Apostolakis, Nikolaos; Chatzieleftheriou, Livia Elena; Bega, Darío; Gramaglia, Marco; Banchs, Albert (2023) Digital Twins for Next-Generation Mobile Networks: Applications and Solutions. *IEEE Communications Magazine*, 61(11), pp.: 80-86.

DOI: https://doi.org/10.1109/MCOM.001.2200854

Digital Twins for Next-Generation Mobile Networks: Applications and Solutions

Nikolaos Apostolakis*†, Livia Elena Chatzieleftheriou*†, Dario Bega‡, Marco Gramaglia†, and Albert Banchs*†
*IMDEA Networks Institute, Spain
†University Carlos III of Madrid, Spain
‡Nokia Bell Labs, Germany

***Abstract*—Digital Twins (DTs) create fully-synchronized virtual representations of real-world systems, which can serve as interactive counterparts for artificial intelligence (AI) and machine learning (ML) algorithms, and hold significant importance for the upcoming 6G mobile networks. In this paper, we argue that DTs can improve all phases of the intelligent networks' workflow, due to their adaptability and scalability properties that would allow them to transparently integrate new AI/ML algorithms faster, more scalably, and more precisely. Our contribution is two-fold: first, we propose three specific application scenarios of DT-enhanced network architectures in the context of 6G. Second, using open-source tools, we implement and evaluate in detail one of them. Our results demonstrate that our DT reflects the characteristics of the physical object, successfully and scalably twinning it, and adapting to changing contextual conditions.**

I. Introduction

Next-generation mobile network systems are rapidly developing to address the increasing need for emerging applications, such as robotics or autonomous vehicles. However, their complexity will increase significantly as they incorporate automation and intelligence, while their management costs have already become a huge impediment for network operators.

In this context, Digital Twins are expected to increase operational efficiency due to their ability to create virtual representations of real physical objects and processes, known as Cyber-Physical Systems (CPSs). CPSs consist [1], [2] of (i) *a physical space* that captures the physical object or entity, also called physical twin (PT), (ii) *a virtual space* that captures a cyber representation of the physical twin, also called Digital Twin (DT), and (iii) *a link* that is used for communication between the physical and the virtual spaces. This link allows for updates of the virtual model when a change in the real object occurs. In fact, this link between the real and the virtual space constitutes the main difference between CPSs and traditional models for predicting entities' behavior in the physical world.

DTs will enable the monitoring or even the optimization of real large-scale physical systems in real-time. By using DTs, the operators can avoid physically performing expensive trials on their real systems that are traditionally performed through human intervention.

The use of DTs is a key technology for the 6th generation of mobile networks, as they will revolutionize the way network automation is performed. Fig. 1 depicts the integration of DTs in a communications provider setting, where DTs coexist with physical objects and reflect their behavior by continuous adaptation to new data. DTs can directly apply network reconfigurations or suggest new ones to the network operator. Similarly, this approach can be applied to the interface between network operators and service providers, allowing the latter to test a service without even having to field-trial it.

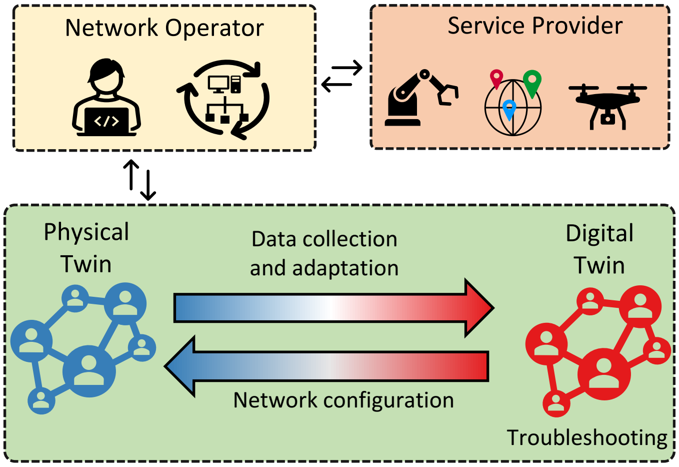

Fig. 1: Interaction of Physical and Digital Twins in a service provider context.

