Deep Learning for Moving Blockage Prediction using Real mmWave Measurements

Shunyao Wu, Muhammad Alrabeiah, Andrew Hredzak, Chaitali Chakrabarti, and Ahmed Alkhateeb
School of Electrical, Computer and Energy Engineering
Arizona State University Tempe, AZ 85287
Email: {vincentw, malrabei, ahredzak, chaitali, alkhateeb}@asu.edu

***Abstract*—Millimeter wave (mmWave) communication is a key component of 5G systems and beyond. Such systems provide high bandwidth and high data rate but are sensitive to blockages. A sudden blockage in the line of sight (LOS) link leads to abrupt disconnection. Thus addressing blockage problems is essential for enhancing the reliability and latency of mmWave communication networks. In this paper, we propose a novel solution that relies only on in-band mmWave wireless measurements to proactively predict future dynamic line-of-sight (LOS) link blockages. The proposed solution utilizes deep neural networks and special patterns of received signal power, that we call *pre-blockage wireless signatures*, to infer future blockages. Specifically, the machine learning models attempt to predict: (i) Whether a blockage will occur in the next few seconds? (ii) At what time instance will this blockage occur? To evaluate our proposed approach, we build a mmWave communication setup with moving blockage and collect received power sequences. Simulation results on a real dataset show that blockage occurrence can be predicted with more than 85% accuracy, and the exact time instance of blockage occurrence can be obtained with less than 2 time instances (1.66s) error for prediction interval of 10 time instances (8.8s). This demonstrates the potential of the proposed solution for dynamic blockage prediction and proactive hand-off.**

***Index Terms*—Millimeter wave, machine learning, blockage prediction, handover**

I. Introduction

Communication in the mmWave frequency range offers high bandwidth and higher data rate demands required by 5G and beyond cellular systems [1]. Unfortunately such systems perform poorly in presence of line-of-sight (LOS) link blockages, which could cause sudden link failures, impacting the reliability and latency of the mobile networks. This problem is particularly important in mmWave/sub-THz systems because of (i) their reliance on LOS communications for sufficient receive signal power and (ii) the high penetration loss of these high-frequency signals (high sensitivity to blockages). One approach to address this challenge is equipping the mmWave system with the capability to predict possible blockage proactively. A successful prediction will allow a base station to take mitigation measures, e.g., user hand-off, before the link is blocked, thereby resolving the reliability and latency problems. In this paper, **we investigate the potential of leveraging *wireless mmWave/sub-THz in-band signatures* to proactively predict future link blockages** and attempt to answer two main questions: (i) Can in-band wireless signals be utilized to predict future blockages? (ii) Can these signals also predict when a blockage will happen in the future?

Many recently published studies have used machine learning to address problems arising from link blockages in MIMO and mmWave communications [2]–[6]. The work in [2], [3], [5] collectively demonstrates the ability of a machine learning algorithm (whether deep [3] or shallow [2]) to identify or differentiate LOS and NLOS links. This identification task could be an interesting ability of a system operating in the sub-6 GHz range. However, the requirements are more stringent for a mmWave system and demand a proactive approach. A step towards doing so is presented in [4], where a proactive solution is proposed to predict stationary blockages. This solution depends on beam sequences alone, and as such, cannot handle dynamic blockages. Another direction addressing blockage prediction relies on Vision-Aided Wireless Communications [6], [7], where proactive blockage prediction is done using images and sub-6GHz channels [7] or mmWave beams [6]. The work requires extra sensory data (images), and it does not take full advantage of the wireless data.

In this paper, we propose a machine learning algorithm to address the dynamic-blockage prediction problem. The algorithm uses sequences of received power to predict whether a blockage is incoming or not. The basic idea behind our algorithm is the ability to recognize *pre-blockage signature*, a sequence of received signal power prior to blockage occurrence. We argue that such a signature could serve as a important clue for incoming blockages. Our contribution is summarized as follows:

1) We propose a new approach that leverages only the in-band mmWave signals to proactively predict future dynamic LOS link blockages before they happen. This approach does not require any extra sensors or out-of-band measurements and can work transparently to the normal operation of the communication systems.
2) We propose a recurrent neural network (RNN) architecture based on Gated-Recurrent Units (GRUs) to predict incoming blockages. The architecture is designed to learn one of two tasks: i) predict whether a blockage is incoming or not, ii) pinpoint the time instance at which the blockage occurs.
3) We develop a mmWave communication setup with a moving blockage. We use that setup to build a dataset of received power sequences and their corresponding future link statuses. The proposed algorithm uses this dataset to predict incoming blockages.

