# Gated Recurrent Units for Blockage Mitigation in mmWave Wireless

Ahmed Almutairi
Portland State University
Portland, OR, USA
almut8@pdx.edu

Alireza Keshavarz-Haddad
Shiraz University
Shiraz, Iran
alireza.keshavarzhaddad@gmail.com

Ehsan Aryafar
Portland State University
Portland, OR, USA
earyafar@pdx.edu

## ABSTRACT

Millimeter-Wave (mmWave) communication is susceptible to blockages, which can significantly reduce the signal strength at the receiver. Mitigating the negative impacts of blockages is a key requirement to ensure reliable and high throughput mmWave communication links. Previous research on blockage mitigation has introduced several model and protocol based blockage mitigation solutions that focus on one technique at a time, such as handoff to a different base station or beam adaptation to the same base station. In this paper, we address the overarching problem: *what blockage mitigation method should be employed?* and *what is the optimal sub-selection within that method?* To address the problem, we developed a Gated Recurrent Unit (GRU) model that is trained using periodically exchanged messages in mmWave systems. We gathered extensive amount of simulation data from a commercially available mmWave simulator, show that the proposed method does not incur any additional communication overhead, and that it achieves outstanding results in selecting the optimal blockage mitigation method with an accuracy higher than 93%. We also show that the proposed method significantly increases the amount of transferred data compared to several other blockage mitigation policies.

## CCS CONCEPTS

• **Networks** → **Network performance evaluation**; • **Computing methodologies** → **Machine learning**.

## KEYWORDS

MmWave Wireless, Blockage Mitigation, Recurrent Neural Networks, Deep Learning

**ACM Reference Format:**
Ahmed Almutairi, Alireza Keshavarz-Haddad, and Ehsan Aryafar. 2023. Gated Recurrent Units for Blockage Mitigation in mmWave Wireless . In *Proceedings of the Int'l ACM Symposium on Mobility Management and Wireless Access (MobiWac '23), October 30-November 3, 2023, Montreal, QC, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3616390.3618280

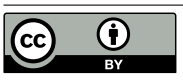
## 1 INTRODUCTION

MmWave communication is a major component of several existing wireless standards such as 5G (cellular) and 802.11 ad/ay (WiFi). It is a key technology to provide very high data rates in a variety of applications such as Industrial Internet of Things (IIoT) [1–4]. However, mmWave systems are susceptible to high path loss, high noise power, and blockages. To address high path loss and noise power challenges, mmWave systems employ beamforming techniques to form narrow directional beams (Fig. 1(a)) at the transmitter (Tx) and/or the receiver (Rx). This significantly increases the signal strength at the receiver but introduces additional challenges such as beam selection. Existing mmWave standards utilize a beam search process that occurs periodically at the beginning of each communication interval (e.g., every 100 msec in mmWave WiFi) to handle beam selection/search.

The other challenge associated with mmWave communication is susceptibility to blockages, e.g., human body alone can block the signal and significantly reduce its strength at the receiver [5–7]. Existing mmWave standards respond to blockages in a reactive manner, and it can take them several communication intervals until the selected beams are switched or the client is handed off to another base station (BS). However, this can significantly reduce the throughput. To address the issue, the research community has introduced several methods in isolation to better handle blockages, e.g., use model-driven methods to pro-actively switch the beams to the same BS before blockages happen [8] or widen the beams so that the signal passes through the blocker [9].

Our goal in this paper is to address the overarching problem: *from the plurality of blockage mitigation techniques, which one should be employed?* Specifically, we focus on three techniques: beam switching on the same BS, handoff, and beam widening. We also address the associated sub-problem within each technique, e.g., what new beam should be selected, which BS should the client handoff to, and how much to widen the beam. To address the problem, we develop a framework that proactively takes the appropriate action in order to minimize the impact of blockages. At its core, our framework uses Gated Recurrent Units (GRUs), a newer generation of Recurrent Neural Networks (RNNs) suitable for learning sequential data, and relies on periodic existing message passing in mmWave standards to decide on the appropriate action. Our key contributions can be summarized as follows:

- **Data Gathering:** We utilized Wireless InSite (WI) simulator to conduct numerous experiments and model different types of blockages in an IIoT setting. We have publicly released all of our data and software code [10] so that other researchers in the community can build on our work.
- **Blockage Mitigation Framework:** We develop a GRU-based framework to mitigate blockages. We show that GRUs have a significantly higher accuracy in selecting the optimal action when compared to Categorical Naive Bayes and Support Vector Machines. We also show that the solution only needs a few time series samples, which slightly increases



This work is licensed under a Creative Commons Attribution International 4.0 License.

*MobiWac '23, October 30-November 3, 2023, Montreal, QC, Canada*
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0367-6/23/10.
https://doi.org/10.1145/3616390.3618280

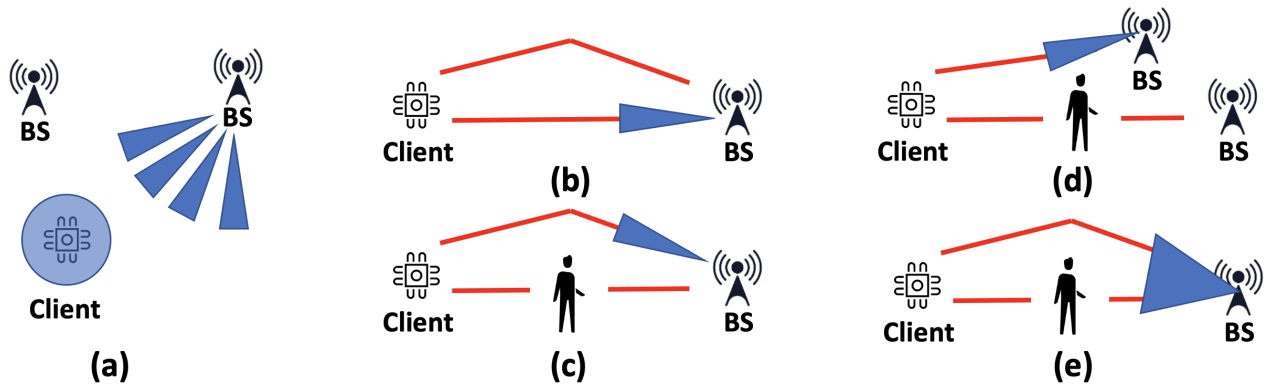



**Figure 1: (a): There are several beams to choose from at the BS. Beam selection typically happens through a search process at the beginning of every communication interval; (b): BS has identified a proper beam for communication with the client. Here the two red paths capture the multi-path nature of the communication channel; (c): In beam switching, BS switches to a different beam when blockage happens; (d): In handoff, the network may change the BS serving the client; (e): In beam widening, BS widens its beam. Energy reaches the client through other paths.**

its accuracy when compared to Long Short Term Memory (LSTM) RNNs in addition to using less memory and being faster.