Related Work. There exists several works studying the impact of DT on future generation 6G mobile networks. In [3], the authors propose a DT-empowered architecture for optimizing traffic in 6G. DTs create digital replicas of physical systems (*e.g.*, Intelligent Reflecting Surfaces, Unmanned Airborne Vehicles *etc.*), and Reinforcement Learning (RL)-aided orchestration agents continuously monitor their performance metrics in order to learn networking policies that meet dynamic application characteristics. In a recent classification of DT-enabled frameworks for 6G related services [4], the authors identify different contexts of DT applications, including IoT, connected vehicles, and 6G network management. In [5], the authors discuss key requirements for designing a DT-enabled 6G system. They describe the components for their DT and present the benefits of adopting a twins-based architecture across the edge and cloud, as well as the implementation challenges that this paradigm brings. In [6], the authors employ DTs as emulators of networking components, such as the radio channel and connected vehicle. The authors in [7] mention multiple application areas of DTs in Radio Access Networks

(RAN). A radio-aware DT can capture fine details of the lower layers of the networking stack, predict the channel conditions, and take proactive radio resource scheduling actions. The authors in [8] augment the training data for their RL agent by designing a Deep Learning (DL)-based DT that simulates the behavior of the real traces they collected from their virtualized RAN's execution environment. In [9], the authors present a set of open-source tools and technologies in order to create a DT of a 5G network, measuring the round-trip time of TCP data transmissions.

Contributions. This paper discusses the integration of DTs as a fundamental element of 6G Networks. Our contribution is two-fold: first, we propose three specific application scenarios of DT-enhanced network architectures in the context of 6G. Second, we implement using open-source tools and evaluate in detail one of these scenarios, namely a DT of a network appliance such as a virtualized RAN component. Our results demonstrate that our DT accurately reflects the characteristics of the physical object, successfully and scalably twinning it, and adapting to changing contextual conditions.

II. Digital Twins in Mobile Networks

We now discuss the DT paradigm and its role in next-generation networks, and present three specific use cases where we envision that DTs will significantly improve next-generation communication networks.

A. Preliminaries

Machine Learning (ML) techniques for DTs. DTs can use a variety of methods, from traditional statistical and analytical models to data-driven techniques such as ML or RL algorithms. Compared to traditional network simulators, DTs have the advantage of being linked and real-time synchronized with the physical twins they represent. ML and RL techniques can leverage this link to accurately model the physical object's behavior by observing its historical and current data, enabling real-time DTs to boost operations and management in next-generation mobile network systems. The use of such methods to build DTs will result in the development of native-AI 6G networks, which can iterate faster and automatically adapt to changes in the underlying physical system by substituting digital with physical components.

Interaction between twins. When the real system faces states that it had not encountered before, a *distribution shift* occurs because the data given as input to the digital twin will be drawn from a different distribution than the one over which it was trained. In these cases, DTs leverage both historical and real-time data provided by the physical twins and adjust their internal model accordingly. This procedure relies on the link between the physical and the digital twin, which we call *data adaptation.* Along with real-time *data collection*, data adaptation captures the actual mobile network status and accurately models the physical twin's behavior. When the DT's internal model captures the physical twin with enough accuracy, the DT may request a resource *reconfiguration* at the physical twin to make it operate more efficiently.

Challenges. However, the instantiation, maintenance, and lifecycle management of DTs in real networks arise multiple open challenges that need to be addressed. DTs should provide an accurate model of the real network and generalize over unseen scenarios to emulate the network behavior with high accuracy. Keeping DTs updated and synchronized with the real network requires identifying heterogeneous data sources and processing and collecting new data in real-time (or every time the real network behavior changes). The data collection process may become costly and lead to the overload of the real network given the large amount of data involved. Furthermore, DTs should be interpretable to assess the decisions taken. Thus, new techniques to evaluate and explain predictions and decisions performed by DTs (especially if built leveraging neural networks) are crucial for allowing their deployment in real environments.

B. DTs in next-generation mobile networks.

DTs can be powerful enablers of new features in next-generation mobile networks, where automation of the operating procedures and self-reconfiguration or system parameters will be required. DTs provide scalable models of physical objects, and thus can be integrated as separate modules across the network architecture, offering faster operation of autonomous and self-learning networking algorithms.

Fig. 2 shows our proposed taxonomy for the application of DTs in the 6G network domain, which encompasses all the hardware and software components that provide network services to application providers. This includes, for instance, an enhanced Mobile Broadband network service used by a Content Delivery Network provider to deliver multimedia files on an Ultra-Reliable Low Latency Communication service that provides industrial services.

The network domain also comprises the infrastructure, the surrounding environment, and sensing services that are, by nature, tightly attached to the environment where they are executed. Network operators and service providers interact through an exposure layer that allows efficient interplay between them.

In this context, we envision three main scenarios for DTs: DTs of Network Appliances, DTs of Network Services, and DTs of the network environment.