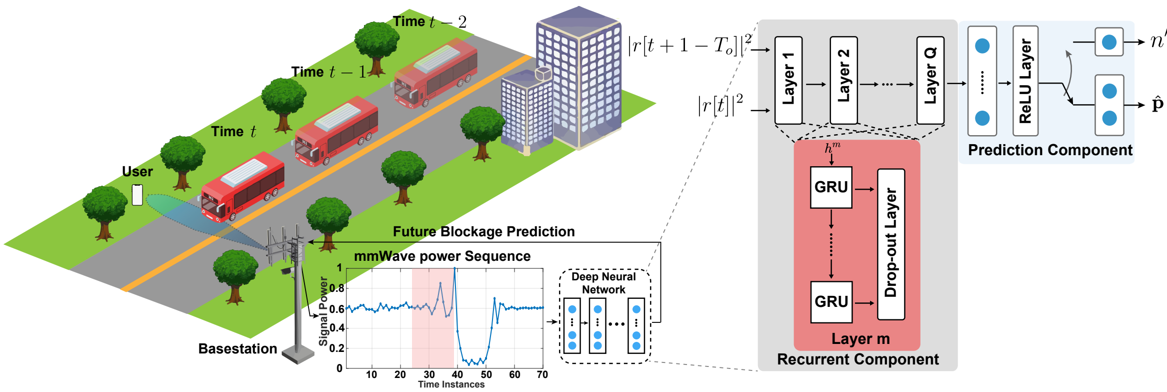


Fig. 1: The left panel illustrates the overall system model where a mmWave/sub-THz basestation utilizes the received mmWave/sub-THz signal power to enable the proposed proactive dynamic link blockage prediction approach. The right panel shows the overall RNN architecture to predict the blockage occurrence. The architecture consists of (i) recurrent component, and (ii) prediction component.

The rest of this paper is organized as follows. Section II presents the system and channel models adopted to study the dynamic-blockage prediction. Section III presents a formulation of the prediction problem. The proposed machine learning model is presented in Section IV. The data collection scenario and setup is introduced in Section V. Proposed algorithm evaluation and simulation results are presented in Section VI, and, finally, the paper is concluded in Section VII.

II. System and Channel Models

System model: The communication system considered in this work is described in Fig.1. It assumes a base station serving a static user who is in the vicinity of a possible moving blockage. The base station employs an M-element Uniform Linear Array (ULA) antenna operating at 60GHz carrier frequency with Orthogonal Frequency Division Multiplexing (OFDM). It also assumes a fully analog architecture with a predefined beam-steering codebook $\mathcal{F} = \{\mathbf{f}_w\}_{w=1}^{W}$, where $\mathbf{f}_w \in \mathbb{C}^{M\times 1}$ is given by:

$$\mathbf{f}_w = \frac{1}{\sqrt{M}}\left[1, e^{j\frac{2\pi}{\lambda}d\sin(\phi_w)}, \dots, e^{j(M-1)\frac{2\pi}{\lambda}d\cos(\phi_w)}\right]^T, \quad (1)$$

where d is the spacing between the ULA elements, λ is the wavelength, and $\phi_w \in \{\frac{2\pi w}{W}\}_{w=0}^{W-1}$ is a uniformly quantized azimuth angle with a step of $1/W$. At any time instance t, the downlink received signal is expressed as follows:

$$r_k[t] = \mathbf{h}_k[t]^T \mathbf{f}_w s_k[t] + n_k \quad (2)$$

where $\mathbf{h}_k[t] \in \mathbb{C}^{M\times 1}$ is the downlink channel at the kth subcarrier, $s_k[t]$ is the symbol transmitted on the kth subcarrier, and, finally, n_k is a complex Gaussian noise sample, $\sim \mathcal{CN}(0,\sigma^2)$ at the kth subcarrier.

Channel model: This work adopts the geometric (physical) channel model, which captures the physical characteristics of signal propagation including the dependence on the environment geometry, materials, frequency band, etc., [1]. With this model, the channel can be expressed as:

$$\mathbf{h}_k = \sum_{d=0}^{D-1}\sum_{\ell=1}^{L} \alpha_\ell e^{-\jmath\frac{2\pi k}{K}d} p\left(dT_\mathrm{S} - \tau_\ell\right) \mathbf{a}\left(\theta_\ell, \phi_\ell\right), \quad (3)$$

where L is number of channel paths, α_ℓ is the path gain (including the path-loss), τ_ℓ is the delay, θ_ℓ is the azimuth angle of arrival, and ϕ_ℓ is the elevation angle, of the ℓth channel path. T_S represents the sampling time while D denotes the cyclic prefix length (assuming that the maximum delay is less than DT_S).