- **Policies:** We model alternative forms of blockage mitigation techniques as policies, formally prove our framework provides higher throughput than them, and show through simulations that it substantially increases the average amount of transferred data across all types of blockages.

The rest of this paper is organized as follows. We discuss the related work in Section 2. Section 3 describes the system model and our GRU-based blockage mitigation method. We present the policy-based definition of our approach along with alternative policies in Section 4. Section 5 presents our data gathering process and performance evaluation results. Finally, we conclude the paper in Section 6.

## 2 RELATED WORK

There is a rich body of work to model and mitigate blockages in mmWave systems. Here, we discuss three of the most prominent techniques to mitigate blockages.

**Beam Switching.** Beam switching/adaptation is one of the key techniques to mitigate blockages [11]. It refers to switching the beam to another direction that is not blocked (Fig. 1(c)). This technique has been studied in several prior works optimized through: (i) leveraging an out-of-band radio [12], (ii) sensing the reflective environment [13], (iii) using model-driven methods [8], and (iv) employing deep learning based on a given client's location and environment [14, 15]. Beam switching can be a reliable blockage mitigation technique in some scenarios and environments. On the other hand, it may fail when the blocker is large (e.g., a truck) or when there are few reflectors in the environment.

**Handoff.** Handoff (BS adaptation, Fig. 1(d)) is another widely used technique to mitigate blockages. It involves handing off the client to another BS when blockage is detected. Several research works (e.g., [16–21]) have optimized the handoff decision making process by using Reconfigurable Intelligent Surfaces (RIS) or employing deep learning models based on channel state information, the client location, and other network parameters. Handoff is an efficient blockage mitigation technique in many scenarios. However, there are many instances in which handoff may fail. For example, when the time to complete handoff is more than the duration of the blockage event, then handoff would be unnecessary. The problem can worsen in higher mobility environments, where the handoff frequency may become too high.

**Beam Widening.** Beam widening refers to increasing the half-power beamwidth of the currently selected beam (Fig. 1(e)). Several works have shown the benefits of beam widening to mitigate blockages [9]. However, beam widening reduces the beamforming gain. Further, beam widening may still fail when blocker is large or close to the client or BS.

While previous works have attempted to tackle blockages, they have generally optimized only one technique at a time. Although a technique may work well in certain scenarios, it may fall short in others, suggesting the need to address the problem holistically. Our goal in this paper is to integrate the above techniques into one system. The resulting framework handles various scenarios and blockage types effectively, providing a more robust approach to mitigating blockages.

## 3 GRUS FOR MMWAVE BLOCKAGE MITIGATION

In this section, we present our GRU-based blockage mitigation framework. We first briefly discuss GRUs and their distinction from LSTM RNNs. Then, we introduce our system model and how we use GRU models to tackle blockages.

### 3.1 Gated Recurrent Units (GRUs)

GRU [22] is a type of RNN architecture that is commonly used for modeling sequential data. It was introduced as an alternative to the more popular LSTM architecture.

Like LSTM, GRU is designed to address the vanishing gradient problem that can occur in traditional RNNs. The vanishing gradient problem refers to the issue where gradients can become extremely small as they propagate through the network, which can make it difficult for the machine learning (ML) model to learn long-term dependencies.

GRU accomplishes this by using gating mechanisms to selectively update and reset the hidden state of the network. Specifically, GRU has two gates: a reset gate and an update gate. The reset gate helps the network decide how much of the past information should be forgotten, while the update gate helps the network decide how much of the current information should be used to update the hidden state.

One of the advantages of GRU over LSTM is that it has fewer parameters, which makes it faster to train and more efficient to store. Additionally, GRU has been shown to perform comparably to LSTM on a wide range of tasks, including language modeling and speech recognition.

### 3.2 System Model

We consider an IIoT setup [1, 2] with fixed BSs and clients, and mobile blockers. Each BS uses a phased array antenna of size $M$ and has access to a set of $B$ directional beams to cover a horizontal range of $x$ degrees. Client devices are much simpler IoT devices and each client device uses a single omni (or quasi-omni) beam for both transmission and reception. The BSs and clients operate at a mmWave band and blockage mitigation is done on the BS or network side.

Our work builds on prior work [23], which leverages an innovative deep learning technique to proactively determine if a blockage is likely to occur in the future time interval. Their proposed solution shows high accuracy in the near future time interval and only relies on undergoing communication between the BS and client without incurring additional communication overhead. This is achieved by leveraging a sequence of in-band wireless data measurements and jointly employing recurrent and convolutional neural networks. The work in [23] makes the assumption that the current connection is line-of-sight (LoS), which can be accurately predicted in mmWave systems. For example, the work in [24] can accurately classify LoS and nLoS channel conditions at the beginning of each communication interval by only using information from the messages that are exchanged between BS and client during the beam search interval (thus incurring no additional overhead).

The proposed framework in this paper builds upon [23] by using their model to predict if a blockage is going to happen, and then decides on the best course of action to eliminate the negative impact of the oncoming blockage.

### 3.3 Blockage Mitigation Framework

Our objective is to mitigate blockages by proactively minimizing their impact, which can lead to increased throughput and reduced latency of communication. Blockages can exhibit similar shapes, velocities, and patterns of trajectories. To address this issue, we propose a model that learns similar features from a sequence of reported signal-to-noise ratio (SNR) values to determine the best action to take. The action space consists of three main actions: beam switching, beam widening, and handoff. Each main action includes multiple sub-actions (e.g., beam switching includes the selection of new beam from all available beams at the BS, and handoff could be to any of the surrounding BSs). Therefore, the total number of actions include the sum of the total number of beams at a BS[1], total levels of beam widening, and all surrounding BSs that a client can be handed off to. Let $W$ be the total number beam widening levels and $H$ be the number of surrounding BSs. Therefore, the total number of actions will be equal to $B + W + H$.