1) Digital Twinning of Network Appliances: Recent trends in network softwarization have led to the creation of Virtual Network Functions (VNF) over *shared* computing network and storage virtual resources. This has increased the complexity of management and orchestration algorithms, which now have to handle a plethora of configuration variables to find optimal operation points. The variables that may be configured range from resource orchestration ones, such as available computing capacity, to the internals of each VNF such as the radio parameters of a RAN VNF.

In this context, the surge of AI/ML algorithms allows for a seamless transition toward this view, leveraging the great availability of data coming from the network. Here, DTs can play a fundamental role in improving the quality of training, monitoring, and governance of the deployed models. In particular, the DTs may be of crucial importance for:

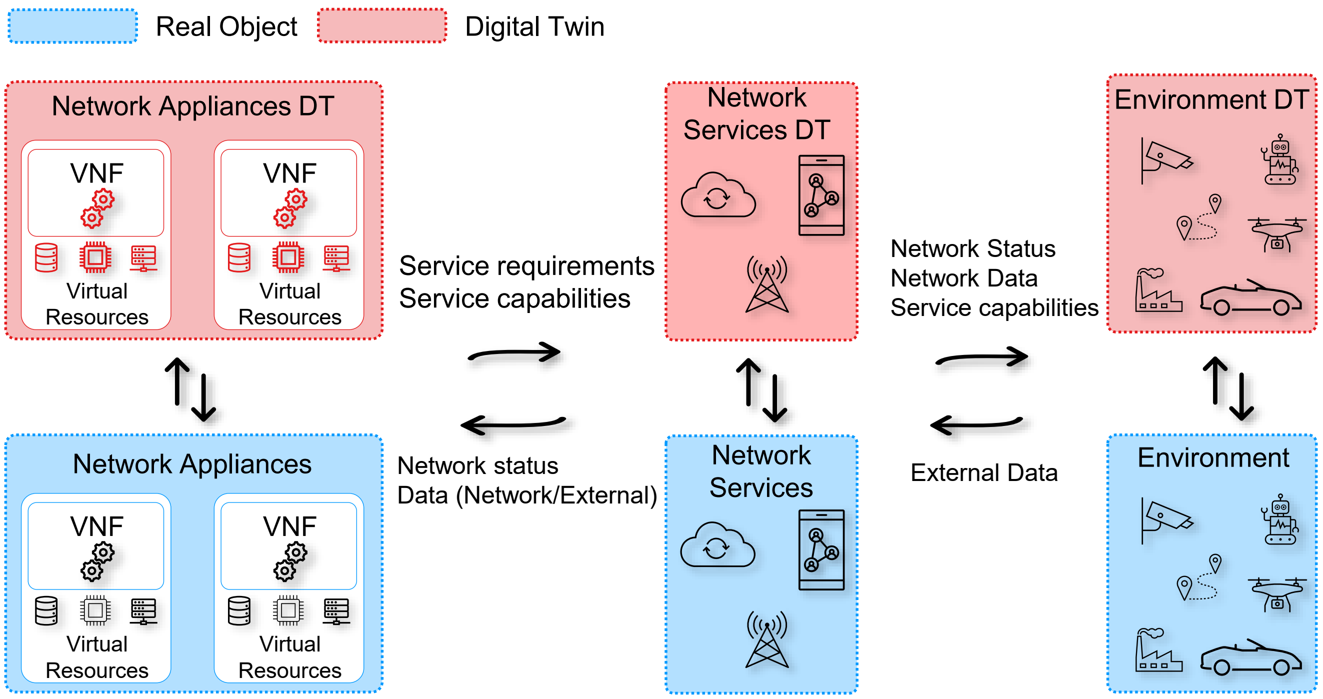

Fig. 2: The applicability of DT in different domains of a network.

- Model **training**, especially for RL-based models, that require a tight exploration of state and policies to find the best solution. Specifically, RL models could take advantage of the DT models they are interacting with, in order to avoid impacting normal mobile network system operations in a production environment.
- **Monitoring** can use DTs to perform *sandboxed* decisions (*i.e.*, those taken in an isolated environment under high supervision and safety). These decisions can then be compared to those of the real operational system, identifying possible drifts between the two models, thus allowing for a fast reaction close to the element that made a decision.
- Model **governance**, to track the interactions between the AI/ML models and their results. Also, new models can be trained in parallel with data coming from either the DT or the physical system, to allow a hot swap when the contextual conditions are changing.

The different VNFs that compose network services (*e.g.*, an enhanced Mobile Broadband network slice for video-on-demand) can, in turn, also be considered DTs for other domains of the framework or the service provider, as discussed next.

