ML-Assisted Beam Selection via Digital Twins for Time-Sensitive Industrial IoT

Nikita Zeulin, Aleksei Ponomarenko-Timofeev, Olga Galinina, and Sergey Andreev
Tampere University, Tampere, Finland
{nikita.zeulin, aleksei.ponomarenko-timofeev, olga.galinina, sergey.andreev}@tuni.fi

***Abstract*—In this article, we propose a machine learning (ML)-assisted beam selection framework that leverages the availability of digital twins to reduce beam training overheads and thus facilitate the efficient operation of time-sensitive IoT applications in dynamic industrial environments. Our approach employs a digital twin of the environment to create an accurate map-based channel model and train a beam predictor that narrows the beam search space to a set of candidate configurations. To verify the proposed concept, we perform shooting-and-bouncing ray (SBR) modeling for a reconstructed 3D model of an industrial vehicle calibrated using the real-world millimeter-wave (mmWave) propagation data collected during a measurement campaign. We confirm that lightweight ML models are capable of predicting the optimal beam configuration while enjoying considerably smaller size compared to the map-based channel model.**

I. Emergence of Industrial IoT

The fast-paced development of Industrial Internet of Things (IIoT) became an enabler for digital twins, which can be used to forecast deviations of the modeled object from its current state and make timely decisions to prevent malfunctions. High fidelity of modeling is achieved by accurate replication of the properties of the original entity, such as shapes, materials, and physics, thus aiding digital twins optimize many industrial tasks, from material tracking to predictive maintenance [1].

Digital twins found their applications in a range of dynamic scenarios of Industry 4.0, from human-robot interaction to warehouse automation, where multiple mobile agents co-exist within a common physical environment and collaboratively perform industrial tasks. The mobile agents, both robots and humans, may continuously stream video or readings from radars and lidars to an edge server tracking the status of operations, which require reliable wireless connectivity to maintain adequate and real-time orchestration of the mobile agents.

The operating quality of dynamic time-sensitive IIoT applications directly depends on the capability of the wireless network to provide high-rate radio connections. The prominent 5G New Radio (NR) [2] and IEEE 802.11ay [3] technologies operating in millimeter-wave (mmWave) spectrum can meet the stringent connectivity requirements such as those imposed by smart manufacturing. However, the high sensitivity of the mmWave links to the propagation environment requires directional transmission and, therefore, careful beam management: the device and the access point (AP) or the base station (BS) employ beam training procedures to select the optimal beam configuration before initiating a data transmission. In the case of highly directional connectivity, cumbersome beam training procedures result in degraded spectral efficiency of the link [4]. Developing efficient beam training algorithms can improve the operation of time-sensitive IIoT applications in dynamic industrial environments with high mobility and frequent link blockages.

Training overhead reduction is one of the matters being actively investigated by the wireless communications community. Among the proposed techniques, the most promising are those leveraging machine learning (ML) assistance and map-based channel models. The concept of ML-assisted beam training is to predict the best-aligned beam based on the previous measurements [5]. The procedure was proposed as a faster alternative to the conventional beam training, which can induce high training overheads, especially for larger antenna arrays. Trained ML models are capable of predicting the channel characteristics or the best antenna configurations, and their prediction accuracy is directly governed by the choice of the ML model, which should be selected to meet the complexity of the modeled environment and the availability of the training data. In contrast, map-based methods employ realistic channel models obtained, e.g., from ray-tracing simulations [6]. Although these methods provide more accurate results, they require pre-constructed 3D models of the environment, which should be precise enough to accurately reconstruct the channel. Moreover, a ray-traced channel model should be aligned with the real environment, which requires synchronizing the model at the reference points measured in the field. These features narrow down the range of practical tasks for the map-based methods, thus making their applicability limited.