**GRU Model.** We employed a GRU model consisting of four layers of GRU cells. The first two layers consist of 128 units each, while the latter two layers have 64 units each. The four GRU layers are followed by a dense output layer with the size equal to the number of actions. The input to the model is a sequence of length $N$ time steps each of which consists of SNR values of the current BS's beams along with the SNR value of the best beam of each surrounding BS and each BS's ID. Formally, let $S$ be a sequence of time steps, and $s \in S$ a time step where $s = \{\{b_1, b_2, \ldots, b_B\}, \{BS_1, BS_{SNR_1}, \ldots, BS_H, BS_{SNR_H}\}\}$. Here, $b_1$ shows the SNR of beam one of the current BS, and $BS_1$, $BS_{SNR_1}$ shows the ID of BS one and the SNR value of the best beam (measured at the client) of BS one. Therefore, the time steps sequence will be $S = \{s_1, s_2, s_3, \ldots, s_N\}$. We refer to the time step sequence $S$ as a sample. All SNR values of current and surrounding BSs (stemming from different beams at each BS), of each time step, are measured at the client side and reported to the serving BS during the periodic measurement report (MR) interval. We feed the GRU with these MRs. We optimized the training of our neural network by using the Adam optimizer with a learning rate of 0.001, cross-entropy loss function, a batch size of 32, a dropout rate of 0.2, and $L2$ regularization with a coefficient of 0.01 to prevent overfitting. Table 1 summarizes our GRU model structure and configuration.

**Table 1: GRU Model Structure and Configurations.**

| Parameter | Value |
|---|---|
| GRU layer1 | 128 Units |
| Dropout | 0.2 |
| GRU layer2 | 128 Units |
| Dropout | 0.2 |
| GRU layer3 | 64 Units |
| Dropout | 0.2 |
| GRU layer4 | 64 Units |
| Dropout | 0.2 |
| Output dense | 46 classes |
| GRU activation | tanh |
| Output activation | Softmax |

**Action Selection Metric.** The dataset samples that are fed to the GRU model must be labeled with the best action, which is not trivial to define. For example, selecting the best action based on

[1]Note that we assume each BS has access to $B$ beams.

the highest reported SNR value can be inefficient. For example, let the handoff action take 1 second to complete and let the human blockage last for 0.4 seconds. Therefore, if the handoff action is selected because of the the highest expected SNR, there will be 0.6 seconds during which no data is transferred. Thus, we need to choose a metric that not only takes into account the SNR but also both the duration of the blockage event as well as the duration needed to perform the blockage mitigation action. To do this, we first define the average throughput of a client associated with a BS with a given SNR value as:

$$R = \frac{\omega}{U} \times \log_2(1 + SNR) \tag{1}$$

here, $\omega$ is the communication bandwidth and $U$ is the total number of clients associated with the BS.

Next, we choose the amount of transferred data ($D$) as the metric based on which we select the best blockage mitigation technique. This metric combines the average throughput metric ($R$) defined in Eq. 1 and both the duration of the blockage event ($T$) and the duration of a given blockage mitigation technique (C) according to the following formula:

$$D = \max\{(T - C), 0\} \times R \tag{2}$$

here, $T$ and $C$ are in seconds. The duration of blockage mitigation technique ($C$) can also be considered as the cost associated with taking that action. We can also estimate $T$ through $T = \frac{L}{V}$, where $L$ is the length of the blocker and $V$ is the blocker velocity. In our simulations, we let the cost associated with handoff, beam switching, and beam widening be equal to 1, 0.01, and 0.015 seconds, respectively.

## 4 BLOCKAGE MITIGATION TECHNIQUES AS POLICIES

In the previous section, we chose and defined the amount of transferred data as the metric to optimize when selecting the best blockage mitigation technique. In this section, we give a formal definition of our action selection mechanism as a policy. We also define alternative policies that model other blockage mitigation mechanisms from the literature.

Assume that there are $K$ types of blockages. Denote the probability of occurrence of blockage type $i$ by $P_i$. Further, assume there are $M$ blockage mitigation techniques, and the cost of each technique $j$ is denoted by $C_j$. Let $T_i$ be the the duration of blockage type $i$, and let $R_{i,j}$ be the rate of a blocked user when the blockage is type $i$ and the used mitigation technique is $j$. Then, we define the amount of transferred data, when the blockage type, selected blockage mitigation technique, and rate are known as:

$$D_{i,j} = \max\{(T_i - C_j), 0\} \times R_{i,j} \tag{3}$$

**Policy 1.** Choose the best mitigation technique when the blockage type and the value of $R_{i,j}$ are given, i.e., use the mitigation technique $j$ with the maximum amount of transferred data as estimated by Eq. 3. This policy is the one that we implemented in our approach. The expected amount of transferred data when policy 1 is employed is:

$$\mathbb{E}[D_{i,j}\ policy\ 1] = \sum_{i=1}^{K} P_i \times \bar{D}_{i,max} \tag{4}$$

where $D_{i,max} = \max(D_{i,1}, D_{i,2}, \ldots, D_{i,M})$ and $\bar{D}_{i,max} = \mathbb{E}[D_{i,max}]$. The expectation in $\mathbb{E}[D_{i,max}]$ is to account for user specific $R_{i,j}$, which in addition to blockage type $i$ and mitigation technique $j$, depends on the client channel.

**Policy 2.** Choose a fixed best mitigation technique for each type of blockage. For example, if the mitigation technique $j$ provides on average the maximum amount of transferred data for blockage type $i$, use this mitigation technique for the specific blockage type $i$. We use the notation $j = \delta(i)$ to distinguish this mitigation technique $j$. The expected amount of transferred data when policy 2 is employed is:

$$\mathbb{E}[D_{i,j}\ policy\ 2] = \sum_{i=1}^{K} P_i \times \bar{D}_{i,\delta(i)} \tag{5}$$

where

$$\begin{aligned}\bar{D}_{i,\delta(i)} = \mathbb{E}[D_{i,\delta(i)}] &= \max(\mathbb{E}[D_{i,1}], \mathbb{E}[D_{i,2}], \ldots, \mathbb{E}[D_{i,M}]) \\ &= \max(\bar{D}_{i,1}, \bar{D}_{i,2}, \ldots, \bar{D}_{i,M})\end{aligned}$$

Note, blockage type is assumed to be known.