2) Digital Twinning of Network Services: The emergence of novel network applications with complex requirements (*e.g.*, AR/VR, Metaverse, Vehicular Networks) makes the traditional human-based network management solutions impractical. Here, DTs can enable the efficient control and management of the mobile network by providing a data-driven virtual model for it. The real-time digital representation of the physical twin can be leveraged by:

- the Management and Orchestration (MANO) framework to perform troubleshooting, network planning, and optimization. An updated and high-fidelity DT, based on real-time data collected from the network, makes the mobile network's behavior predictable and enables proactive testing of novel optimization algorithms (*e.g.*, to optimize radio resource management or energy efficiency), measurement of network infrastructure and software updates/upgrades impacts or anomalies detection without affecting current physical system operations. Furthermore, integrating DTs in the 6G mobile network can be a key enabler for network automation.
- The service providers can benefit by interacting with DT services and capabilities to optimize the provided network service. For instance, the interaction between the service provider and DT may result in service requirements negotiation to accommodate the network service that would otherwise not be served given the predicted future physical twin status. On the other side, the service provider may leverage the interaction with the DT to drive and tune network services as network analytics services to ensure the optimization of its private metric.

The digital twinning of network services may be provided as a unique virtual entity that models the behavior of the entire mobile network or as multiple DTs that capture single physical twins and interact with each other to mimic the behavior of the whole physical system. Nevertheless, depending on the use case and scenario, the DT maybe not be limited to mimicking only mobile network-related aspects, but it could be extended to cover, as discussed next, also environment properties, offering a more complete view of the physical twin.

3) Digital Twinning of the environment: Next-generation networks will provide more than data transfer between terminals and service providers, additionally offering (remote) sensing capabilities of the underlying environment. High-level, this scenario provides an interoperation of wireless communication and sensing capabilities that can empower (especially IoT) service providers with more contextual information about the surrounding environments or even enrich the services with more use cases, such as event detection at home or in a vehicular environment. Other environmental metrics could be used, such as coarse user location and trajectories.

Hence, this functionality could also be offered as a DT to service providers to improve their business intelligence processes with network sensing data, but also to avoid possible leakages of private or confidential data from the network. That is, the interaction with a DT of the sensing environment may happen with privacy-preserving guarantees for the end-users and the infrastructure provider.

III. A case study for DT in vRAN environments

Motivated by the scenarios discussed in Sec. II, we now present a case study of DT application to Network Appliances, specifically focusing on virtualized Radio Access Networks (vRAN) environments. Interaction with AI/ML algorithms may be time-consuming and resource-expensive in such systems, especially due to the difficulty of common software models in exposing data at a very high pace. In particular, we focus on the lower layers of the protocol stack.

A. Problem Formulation: Virtualized RANs

processing is the most computing-expensive operation in a mobile network stack [10]. Thus, in such systems, the amount of computing resources plays a significant role in the overall performance since, under shortages, the users can experience detrimental effects in their perceived throughput.

This aspect becomes crucial when synchronization constraints come into play, such as in the case of Hybrid Automatic Repeat reQuest (HARQ) processing [11], which imposes stringent deadlines on the decoding of uplink (UL) wireless frames [8]. This task is not only computationally expensive but also far from having a deterministic execution time, as it is affected by several factors such as (i) the allocation of the Physical Resource Blocks (PRB), (ii) the selection of a Modulation and Coding Scheme (MCS), and (iii) the perceived user's signal-to-noise ratio (SNR). Moreover, it also depends on the available computing capacity at the base station.

As resources in vRAN systems are pooled among different processes [8], RAN procedures such as MAC scheduling should be *computationally-aware* [12], to avoid disruptions caused by computing capacity shortages. Allowing users to send more data also yields higher computation times, which may result in decoding deadline violations. More data can be sent by using more complex modulation or transmitting over wider bands. While the above deadline is configurable in 5G systems, the default for 5G user traffic equals 3ms.

Thus, to ensure the reliability of the decoding times, the overall throughput may decrease. However, due to the lack of deterministic behavior, the impact of the computing capacity on scheduling decisions is hard to predict. This calls for data-driven approaches such as AI/ML ones, as we discuss next.

B. Challenges on the application of Machine Learning

The design of an AI/ML scheduling algorithm entails the electing, training, and deploying DL models in a production environment (within the pipeline, workflow, *etc.*), which can be challenging given the complexity of real systems. In our case study, grant assignment decisions are made in the MAC layer, the input about the CPU resources comes from the MANO, and the decoder performance is recorded at the Physical layer.

