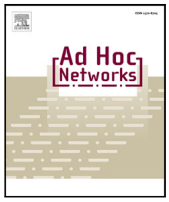
Ad Hoc Networks 162 (2024) 103562

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc



Survey paper

A deep learning framework for blockage mitigation and duration prediction in mmWave wireless networks

Ahmed Almutairi [a], Alireza Keshavarz-Haddad [b,*], Ehsan Aryafar [a]

[a] Computer Science Department, Portland State University, Portland, OR, USA
[b] School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran

ARTICLE INFO

Keywords:
MmWave wireless
Blockage
Recurrent Neural Networks
Deep learning

ABSTRACT

Millimeter-Wave (mmWave) communication can be highly affected by blockages, which can drastically decrease the signal strength at the receiver side. To overcome the impact of blockages, predicting the optimal mitigation technique and accurately estimating the duration of the blockage events are crucial for maintaining reliable and high-performance mmWave communication systems. Prior works on mitigating blockages have proposed a variety of model and protocol-based blockage mitigation solutions that concentrate on a singular technique at a time, like switching the current beam to an alternative beam at the current base station or client. In this paper, we tackle the overarching question: *what blockage mitigation technique should be employed?* and *what is the optimal sub-selection within that technique?* We also address the blockage duration estimation problem. We solve these problems by developing a Gated Recurrent Unit (GRU) model, trained on data from periodic message exchanges in mmWave systems. We tested our neural network models by utilizing a mmWave simulator that is commercially available and widely used in wireless communication to compile a large amount of dataset for this purpose. Our findings reveal that our proposed method introduces no extra communication overhead, while achieving remarkable accuracy, exceeding 91%, in predicting the optimal blockage mitigation technique. Moreover, the blockage duration estimation model achieves a very high accuracy with a residual mean error of less than 0.04 s. Finally, we demonstrate that our proposed blockage mitigation method substantially boosts the volume of data transferred in comparison to various other blockage mitigation strategies.

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) [1–5]. 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 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 [6–8]. 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 [9] or widen the beams so that the signal passes through the blocker [10].

Our objectives in this paper are to address the negative impact of blockages and predict their duration. For blockage mitigation, we aim to address the overarching problem: *from the plurality of blockage mitigation techniques, which one should be employed?* In our prior work [11], we investigated this overarching problem by focusing on three techniques (beam switching, beam widening, and handoff), but we assumed that (i) the client has access to only a single omni-directional beam, and (ii) only the BS takes the blockage mitigation action. In this extended work,

* Corresponding author.
E-mail addresses: almut8@pdx.edu (A. Almutairi), keshavarz@shirazu.ac.ir (A. Keshavarz-Haddad), earyafar@pdx.edu (E. Aryafar).

https://doi.org/10.1016/j.adhoc.2024.103562
Received 30 January 2024; Received in revised form 7 May 2024; Accepted 22 May 2024
Available online 25 May 2024
1570-8705/© 2024 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

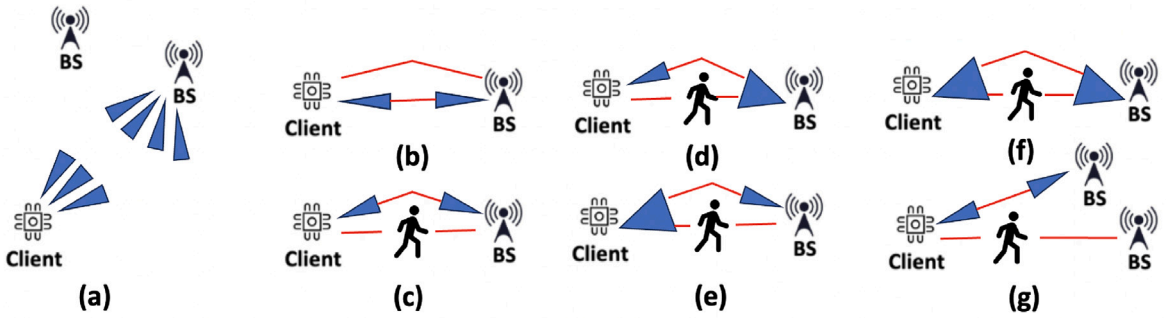


Fig. 1. (a): There are several beams to choose from at both the BS and the client. Beam selection typically happens through a search process at the beginning of every communication interval; (b): BS and client have identified proper beams for communication with each other. Here the two red paths capture the multi-path nature of the communication channel; (c): In BeSw_BeSw, BS and client switch to different beams when blockage happens; (d): In BeWi_BeSw, BS widens its beam and client switches to a different beam; (e): In BeSw_BeWi, BS switches to a different beam and client widens its beam; (f): In BeWi_BeWi, BS and client widen their beams. Energy reaches the client through other paths; (g): In Ho_BeSw, the network may change the BS serving the client and client switches to a different beam.

we assume that the client has also access to a plurality of directional beams and can collaborate with the BS for better blockage mitigation. In addition, we also address the blockage duration prediction problem.

Specifically, we focus on five blockage mitigation techniques as depicted in Figs. 1(c)–(g): beam switching on both the BS and client (BeSw_BeSw), beam switching on the BS and beam widening by the client (BeSw_BeWi), beam widening by the BS and beam switching by the client (BeWi_BeSw), beam widening on both the BS and client (BeWi_BeWi), and client handoff to a new BS, which results in new beam selections on both sides (Ho_BeSw). 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. For the blockage duration prediction, we aim to estimate the exact time interval of the blockage event. In this way, we can estimate when the blockage event ends and hence the communication can revert back to the original settings (*i.e.*, before applying the mitigation technique). To address these problems, we develop frameworks that proactively take the appropriate actions 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 make predictions. Our key contributions can be summarized as follows:

- **Data Gathering:** We utilize Wireless InSite (WI) simulator [12] to conduct numerous experiments and model different types of blockages in an IIoT setting. Prior to publication, we will publicly release all of our data and software code so that other researchers in the community can use our results and build on our work.
- **Blockage Mitigation Framework:** We develop a new GRU-based framework to mitigate blockages. We show that GRUs have a significantly higher accuracy with over 91% in selecting the optimal action when compared to Categorical Naive Bayes and Transformer Networks. We also show that the solution only needs a few time series samples, which slightly increases its accuracy when compared to Long Short Term Memory (LSTM) RNNs in addition to using less memory and being faster.
- **Blockage Duration Prediction Framework:** We develop a GRU-based framework to predict the blockage duration. We show that our model can predict the blockage duration with residual error mean of less than 0.04 s, indicating that the difference between the actual and predicted blockage duration values is very small.
- **Policies:** We model alternative forms of blockage mitigation techniques as policies, formally prove our solution provides a higher throughput than them, and show through simulations that it substantially increases the amount of transferred data across all types of blockages.
- **Average Loss Ratio:** We develop a theoretical model to derive an upper bound on the average data loss ratio. This metric models the impact of imperfect blockage duration estimation (and the resulting sub-optimal mitigation technique selection) by dividing the total amount of data loss (due to imperfections) to the total amount of transferred data in the ideal setting (*i.e.*, when there is no error in duration prediction). We verify this upper bound through simulations and also show that the average loss ratio is under 6% of the ideal setting.