III. Problem Formulation

Proactively identifying Line of Sight (LOS) link status has significant advantages both at the physical and network layer levels. In this paper, we focus on two specific problems: (i) How to use the received mmWave signal power information to predict whether there is a blockage in the near future or not, and (ii) in case there is a blockage, how to use the received mmWave signal power information to predict when that blockage will occur.

Problem 1: To formulate the presence of a blockage in the near future, let $t \in \mathbb{Z}$ be the index of the discrete time instance, $x[t]$ be the link status at the tth time instance, and $S_{ob} = \{|r[t+n]|^2\}_{n=-T_o+1}^{0}$ be the sequence of received signal power for the observation interval of T_o instances. Note that for simplicity, the subcarrier index k is omitted from $r[t+n]$. Given a signal power based observation sequence, we want to predict the occurrence of blockage within a future time interval extending over T_P instances. We use b_{T_P} to indicate whether there is a blockage occurrence within that interval or not. More formally, b_{T_p} is defined as follows:

$$b_{T_P} = \begin{cases} 0, & x[t+n'] = 0 \quad \forall n' \in \{1,\dots,T_P\} \\ 1, & \text{otherwise} \end{cases} \quad (4)$$

where 1 indicates the occurrence of a blockage and 0 is the absence of blockage. The goal of this problem is to predict b_{T_p} with high accuracy, i.e., high success probability $\mathbb{P}(\hat{b}_{T_p} = b_{T_p}|S_{ob})$ where $\hat{b}_{T_p}$ is the predicted link status. To that end, a prediction function $f_\Theta(S_{ob})$ parameterized by a set of parameters Θ could to be learned using a machine learning algorithm such that it maximizes $\mathbb{P}(\hat{b}_{T_p} = b_{T_p}|S_{ob})$.

Problem 2: Given signal power based observation sequence S_{ob} and the knowledge that there is a blockage in the future T_p instances, the goal is to predict n' at which $x[t+n'] = 1$. This represents the exact instance at which the blockage

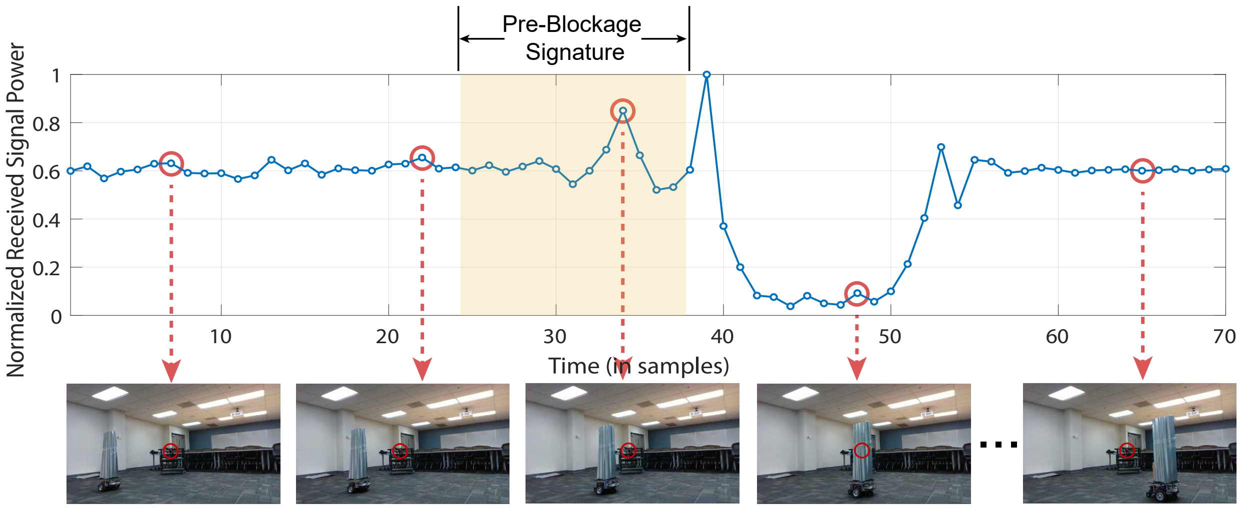


Fig. 2: Example of indoor pre-blockage signature: The upper subfigure shows the received signal power vs time. The bottom panel shows images captured by the camera.