A second challenge is the high complexity of carrying out the training. In order to learn a good model, an AI/ML algorithm needs to visit all the possible input combinations several times. This may turn into a very long operation in a real system, as (i) real scheduling decisions are limited to once per every scheduling interval (1ms by default), and (ii) it may be difficult to reach certain combinations of the inputs, due to channel conditions. This cannot be performed on the fly in a production system, as for example, Open RAN forces the deployment of offline pre-trained models [13].

These aspects make it impractical to learn from a real system, as a number of aspects such as model architectures and hyperparameter configurations, analysis, and model interpretability have to be taken into account.

To overcome these issues, we now present a DT of a vRAN network appliance that can improve the operation of computationally-aware MAC schedulers.

C. Digital Twin design

In the above scenario, the physical system that has to be controlled is the Forward Error Correction decoder, since it provides feedback on both the decoding time and the Cyclic Redundancy Check (CRC) result. Therefore, the DT of such a system must replicate the physical system's distribution of decoding times and decoding success and failures. For this purpose, we create a dataset with real-trace performance measurements, and we then train a supervised model to capture them in the DT.

Deciding what to twin and how. We query a real decoder under different combinations of the input space, namely the CPU Capacity (in % of maximum CPU resources), the user's SNR (in dB), the MCS index, and the number of PRBs. We observe the decoder's response, and capture the decoding result with a binary variable, *i.e.*, $CRC = 1$ if the frame is successfully decoded, or $CRC = 0$ otherwise, and the decoding time, which we model as a continuous normally distributed variable, truncated to positive values. Our DT outputs i) the probability with which a frame is successfully decoded, and ii) the mean and standard deviation of the normal distribution that models the decoding time.

D. Dataset Collection

Software. As 6G is still being specified by standardization bodies and no open-source reference architecture is available, we used srsRAN [14], an open-source software that implements the 4G/5G functionality of the whole mobile networking stack of the eNB/gNB. We used its 22.04 version, which implements the Rel.15 of the 3GPP standard.

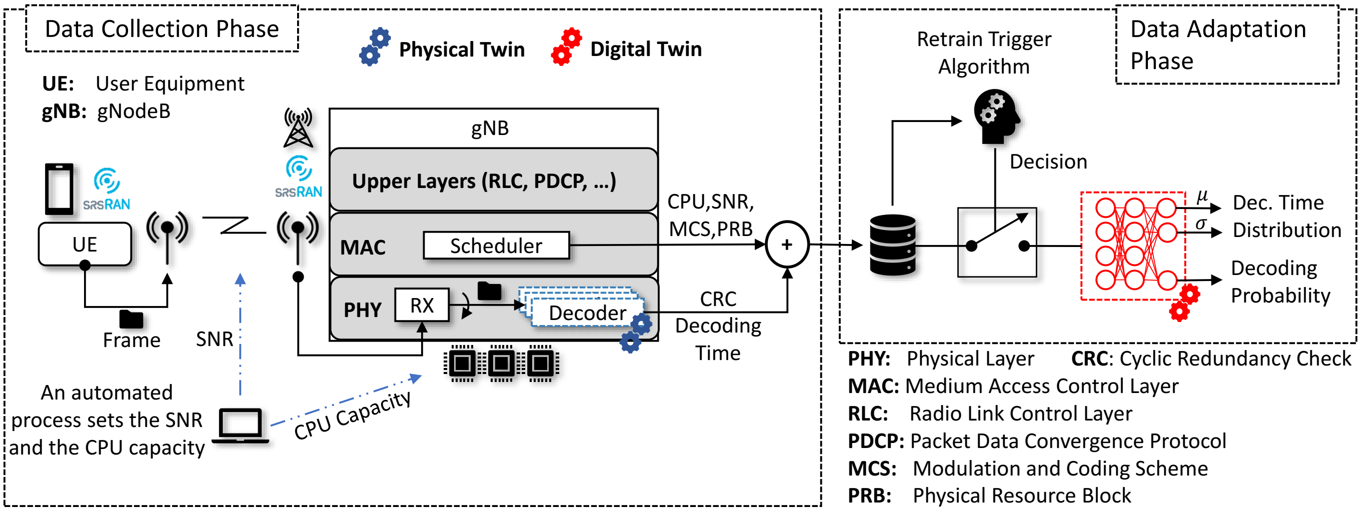

Fig. 3: The building blocks of the proposed DT for vRAN systems.

Data. We measured the decoding time of the UL frame by recording the wall time of the decoding process. To eliminate the noise from other processes, we pinned the decoder threads to a specific CPU set and prevented the default Linux scheduler from preempting those threads in favor of other jobs. The CRC result was directly retrieved by the decoder.