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 and duration prediction methods. We present the policy-based definition of our approach along with alternative policies in Section 4 and present a theoretical framework to model average loss ratio in Section 5. Section 6 presents our data gathering process and performance evaluation results. Section 7 discusses the limitations of our work and our plans for future extensions. Finally, we conclude the paper in Section 8.

2. Related work

The blockage mitigation problem has been intensively investigated in prior work. In this section, we discuss the mmWave blockages and their impacts. Then, we discuss the studies that investigate different methods to mitigate blockages. Next, we discuss the previous studies in predicting blockage duration.

2.1. mmWave blockages

MmWave signals can be easily degraded or completely blocked by a variety of blockages [13]. These blockages can vary depending on the environment. For example, in urban scenarios, blockages such as buildings and vehicles can be prominent, while human and furniture can be prominent for indoor scenarios. These different type of blockages can adversely affect the mmWave link (*e.g.*, human body alone can highly decrease the overall rate or throughput of a mmWave communication link). Many prior works have focused on studying the impact of these blockages on the mmWave link in different environments. Some of these works have studied impact of human body on the performance of mmWave communication links and systems [14–16]. Other works have investigated the effects of different blockages (*e.g.*, roads, bridges, and buildings) in a variety of urban scenarios [17–20]. Mmwave communication is not only affected by larger blockages, but it could be also affected by smaller blockages (*e.g.*, road signs) [21].

2.2. Methods to mitigate blockages

Many of prior works have stated that the negative impact of blockages on mmWave communication can be mitigated with three main techniques (beam switching, handoff, and beam widening). Each of those works has studied one technique at a time as a solution to mitigate the impact of the blockages.

Beam Switching. Beam switching is a crucial method for mitigating the impact of blockages [22]. Here, when the current beam is blocked, another beam can be selected for communication (Fig. 1(c)). This technique has been studied in several prior works optimized through: (i) utilizing an out-of-band radio such as legacy sub-6 GHz WiFi [23], (ii) sensing the reflective environment by leveraging the observation that mmWave channels at nearby locations are highly correlated [24], (iii) using model-driven methods by identifying the hidden relationships between 3D beams and mmWave channels to reduce the beam adaptation overhead [9], and (iv) employing deep learning based on a given client's location and environment [25,26]. Beam switching is a useful technique for mitigating blockages in specific cases. However, in other cases (*e.g.*, when blockers are large), it might not be able to help with blockage mitigation.

Handoff. Another solution for mitigating blockages is handoff (Fig. 1(g)). Here, when the current connection between the BS and client is blocked, the client can be handed off to another BS. A number of prior studies (*e.g.*, [27–32]) have enhanced the handoff decision making process by using machine learning models (particularly deep reinforcement learning) based on channel state information, the client location, and other network parameters. Handoff is an effective solution for mitigating blockages in numerous situations. However, it can fail for other many scenarios. For instance, if the time required to complete handoff exceeds the duration of the blockage event, then handoff becomes insufficient. This issue can become more pronounced in environments with higher mobility, where the frequency of handoffs may be excessively high.

Beam Widening. When a blockage happens, one or both ends of the communication link (*e.g.*, BS and client) can increase the half-power beamwidth of their current beams (Fig. 1(d)–(f)). This technique has been investigated in prior works for mitigating blockages, *e.g.*, the work in [10] has experimentally shown the benefits of beam widening to mitigate blockages in indoor environments, the work in [33] proposes a theoretical model based on multi-armed bandits to adjust the beamwidth on both clients and BSs, and the work in [34] uses partial activation of antenna arrays to widen the beams. However, beam widening reduces the beamforming gain. Additionally, when the blocker is large or located close to the client or the base station, beam widening might still be ineffective.

Prior works have attempted to address the blockage mitigation problem by considering only one technique at a time. However, considering one technique might be beneficial for specific scenarios but fails in others. Different scenarios and environments require different techniques. Hence, we aim here to integrate the above techniques into one system. Therefore, our framework tackles different scenarios and blockages effectively, providing a more robust approach to mitigating blockages.

2.3. Blockage timing estimation

Blockage Duration Prediction. Identifying the blockage duration can help in maximizing the overall wireless network performance by incorporating it an overall blockage mitigation framework. For example, the solution can revert back to the original settings (BS associations and beams) once the blockage event ends, increasing the amount of data transferred. This blockage duration prediction problem is less addressed by the research community. The work in [35] uses a Markov chain model to determine the average blockage duration. Similarly, the work in [36] uses a theoretical model to estimate the average fraction of time that the LoS path is blocked. Our work aims to determine the exact blockage duration, which can be very different across different types of blockers. The work in [37] tries to proactively identify the type of incoming blockage based on how severe it is. The severity is quantified using a discrete blockage-severity index, which is derived from the average blockage time interval measured for each object. Next, authors group objects based on each object's average blockage time interval, each of which represents a class label. However, predicting the time interval of blockages in a discrete manner and tied to the blocker type, specifically in a time sensitive system such as mmWave wireless, can have severe downside effects. For example, predicting the blocker as a bus with an average 1300 msec duration can result in significant drop in the amount of transferred data when the true blocker is a skateboard with an average 180 msec blockage duration. In our work, we consider a continuous time GRU-based framework that solves the blockage duration prediction as a regression problem. We show that our approach results in a very low mean residual error across all different types of blockers, speeds, and antenna array configurations at both the BS and client.

3. GRUs for mmWave blockage mitigation

In this section, we introduce our GRU-based frameworks for blockage mitigation and duration prediction. We begin by providing a brief overview of GRUs and how they differ from LSTM RNNs. Following this, we describe our system model and explain how we employ GRU models to address blockages and forecast their duration.

3.1. Gated recurrent units (GRUs)

GRU [38] is an architecture within the recurrent neural network (RNN) family, specifically designed for processing and predicting outcomes in sequential data. Introduced as a streamlined alternative to the more complex LSTM (Long Short-Term Memory) networks, GRUs have gained popularity due to their simplicity and efficiency.

The design of GRU addresses a critical challenge in traditional RNNs known as the vanishing gradient problem. This problem arises when the gradients, which are used during the training of the network to update the weights, become exceedingly small, effectively halting the network from learning further. This can severely impact the network's ability to remember information over longer sequences, which is crucial for tasks that involve understanding context over time. GRU mitigates this issue through its unique gating mechanisms, which help maintain the flow of gradients during training.

GRU utilizes two gates, as shown in Fig. 2, within its architecture: the reset gate and the update gate. These gates determine the extent to which information is passed through the network. The reset gate decides how much of the previous information to forget, and the update gate decides what portion of the new information will be used to update the hidden state. This selective remembering and forgetting allow GRUs to capture dependencies over different time spans more effectively.

Compared to LSTMs, GRUs offer a more compact architecture, primarily because they use fewer parameters. This reduction in parameters not only speeds up the training process but also reduces the computational burden, making GRUs a more efficient choice in many scenarios. Despite their simplicity, GRUs have demonstrated performance on par with LSTMs in various applications. These include tasks in natural language processing, such as language modeling and machine translation, and in other domains like time-series prediction and speech recognition.

Moreover, the efficiency of GRUs makes them particularly well-suited for use in mobile applications and other environments where computational resources are limited. Their ability to achieve similar or even superior results to LSTMs with fewer parameters means that they can be deployed effectively in systems where fast execution and low memory footprint are critical. As research continues to evolve, GRUs are being further optimized and adapted to a wider range of tasks. In this paper, we also show their benefit as a blockage mitigation tool in mmWave wireless networks.

