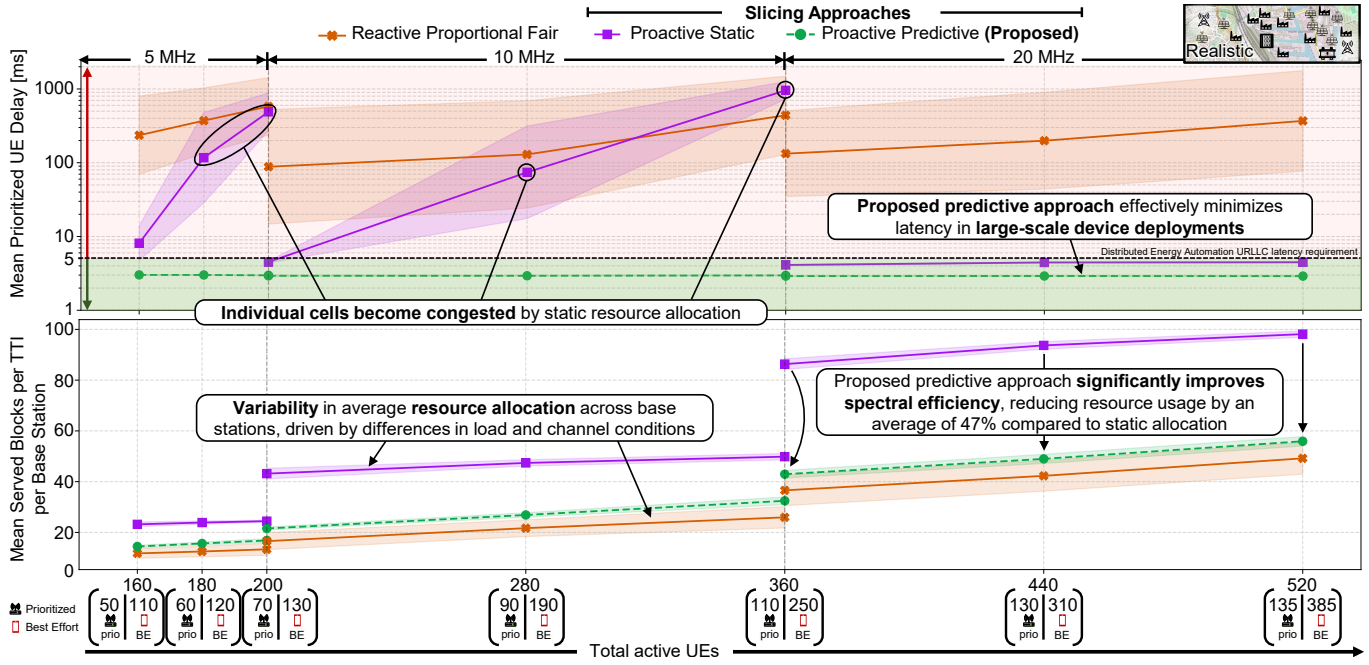

Fig. 5: Realistic scenario: Scalability of serving prioritized devices throughout a network with multiple base stations. Resource blocks are averaged over all base stations. UE counts are comparable only within each bandwidth. Shaded bands: 95 % confidence intervals.

VI. Conclusion and Outlook

Our enhanced SAMUS approach shows the feasibility of network slices in public infrastructure as a complement to dedicated wired networks, demonstrated for a Smart Grid use case modeled with our novel scenario generation tool. We maintained low latency below the 5 ms threshold for distributed energy automation [16] throughout network congestion phases. At the same time, our approach maintains spectral efficiency from a network operator's view, adding only minor load compared to reactive scheduling. Relying on standard DCI grants only, the approach also suits legacy RANs without UE modifications. However, the presented results are obtained in simulation, thus real-world factors such as vendor-specific hardware behavior may affect performance. Future work will focus on integrating external information into the ML-based scheduler to enable fluid network slices, and on transferring the approach to further critical infrastructure use cases such as intelligent transportation systems.

Acknowledgment

This work has been funded by the Federal Ministry for Economic Affairs and Energy (BMWE) via the project ReCoDE under funding reference 03EI6093B.

References

[1] D. Overbeck, N. A. Wagner, et al., "Data-Driven Proactive Uplink Slicing Enabling Real-Time Control within an Open RAN Testbed," in *Proc. IEEE INFOCOM Wkshps*, 2024.

[2] D. Bennett, A. Hasandka, M. Touhiduzzaman, and E. Vaughan, "Service-based, segmented, 5G network-based architecture for securing distributed energy resources," in *Proc. IEEE Power & Energy Soc. Gen. Meeting (PESGM)*, 2021.

[3] Y. Hu, L. Gong, et al., "A Carrying Method for 5G Network Slicing in Smart Grid Communication Services Based on Neural Network," *Future Internet*, vol. 15, no. 7, 2023.

[4] A. Ehsanian and T. Spyropoulos, "Distributed LSTM-based Slice Resource Allocation for Beyond 5G Networks," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, 2024.

[5] M. S. Abood, H. Wang, et al., "An LSTM-Based Network Slicing Classification Future Predictive Framework for Optimized Resource Allocation in C-V2X," *IEEE Access*, vol. 11, 2023.

[6] M. Gajić, M. Bosk, S. Schwarzmann, S. Lange, and T. Zinner, "Demonstrating QoE-aware 5G Network Slicing Emulated with HTB in OMNeT++," in *Proc. IEEE INFOCOM Wkshps*, 2022.

[7] M. Farreras, J. Paillissé, L. Fàbrega, and P. Vilà, "GNNetSlice: A GNN-based performance model to support network slicing in B5G networks," *Comput. Commun.*, vol. 232, 2025.

[8] B. Reveron Baecker, S. Candas, D. Tepe, and A. Mohapatra, "Generation of low-voltage synthetic grid data for energy system modeling with the pylovo tool," *Sustain. Energy Grids Netw.*, vol. 41, 2025.

[9] S. Meinecke, D. Sarajlić, et al., "SimBench—A Benchmark Dataset of Electric Power Systems to Compare Innovative Solutions Based on Power Flow Analysis," *Energies*, vol. 13, no. 12, 2020.

[10] 3rd Generation Partnership Project (3GPP), "NR; Medium Access Control (MAC) protocol specification," 3GPP, Technical Specification TS 38.321, 2024.

[11] C. Bektas, D. Overbeck, and C. Wietfeld, "SAMUS: Slice-Aware Machine Learning-based Ultra-Reliable Scheduling," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2021.

[12] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are Transformers Effective for Time Series Forecasting?" In *Proc. AAAI Conf. Artif. Intell.*, AAAI Press, 2023.

[13] N. A. Wagner, J. Eßer, et al., "Real-Time Predictive Scheduling for Networked Robot Control Using Digital Twins and OpenRAN," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2024, pp. 1–6.

[14] MZ Automation GmbH, *libIEC61850 – An open-source library for the IEC 61850*, Version 1.6.0, 2025.

[15] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, "Simu5G – an OMNeT++ library for end-to-end performance evaluation of 5G networks," *IEEE Access*, vol. 8, pp. 181 176–181 191, 2020.

[16] 3rd Generation Partnership Project (3GPP), "Service requirements for cyber-physical control applications in vertical domains," 3GPP, Technical Specification TS 22.104, 2024.

[17] D. Overbeck, H. Schippers, F. Kurtz, and C. Wietfeld, "Reliable 5G Network Slicing for Distributed Energy Systems: Validation of Live Performance during High-Load Event," in *Proc. CIRED Chicago Workshop: Resilience Electr. Distrib. Syst.*, 2024.