**Policy 3.** Unlike policy 2, this policy chooses a single fixed technique for all type of blockages. The chosen technique, which we denote as $j\star$, is the technique that provides on average the maximum amount of transferred data across all types of blockages. We can compute the expected amount of transferred data when policy 3 is employed as follows:

$$\mathbb{E}[D_{i,j}\ policy\ 3] = \sum_{i=1}^{K} P_i \times \bar{D}_{i,j\star} \tag{6}$$

where $\sum_{i=1}^{K} P_i \times \bar{D}_{i,j\star} = \max(\sum_{i=1}^{K} P_i \times \bar{D}_{i,1}, \sum_{i=1}^{K} P_i \times \bar{D}_{i,2}, \ldots, \sum_{i=1}^{K} P_i \times \bar{D}_{i,M})$ and $\bar{D}_{i,j} = \mathbb{E}[D_{i,j}]$.

**Policy 4.** Choose an arbitrary mitigation technique for all type of blockages, i.e., choose and always use a fixed randomly selected technique $j$ for all of the blockage types. The expected amount of transferred data when policy 4 is employed is:

$$\mathbb{E}[D_{i,j}\ policy\ 4] = \sum_{i=1}^{K} P_i \times \bar{D}_{i,j} \tag{7}$$

We have the following proposition on the theoretical performance of these four policies. In the following section, we quantify these theoretical results through simulations.

**Proposition.** The order of performance (in terms of the average amount of transferred data) among the four aforementioned policies is as follows:

$$\mathbb{E}[D_{i,j}\ policy\ 1] \geq \mathbb{E}[D_{i,j}\ policy\ 2] \geq \mathbb{E}[D_{i,j}\ policy\ 3] \geq \mathbb{E}[D_{i,j}\ policy\ 4]$$

with policy 1 being the best policy in mitigating the blockages and providing the maximum average amount of transferred data. The degree of difference between these policies measured in terms

of the average amount of transferred data depends on the distribution of $R_{i,j}$s for various cases of $i$ and $j$, the value of $T_i$, and the value of $C_j$.

**Proof.** From the definition, $D_{i,max} \geq D_{i,\delta(i)} \Rightarrow \bar{D}_{i,max} = \mathbb{E}[D_{i,max}] \geq \mathbb{E}[D_{i,\delta(i)}] = \bar{D}_{i,\delta(i)}$.

$\mathbb{E}[D_{i,j}\ policy\ 1] = \sum_{i=1}^{K} P_i \times \bar{D}_{i,max} \geq \sum_{i=1}^{K} P_i \times \bar{D}_{i,\delta(i)} = \mathbb{E}[D_{i,j}\ policy\ 2]$.

Next, based on the definition of $\delta(i)$, $\bar{D}_{i,\delta(i)} \geq \bar{D}_{i,j\star}$, hence

$\mathbb{E}[D_{i,j}\ policy\ 2] = \sum_{i=1}^{K} P_i \times \bar{D}_{i,\delta(i)} \geq \sum_{i=1}^{K} P_i \times \bar{D}_{i,j\star} = \mathbb{E}[D_{i,j}\ policy\ 3]$

Finally, from the definition of $j\star$,

$\mathbb{E}[D_{i,j}\ policy\ 3] = \sum_{i=1}^{K} P_i \times \bar{D}_{i,j\star} \geq \sum_{i=1}^{K} P_i \times \bar{D}_{i,j} = \mathbb{E}[D_{i,j}\ policy\ 4]$.

## 5 PERFORMANCE EVALUATION

In this section, we discuss our performance evaluation results. First, we discuss our simulated environment and our methodology to gather and label data. Next, we discuss the performance of our ML-based blockage mitigation technique considering both ML metrics (e.g., accuracy, F1 score) and networking metrics (e.g., transferred data, throughput).

### 5.1 Measurement Campaign

**Simulator.** We used a commercially available wireless simulator named Wireless InSite (WI) to simulate an IIoT use case scenario with a size of $350 \times 150\ m^2$. IIoT is a cutting-edge technology that connects Internet-connected devices, sensors, and machines with industrial processes and systems. It is considered as one of the critical technologies for the development of Industry 4.0 [25, 26]. MmWave wireless technologies play a significant role in IIoT by providing high data rates and assisting with positioning [1, 2, 27, 28], among others.

**Simulation Setup.** Our simulation scenario consists of six BSs and one hundred clients. BSs are distributed along both sides of the environment with three BSs at each side with 75 $m$ distance between them (see Fig. 2). Each BS has access to a uniform linear array (ULA) that consists of 16 antennas, which provide 36 beams and cover a horizontal range of 120°. Clients are randomly distributed within the environment. Each client has access to a single beam with an omni-directional radiation pattern. All BSs and clients operate at 28 GHz band and are on a LoS channel condition prior to blockage. BS height is 2.5 $m$, while the client height is 1.5 $m$. Table 2 summarizes the simulation parameters.

We included four different types of blockers: human, cart, truck, and pickup. These particular blockage types are the most commonly encountered blockers in a factory setting. We modeled a human as a cylinder with radius of 30 $cm$ that can move at 1.4 $m/s$ speed. The size of carts, pickups, and trucks are $2.70\times1.21\times1.8\ m^3$, $5.4\times2.1\times1.9\ m^3$, and $12.3 \times 2.7 \times 2.3\ m^3$, respectively. We assumed the velocity of each of these three blockers can range from one to five miles per hour (mph), which equals to 0.89 to 2.2 $m/s$.

**Data Gathering.** As discussed in Section 3.3, our GRU-based model takes a sequence of data as input to decide on what action to take as an output. This data consists of multiple time steps each of which contains SNR values for all the beams of the current BS along with best SNR (stemming from the best beam) of each of the surrounding BSs along with BS IDs. We run our simulation based on the above setup and record the SNR measurements at the clients side every 75 $ms$. In other words, the difference in time between two consecutive time steps is 75 $ms$. The run continues as the blockers move until the current connection is completely blocked. At the end of each run, we extract the measured SNR values of the current BS and surrounding BSs' best SNR values and their IDs for every time step, including the time step in which the connection is blocked. The last time step (when the connection is blocked) is repeated with varying number of antennas at the current BS to get information about how beam widening performs when the connection is blocked[2]. This information is used when we label the dataset samples. Through these simulation, we gathered a dataset comprising 20,000 samples, each consisting of nine time steps. Within each time step, there are 36 SNR values that represent the current BS's 36 beams along with the best SNR values for the five surrounding BSs and the BS IDs. The dataset has been evenly distributed across the four different types of blockages, with 5,000