occurs within a window of T_p instances. Similar to **Problem 1**, the future instance is predicted by a parameterized function $g_\Theta(\mathcal{S}_{ob})$ that could be learned using a ML algorithm. The aim of the ML algorithm is to maximize the prediction accuracy $\mathbb{P}(\hat{n}' = n'|\mathcal{S}_{ob}, b_{T_p} = 1)$.

IV. Moving Blockage Prediction using Recurrent Neural Networks

A. Key Idea

Any wireless environment can be broadly broken down into two categories of objects, stationary and dynamic. Both shape the characterization of the wireless channel [8], and as a result, the behavior of these objects (whether stationary or dynamic) affect the behavior of the wireless channel. Building on that, the key blockage prediction approaches in this paper are based on the following simple observation: Consider a fixed transmitter and receiver in a certain environment with a LOS path. If an object moves in this environment till it blocks this LOS path, then, during the movement, the object acts like a scatterer for the signal propagating from the transmitter to the receiver. The received signal during this interval will experience a constructive and destructive interference from the LOS ray and the ray scattered on the moving object. Further, the contribution of the moving blockage/scatterer will change as the scatterer approaches the LOS link and before it blocks the link. **We call this receive signal pattern that precedes the occurrence of a blockage and reflects the behavior of the blocking object the *pre-blockage wireless signature*.**

Fig. 2 illustrates the example in indoor scenario. It shows a sequence of received power versus time instances, and the corresponding photos show how far the blockages (the metal object in the photos) is from the transmitter (the object circled with red in the photos). The received power starts with smooth fluctuations (between the 1-st and 30-th instances in Fig. 2), for the blockage is far from both the receiver and the transmitter. However, as the blockage advances, the received power sequence changes, as shown in the red-shaded region of Fig. 2. The sharper fluctuations is what we refer to as the pre-blockage signature, and it could be utilized to identify incoming blockages.

B. Deep Learning Model

Neural Network Architecture: Learning the pre-blockage signature from a sequence of observed received signal power requires a neural network that processes input data samples over time such as recurrent neural networks. We design a Gated Recurrent Unit (GRU) network [9] of Q-layers that takes in a sequence of observed received signal power (i.e., S_{ob}) and learns to predict the link status b_{T_p}. Fig.1 depicts the schematic of such a network. Each layer in the network consists of T_o GRUs, where T_o is the length of the observation time interval. The output of the last GRU of the last layer is fed to a Fully Connected (FC) layer followed by either a classifier for **Problem 1** or a regressor for **Problem 2**. The classifier outputs a probability vector ($\hat{\mathbf{p}}$) of whether the link status is blocked or not in T_P future time instances. For **Problem 2**, the regressor outputs the predicted time instance $\hat{n}'$ indicating the time instance when the blockage will occur.

Pre-Processing: The input data is pre-processed to make it suitable for our model to learn, see [10] for more details. We choose to standardize the inputs by subtracting the mean μ of the dataset and dividing by its standard deviation σ. Let $\mathbf{A} \in \mathbb{R}^{U \times N}$ be the dataset matrix with U rows and N columns to represent power sample data of U data points. Each row represents a data point and N is the number of received power samples of each data point. Data standardization is done by computing:

$$\hat{\mathbf{A}}_{u,n} = \frac{\mathbf{A}_{u,n} - \mu}{\sigma}, \tag{5}$$

where:

$$\mu = \frac{1}{N \times U} \sum_{u=1}^{U} \sum_{n=1}^{N} \mathbf{A}_{u,n} \tag{6}$$

$$\sigma = \sqrt{\frac{1}{N \times U} \sum_{u=1}^{U} \sum_{n=1}^{N} (\mathbf{A}_{u,n} - \mu)^2} \tag{7}$$

$$\forall u \in \{1, \ldots, U\} \text{and } n \in \{1, \ldots, N\}, \tag{8}$$

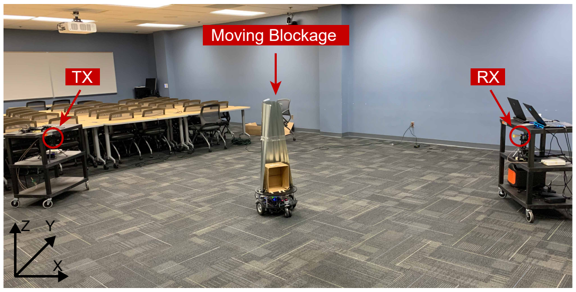


Fig. 3: Data collection setup for indoor scenario, where a blockage moves along a trajectory between the transmitter (TX) and the receiver(RX).