In this article, we propose a framework that combines the advantages of ML-assisted beam training and map-based channel models. The core principle of our framework is to employ a map-based channel model as a source of the training data for an ML-based *beam predictor*, i.e., an ML model predicting a set of beam configurations between the IIoT devices and the AP/BS. Digital twins naturally integrate an estimation of the channel map and learning, which allows IIoT devices to readily use a pre-trained model for ML-assisted beam selection without preliminary data collection. Our framework seamlessly leverages digital twins to centralize the ML-related routines at the edge server, thus substantially reducing the device workload. The proposed approach can be combined with desired ML-based beam selection methods, while its envisaged applications may go beyond industrial scenarios, as discussed in [7].

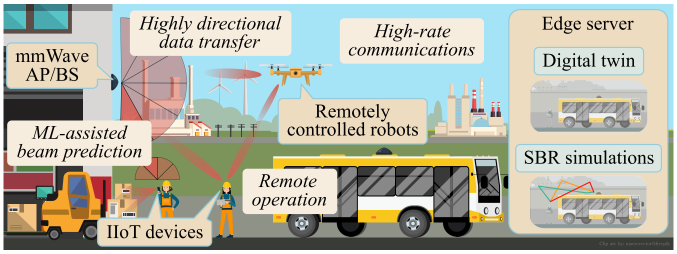

Fig. 1. Envisioned interplay between time-sensitive IIoT applications and ML-assisted mmWave radio.

II. Challenges in Time-Sensitive IIoT

In this section, we review the challenges of time-sensitive IIoT applications in dynamic industrial environments and discuss how mmWave and ML can facilitate their successful operation (see Fig. 1).

A. High-Rate Connectivity for Industrial Collaboration

Collaborative operation of mobile industrial agents — for example, semi-autonomous or remotely controlled robots — is sensitive to the throughput of the wireless links, which should be sufficient to handle the large volumes of streaming data, such as real-time video or high-resolution sensor readings [8]. Radio networks at the factory premises play a crucial role as a medium for industrial orchestration and monitoring and, therefore, should provide sufficient quality of service throughout the entire time of operation.

Despite the impressive achievable data rates, mmWave communication is susceptible to notable challenges. The intrinsic features of extremely high frequencies require careful alignment of the transmitter and receiver directional antennas to achieve the desired rates. The optimal beam pair is selected by running a beam training procedure based on sequentially measuring the channel quality at the receiver for all the beam configurations. Beam training overhead associated with the procedure can readily reach orders of milliseconds for larger antenna arrays [9], which is comparable to the latencies required by mission-critical IIoT applications [10].

The periodicity of beam training and the related reduction in spectral efficiency directly depend on the channel coherence. Dynamic environments with non-coherent, continuously changing channels require frequent beam training, thus yielding high training overheads and ineffective transmission. While one of the possible ways to mitigate this issue is to initiate beam training only in the cases of link outage [4], that strategy may not be able to resolve link drops due to blockages or changes in device orientation. Therefore, the requirements imposed by time-sensitive and dynamic IIoT applications call for developing an efficient framework that alleviates the current issues of mmWave connectivity.

B. Communication Reliability in Dynamic IIoT Applications

Reliability is one of the most important aspects of dynamic collaborative operations. In human-to-machine applications, robots may assist human workers in physically demanding tasks, thereby saving human time and energy as well as preventing potential injuries. However, a moving robot is a source of danger itself: irregular wireless connectivity may result in unpredictable or abrupt movements that may cause severe injuries to the nearby workers. Remote operations impose similar demands on the connection reliability: data transmission should be made seamless to achieve high precision of remote control. Hence, smoothness of operation largely depends on the underlying wireless network, which should be able to quickly (re-)establish new links and avoid breakdowns.

One of the possible ways to improve the reliability of mmWave connectivity is to reduce the time to recover from blockage, that is, decrease the overheads of beam training. The attention of the communications community is currently attracted to ML-assisted beam training [5], [9], [11]. In essence, these methods train the appropriate ML models to predict the (sub-)optimal beam(s) from the channel measurements, device coordinates, or other relevant parameters. In other words, a trained ML model may be thought of as an approximation of a full map-based channel model. What makes this approach attractive is the fact that the beam prediction procedure may take significantly less time as compared to the conventional beam training, especially when using exhaustive beam search for larger antenna arrays [9].