**Table 2: Simulation Setup.**

| Parameter | Value |
|---|---|
| BS antenna array size | 16 (ULA) |
| Client radiation pattern | Omni |
| Number of beams at the BS | 36 |
| Frequency | 28 GHz |
| Bandwidth | 1 GHz |

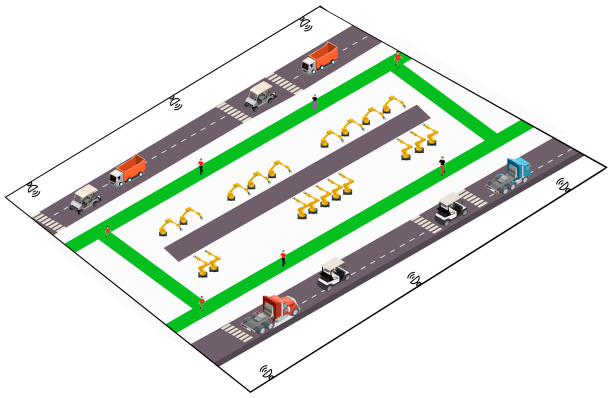


**Figure 2: IIoT scenario with six BSs on the two sides, clients, and different types of blockers. We included four different types of blockers: human, cart, truck, and pickup, which are the most commonly encountered blockers in an industrial IoT (e.g., factory) setting.**

[2]In a uniform linear array (ULA) with $M$ antennas and $\frac{\lambda}{2}$ spacing distance ($\lambda$ is the carrier wavelength), the main lobe beamforming gain is equal to $10\times\log_{10}^{M}$ (in dB) with $\frac{102}{M}$ (in degrees) half power beamwidth. Thus, we varied the number of active antennas from 16 to 14, 12, 10, and 8 to provide five different levels of beam widening.

samples collected for each blockage type. We use 70% of dataset for training and 30% for test.

**Data Labeling.** Labeling refers to determining the best action that could have mitigated each blockage occurrence. In order to label our dataset samples, we began by determining the number of classes, or actions, which totaled 46. These 46 actions are distributed as follows: 36 classes (ranging from 0 to 35) were assigned to the current BS's 36 beams to accommodate the beam switching action, 5 classes (ranging from 36 to 40) were assigned to the number of surrounding BSs to which a client can be handed off to, and 5 classes (ranging from 41 to 45) were assigned to beam widening. As detailed in Section 3.3, we labeled each sample in our dataset based on the optimal action that would result in the highest amount of transferred data during a blockage event. To do this, we utilized Eq. 3 to label each sample appropriately.

To gain a deeper insight into our labeled dataset, we conducted two distinct analyses. First, we calculated the percentage of samples labeled with each action across the entire dataset, regardless of the type of blockage. Fig. 3(a) shows the corresponding result. We observe that handoff was the best action for 43% of our samples, beam widening was the best action for 30% of our samples, and beam switching was the best action for 27% of our samples.

Next, we conducted an analysis to examine the relationship between the best action and the type of blockage. Fig. 3(b) demonstrates the corresponding result. Our results indicate that beam switching and beam widening are commonly best actions for smaller blockers (e.g., human or cart), whereas handoff is a better action for larger blockers (e.g., pickup or truck). Additionally, we discovered that beam switching and beam widening could sometimes better handle large blockages compared to handoff. However, handoff was never a good solution to handle small blockers due to its cost exceeding the blockage event time.

## 5.2 Results

We next discuss the performance of our blockage mitigation framework considering both ML and networking metrics.

**GRU Evaluation.** Accuracy is a popular metric to evaluate the performance of a machine learning model. It is defined as the ratio of correct predictions made by the model to the total number of predictions made. Since our model takes a sequence of data as input to make a decision, we need to determine how many sequences (number of time steps) the model needs to give the optimal accuracy result. To achieve this, we evaluate our model over the last 3, 5, 7, and 9 time steps. We observed that the model gives similar accuracy results for all number of time of steps. Hence, to reduce the computational complexity and increase memory efficiency, we consider the least number of time steps, which is 3.

Fig. 3(c) shows the accuracy of our GRU model across all samples, which is 93% and demonstrates that the model has learned to make accurate predictions.

Next, to obtain a more complete picture of the model's accuracy, we consider Top-K accuracy. Top-K accuracy is a more informative metric that measures the proportion of instances in which the correct label is among the top K predicted labels. For instance, Top-1 accuracy measures the proportion of instances in which the correct label is the top prediction made by the model. In our case, the Top-1 accuracy of our GRU model is 93%. This result is a testament to the effectiveness of our model, as it is making accurate predictions in majority of cases.

Moreover, our model has achieved a Top-2 accuracy of 98% (Fig. 3(c)), meaning that the correct label is among the top two predictions for 98% of instances. Furthermore, our model has achieved a Top-3 accuracy of 99%, which indicates that the correct label is among the top three predictions for 99% of instances. Note that to be counted as a correct action, details of the action must be correct too. For example, when beam switch to a particular beam is the correct action, the model not only has to select beam switching as the appropriate blockage mitigation technique but it must also correctly select the beam to switch to (out of 36 available beams) to match the label.

**Table 3: Recall and Precision of Beam Switching, Beam Widening, and Handoff.**

| | Recall | Precision |
|---|---|---|
| Beam Switching | 86.6% | 86.7% |
| Beam Widening | 96.3% | 96.9% |
| Handoff | 92.6% | 92.1% |

In order to gain a better insight into the performance of our model, we analyzed each action using recall and precision metrics. For every action, we calculate both recall and precision scores. The beam switching action yielded recall and precision scores of 86.6% and 86.7%, respectively. Beam widening demonstrated a recall score of 92.6% and a precision score of 92.1%. However, the highest recall and precision scores among all actions were achieved by the handoff action, with recall and precision scores of 96.3% and 96.9%, respectively.

Despite the fact that beam switching has the lowest recall and precision scores compared to other actions, it is still performing well with over 85% accuracy. The lower scores are due to the action comprising a much larger number of sub-actions (36 actions corresponding to 36 beams), which increases the confusion between the sub-actions. Table 3 shows a summary of these results.

**Comparison with Other ML Models.** To demonstrate the effectiveness of GRUs, we compared it with four baseline techniques: random selection, Categorical Naive Bayes (CategoricalNB), support vector machine (SVM), and LSTM.