Method. To sweep the entire input space we went through the following process: We set up the srsRAN's UE and gNB processes on different host machines. We used 10 MHz of bandwidth (up to 45 PRB for data transmission in the UL) with Transmission Time Interval (TTI) of 1 ms and 1 user/TTI. To span the PRB and MCS sets, we replaced the default srsRAN's scheduler with a custom one that randomly picks the number of PRBs and the MCS index and directs the user to transmit a new frame. The UL frame is transmitted over an Additive White Gaussian Channel [15] and decoded by the gNB's decoder threads, which output the decoding time and CRC. In order to span across different SNR levels, the wireless channel is controlled using an automated process that selects the target SNR. This process also controls the CPU capacity of the decoding threads, by adjusting the allocated CPU cycles. We depict this procedure (*data collection phase*) in the left block of Fig. 3. We collected $\approx$ 14 million samples, with extensive combinations of the input parameters.

E. Building the DT model

We model the decoder through a two-headed Neural Network (NN) with common hidden feed-forward fully-connected layers. The last common hidden layer is divided into two branches, one for the decoding time prediction task and another for the decoding success prediction task. The last layer comprises three independent neurons; the decoding time's mean and standard deviation, and the probability of successful decoding. Their activation functions are linear, soft-plus (to ensure positivity), and sigmoid (to produce values between 0 and 1), respectively.

Similarly, we use two loss functions for the different tasks. To predict the decoding success, which is a binary classification task, we used the *Binary Cross Entropy* (BCE) loss, which is the negative of the log of corrected predicted probabilities. To predict the mean and standard deviation parameters of the normal distribution, we used the *Negative Log-Likelihood* (NLL) loss function.

We trained the NN using backpropagation and the Adam optimizer with a learning rate of 10^{-4}. The training of the DT with new samples is controlled by a *Retrain Trigger Algorithm*, which is fed by the newly collected samples and decides whether to trigger new retraining or not. As we explain in Sec. IV, we used a simple heuristic that computes a distribution similarity of the new samples against the old ones and triggers retraining if this metric drops below a certain threshold. While more complex re-training procedures may be used, for the sake of conciseness, we limit this analysis to this method only. Alternative algorithms could depend on the amount of newly collected samples or occur on a fixed frequency. For NN-backed DTs, identifying peaks on the loss of the model can be used. In the right block of Fig. 3, we depict the training procedure (*data adaptation phase*).

IV. Evaluation

We now evaluate the capability of the DT to replicate the physical system by (i) generating similar data distributions and (ii) being able to react to distribution shifts.

A. Digital Twin Performance

We divide the evaluation of the DT's performance into the performance of the decoding time distribution prediction task and the decoding success probability task.

1) Decoding time prediction task: Using the Kernel Density Estimation method, we approximate the normalized Probability Density Function (PDF) of the real distribution, which we plot in Fig. 4, together with the predicted one for 15 dB of SNR, 40 PRBs, 100% CPU Capacity, and MCS index $\in \{2, 12, 20\}$.

Looking at the real PDF, lower MCS indices, imply lower decoding complexity, yielding smaller mean decoding times.

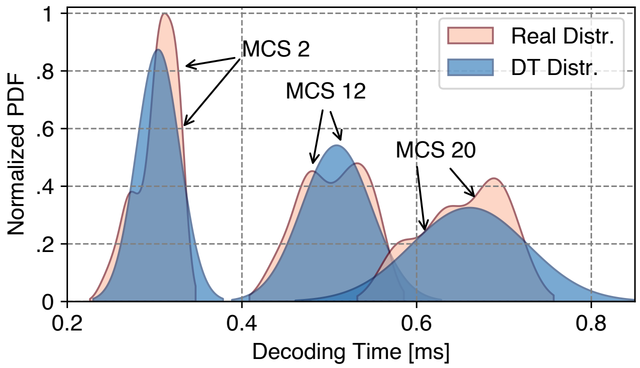

Fig. 4: The PDF of the real and the predicted decoding time distributions, truncated to positive values.

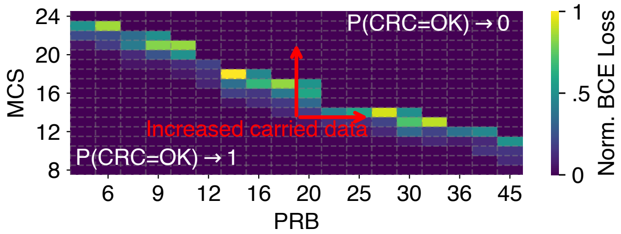

Fig. 5: The real and DT's prediction of decoding probability, and the BCE loss of the prediction task.