However, ML-based methods may fail to learn effectively in complex environments with irregular channel conditions, that is, with low-to-moderate correlation in two proximate positions. Complex ML models may have such potential, but their training process might be computationally infeasible, especially in online regimes, where the model is being continuously retrained on the channel measurements collected, e.g., from the user devices. Moreover, such ML models would require large volumes of training data collected in advance from across the entire environment. Therefore, ML-assisted beam training may require external assistance, which could be readily provided by the server.

III. ML-Assisted Beam Selection Framework

We envision a framework, where to create a map-based channel model, the server uses a precise 3D replica of the environment, which is already pre-constructed within the digital twin. In our framework, the generated channel map serves as a training set for the ML-based *beam predictor*, i.e., an ML model predicting beam configurations for the IIoT device and the AP/BS. As a result, the devices utilize a pre-trained ML model predicting a list of beam pairs without prior data collection. The key aspects of the proposed framework are discussed below.

A. Digital Twins and mmWave Channel Modeling

The intrinsic accuracy of a digital twin to the physical environment can be leveraged to create realistic channel models. Map-based channel estimation can be naturally integrated into the manufacturing processes since having a digital twin eliminates the need to create and maintain a separate digital model. In our proposed framework, the IIoT system considers a digital twin of the physical environment (Fig. 2, ①) as a

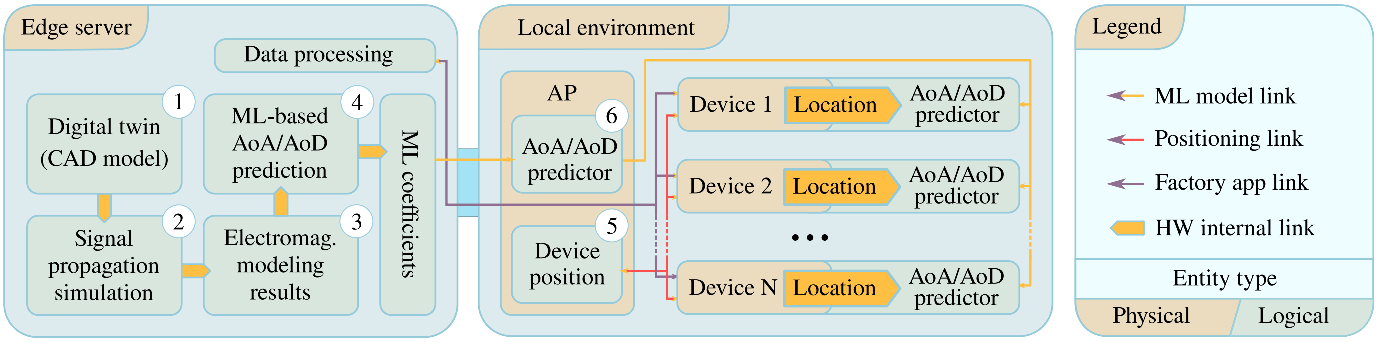

Fig. 2. Conceptual structure of proposed ML-assisted beam selection framework.

source of an accurate 3D model, which is used to reconstruct the corresponding map-based channel model. The computer-aided design (CAD) model incorporated into the digital twin is uploaded into the signal propagation simulator (Fig. 2, ②). Alternatively, when CAD models are unavailable, less precise virtual replicas can be constructed via photogrammetry methods from the photos of the objects or the environment. The results of simulations are transformed into a channel map containing pairs of angles of arrival (AoA) and departure (AoD) for each coordinate within the environment (Fig. 2, ③). The values of AoA and AoD are further converted into the optimal beam pairs, which are used by the devices to establish wireless links. The CAD model is static and, therefore, the described procedure can be performed once to create a static channel map.