Random selection is a simple technique that randomly assigns class labels to the data, making it a useful baseline for determining the performance of a model by chance, while CategoricalNB is a popular ML algorithm that works well for text classification tasks and assumes that the features are categorical. SVM works by finding the best possible boundary that separates data points into different classes. In other words, it tries to find the hyperplane that maximizes the margin between the different classes in the dataset. We evaluated each technique on the same dataset and considered accuracy as the performance evaluation metric.

We used scikit-learn, the open-source ML library, to implement CategoricalBN and set the hyperparameters *alpha* and *fit_prior* to 1 and true, respectively. We used the same scikit-learn library to implement an SVM model with "radial basis function (rbf)" *kernel*

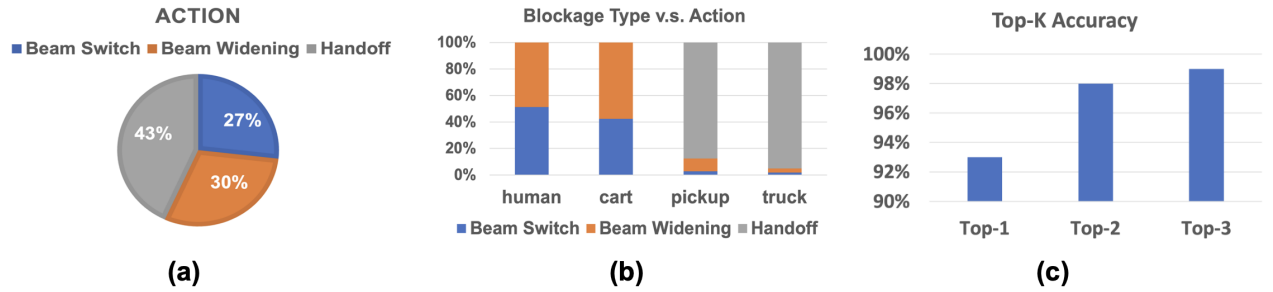



**Figure 3: (a): Fraction of samples in the dataset labeled with each action; (b): Fraction of samples labeled with each action for each blockage type; (c): Top-1, Top-2, and Top-3 accuracy results of our GRU-based blockage mitigation framework. The correct label is the predicted label in 93% of instances and is among the top three predictions for 99% of instances.**

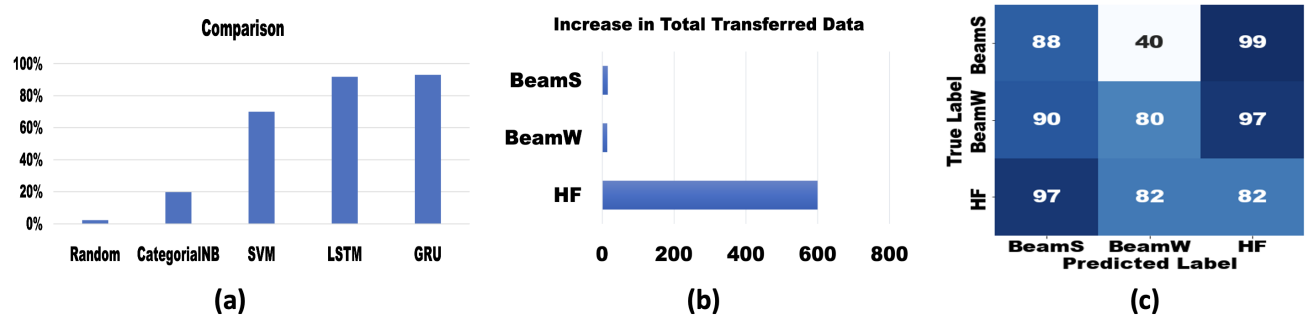



**Figure 4: (a): Accuracy of different ML models. GRU achieves the highest accuracy; (b): Increase in total transferred data for different actions. Here BeamS, BeamW, and HF denote beam switching, beam widening, and handoff, respectively; (c): The average drop in throughput when selected action is not optimal. Y-axis shows the true label and x-axis shows the incorrect predicted label. When the true and predicted label are the same, the selected sub-action is not optimal. The overall average drop in throughput across all actions is 92%.**

and parameter *gamma* set to "scale". We implemented the LSTM NN with the same structure and hyperparameters of GRU as described in Section 3.1.

Fig. 4(a) shows the accuracy results. The results show that the GRU model outperforms all baseline techniques, achieving an accuracy of 93%. In comparison, random selection achieves an accuracy of only 2.2%, CategoricalNB 19.7%, and SVM 70%. LSTM, which is very similar to GRU, gets a very close accuracy at 91.8%.

We chose the GRU model over LSTM for several reasons. First, GRU models require fewer parameters, which reduces the training time and improves computational efficiency. This is especially important when working with large datasets or when there are time constraints for model training. Additionally, we found that the GRU model has a slightly higher accuracy than the LSTM model, further supporting our decision to choose GRU.

**Network Evaluation.** We next evaluate our model performance in terms of increase in the amount of transferred data when an action is taken compared to the blocked connection. Note that we measure throughput according to Eq. 1, which depends on the client SNR. When blockage happens, SNR drops, which reduces the throughput (but is non-zero due to Eq. 1). Fig. 4(b) shows the corresponding results where beam switching, beam widening and handoff are denoted by BeamS, BeamW, and HF, respectively. Beam switching increases the total amount of transferred data by x16, while beam widening increases that by x14. Handoff increases the total amount of transferred data by about x600, much higher than beam widening and beam switching. Note, that the results do not mean that beam switching and beam widening are not effective actions. Handoff is commonly used when blocker is large (e.g., with a pickup or truck), which has the most negative impact on the underlying connection. Therefore, the gap between the new throughput (as a result of handoff) and baseline throughput (throughput of the connection blocked by a large blocker) is very large. Additionally, with handoff, the new BS will likely observe no negative impact from the current blocker. Beam switching and beam widening are still efficient actions when the blocker is small (e.g., human or cart as we showed in Fig. 3(b)). They also result in more increase in throughout than handoff when blocker is small, due to the shorter duration of the blockage event and high cost of handoff.