TABLE I: Parameters for mmWave Communication System

Name	Value
Carrier Frequency	60GHz
Signal Bandwidth	20MHz
number of subcarriers	64
Horn Antenna Gain	20dBi
Transmit Power	30dBm

Training loss: For **Problem 1** the future link-status prediction is posed as a binary classification problem, in which the classifier attempts to determine whether the link is blocked or not within the future time interval. As such, the network training is performed with a cross entropy loss function [11] computed over the outputs of the network:

$$l_{\text{CH}} = \sum_{c=1}^{2} p_c \log \hat{p}_c, \tag{9}$$

where $\mathbf{p} = [p_1, p_2]^T$ is the one-hot vector which represents the categorical variables as binary vector; the category with highest probability is encoded as 1 others are encoded as 0's. It takes one of two values: $[1, 0]^T$ for the case when $b_{T_p} = 0$ and $[0, 1]^T$ for the case when $b_{T_p} = 1$, and l_{CH} is the training loss computed for one data point.

For **Problem 2**, we pose the problem of predicting the blockage instance as a regression problem. Our model tries to determine the exact time instance at which the blockage occurs. We use Mean Square Error (MSE) loss as training function. In formal terms, we aim to minimize the difference between the predicted instance and groundtruth instance [11]

$$l_{\text{MSE}} = (n'^{(u)} - \hat{n}'^{(u)})^2 \tag{10}$$

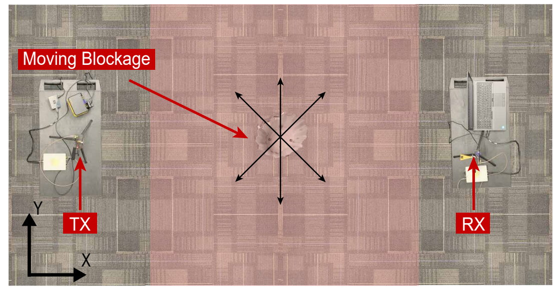
where $n'^{(u)}$ and $\hat{n}'^{(u)}$ are ground truth time instance and predicted time instance, respectively.

V. Experimental Setup and Evaluation Dataset

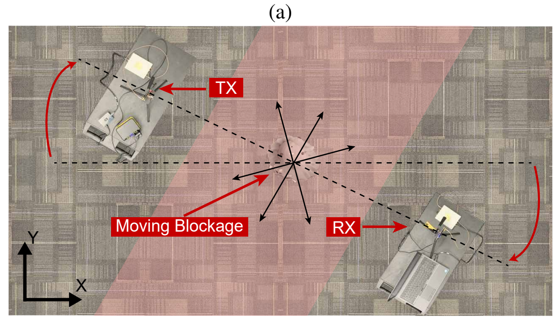
To evaluate our approach, we build a mmWave testbed and create a dataset, the details of which are explained below.

A. Communication Scenario & Testbed

We build a mmWave communication system comprising of a transmitter with an omnidirectional antenna that communicates with a receiver that has a 10-degree beamwidth horn antenna. The operating parameters for our mmWave communication system are shown in Table I. To simulate a moving blockage,



(a)



(b)

Fig. 4: Top-view of the experimental scenario. (a) Directions and area of the moving blockage and the relative positions of transmitter (TX), receiver (RX) and moving blockage. (b) Rotated TX-RX setup.

we use a metal cylinder with a height of 1m, which can completely block the LOS link between the transmitter and receiver. Then, we mount that cylinder onto a programmed robot that moves along a pre-defined trajectory to simulate the moving blockage. The speed of the robot is 0.0625m/s and the sampling rate of our system is 1.13 sample/s, which means a single time instance takes about 0.88s. Fig. 3 depicts our experimental system.