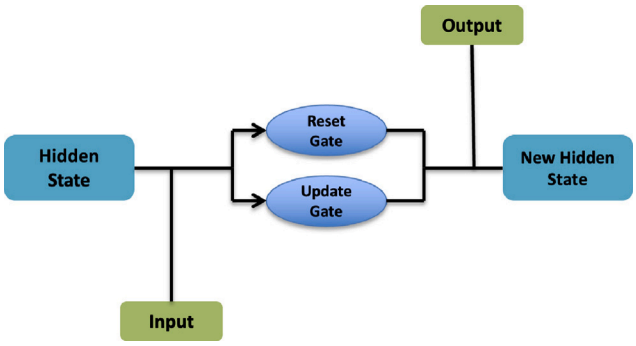


Fig. 2. GRU Architecture diagram. It shows the input being processed by two gates: the reset gate, which decides the amount of past information to be forgotten, and the update gate, which determines the amount of current information to be used in a new hidden state, influencing the output.

3.2. System model

We consider an IIoT scenario [1,2] that comprises stationary BSs and clients, alongside moving blockers. The BSs and clients are equipped with phased arrays of multiple antennas that function in the mmWave band, utilizing a set of directional beams to span a specified horizontal range. The action to mitigate blockages are collaboratively formulated by both BSs and clients, although the prediction of these actions and the duration of blockages are managed by the BS/network side.

Our work builds on previous work [37] that employs a cutting-edge deep learning approach to proactively determine the likelihood of future blockages and the time of their occurring. The methodology demonstrated in their work achieves remarkable precision in predicting oncoming blockages and depends solely on the existing communications between the BS and client, thus avoiding extra communication overhead. This efficiency is attained through utilizing a sequence of in-band wireless data along with a combination of recurrent and convolutional neural networks. Furthermore, this work [37] assumes that the current connection between the BS and the client is line-of-sight (LoS), which can be accurately predicted in mmWave systems. For instance, the study in [39] successfully classifies LoS and non-LoS (nLoS) conditions at each communication interval's onset, leveraging only the data exchanged during the beam search phase, which does not introduce additional overhead.

Our frameworks in this paper build upon the work in [37], which predicts future blockages and when they are going to occur. Following their model prediction, our proposed blockage mitigation framework determines the optimal actions to counteract the impending blockage's effects, while our blockage duration prediction model estimates the time required for a blockage to clear.

3.3. Blockage mitigation and duration prediction

Our objectives are to mitigate blockages by proactively minimizing their impact and predict duration of the blockages, which can collaboratively lead to increased throughput and reduced latency of communication. Blockages can exhibit similar shapes, velocities, and patterns of trajectories. To address these issues, we propose two models that learn similar features from a sequence of reported signal-to-noise ratio (SNR) values to determine the best action for mitigating the impact of blockages and estimating the blockage duration. The blockage duration is a continuous time that depends on type, size, and speed of a blocker, while the action space consists of five main joint actions. The first part of the joint actions is taken by the BS and the latter is the client's action. These joint actions include beam switching_beam switching (BeSw_BeSw), beam widening_beam switching (BeWi_BeSw), beam switching_beam widening (BeSw_BeWi), beam widening_beam widening (BeWi_BeWi) and handoff_beam switching (Ho_BeSw). Each main joint action includes multiple sub-actions, *e.g.*, BeSw_BeSw includes the selection of new beams from all available beams at the BS and the client, and Ho_BeSw includes handing of the client to any of the surrounding BSs and the selection of new beam from all available beams at the client. Therefore, the total number of actions includes sum of all sub-actions of the main joint actions. Let TNA be the total number of actions. Note that BeSw_BeSw involves all combinations between the BS and the client beams, BeWi_BeSw includes all combinations between the total number of beam widening levels at the BS and total number of beams at the client, BeSw_BeWi includes combinations between total number of beams at the BS and total number of beam widening levels at the client, BeWi_BeWi includes all combinations between the total number of beam widening at both BS and client, and Ho_BeSw includes all combinations between the total number of surrounding BSs and total number of beams at the client. Let the symbol # denote the number of sub-actions in a joint action, *e.g.*, #BeSw_BeSw shows the total number of BS and client beam combinations. Then, $TNA = \#BeSw\_BeSw + \#BeWi\_BeSw + \#BeSw\_BeWi + \#BeWi\_BeWi + \#Ho\_BeSw$.

GRU Model. We utilized a pair of GRU models, one dedicated to blockage mitigation (*i.e.*, technique selection) and the other focused on predicting the blockage duration. Each of those models consists of four layers of GRU cells with 128 units for each of the first hidden layers and 64 units for each of the latter two layers. The four GRU layers in each model are followed by a dense output layer with the size equal to the number of actions for the blockage mitigation and a single unit for the blockage duration prediction. The input to both models is a sequence of length N time steps each of which consists of SNR values of the current BS-client connection'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 $s = \{\{b_1, b_2, \ldots, b_B\}, \{BS_1, BS_{SNR_1}, \ldots, BS_H, BS_{SNR_H}\}\}$. We assume there are B beam pair combinations between the serving BS and client and each b_i shows the client SNR for beam pair combination i.[1] Further, BS_1, BS_{SNR_1} shows the ID of BS one and the SNR value of the best BS-client beams (measured at the client) of BS one. Note the H here is the maximum number of surrounding BSs. Then, 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 and client), 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 both GRU models with these MRs. For the blockage mitigation model, we optimized the training of our neural network model 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. For the blockage duration prediction model, we used similar configurations as the first model except for using mean squared error (MSE) as the loss function and linear as the activation function of the output layer. Table 1 summarizes our GRU model configurations and Fig. 3 shows the GRU architectures we used in our models.

Action Selection Metric. The dataset samples that are fed to the GRU models must be labeled with the actual blockage duration and the best action. The blockage duration can be achieved through $T = \frac{L}{V}$, where L is the length of the blocker and V is the blocker velocity. Defining the best action label is more challenging. For example, selecting the best action based on the highest reported SNR value can be inefficient. For example, let the Ho-BeSw action take 1 s to complete and let the human blockage last for 0.4 s. Therefore, if the Ho-BeSw action is

[1] It is possible to reduce this overhead by using compressive sensing or other techniques. We leave development of such techniques and their impact on the accuracy of our neural networks as part of our future work.

Table 1
GRU models structure and configurations.