**Drop in Throughput when Selected Action is not Optimal.** Our results in Fig. 3(a) showed that the GRU model selects the optimal action in 93% of the times. We next study the drop in performance (in terms of throughput) when the selected action is not optimal, which has a 7% probability. To do so, we calculate the

client throughput during the blockage event had we selected the optimal action as well as the action selected by the GRU model. We then calculate the percentage drop in performance (from the previous two calculations) and find its average value across all clients and blockage events and found that to be equal to 92%. This shows that there is a large average penalty when the optimal action is not taken, or in other words optimal action can result in a very high increase in throughput. Note that while this analysis is specific to our GRU model, other methods also have a similar performance, as we will show through average throughput results later in this section.

To gain more insights into this average number, we extend our examination to compare the percentage drop in performance for each specific action. Fig. 3(c) shows the corresponding results. Here the y-axis shows the true label and the x-axis shows the incorrect predicted label. For example, when the optimal action is identified as BeamS (top row in Fig. 3(c)), we derive the average percentage drop in throughput that would be achieved if the selected action is BeamS to a non-optimal beam ID, BeamW, or HF.

By examining the results across all combinations, we observe that when the true label is BeamS, handoff (HF) would result in the highest (99%) drop in throughput. Further, if BeamS is the true label, BeamW would result in the lowest (40%) average drop in throughput. This type of a cost analysis could also be used by a network operator to devise more sophisticated blockage mitigation policies by associating different costs to different actions.

**Average Throughput Across Different ML Models.** Throughput is a critical metric for evaluating overall network performance. To assess the effectiveness of our model, we conducted an evaluation based on the average throughput across all ML models (GRU, LSTM, SVM, CategoricalNB, and Random). We measure throughput for the duration of the blockage event taking into account blockage type and velocity, and action delay, among others.

**Table 4: Average throughput across all clients, BSs, and blockage events achieved by different ML models.**

| | Random | CatNB | SVM | LSTM | GRU |
|---|---|---|---|---|---|
| Throughput (Mbps) | 0.52 | 4.62 | 16.32 | 21.8 | 22.7 |

Table 4 summarizes the average throughput results for each model. Note that these throughput results are averaged across all clients, BSs, and blockage events. Our evaluation shows that the GRU model outperforms all the baselines with an average throughput of 22.7 Mbps. The ratio of increase in throughput compared to different schemes is 43.7 with respect to (w.r.t.) Random, 4.91 w.r.t. CatNB, 1.39 w.r.t. SVM, and 1.04 w.r.t. LSTM.

**Comparison with Other Policies.** The GRU-based blockage mitigation framework exhibits a high level of performance, as observed in terms of both ML-based and networking metrics. We next compare the performance of our approach with three alternative policies (policies 2, 3, and 4 from Section 3.3) using the average amount of transferred data as metric. For this purpose, we utilized the same dataset that was used for both training and testing the GRU model. Subsequently, we computed the average amount of transferred data for each of the policies. Table 5 summarizes the decline in performance when implementing policies 2, 3, and 4 in comparison to our approach, which employs policy 1.

**Table 5: Comparison of GRU based model against three alternative policies (policies 2, 3, 4) in terms of average percentage drop in the amount of transferred data.**

| | Human | Cart | Pickup | Truck |
|---|---|---|---|---|
| Policy 2 | -25% | -5% | -29% | -16% |
| Policy 3 | -27% | -5% | -99% | -99% |
| Policy 4 | -100% | -100% | -29% | -16% |

In Policy 2, a fixed best technique is employed to maximize the average amount of data for each blockage type. Specifically, we found that the optimal technique for blockage type "human" is beam switching, while the best technique for "cart" is beam widening. For "pickup" and "truck," the ideal approach is handoff. To determine the average amount of data transferred for each blockage type, we applied the corresponding best technique and compared the results with our own approach.

The first row in Table 5 depicts the results. Our results show that when employing only beam switching for human blockage, the average amount of transferred data decreases by 25% compared to our approach. Similarly, for cart, the use of beam widening only resulted in a 5% decrease in the average amount of transferred data. For pickup, using handoff only, resulted in a 29% decrease in the average amount of transferred data. Finally, for a truck blocker, the average decrease is 16%.

In Policy 3, a fixed technique is employed to maximize the average amount of transferred data across all blockage types. To determine the optimal technique, we calculated the average amount of transferred data for the three techniques for each type of blockage and compared the results to identify the technique that provided the maximum performance across all blockage types. Our analysis revealed that beam widening was the optimal technique that provided the maximum average amount of transferred data across all blockage types.

To further evaluate the effectiveness of Policy 3, we compared its performance with our approach in the same manner as we did for Policy 2. The results of our analysis (depicted in second row of Table 5) indicates a substantial reduction in the average amount of transferred data for larger blockers, i.e., pickups and trucks. Specifically, the average amount of transferred data for pickups and trucks decreased by approximately 99%, while the performance drop for the cart blocker remained at the same level as in Policy 2. Finally, for the human blockage type, the average amount of transferred data decreased by 27%, which was 2% less than the decrease observed in Policy 2.

For policy 4, where an arbitrary technique is chosen for all types of blockages, we found that the results of the chosen technique might do well with some blockage types and poorly with other. For example, handoff works well with larger blockages, but decreased the performance by 100% for smaller blockages since the cost of taking the handoff exceeds the small blockage duration. Therefore, if all of the blockages in an environment are small (e.g., carts or

humans), handoff provides no gains. Hence, handoff decreased the performance for carts and human blockages by 100% and stayed at the same level of policy 2 for the other blockages. The last row in Table 5 captures these results.

## 6 CONCLUSION AND FUTURE WORK

In this paper, we addressed the following problem: *From the plurality of blockage mitigation techniques, which blockage mitigation method should be employed?* and *What is sub-selection within that method?* We then introduced a GRU-based framework to solve the problem. We showed that the model provides a high level of accuracy by using only SNR values that are readily available as part of the underlying communication. We also showed substantial increase in throughput compared to alternative policies and ML models.

This paper focused on IIoT scenarios with low cost/power communication chips on fixed clients, which gives them access to a single omni/quasi-omni beam. For our future work, we plan to extend the work to support client mobility as well as when the client also has access to many communication beams. We will also investigate the possibility of taking paired/joint actions (e.g., when one action is taken by the BS and the other action is taken by the client) to address these more challenging network settings.

## 7 ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their valuable feedback, which helped in improving the presentation of the paper. This research was supported in part by NSF grant CNS-1942305.

## REFERENCES

[1] 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," in *IEEE Network*, 2019.