Instead, higher MCS yield higher variability of the decoding times. We explain this due to the more complex task the decoder software implementation has to solve. This may incur glitches such as cache misses that reduce the overall performance unevenly, producing higher uncertainty.

We corroborate our modeling assumption to capture the decoding time with a normal distribution (truncated to positive values) by performing the Kolmogorov-Smirnov test for various scenarios. For all the tested combinations of selected MCS, the null hypothesis was accepted with a 99% confidence interval.

2) Decoding success probability task: Then, we evaluate the predictions of the decoding success probability task. We observe that the decoding probability is high in low MCS index and number of PRBs, since the data rate is low. When either the MCS or the number of PRBs increases, the carried data increases, the Shannon capacity of the channel for this SNR level is reached, and the probability gradually drops to 0. In Fig. 5, we plot the BCE loss for 10 dB of SNR, a variety of MCS indices, and number of PRBs. We observe that our DT manages to learn the trend when the probability is either very high or low, while it gives a small prediction error in the transition region.

B. Distribution Shift

In this experiment, we study the interaction between the PT and the DT, and the capability of the latter to adapt to previously unseen inputs. This may happen when the available historical data does not cover regions of the input space, *e.g.*, due to sudden failures in the CPU capacity or a new implementation or computing infrastructure that was not used to create the DT.

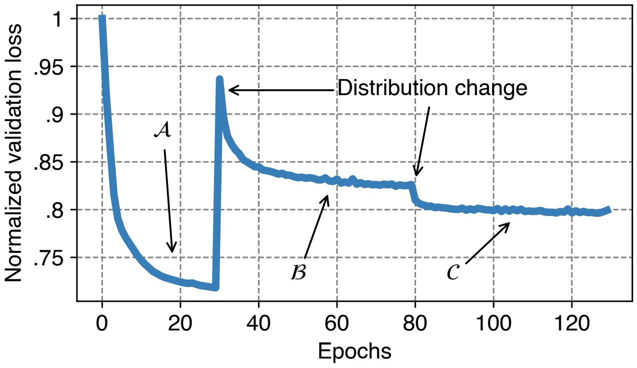

Fig. 6: The DT validation loss (sum of BCE and NLL losses) when distribution changes happen.

To simulate the former scenario, we draw the available CPU capacity of the PT in the interval 90-100% of the maximum achievable, *i.e.*, considering only very high CPU capacity (distribution $\mathcal{A}$). We divide the collected dataset into the training set, used for training the DT, and the validation set, used for evaluating its performance across the training epochs. In Fig. 6, we plot the normalized validation loss where we trained the DT for 30 epochs.

Then, we draw the available CPU capacity in the interval 10-90% (distribution $\mathcal{B}$), *i.e.*, considering a wide range for the CPU capacity, in any case, lower than the previously used one. The retraining procedure can be triggered either at a certain frequency or when the similarity between the DT predictions and recent observations drops between a certain threshold. In our case, a distribution change $\mathcal{A} \to \mathcal{B}$ took place, which caused a drop in similarity and issued a retraining of the DT. We observe a spike in the validation loss as previously unseen observations give high prediction error, which later decreases as the model readjusts. Finally, we introduce a further distribution change by modifying again the CPU capacity to the full range of 10-100% (distribution $\mathcal{C}$). Even though the new distribution has changed ($\mathcal{B} \to \mathcal{C}$), it comprises inputs that have been already seen in the past and are already incorporated in the model, which explains the drop in the loss instead of a sudden increase.

C. Complexity and time of inference gain

We quantify the gain of using NN for predicting the decoding probability against a pure simulation approach by evaluating the *complexity* and *inference time* of getting a new data sample. Using a 5G testbed, a new data sample can be generated at least every 1 ms, considering a TTI of 1 ms. Besides, this adds extra engineering complexity to build a reliable end-to-end system to take measurements. On the contrary, having this NN-backed DT, the time of generation of a single sample is the inference time of the forward pass of the NN. We queried the DT on our computing platform with 1 million combinations of the input features, and the average inference time was measured at 2.8 μs per sample, yielding a 350× speed increase when compared with a non DT-based training. Finally, the complexity of setting up this DT comes entirely from starting up a new process and loading the weights of the NN, which has a negligible cost compared to the training of the algorithm.

V. Conclusions

In this paper, we discussed the advantages and benefits of the usage of a Digital Twin in the context of next-generation mobile networks. We proposed three possible application scenarios and then provided the design and implementation details for one of them, namely a DT for a virtualized RAN system. Our results demonstrate that our DT reflects the characteristics of the physical object, successfully and scalably twinning it, and adapting to changing contextual conditions.