The environment may have dynamic objects, such as human workers and robots, which act as reflectors and alter the channel model. Another feature of our framework is to periodically run the described pipeline to construct snapshots of the channel map based on the actual state of the environment. In this case, a static base layer is refined by adding dynamic layers containing pre-defined 3D models, or phantoms, of the objects, which were not originally included in the static CAD model. The phantoms may have different levels of detalization depending on the modeled object: the system may store detailed replicas of predictable agents while applying less specific primitives (e.g., a cylinder for a moving human) for unknown or volatile entities. The edge server may run the refinement procedure for each snapshot of the dynamic environment, which can be, however, computationally intensive depending on the parameters of the modeled space. Although such refinement of the channel map can be omitted, it may result in a better selection of beam pairs, especially when the dynamic objects are made of highly-reflective materials.

To preserve privacy, access to the CAD model can be restricted solely to the group of trusted parties. Additional security or privacy mechanisms could be applied to prevent a disclosure of the plain CAD model over the network. While our framework is compatible with both types of mechanisms, the security-centric approaches such as data encryption might be preferred since they do not impact the data fidelity. In contrast, privacy-centric mechanisms may introduce additional noise to the CAD model, thus potentially degrading the prediction quality.

The constructed map-based channel model can be employed as a beam predictor in the form of a table containing a list of the candidate directions to explore. Iterating over the selected candidates reduces the training overheads as compared to performing an exhaustive search. The main disadvantage of map-based beam predictors is in the need to explicitly store the selected beams for each coordinate in 2D or 3D space. The size of the stored table increases with that of the environment, and a map-based beam predictor may face storage limitations discernible for the IIoT devices. Additionally, in dynamic environments, a map-based channel model should be continuously recomputed to achieve better predictive performance. When such periodic updates are not feasible, the device can use outdated map-based beam predictors at the cost of degraded accuracy. These issues can be mitigated by using an ML-based beam predictor.

B. Improving Beam Selection with ML Models

The existing ML-assisted beam training methods share a common feature: their operation solely relies on the channel measurements collected by the devices in the considered environment. Representativeness and distribution of the training data are largely determined by the variety of locations where the measurements were collected. The training data may be spatially heterogeneous, and the ML model may fail to robustly predict the beam pair in less represented areas, which leads to lower transmission rates as compared to the results of an exhaustive beam search.

Our framework resolves this issue by utilizing the map-based channel model as a source of training data for the ML-based beam predictor. Since computational capabilities of the IIoT devices may be insufficient for ML-related processing, the edge server leverages the simulated channel measurements to train the ML model without assistance of the devices. In our framework, the ML model infers a set of the best-aligned beam pairs from the device coordinates (Fig. 2, ⑤). Importantly, the AP/BS and the IIoT device use the same ML-based beam predictor (Fig. 2, ⑥) to avoid beam misalignment. In contrast to the state-of-the-art approaches, our framework enables direct control over the size of the training set determined by the resolution of the map-based channel model. Hence, the performance of ML-based beam predictors is not limited by non-representative training data.

The complexity of the mobile environment directly affects the accuracy of ML-based beam predictors. To ensure the required prediction accuracy in highly dynamic scenarios, ML-based beam predictors could be retrained periodically. The frequency of retraining procedures directly depends on the changes in the environment, the capabilities of the digital twin to track them, and the computational resources of the server. The periodicity of retraining should also be adjusted to the level of mobility: the difference in accuracy from intensive model updates is marginal, while infrequent model updates may not capture the dynamics and, as a result, impair the beam prediction performance. Our framework is flexible to the choice of ML models; their complexity can be readily adjusted to balance between the effective resource constraints and the required accuracy levels.

The size of the ML-based beam predictor depends only on the ML model architecture. Lightweight models may enjoy sizes considerably smaller than those of map-based beam predictors. However, in terms of computation overheads, the map-based approach may outperform the ML-based methods if the entire dataset can be stored in the on-device memory and the training environment is an accurate and stable approximation of the real-world layout. In this case, the cost of a reduced size is a less accurate beam selection procedure. Although lightweight solutions can naturally compress flat channel maps, they may fail in predicting beam pairs for irregular environments. Alternatively, complex ML models could increase beam prediction accuracy, but their training may be computationally intensive, while their size is comparable to that of the channel maps. Hence, ML-based beam predictors have a complexity–performance tradeoff, which can be controlled by selecting the ML model architecture.