Parameter	Blockage mitigation	Blockage duration
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	TNA Units	1 Unit
GRU activation	ReLU	
Output activation	Softmax	Linear

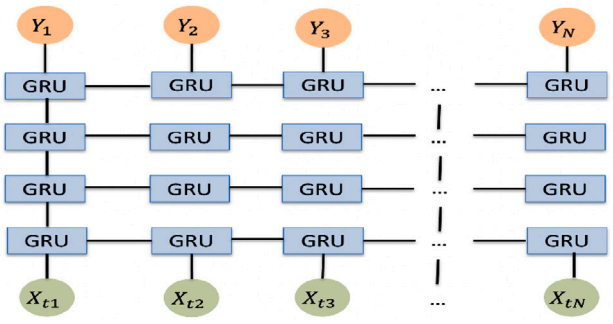


Fig. 3. GRU model structure with four GRU layers. X_{t1} to X_{tN} are the time steps that are used as the input, while Y_1 to Y_N are the outputs of the model, corresponding to the probability of each class (blockage mitigation strategy). Blockage duration prediction uses the same architecture but with only one scalar value corresponding to the blockage duration.

selected because of the highest expected SNR, there will be 0.6 s 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 throughput of a client associated with a BS with a given SNR value as:

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

here, ω is the communication bandwidth of 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 estimated throughput metric (R) defined in Eq. (1) and both the duration of the blockage event (T) and the cost 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. In our simulations, we let the cost associated with Ho_BeSw be 1 s, BeSw_BeSw be .01 s, and BeSw_BeWi, BeWi_BeSw, or BeWi_BeWi be 0.015 s.

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, which are characterized by their size.[2] Denote the probability of occurrence of blockage type i by P_i. Further, assume there are U blockage mitigation techniques, and the cost of each technique j is denoted by C_j. Let T_i be the duration of blockage type i, and let $R_{i,j}$ be the achievable rate of the user when blockage type is i and mitigation technique is j, in the given network. Note that $R_{i,j}$ is a random variable. 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 and uses neural networks to implicitly identify blockage type and data rate. 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

$$D_{i,max} = \max(D_{i,1}, D_{i,2}, \ldots, D_{i,U})\ \&\ \bar{D}_{i,max} = \mathbb{E}[D_{i,max}]$$

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

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

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:

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

[2] We assume that different blockers have different sizes and we have a discrete number of them such as human, car, truck, and pickup. Further, the velocity of the blocker would impact the blockage duration and data rate, that is why we consider these variables as random variables. There could, however, be randomness in the size of blockers that belong to the same type as well, *e.g.*, humans of different heights. There are several ways this can be accommodated in our theoretical analysis, *e.g.*, just considering the same number of blockers but with random variables (such as blockage duration and data rate) with a higher variance, or introducing a higher number of blocker types, *e.g.*, tall vs medium vs short humans. In the most extreme case, we can consider each unique blocker as a different type. In theory, even a clustering method can be used to group blockers into types.

where

$$\sum_{i=1}^{K} P_i \times \bar{D}_{i,j\star} = \max\left(\sum_{i=1}^{K} P_i \times \bar{D}_{i,1}, \sum_{i=1}^{K} P_i \times \bar{D}_{i,2}, \ldots, \sum_{i=1}^{K} P_i \times \bar{D}_{i,U}\right)$$

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 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 Section 6, 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. Theoretical model on average loss ratio

Estimating the blockage duration can be a useful tool in an overall blockage mitigation framework. For example, consider a high throughput link that undergoes a blockage event. If the system has an estimate of blockage duration, the BS and client can fall back to the original link once blockage ends. However, over or under estimation of the blockage duration coupled with sub-optimal blockage mitigation technique selection can limit the performance. In this section, we develop a theoretical framework to model the average data loss ratio, a metric that captures these inaccuracies. In short, the metric captures the loss in data rate due to inaccuracies divided by the total amount of data that can be transferred assuming optimal blockage duration estimation and mitigation strategy selection. In Section 6, we show through simulations that the loss ratio is indeed very small and the upper bound derived in this section is very tight.

Let T, $\hat{T}$, C_k, and R_k be the actual blockage duration, predicted blockage duration, the cost of applying the kth mitigation technique, and the data rate (throughput) after applying the kth mitigation technique, respectively. Then, we define the index of predicted mitigation technique (p), the index of the optimal technique (o), and the index of the maximum achievable rate (q) for each given blockage as follows:

$$\begin{cases} p = \underset{k}{\text{argmax}}(\hat{T} - C_k) \times R_k \\ o = \underset{k}{\text{argmax}}(T - C_k) \times R_k \\ q = \underset{k}{\text{argmax}}(R_k) \end{cases}$$

Note that we expect $p = o$ for the majority of the cases. If not, we must have:

$$\begin{cases} (\hat{T} - C_p) \times R_p > (\hat{T} - C_o) \times R_o \\ (T - C_o) \times R_o > (T - C_p) R_p \end{cases}$$

To derive an upper bound on the average loss ratio, we first derive an upper bound on the data loss as follows:

$$\begin{aligned} loss\_data =& (T - C_o) \times R_o - (T - C_p) \times R_p \\ \leq& (T - C_o) \times R_o - (T - C_p) \times R_p + (\hat{T} - C_p) \\ & \times R_p - (\hat{T} - C_o) \times R_o = (T - \hat{T}) \times (R_o - R_p) \\ \leq& |T - \hat{T}| \times |R_o - R_p| \leq |T - \hat{T}| \times R_q \end{aligned} \tag{8}$$

Next, we derive a lower bound on the total amount of data that can be transferred in the optimal case as follows:

$$total\_data = (T - C_o) \times R_o \geq (T - C_q) \times R_q = T \times R_q - C_q \times R_q \tag{9}$$

Note that variables T and R_q are independent. Similarly, variables $T - \hat{T}$ and R_q are independent. Thus, leveraging Eqs. (8) and (9) we can derive the following upper bound on the average loss ratio:

$$\begin{aligned} \overline{Loss - Ratio} =& \mathbb{E}\left(\frac{loss\_data}{total\_data}\right) \leq \frac{\mathbb{E}(|T - \hat{T}|) \times \mathbb{E}(R_q)}{\mathbb{E}(T) \times \mathbb{E}(R_q) - \mathbb{E}(C_q \times R_q)} \\ =& \frac{\mathbb{E}(|T - \hat{T}|)}{\mathbb{E}(|T|) - \frac{\mathbb{E}(C_q \times R_q)}{\mathbb{E}(R_q)}} \end{aligned} \tag{10}$$

6. 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 GRU-based blockage mitigation and duration estimation techniques. We consider both ML metrics (*e.g.*, accuracy in technique selection and mean squared error in duration prediction) as well as networking metrics (*e.g.*, transferred data, throughput). We also do comparisons against other ML methods (*e.g.*, LSTM, Transformers) and blockage mitigation policies. Finally, we empirically evaluate the upper bound we derived in the previous section and show its tightness.

6.1. Measurement campaign

Simulator. We used a commercially available wireless simulator named Wireless InSite (WI) [12][3] to simulate an IIoT use case scenario with a size of 350×150 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 [40,41]. MmWave wireless technologies play a significant role in IIoT by providing high data rates and assisting with positioning [1,2,42,43], among others.

Simulation Setup. In our simulation setting, we deploy a total of six BSs and one hundred clients. The BSs are positioned along both sides of the environment, with three BSs on each side, maintaining a distance of 75 m between each one another. Clients are scattered randomly throughout the environment (as depicted in Fig. 4). Each BS and client deploys a uniform linear array (ULA) consisting of either 16, 8, or 4 antennas, enabling the creation of 18, 9, or 4 beams, respectively.

[3] Wireless Insite [12] is a commercial software used for site-specific wireless performance analysis. It uses a combination of RF propagation models, 3D ray-tracing, fast ray-based methods, and empirical models for the analysis of site-specific radio wave propagation and wireless communication systems. Potential biases can arise from various sources such as fidelity of models used to mimic wave interaction with different materials, combination of ray tracing and statistical models (which create computational complexity versus accuracy tradeoffs), and fidelity of antenna models used in the simulator.