Acknowledgement

The work of University Carlos III of Madrid has been funded by the H2020 Project DAEMON (Grant Agreement No. 101017109), the Horizon Europe Project TrialsNet (Grant Agreement No. 101095871), and by the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union-NextGenerationEU through the UNICO 5G I+D project 6G-CLARION.

References

[1] Mihai *et al.*, "Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 9 2022.

[2] Fuller *et al.*, "Digital Twin: Enabling Technologies, Challenges and Open Research," *IEEE Access*, vol. 8, pp. 108 952–108 971, 2020.

[3] Tariq *et al.*, "Toward Experience-Driven Traffic Management and Orchestration in Digital-Twin-Enabled 6G Networks," *CoRR*, vol. abs/2201.04259, 2022. [Online]. Available: https://arxiv.org/abs/2201.04259

[4] Masaracchia *et al.*, "Digital Twin for 6G: Taxonomy, Research Challenges, and the Road Ahead," *IEEE Open Journal of the Communications Society*, vol. PP, pp. 1–1, 01 2022.

[5] Khan *et al.*, "Digital-Twin-Enabled 6G: Vision, Architectural Trends, and Future Directions," *IEEE Communications Magazine*, vol. 60, no. 1, pp. 74–80, 2022.

[6] Nguyen *et al.*, "Digital Twin for 5G and Beyond," *IEEE Communications Magazine*, vol. 59, no. 2, pp. 10–15, 2021.

[7] Kuruvatti *et al.*, "Empowering 6G Communication Systems With Digital Twin Technology: A Comprehensive Survey," *IEEE Access*, vol. 10, pp. 112 158–112 186, 2022.

[8] Ayala-Romero *et al.*, "VrAIn: A Deep Learning Approach Tailoring Computing and Radio Resources in Virtualized RANs," in *The 25th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '19. New York, NY, USA: Association for Computing Machinery, 2019.

[9] Ramirez *et al.*, "5G Digital Twin: A Study of Enabling Technologies," *Applied Sciences*, vol. 12, no. 15, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/15/7794

[10] Bhaumik *et al.*, "CloudIQ: A Framework for Processing Base Stations in a Data Center," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, ser. Mobicom '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 125–136.

[11] Garcia-Aviles *et al.*, "Nuberu: Reliable RAN Virtualization in Shared Platforms," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 749–761.

[12] Bega *et al.*, "CARES: Computation-Aware Scheduling in Virtualized Radio Access Networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 12, pp. 7993–8006, 2018.

[13] Bonati *et al.*, "OpenRAN Gym: An Open Toolbox for Data Collection and Experimentation with AI in O-RAN," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE Press, 2022, p. 518–523. [Online]. Available: https://doi.org/10.1109/WCNC51071.2022.9771908

[14] Software Radio Systems (SRS), "srsRAN." [Online]. Available: https://www.srslte.com/

[15] Apostolakis *et al.*, "Design and Validation of an Open Source Cloud Native Mobile Network," *IEEE Communications Magazine*, vol. 60, no. 11, pp. 66–72, 2022.

Nikolaos Apostolakis is a Ph.D. student, affiliated with the University Carlos III of Madrid (UC3M) and the IMDEA Networks Institute. He holds an M.Sc. (2017) in electrical and computer engineering and a master's degree (2021) in NFV/SDN for 5G Networks. His interests include virtualized mobile networks, reinforcement learning, and explainable AI.

Livia Elena Chatzieleftheriou is a post-doctoral researcher at the IMDEA Networks Institute. She holds an M.Sc. (2015) in applied mathematics and a Ph.D. (2022) in Computer science. Her current research interests are in online learning and explainable AI for next-generation mobile networks.

Dario Bega is a Network System Automation Researcher at Nokia Bell Labs. He received his M.Sc. (2013) in Telecommunications Engineering from the University of Pisa, Italy, and his Ph.D. (2020) in Telematics Engineering from the University Carlos III of Madrid and IMDEA Networks Institute.

Marco Gramaglia is a visiting professor at the University Carlos III of Madrid, where he received M.Sc. (2009) and Ph.D. (2012) degrees in Telematics Engineering. His current research interests are software mobile network stacks and the application of AI/ML solutions for next-generation mobile networks.

Albert Banchs has a double affiliation as a professor at the University Carlos III of Madrid and deputy director of IMDEA Networks Institute. He is an author of more than 100 publications, has been the Principal Investigator of 9 European Projects, and has served on many TPCs and Editorial Boards. He received his M.Sc. and Ph.D. degrees from the Polytechnic University of Catalonia (UPC) in 1997 and 2002.