We consider an indoor scenario where the transmitter and receiver are placed 8 m apart from each other and the robot moves between them in different trajectories. The moving area of the robot is shown as red-shaded region in both Fig. 4a and Fig. 4b. To illustrate this further, let's consider the coordinate system in Fig. 3, where the x-axis extends along the LOS between the transmitter and receiver and y-axis extends perpendicular to the LOS between the transmitter and receiver; z-axis is perpendicular to the ground. The robot moves back and forth on the y-axis to create multiple back and forward trajectories with a spacing of 1 m. In order to make our dataset diverse, we also program the robot to create trajectories with different angles. In Fig. 4a and Fig. 4b, arrows are used to show those trajectories. Furthermore, the whole testbed is rotated with a small angle on the x-y plane such that the background is slightly changed, and the robot is programmed to do another back and forth cycle. See Fig. 4a and Fig. 4b for an example of the robot motion in the rotated testbed. By continuously moving and rotating, we collect a dataset of power sequences for different motion and propagation patterns.

B. Dataset Generation

Every robot trajectory provides us with a single received-power sequence, which is manually annotated to create the link status labels. We use 1 to indicate the time instances

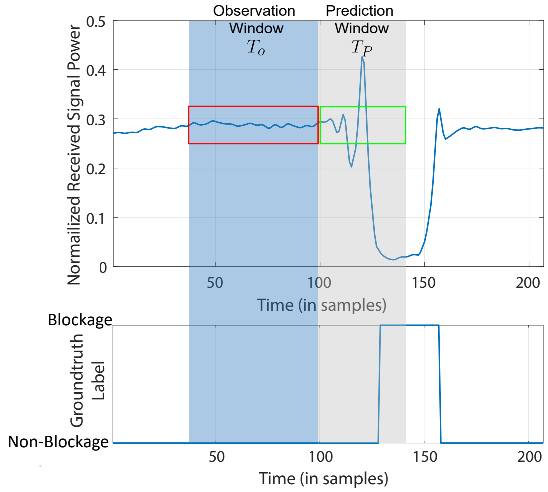


Fig. 5: Sequence generation using sliding window. The top image shows the received power for a raw sequence. The bottom image shows the corresponding link status.

at which the LOS link is blocked and 0 otherwise. We call the pair of received power and link status sequence a raw sequence pair. Furthermore, since the number of raw sequence pairs we collected from the experiment is limited, data augmentation is used to increase the dataset size. Originally, we conducted 158 experiments and generated 158 sequence pairs, i.e. $\mathcal{S}_1 = \{(S_{d1}, x_{d1})^{(u)}\}_{u=1}^{U_d}, U_d = 158$, d_1 means the original input. We generate additional pairs by dropping samples at rates 2, 3, and 4, which results in $\mathcal{S}_2 = \{(S_{d2}, x_{d2})^{(u)}\}_{u=1}^{U_d}$, $\mathcal{S}_3 = \{(S_{d3}, x_{d3})^{(u)}\}_{u=1}^{U_d}$, $\mathcal{S}_4 = \{(S_{d4}, x_{d4})^{(u)}\}_{u=1}^{U_d}$, respectively. In reality, this procedure means the blockage moves along the same trajectory at 2, 3 or 4 times the original speed. Next we concatenate these sequences together as $\mathcal{S} = \mathcal{S}_1 \bigcup \mathcal{S}_2 \bigcup \mathcal{S}_3 \bigcup \mathcal{S}_4 = \{(S, x)^{(u)}\}_{u=1}^{U}$ where $U = 4U_d$.

The method to generate the data points for **Problem 1** and **Problem 2** are as follows.

Problem 1: A data point consists of an input received signal power sequence S_{ob} and an input label b_{T_p}. To generate the received signal power sequence, we use a sliding window method, shown in Fig. 5. For example, for current time t, we generate S_{ob} by extracting the received power sequence from time instance $t - T_o + 1$ to t, shown as red box in the top subplot in Fig. 5. For input label generation, we first extract the label sequence from time instance $t+1$ to $t+T_P$, shown as green box in bottom subplot in Fig. 5, to generate the sequence $\{x[t+n]\}_{n=t}^{t+T_P}$. Then we generate input labels b_{t_P} using eqn. 4. Finally, we pair the received signal power sequences with input labels, expressed as $S_{P1} = \{(S_{ob}, b_{T_P})^{(u)}\}_{u=1}^{U_{P1}}$, where U_{P1} is the total number of samples that are input to our model. In this problem, S_{P1} is a mixture of two categories, non-transition sequence pairs with $b_{T_P} = 0$ and transition sequence pairs with $b_{T_P} = 1$. To eliminate the dataset bias, we keep the ratio of transition and non-transition sequence pairs to 1:1.