Table 2
Simulation setup.

Parameter	Value
BS antenna array sizes	16 and 8 (ULA)
Client antenna array sizes	8 and 4 (ULA)
Number of beams at the BS	18 and 9
Number of beams at the client	9 and 4
Frequency	28 GHz
Bandwidth	1 GHz

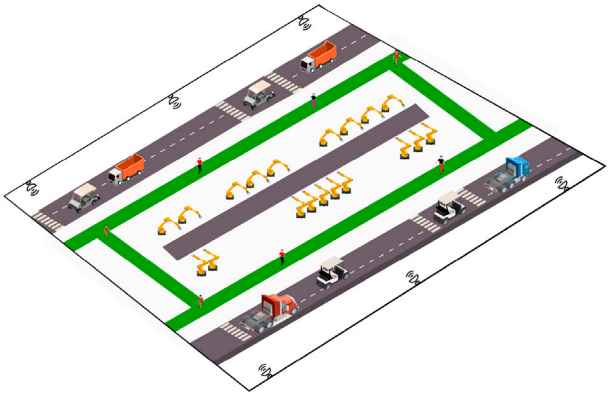


Fig. 4. IIoT scenario with six BSs on the two sides, clients, and different types of blockers. We included four different types of blockers: human, cart, truck, and pickup, which are the most commonly encountered blockers in an industrial IoT (*e.g.*, factory) setting.

These beams cover a horizontal expanse of 120°. All BSs and clients are configured to operate within the 28 GHz frequency band, with a line-of-sight (LoS) channel condition prior to any blockage events. The BSs stand at an elevation of 2.5 m, while the clients are positioned at a height of 1.5 m. Refer to Table 2 for a summary of the simulation parameters.

We considered four distinct types of blockers: human, cart, truck, and pickup. These blockage categories represent the most frequently encountered obstacles in a factory environment. We represented a human as a cylinder with a radius of 30 cm, capable of moving at speeds ranging from 0.89 to 1.4 m/s. The dimensions of carts, pickups, and trucks are $2.70\times1.21\times1.8$ m^3, $5.4\times2.1\times1.9$ m^3, and $12.3\times2.7\times2.3$ m^3, respectively. For these three types of blockers, we assumed velocities ranging from one to five miles per hour (mph), equivalent to 0.89 to 2.2 m/s.

Data Gathering. As outlined in Section 3.3, our GRU-based model processes a sequential dataset to determine the appropriate action. This dataset comprises multiple time steps, each containing SNR values for all beams of the current BS and client, along with the best SNR (stemming from the optimal beam) of each neighboring BS, accompanied by their respective IDs. We examine four array configuration scenarios: 16 antennas at the BS paired with 8 or 4 antennas at the client, and 8 antennas at the BS paired with 8 or 4 antennas at the client. Our simulation operates based on this setup, with SNR measurements recorded at the client's side every 100 msec. Hence, the time difference between consecutive steps remains constant at 100 msec. The simulation continues as blockers move until the ongoing connection is entirely obstructed. At the conclusion of each simulation, we extract the SNR values measured at the current BS, the best SNR values of surrounding BSs, along with their corresponding IDs for each time step, including the step where the connection becomes obstructed. Subsequently, we repeat the last time step (when the connection is blocked), while varying the number of antennas at the current BS or client to assess the performance of beam widening under blocked conditions.[4] This information is integral to labeling the dataset samples. Our simulations yielded a dataset comprising 30,000 samples, each consisting of five time steps. Within each time step, there are 36, 81, 72, or 162 SNR values, representing all combinations of serving BS and client beams, which are 9×4, 9×9, 18×4, or 18×9 beam combinations, respectively. Additionally, each time step includes the best SNR values for the five surrounding BSs and their respective IDs. The dataset is evenly distributed across four different types of blockages, with 7500 samples collected for each blockage type. For model training, we allocate 70% of the dataset, reserving the remaining 30% for testing purposes.

Table 3
Antenna size configurations scenarios.

BS antenna	BS beams	Client antenna	Client beams	No. of actions
8	9	4	4	194
8	9	8	9	384
16	18	4	4	536
16	18	8	9	870

Data Labeling. Our dataset samples must have two type of labels. First label is the actual time of the blockage duration. The second label is the best action to mitigate the impact of the blockage. To select the best action label for each sample, we start by establishing the number of classes or actions. This count is contingent on the quantity of antennas and beams present at the BSs and clients. As elaborated in Section 3.3, each sample in our dataset was labeled based on the optimal action that would maximize data transfer during a blockage event, employing Eq. (2) for this purpose. Table 3 provides a summary of the four configurations resulting from varying numbers of antennas at the BS/client, along with the total count of actions.

To delve deeper into our labeled dataset, we conducted two distinct analyses. First, we computed the percentage of samples labeled with each action across the entire dataset, irrespective of the blockage type. Fig. 5(a) illustrates the corresponding findings. We note that Ho_BeSw was identified as the optimal action for 64% of our samples, BeSw_BeWi for 15% of our samples, BeSw_BeSw for 12% of our samples, and BeWi_BeWi for 9% of our samples.

Subsequently, we conducted an analysis to explore the correlation between the best action and the type of blockage, depicted in Fig. 5(b). Our findings reveal that combinations involving beam switch and widening (BeSw_BeSw, BeSw_BeWi, and BeWi_BeWi) are frequently optimal for mitigating blockages caused by smaller obstacles (*e.g.*, human). Conversely, handoff (Ho_BeSw) emerges as a preferable action for larger obstructions (*e.g.*, pickup or truck). Moreover, we observed instances where BeSw_BeSw, BeSw_BeWi, and BeWi_BeWi outperformed Ho_BeSw in managing significant blockages. However, Ho_BeSw consistently proved ineffective in addressing small obstacles, primarily due to its higher cost relative to the duration of the blockage event.

6.2. Results

Next, we delve into the evaluation of our blockage mitigation framework, examining both ML and networking metrics. Additionally, we assess the efficacy of our blockage duration framework by analyzing the residual error mean between the actual and predicted blockage duration.

GRU for Blockage Mitigation Evaluation. Accuracy serves as a prevalent metric for assessing the performance of a machine learning model, representing the proportion of correct predictions relative to the total predictions made. We opt to assess our model's performance based on the last 5 time steps. Given four distinct array configurations, we

[4] In a uniform linear array (ULA) with M antennas and $\frac{\lambda}{2}$ spacing distance (λ 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 at the BS or client to model beam widening.