In dynamic scenarios where the channel map of the environment is refined periodically, the online operation of our framework can be improved further with transfer learning, representation learning, or model distillation. The edge server may employ a static channel map to train a heavier deep neural network model, which would predict the optimal beam configuration with very high accuracy. This network can be compressed, or distilled, into a smaller ML-based beam predictor, the architecture of which is selected depending on the prediction accuracy requirements as well as the computational and storage constraints of the IIoT device.

Representation learning may be employed to learn the representation of the data beyond the 3D coordinates suggested by our framework by default. Trajectories of mobile agents can be tracked to extract useful information on the temporal dynamics of their coordinates. In this case, the decision that the device makes at the inference stage could be based not only on the current coordinates but also on the history of its motion. The devices may benefit from previous information, thus making different decisions on the best beam configuration, for example, if they move along straight lines or have suddenly changed their direction. This approach has the potential to train a dynamic map instead of continuously retraining the static one, which reduces the complexity of retraining and broadcasting the model updates.

Transfer learning can further accelerate online retraining by reducing the complexity of ray-tracing simulations. In dynamic scenarios, two consequent snapshots of the environment have distinct yet similar channel models or domains in terms of the transfer learning. Then, the new ML-based beam predictor can adjust the weights of its outdated version to accurately predict in the new domain. Since the domains are similar, the edge server should compute only a small subset of the channel measurements used for refinement, thus decreasing the computational complexity of ray-tracing simulations.

IV. Selected Numerical Illustrations

In this section, we present a proof-of-concept prototype of the discussed framework, where we use a manually designed realistic 3D model of the environment. We briefly describe the selected scenario, discuss the calibration procedure using channel measurements, and present selected numerical examples for dynamic environments.

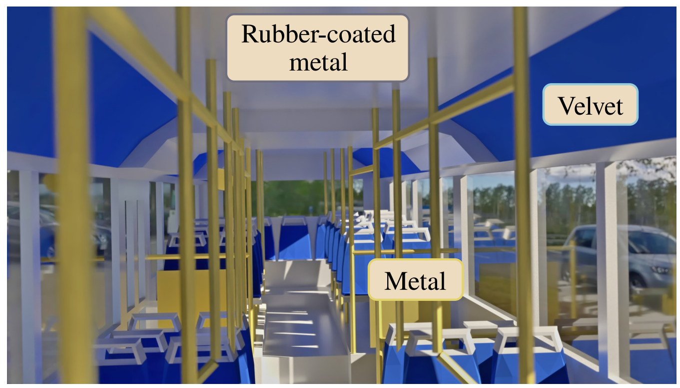

Fig. 3. Rendered reconstruction of utilized factory transport model.

A. Scenario and Metrics of Interest

We consider an industrial scenario with a mmWave network of IIoT devices deployed in a factory vehicle (Linkker electric bus in our example), which may transport workers engaged in collaborative remote control operation. As direct cellular connections might be hindered by severe signal attenuation caused by metal sheathing, an intra-vehicular AP acts as an aggregator connecting devices to the external network.

We assume that the factory has a digital twin utilized by the factory edge server to perform a shooting-and-bouncing ray (SBR) modeling. The output serves as a dataset for training an ML model, which is then made available to an intra-vehicle AP and its connected devices. The model is trained at the edge server using a static 3D model, while inference is performed locally at the devices, and its accuracy is assessed in a dynamic scenario with randomly moving blockers. To predict a set of the candidate beams, both AP and connected devices employ the trained ML model that provides a set of AoA/AoD pairs, which are translated into a beam configuration using the device position and antenna orientation. Predictors of the AP and the devices produce identical AoA/AoD pairs for equal inputs.