Problem 2: A data point is represented by input received signal power sequence S_{ob} with the ground-truth time instance n' instead of the label. So $S_{P2} = \{(S_{ob}, n')^{(u)}\}_{u=1}^{U_{P2}}$, where U_{P2} is the total number of samples that are input to our model. For this problem, we only select transition-sequence pairs for the input dataset.

TABLE II: Parameters for Deep Learning Model

Name	**Value**	
	Problem 1	**Problem 2**
Input sequence length	10	10
Predicted future time steps	1-40	1-40
Hidden state of RNN	20	20
Output dimension	2	1
Number of RNN layer	1	1
Dropout rate	0.2	0.2
Epoch	1000	1000

In our experiments, we pick input sequence length (T_o) as 10 and prediction interval (T_p) from 1 to 40. For each T_p, each raw sequence pair generates one transition-sequence pair and multiple non-transition sequence pairs. Totally, we get 632 transition-sequence pairs for each T_p.

VI. Experimental Results

In this Section, the metrics used to evaluate the performance is introduced in subsection VI-A, and the details of parameters for our neural network are shown in subsection VI-B. The analysis of the results is presented in subsection VI-C.

A. Evaluation Metrics

Since **Problem 1** is a classification problem, we use Top-1 accuracy as our evaluation metric. It is defined as the compliment of the classification error given in [12].
Problem 2 is posed as a regression problem, and so we use Mean Absolute Error (MAE) and its standard deviation to evaluate the quality of our model predictions. The MAE is the mean absolute error between ground-truth value and predicted value. For each prediction interval T_P, we calculate $\bar{e}_{T_P}$, the MAE over all the samples, and std_{T_P}, the standard deviation of the absolute errors.

$$e_{T_P}^{(u)} = |n'^{(u)} - \hat{n}'^{(u)}|, \quad \forall u \in \{1, \ldots, U_{v2}\}, \tag{11}$$

$$\bar{e}_{T_P} = \frac{1}{U_{v2}} \sum_{u=1}^{U_{v2}} |n'^{(u)} - \hat{n}'^{(u)}|, \tag{12}$$

$$\mathrm{std}_{T_P} = \sqrt{\frac{1}{U_{v2}} \sum_{i=1}^{U_{v2}} \left(e_{T_P}^{(u)} - \bar{e}_{T_P}\right)^2}, \tag{13}$$

where, $e_{T_P}^{(u)}$ is the absolute error for uth sample, U_{v2} is the total numbers of samples in validation set, $n'^{(u)}$ and $\hat{n}'^{(u)}$ are target and predicted time instances between current time and the time of blockage occurrence, for prediction interval is T_P.

B. Network Training

We build the deep learning model described in Section IV using Pytorch. It consists of 1 GRU layer with a dropout layer, which has been chosen empirically by experimenting on the dataset. The details of the model are listed in Table II. We input 10 successive time-instances of received signal power and the corresponding labels for training.

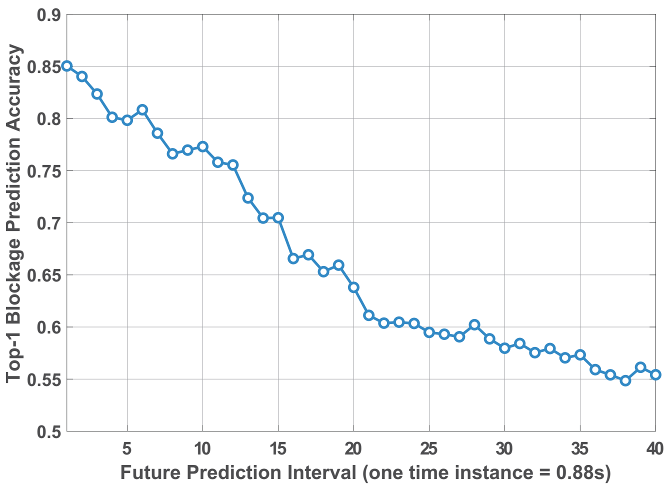


Fig. 6: Top-1 blockage prediction accuracy for different future prediction intervals.