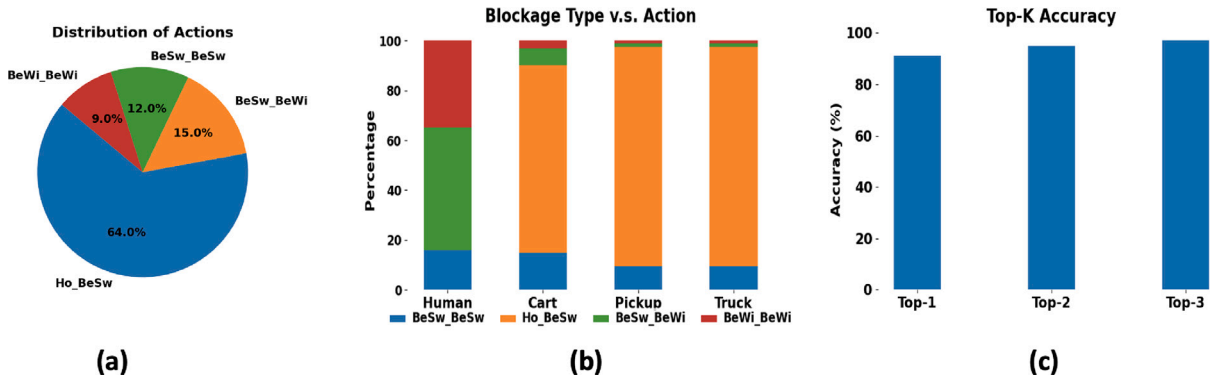


Fig. 5. (a): Fraction of samples in the dataset labeled with each action; (b): Fraction of samples labeled with each action for each blockage type; (c): Average op-1, Top-2, and Top-3 accuracy results (across the different array sizes) of our GRU-based blockage mitigation framework. The correct label is the predicted label in 91% of instances and is among the top three predictions for 97% of instances.

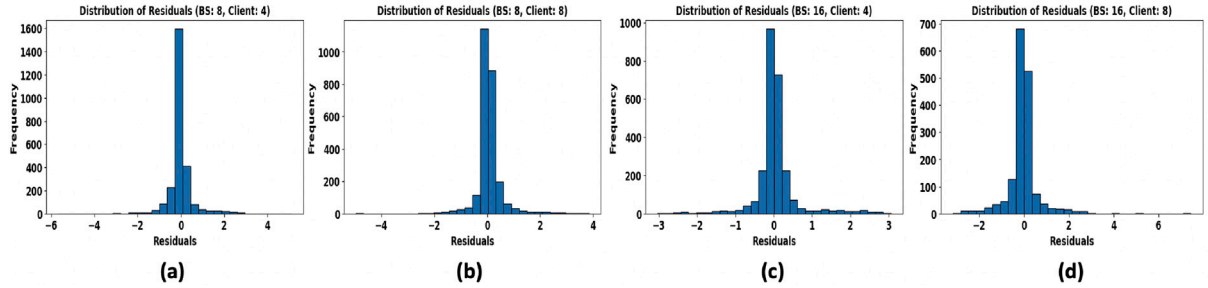


Fig. 6. The residual results represent the disparity between the actual and predicted blockage duration across the four array configuration scenarios: (a) BS: 8 - Client: 4; (b) BS: 8 - Client: 8; (c) BS: 16 - Client: 4, and (d) BS: 16 - Client: 8.

Table 4
GRU model accuracy for different antennas size configurations.

Array size	Accuracy
BS: 8, Client: 4	90.2%
BS: 8, Client: 8	91.4%
BS: 16, Client: 4	91.7%
BS: 16, Client: 8	91.5%

train and evaluate our model across these varied datasets. Remarkably, our GRU model exhibited robust performance across all configurations, boasting accuracy rates exceeding 90%. Notably, the model showcased its highest accuracy of 91.7% when base stations (BSs) were equipped with 16 antennas and clients with 4 antennas. Table 4 outlines the accuracy outcomes across the different array configuration scenarios.

Next, to obtain a more complete picture of the model's accuracy, we consider Top-K accuracy as depicted in Fig. 5(c). 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. To investigate the top K accuracy, we consider the average of the four array configuration scenarios. In our case, Top-1 accuracy of our GRU model is 91%. 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 95% (Fig. 5(c)), meaning that the correct label is among the top two predictions for 95% of instances. Furthermore, our model has achieved a Top-3 accuracy of 97%, which indicates that the correct label is among the top three predictions for 97% of instances. Note that to be counted as a correct action, details of the action must be correct too. For example, when BeSw_BeSw to a particular beams is the correct action, the model not only has to select beam switching for the BS and the client as the appropriate blockage mitigation technique, but it must also correctly select the beam to switch to (out of all the available beams at the BS and the client) to match the label.

GRUs for Blockage Duration Prediction. Since we are dealing with a regression problem here, and the aim is to predict continuous values, we need to use a metric that gives a clearer view by measuring the difference between the actual and predicted values. Thus, we use residual errors, which are the differences between the actual and predicted values. The use of residuals as a metric for performance evaluation involves assessing how well a model's predictions align with the actual data points. A residual plot or analysis helps to identify patterns or trends in the errors made by the model. Fig. 6 shows the residual plots for the four array configuration scenarios. In examining the residual plots of our model, it is evident that, on the whole, our predictions align well with the actual values. The majority of data points cluster around zero, indicating that, on average, our model is providing accurate estimates. The mean of the overall residual across all the four different antennas configuration scenarios is −0.04 s. While assessing the distribution of residuals, we observe a slight right-skewness, suggesting that there are several instances where our model tends to underestimate the target variable. However, it is important to note that our model deviations are generally modest.

Comparison with Other ML Models. To evaluate the efficacy of GRUs in mitigating blockages, we conducted a comparative analysis with three other machine learning techniques: Categorical Naive Bayes (CatNB), Transformer, and LSTM. CatNB, a variant of the Naive Bayes algorithm, is widely used for its proficiency in handling categorical data, particularly in text classification tasks. It operates under the assumption that all features are independent of each other given the class label. In contrast, Transformers [44] represent a revolutionary architecture in deep learning. These models differ from conventional sequential approaches by employing attention mechanisms that process all input data simultaneously, rather than sequentially. This capability

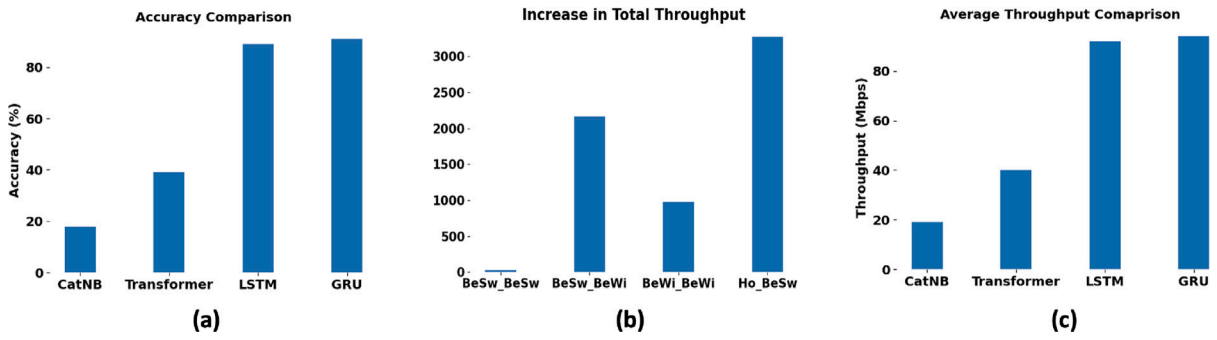


Fig. 7. (a): Accuracy of different ML models. GRU achieves the highest accuracy; (b): Increase in total transferred data for different actions. (c): Average throughput across all clients, BSs, and block- age events achieved by different ML models.

allows for significant gains in processing efficiency, making Transformers especially suitable for complex tasks like natural language understanding.