To create a map-based channel model of the vehicle, we perform an SBR simulation in Wireless InSite software [12]. To calibrate the modeled scenario, we utilize a dataset [13] (latency and power delay profile) produced during a prior measurement campaign for studying intra-vehicular mmWave signal propagation. We calibrate the positions of the intra-vehicular AP, devices, materials, and dimensions of the bus parts. The render of the resultant environment is illustrated in Fig. 3. The devices are located one meter above the floor, while the AP is located two meters above the floor in the front section. In our SBR setup, 10890 device positions form a 2D grid, which covers the entire bus interior. Parameters of the simulations are given in Table I.

For the metrics of interest, we consider (i) the norm of the difference between the predicted and the true AoA/AoD and (ii) the probability of the operational beam being in the predicted configuration list. The norm of AoA/AoD prediction error allows assessing the general predictive ability of the system regardless of the antenna parameters at both the AP and the device. The probability of the event where the model captures the operational beam within the half-power beamwidth (HPBW) corresponds to a reduction in the overheads related to the beam training procedure.

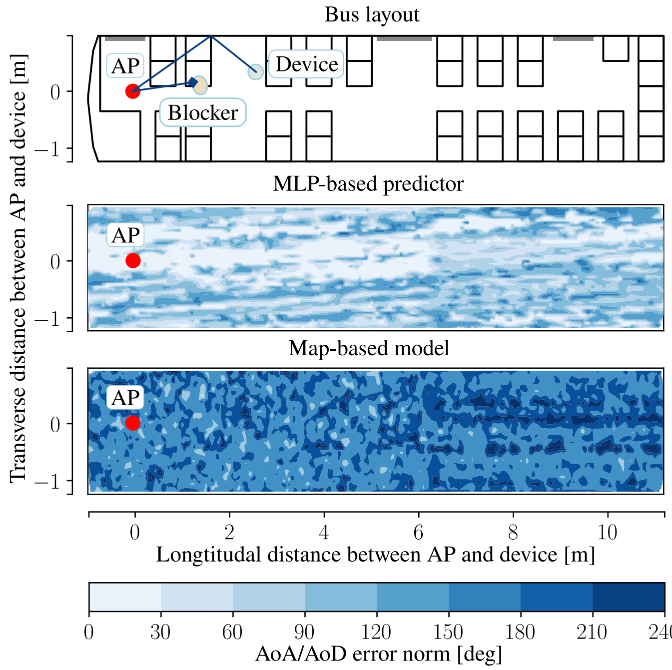

Fig. 4. Angular error norm for MLP- and map-based models.

TABLE I
SIMULATION SETTINGS

Parameter [Unit]	Value
Center frequency [GHz]	61
Bandwidth [GHz]	4
Radiated power [dBm]	0
MPC number	5
AP height [m]	2
Device height [m]	1
Number of device locations	10890
Bus dimensions (W, H, L) [m]	(2.5, 2.5, 12.5)

B. Interpreting Simulation Results

We consider an ML-based beam predictor based on a multi-layer perceptron (MLP) regressor (one hidden layer of 20 neurons), which has previously been successfully applied for beam prediction [14]. Our testing procedure relies on 3000 random locations in a dynamic environment with blockages, while the rest is used for training on the static data. The number of strongest multipath components is limited to 5.

The heatmaps in Fig. 4 illustrate an estimate of an angular error norm between the AoA/AoD values predicted by the ML-based vs. the map-based beam predictors. The ML-based predictor is trained on full SBR data, while the map-based option employs sparse data and interpolation to maintain a similar model size and mimic the use of data collected by one device. The MLP-based predictor significantly outperforms its map-based counterpart in terms of the error norm, thus providing a sensible balance between model size and prediction accuracy. We note that the range of acceptable norm values cannot be defined explicitly as it depends on the system parameters, such as HPBW and mobility level. For example, norm values over 120° correspond to the prediction offset of more than 60° at both devices ($120 = \sqrt{(60^2+60^2)+(60^2+60^2)}$, based on squared azimuth and elevation angle errors for both the AP and the device), which would result in performance degradation for narrower beams.