[2] Y. Lu, P. Richter, and E. S. Lohan, "Opportunities and Challenges in the Industrial Internet of Things based on 5G Positioning," in *Proceedings of IEEE 8th International Conference on Localization and GNSS (ICL - GNSS)*, 2018.

[3] P. Kortoci, A. Mehrabi, C. Joe-Wong, and M. Di Francesco, "Incentivizing Opportunistic Data Collection for Time-Sensitive IoT Applications," in *Proceedings of IEEE SECON*, 2021.

[4] P. Kortoci, L. Zheng, C. Joe-Wong, M. Di Francesco, and M. Chiang, "Fog-based Data Offloading in Urban IoT Scenarios," in *Proceedings of IEEE INFOCOM*, 2019.

[5] M. Gapeyenko, A. Samuylov, M. Gerasimenko, D. Moltchanov, S. Singh, M. R. Akdeniz, E. Aryafar, S. Andreev, N. Himayat, and Y. Koucheryavy, "Spatially-Consistent Human Body Blockage Modeling: A State Generation Procedure," in *IEEE Transactions on Mobile Computing*, 2020.

[6] Y. Liu and D. M. Blough, "Environment-Aware Link Quality Prediction for Millimeter-Wave Wireless LANs," in *Proceedings of ACM MobiWac*, 2022.

[7] A. Ichkov, P. Mähönen, and L. Simić, "End-to-End Millimeter-Wave Network Performance and Mobility Management Overhead in Urban Cellular Deployments with Realistic Pedestrian Traffic and Blockages," in *Proceedings of ACM MobiWac*, 2020.

[8] A. Zhou, L. Wu, S. Xu, H. Ma, T. Wei, and X. Zhang, "Following the Shadow: Agile 3-D Beam-Steering for 60 GHz Wireless Networks," in *Proceedings of IEEE INFOCOM*, 2018.

[9] O. Bshara, Y. Liu, I. Tekin, B. Taskin, and K.R. Dandekar, "mmWave Antenna Gain Switching to Mitigate Indoor Blockage," in *Proceedings of IEEE USNC/URSI*, 2018.

[10] "Code and data repository," https://mmw.cs.pdx.edu/repository.html.

[11] S. Srinivasan, X. Yu, A. Keshavarz-Haddad, and E. Aryafar, "Fair Initial Access Design for mmWave Wireless," in *Proceedings of IEEE ICNP*, 2020.

[12] S. Sur, I. Pefkianakis, X. Zhang, and K. Kim, "WiFi-Assisted 60 GHz Wireless Networks," in *Proceedings of ACM MOBICOM*, 2017.

[13] A. Zhou, X. Zhang, and H. Ma, "Beam-forecast: Facilitating mobile 60 GHz networks via model-driven beam steering," in *Proceedings of IEEE INFOCOM*, 2017.

[14] S. Rezaie, C. N. Manchón, and E. de Carvalho, "Location- and Orientation-Aided Millimeter Wave Beam Selection Using Deep Learning," in *Proceedings of IEEE ICC*, 2020.

[15] A. O. Kaya and H. Viswanathan, "Deep Learning-based Predictive Beam Management for 5G mmWave Systems," in *Proceedings of IEEE Wireless Communication and Networking Conference (WCNC)*, 2021.

[16] L. Jiao, P. Wang, A. Alipour-Fanid, H. Zeng, and K. Zeng, "Enabling Efficient Blockage-Aware Handover in RIS-Assisted mmWave Cellular Networks," in *IEEE International Journal of Advanced Computer Science and Applications*, 2022.

[17] M. S. Mollel, S. Kaijage, and M. Kisangiri, "Deep Reinforcement Learning based Handover Management for Millimeter Wave Communication," in *International Journal of Advanced Computer Science and Applications*, 2021.

[18] Y. Kumar S. and T. Ohtsuki, "Influence and Mitigation of Pedestrian Blockage at mmWave Cellular Networks," in *IEEE Transactions on Vehicular Technology*, 2020.

[19] Z. Wang, L. Li, Y. Xu, H. Tian, and S. Cui, "Handover Control in Wireless Systems via Asynchronous Multiuser Deep Reinforcement Learning," in *IEEE Internet of Things Journal*, 2018.

[20] Y. Sun, G. Feng, S. Qin, Y. C. Liang, and T. P. Yum, "The SMART Handoff Policy for Millimeter Wave Heterogeneous Cellular Networks," in *IEEE Transactions on Mobile Computing*, 2018.

[21] C. L. Vielhaus, J. V. S. Busch, P. Geuer, A. Palaios, J. Rischke, D. F. Külzer, V. Latzko, and F. H. P. Fitzek, "Handover Predictions as an Enabler for Anticipatory Service Adaptations in Next-Generation Cellular Networks," in *Proceedings of ACM MobiWac*, 2022.

[22] K. Cho, B.V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," in *arXiv:1406.1078*, 2014.

[23] S. Wu, M. Alrabeiah, Chakrabarti C, and A. Alkhateeb, "Blockage Prediction Using Wireless Signatures: Deep Learning Enables Real-World Demonstration," in *IEEE Open Journal of the Communications Society*, 2022.

[24] A. Almutairi, S. Srinivasan, A. Keshavarz-Haddad, and E. Aryafar, "Deep Transfer Learning for Cross-Device Channel Classification in mmWave Wireless," in *Proceedings of IEEE MSN*, 2021.

[25] S. Li, Q. Ni, Y. Sun, G. Min, and S. Al-Rubaye, "Energy-Efficient Resource Allocation for Industrial Cyber-Physical IoT Systems in 5G Era," in *IEEE Transactions on Industrial Informatics*, 2018.

[26] H. Ren, C. Pan, T. Deng, M. Elkashlan, and A. Nallanathan, "Joint Pilot and Payload Power Allocation for Massive-MIMO-Enabled URLLC IIoT Networks," in *IEEE Journal on Selected Areas in Communications*, 2020.

[27] S. Zeb, A. Mahmood, H. Pervaiz, S. A. Hassan, M. I. Ashraf, Z. Li, and M. Gidlund, "On TOA-based Ranging over mmWave 5G for Indoor Industrial IoT Networks," in *IEEE Globecom Workshops*, 2020.

[28] Q. Chen, X. Xu, H. Jiang, and X. Liu, "An Energy-Aware Approach for Industrial Internet of Things in 5G Pervasive Edge Computing Environment," in *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, 2021.