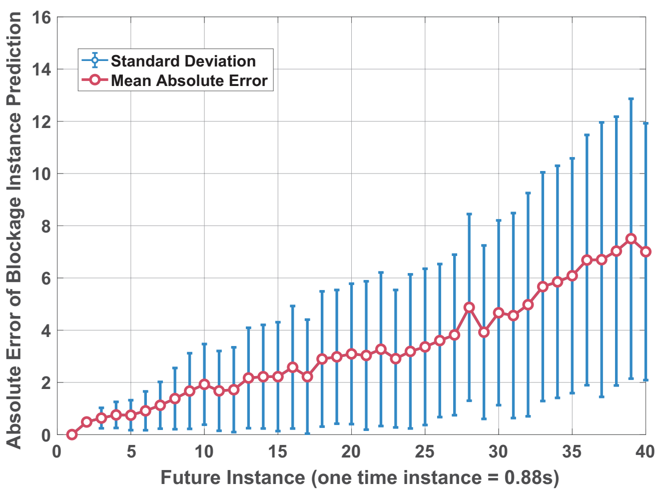


Fig. 7: Mean absolute error between the target (the exact time instance when the blockage occurs) and the prediction of this blockage time instance for different sized future prediction intervals.

C. Performance Evaluation

Problem 1: Fig. 6 plots the Top-1 accuracy as a function of the prediction interval. We see that as the prediction interval increases the accuracy decreases; the Top-1 accuracy decreases sharply at first, and then flattens out. Our model achieves high accuracy when predicting the occurrence of blockage in the near future. i.e. we can achieve above 80% accuracy when predicting the occurrence of blockage in the future 6 time instances (around 5s in the future). This is because when the prediction interval is small, our training dataset contains a large number of sequences with a clear pre-blockage signature, resulting in a high ratio of these sequences in the training dataset. The prediction accuracy is good until the prediction interval goes beyond 15 time instances. It finally converges to the "random guess" or 50% accuracy as the prediction interval approaches 40 time instances. This is because pre-blockage signature is less effective when the blockage is far away from the transmitter and receiver.

Problem 2: Fig. 7 plots the mean absolute error between prediction and ground-truth as a function of prediction interval. For each prediction instance, we show the standard deviation as an error bar. For prediction interval of 15 time instances (T_p = 15), our model can predict all the blockage transitions with an average error of 2 time instances. This prediction also happens with relatively low volatility (± 1.5). As the blockage happens further in the future, the blockage instance prediction error increases. Nevertheless, even when the prediction interval is as large as 40 time instances, we can still predict the exact time of blockage occurrence with the mean absolute error of under 8 time instances.

VII. Conclusion

In this paper, we explored the potential of utilizing mmWave received power data to proactively predict dynamic blockages in mmWave systems, thereby allowing the network to proactively manage hand-off/beam-switching decisions. We formulated the wireless signature blockage prediction problem and developed an efficient machine learning model for this task based on our RNN architecture. Simulation results on real data showed that our model can achieve good performance (85%) for moving blockage prediction for short prediction interval (13.2s). This allows the user to be proactively handed over to another base station without disconnecting the session with a high success probability.

References

[1] R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave mimo systems," *IEEE journal of selected topics in signal processing*, vol. 10, no. 3, pp. 436–453, 2016.

[2] J.-S. Choi, W.-H. Lee, J.-H. Lee, J.-H. Lee, and S.-C. Kim, "Deep learning based nlos identification with commodity wlan devices," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3295–3303, 2017.

[3] C. Huang, A. F. Molisch, R. He, R. Wang, P. Tang, B. Ai, and Z. Zhong, "Machine learning-enabled los/nlos identification for mimo system in dynamic environment," *IEEE Transactions on Wireless Communications*, 2020.

[4] A. Alkhateeb, I. Beltagy, and S. Alex, "Machine learning for reliable mmwave systems: Blockage prediction and proactive handoff," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2018, pp. 1055–1059.

[5] M. Alrabeiah and A. Alkhateeb, "Deep learning for mmWave beam and blockage prediction using sub-6 GHz channels," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5504–5518, 2020.

[6] G. Charan, M. Alrabeiah, and A. Alkhateeb, "Vision-aided 6G wireless communications: Blockage prediction and proactive handoff," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 10 193–10 208, 2021.

[7] M. Alrabeiah, A. Hredzak, and A. Alkhateeb, "Millimeter wave base stations with cameras: Vision aided beam and blockage prediction," *arXiv preprint arXiv:1911.06255*, 2019.

[8] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[9] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[10] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[11] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," 2016, book in preparation for MIT Press. [Online]. Available: http://www.deeplearningbook.org

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.