The dependency between the probability of predicting a valid beam configuration and the antenna HPBW is assessed in Fig. 5. For example, valid beams can be captured with the probability of at least 0.97 if the antenna HPBW is greater than 60°. The probability of successful beam prediction does not reach one even for higher values of HPBW, which is due to blocked beam configurations. The results in Fig. 5 correspond to the use of full training set since reducing the training set size (decreasing the channel model resolution) significantly degrades the MLP-based model performance.

The benefits of our framework in terms of reduced overhead as compared to exhaustive search are illustrated in Fig. 5. Here, the inference time is 305 ns, which is negligibly small compared to, e.g., 15.91 μs and 9.96 μs for one sector level sweep (SLS) and short sector sweep (SSW) of IEEE 802.11ad/ay [3]. For the beam refinement protocol (BRP), the overhead reduction is slightly lower in absolute values due to shorter frames. We may conclude that application of the ML-based predictor is most advantageous for SSW and SLS frames, while for BRP, its use offers marginal benefits in terms of absolute values.

In addition to faster beam selection, our framework reduces the amount of data stored at the devices. A full map-based model provides superior accuracy as the AoA/AoD data are stored for all the locations but rapidly grows in size. Here, the full map-based model has the volume of 3.3 MB, while the ML model is only 250 KB. These numbers hold for one plane of 3D device positions. In reality, the entire volume of the bus interior may be sampled in the SBR tools, and the size of data increases as more layers are added to the modeling. For the bus of 2.5 meters height, adding a new layer every

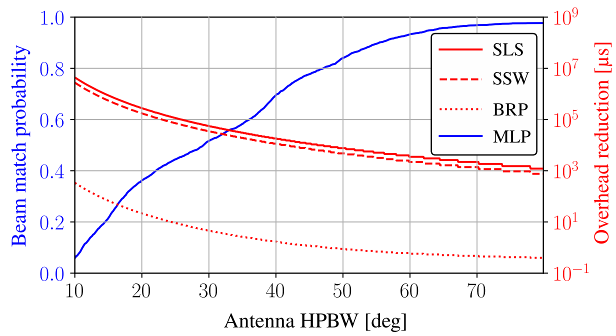

Fig. 5. Successful beam prediction probability and overhead reduction.

10 cm would yield a 25-times increase in the model size up to 83 MB. Placing a new layer every cm would result in the data scaling up to 830 MB.

While the data may theoretically be compressed and processed by the device locally (as a map-based beam predictor), there is an issue of higher power consumption for mobile devices. Further, even if network resource constraints are satisfied, the inference should be performed within a certain deadline, which is challenging to achieve in the presence of high mobility. Devices may also pre-cache parts of the map-based models for inference, but with frequent mobility, data may lose relevance rapidly. The ML-based beam predictor maintains a constant size regardless of the data that the model employs for training, which is especially important for dynamic scenarios where the model can be stored in the device memory.

V. Discussion and Conclusions

IIoT enables multiple applications and utilities by linking physical and digital domains. To maintain usability, these require a reliable communication channel with small downtimes. The mmWave networks provide higher data rates but are susceptible to beam training overheads, which can be effectively reduced if only a subset of beam configurations is considered. In this article, we proposed a framework that employs digital twin-based SBR modeling results to train an ML model for predicting a subset of candidate AoA/AoD options. This strategy allows the utilization of information from ray-tracing of the environment blueprints without the need for measurements, which are expensive, time-consuming, and in some cases impossible. With the proposed approach, beam selection times can be reduced, thus improving the overall network performance. Moreover, the devices may first receive a pre-trained model based on the digital twin and then better adapt it to their environment.