For our experiments, the implementation of CatNB was carried out using the scikit-learn library, with hyperparameters *alpha* and *fi_prior* set to 1 and true, respectively, to smooth probabilities and adjust class priors based on the training data. The Transformer model was developed using TensorFlow and Keras, featuring a design that includes four layers of multi-head self-attention and four layers of position-wise feedforward networks, with a global average pooling layer to aggregate sequence information. We configured the LSTM neural network similarly to the GRU, as outlined in Section 3.1, to maintain consistency in comparative evaluation.

The accuracy results, depicted in Fig. 7(a), demonstrate that the GRU model surpasses all compared techniques, achieving an average accuracy of 91% across various array configuration scenarios. CatNB and Transformers recorded accuracies of 19.7% and 39%, respectively, while LSTM, which shares similarities with GRU, achieved an 89% accuracy.

The preference for GRU over LSTM was based on several factors. Primarily, GRUs require fewer parameters than LSTMs, which reduces both the storage needs and the computational demand during training. This aspect is crucial when dealing with extensive datasets or under stringent time constraints for model training. Additionally, the slightly superior accuracy observed with GRUs compared to LSTMs further validated our decision to employ GRU models. This combination of efficiency and performance underlines the GRU's suitability for tasks requiring effective long-term dependency modeling.

Network Evaluation. We next evaluate the performance of our GRU model for blockage mitigation 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. 7(b) shows the corresponding results. Note that BeSw_BeWi includes both BeSw_BeWi and BeWi_BeSw. BeSw_BeSw increases the total amount of transferred data by x32, while BeSw_BeWi increases that by x2161. BeWi_BeWi increases the total amount of transferred data by about x971, while Ho_BeSw increases that by x3275, much higher than other mitigation techniques. Note, that the results do not mean that BeSw_BeSw is not an effective action. Ho_BeSw, for example, 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 Ho_BeSw) and baseline throughput (throughput of the connection blocked by a large blocker) is very large. Additionally, with Ho_BeSw, the new BS will likely observe no negative impact from the current blocker. BeSw_BeSw is still efficient actions when the blocker is small (*e.g.*, human as we showed in Fig. 5(b)). It also results in more increase in throughout than handoff when blocker is small, due to the shorter duration of the blockage and high cost of handoff.

Table 5
Comparison of GRU-based model against policies 2, 3, 4 in terms of average percentage drop in the amount of transferred data.

	Human	Cart	Pickup	Truck
Policy 2	−11%	−31%	−7.75%	−2%
Policy 3	−11%	−51.25%	−76%	−67%
Policy 4	−100%	−31%	−7.75%	−2%

Throughput Across Different ML Models. Throughput is a critical metric for evaluating the 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, Transformer, CatNB). We measure throughput for the duration of the blockage event taking into account blockage type and velocity, and action delay, among others. Fig. 7(c) summarizes the average throughput results for each model. Note that these throughput results are averaged across all array configuration scenarios (across all clients and BSs) and blockage events. Our evaluation shows that the GRU model outperforms all the baselines with an average throughput of 89.9 Mbps. The ratio of increase in throughput is 4.56 with respect to (w.r.t.) CatNB, 2.51 w.r.t. Transformer, and 1.02 w.r.t. LSTM.

Comparison with Other Policies. The GRU-based blockage mitigation framework exhibits a high level of performance in terms of both accuracy and throughput 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.

In Policy 2, a fixed best technique is employed to maximize the average amount of transferred data for each blockage type. In other words, for each blockage type of human, cart, pickup, and truck, Policy 2 uses a fixed technique that would result in the highest amount of transferred data for that type of blockage.

Through offline analysis, we found that the optimal technique for blockage type "human" is BeSw_BeWi, while the best technique for "cart", "pickup" and "truck", is Ho_BeSw. To determine the average amount of data transferred for each blockage type, we applied the corresponding best technique and compared the results with our proposed approach. The first row in Table 5 depicts the results. Our results show that when employing only BeSw_BeWi for human blockage, the average amount of transferred data decreases by 11% compared to our approach. Similarly, for cart, the use of Ho_BeSw only, resulted in a 31% decrease in the average amount of transferred data. For pickup, using Ho_BeSw only, resulted in a 7.75% decrease in the average amount of transferred data. Finally, for a truck blocker, the average decrease is 2%. As we showed in the results of Fig. 5(b) Ho_BeSw is the

dominant best action when the blockage type is pickup or truck. Thus, an approach that can implicitly/explicitly infer the blockage type and use a fixed technique would result in a lower amount of transferred data compared to our approach (Policy 1).

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 that can be employed in Policy 3, 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 BeSw_BeWi 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 76% and 67%, respectively, while the performance drop for the cart blocker decreased by approximately 51.25%. Finally, for the human blockage type, the average amount of transferred data remained at the same level as in Policy 2. As we showed in the results of Fig. 5(b), for the three blocker types of car, pickup, and truck, Ho_BeSw has the highest percentage as the optimal blockage mitigation technique. Thus, choosing BeSw_BeWi across all blockage types can result in large average drop in the amount of transferred data for these types of blockers. Note that using only BeSw_BeWi for blockage mitigation still provides a higher average amount of total transferred data than only using Ho_BeSw as we discuss next.

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, Ho_BeSw works well with larger blockages, but decreased the performance by 100% for smaller blockages since the cost of taking the Ho_BeSw exceeds the small blockage duration. Therefore, if all of the blockages in an environment are small (*e.g.*, humans), Ho_BeSw provides no gains. Hence, Ho_BeSw decreased the performance for human blockages by 100% and stayed at the same level of policy 2 for the other blockages. The last row in Table 5 captures this.

Empirical Evaluation of Average Loss Ratio. The GRU-based blockage duration prediction framework shows a high level of performance, as observed in terms of residual error values, which exhibit the difference between the actual and predicted blockage duration. To gain a better insight into the impact of model inaccuracies in terms of blockage duration prediction (and the resulting sub-optimal action selection), we simulate the theoretical upper bound on average loss ratio introduced in Section 5. For this purpose, we utilized the same dataset that was used for both training and testing the GRU model. Subsequently, we computed the upper bound of data loss using Eq. (8). We also computed the total amount of data using Eq. (9). Lastly, we calculated the actual and predicted average loss ratio by applying Eq. (10).

We calculated the upper bound on loss ratio, total data, and actual average loss ratio by taking into account all the four different antenna configurations across the BS and clients. Then, we take the average across all the results. Table 6 summarizes our computation results. We observe that the theoretical model indeed provides an upper bound on the actual predicted data loss and average loss ratio. We further observe that the predicted loss ratio is small and close to the upper bound, indicating a small penalty in terms of transferred data loss due to the inaccuracies.

Table 6
Analysis of Upper Bound Loss, Total Data, and Average Loss Ratio. PUB stands for Predicted Upper Bound based on our model in Section 5.

	Value
Actual Data Loss	22.83 Mb
PUB on Data Loss	25.35 Mb
Total Data	376.5 Mb
Actual Average Loss Ratio	6%
PUB on Avg. Loss Ratio	6.7%

7. Limitations and future work

In this Section, we discuss some of the limitations and challenges we faced in our evaluation as well as some ideas we plan to explore in our future work.