Importantly, SBR tools employed to produce results for ML model training should be sufficiently accurate; otherwise, the ML model may train on unrealistic data and offer incorrect AoA/AoD pairs, thus jeopardizing the entire framework. While more accurate modeling may increase compute complexity, the edge server might have higher computing capacity and can produce more accurate ML models. For example, with our available hardware (GTX 550Ti, 2012), the experiment completed in two days. With more powerful equipment, the SBR simulations may run even quicker. Furthermore, our proposed framework is suitable for indoor/outdoor deployments and may be extended to permit even more complex and precise simulations such as Finite Domain Time Difference or Finite Element Method. In addition to more accurate channel simulations, alternative or tailor-made ML models could also be explored. One of the future research directions might be selection and design of the optimal ML model utilized to predict the AoA/AoD pairs in a specific environment. The optimization may be performed in terms of the prediction accuracy and inference/training complexity.

Going forward, our framework can be extended to fully embrace the non-stationary nature of real-world deployments. If a scenario is highly dynamic, the model should be updated regularly; otherwise, devices might fall back to the conventional beam training. Several approaches can be taken for retraining the model. If the dynamics of the environment follow a cyclic pattern (e.g., a subway), it may be reasonable to train a series of predictors for each state of the environment. If no pattern can be established (e.g., a public event), then one may apply online learning mechanisms instead of retraining the model from scratch. As digital twins for buildings and outdoor areas are becoming more common, they facilitate signal propagation modeling of these areas and help produce data that devices cannot obtain in the field. The challenge of building an efficient framework for predicting the AoA/AoD is open and, with recent advancements in mixed reality, offers promising perspectives for many IIoT services.

References

[1] J. Cheng, W. Chen, F. Tao, and C.-L. Lin, "Industrial IoT in 5G environment towards smart manufacturing," *Journal of Industrial Information Integration*, vol. 10, pp. 10–19, 2018.

[2] 3GPP working group, "3GPP TR 21.916 V16.1.0," 3rd Generation Partnership Project (3GPP), Tech. Rep.

[3] IEEE LAN/MAN Standards Committee, "IEEE Std 802.11-2020: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Computer Society, Tech. Rep.

[4] V. Va, J. Choi, and R. W. Heath, "The impact of beamwidth on temporal channel variation in vehicular channels and its implications," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 5014–5029, 2017.

[5] Y. Heng and J. G. Andrews, "Machine learning-assisted beam alignment for mmWave systems," *IEEE Transactions on Cognitive Communications and Networking*, 2021.

[6] Y.-G. Lim, Y. J. Cho, M. S. Sim, Y. Kim, C.-B. Chae, and R. A. Valenzuela, "Map-based millimeter-wave channel models: An overview, data for B5G evaluation and machine learning," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 54–62, 2020.

[7] R. Kerris. Ericsson Builds Digital Twins for 5G Networks in NVIDIA Omniverse. [Online]. Available: https://blogs.nvidia.com/blog/2021/11/09/ericsson-digital-twins-omniverse/

[8] F. Alriksson, D. H. Kang, C. Phillips, J. L. Pradas, and A. Zaidi, "XR and 5G: Extended reality at scale with time-critical communication," *Ericsson Technology Review*, 2021.

[9] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37 328–37 348, 2018.

[10] J. Yang, B. Ai, I. You, M. Imran, L. Wang, K. Guan, D. He, Z. Zhong, and W. Keusgen, "Ultra-reliable communications for industrial Internet of Things: Design considerations and channel modeling," *IEEE Network*, vol. 33, no. 4, pp. 104–111, 2019.

[11] H. Echigo, Y. Cao, M. Bouazizi, and T. Ohtsuki, "A deep learning-based low overhead beam selection in mmWave communications," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 682–691, 2021.

[12] RemCom Incorporated, "RemCom Wireless InSite software," https://www.remcom.com/wireless-insite-em-propagation-software.
[13] W.I.N.T.E.R. group. (2018) WiFiUS project Intra-vehicular measurement data. [Online]. Available: https://cloud.winter.rd.tuni.fi/index.php/s/NAKqarUok8jYxyo
[14] G. Charan, T. Osman, A. Hredzak, N. Thawdar, and A. Alkhateeb, "Vision-position multi-modal beam prediction using real millimeter wave datasets," *arXiv preprint arXiv:2111.07574*, 2021.