First, the simulator's ability to mirror the nuances of real-world IIoT environments is limited. Factors such as varying interference patterns, unexpected equipment behavior, and environmental influences can significantly affect signal blockage, which may not be fully captured by the simulator. To mitigate this limitation, we calibrated the simulator using parameters from typical industry-standard models to enhance its realism.

Second, gathering real-world IIoT data to further validate our findings can be limited since real IIoT systems may have proprietary, security, and privacy issues. While simulation is a practical approach, securing partnerships, in future, with IIoT vendors and operators could provide access to real-world data and environments to further validate our findings.

Third, the results obtained may not directly cover all possible IIoT scenarios such as more complex IIoT environment due to the computational limitations that restrict the size of the simulated environment. We recommend future studies to use distributed simulation techniques to overcome these computational challenges and simulate larger networks.

Finally, our work in this paper assumed static clients and base stations in an IIoT setting, which are initially in LoS condition with respect to one another. We captured network dynamics by introducing mobile blockers of different sizes and speeds. As part of our future work, we plan to investigate the more complex deployments such as urban scenarios with mobile clients that may not initially be in LoS condition with respect to base stations.

8. Conclusion

In this paper, we addressed the following problems: *From the plurality of blockage mitigation techniques, which blockage mitigation method should be employed?*, *What is the optimal sub-selection within that method?*, and *what is the predicted blockage duration event?* We then introduced a GRU-based framework to solve these problems. We showed that the models provide a high level of accuracy for choosing the best blockage mitigation action to take and have a small mean residual error in predicting the blockage duration. The models only use SNR values that are readily available as part of the underlying wireless communication. We also showed substantial increase in throughput compared to alternative policies and ML models. Further, we introduced a theoretical model for calculating the average data loss ratio based on the predicted blockage duration. We showed through simulations that the upper bound is very tight and is on average less than 7% of total transferred data assuming ideal duration prediction. The results indicated the small loss in transferred data when there are inaccuracies in our blockage duration prediction model.

CRediT authorship contribution statement

Ahmed Almutairi: Writing – original draft, Software, Methodology, Investigation, Data curation, Conceptualization. **Alireza Keshavarz-Haddad:** Writing – review & editing, Visualization, Supervision, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Ehsan Aryafar:** Writing – review & editing, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research was supported in part by the United States National Science Foundation grant CNS-1942305 and CNS-1910517.

References

[1] J. Yang, B. Ai, I. You, M. Imran, L. Wang, K. Guan, D. He, Z. Zhong, W. Keusgen, Ultra-reliable communications for Industrial Internet of Things: Design considerations and channel modeling, IEEE Netw. (2019).
[2] Y. Lu, P. Richter, 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, M.D. 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.D. Francesco, M. Chiang, Fog-based data offloading in urban IoT scenarios, in: Proceedings of IEEE INFOCOM, 2019.
[5] R.W.L. Coutinho, A. Boukerche, Transfer learning for disruptive 5G-enabled Industrial Internet of Things, IEEE Trans. Ind. Inform. (2022).
[6] M. Gapeyenko, A. Samuylov, M. Gerasimenko, D. Moltchanov, S. Singh, M.R. Akdeniz, E. Aryafar, S. Andreev, N. Himayat, Y. Koucheryavy, Spatially-consistent human body blockage modeling: A state generation procedure, IEEE Trans. Mob. Comput. (2020).
[7] Y. Liu, D.M. Blough, Environment-aware link quality prediction for millimeter-wave wireless LANs, in: Proceedings of ACM MobiWac, 2022.
[8] A. Ichkov, P. Mahonen, L. Simic, 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.
[9] A. Zhou, L. Wu, S. Xu, H. Ma, T. Wei, X. Zhang, Following the shadow: Agile 3-D beam-steering for 60 GHz wireless networks, in: Proceedings of IEEE INFOCOM, 2018.
[10] O. Bshara, Y. Liu, I. Tekin, B. Taskin, K. Dandekar, mmWave antenna gain switching to mitigate indoor blockage, in: Proceedings of IEEE USNC/URSI, 2018.
[11] A. Almutairi, A. Keshavarz-Haddad, E. Aryafar, Gated recurrent units for blockage mitigation in mmWave wireless, in: Proceedings of ACM MobiWac, 2023.
[12] Wireless insite software: https://www.remcom.com/wireless-insite-propagation-software.
[13] G.R. MacCartney, T.S. Rappaport, S. Rangan, Rapid fading due to human blockage in pedestrian crowds at 5G millimeter-wave frequencies, in: IEEE GLOBECOM, 2017.
[14] A. Alyosef, S. Rizou, Z.D. Zaharis, P.I. Lazaridis, A.M. Nor, O. Fratu, S. Halunga, T.V. Yioultsis, N.V. Kantartzis, A survey on the effects of human blockage on the performance of mm wave communication systems, in: IEEE BlackSeaCom, 2021.
[15] A.B. Zekri, R. Ajgou, M. Hettiri, Impact of azimuth and elevation half power beam width on human blockage scenarios in mmWave channels, in: IEEE 1st International Conference on Communications, Control Systems and Signal Processing, CCSSP, 2020.
[16] X. Liu, Y. Zhang, T. Jiang, L. Yu, J. Zhang, L. Xia, Multi-person blockage loss modeling at millimeter-wave band, in: IEEE 95th Vehicular Technology Conference, 2022.
[17] J. Kim, D. Yan, K. Guarr, D. He, G. Noh, S. Choi, H. Chung, Effects of signal blockage by a road bridge on mmWave vehicular communications, in: IEEE 13th International Conference on Information and Communication Technology Convergence, ICTC, 2022.
[18] I.K. Jain, R. Kumar, S.S. Panwar, The impact of mobile blockers on millimeter wave cellular systems, IEEE J. Sel. Areas Commun. (2019).
[19] C. Garcia-Ruiz, O. Munoz, A. Pascual-Iserte, Effect of correlated building blockages on the ergodic capacity of mmWave systems in urban scenarios, IEEE Trans. Veh. Technol. (2022).
[20] C.G. Ruiz, A. Pascual-Iserte, O. Munoz, Analysis of blocking in mmWave cellular systems: Application to relay positioning, IEEE Trans. Commun. (2020).
[21] F. Alsaleem, J.S. Thompson, D.I. Laurenson, S.K. Podilchak, C.A. Alistarh, Small-size blockage measurements and modelling for mmWave communications systems, in: IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, 2020.
[22] S. Srinivasan, X. Yu, A. Keshavarz-Haddad, E. Aryafar, Fair initial access design for mmWave wireless, in: Proceedings of IEEE ICNP, 2020.
[23] S. Sur, I. Pefkianakis, X. Zhang, K. Kim, WiFi-assisted 60 GHz wireless networks, in: Proceedings of ACM MOBICOM, 2017.
[24] A. Zhou, X. Zhang, H. Ma, Beam-forecast: Facilitating mobile 60 GHz networks via model-driven beam steering, in: Proceedings of IEEE INFOCOM, 2017.
[25] S. Rezaie, C.N. Manchón, E. de Carvalho, Location- and orientation-aided millimeter wave beam selection using deep learning, in: Proceedings of IEEE ICC, 2020.
[26] A.O. Kaya, H. Viswanathan, Deep learning-based predictive beam management for 5G mmWave systems, in: Proceedings of IEEE Wireless Communication and Networking Conference, WCNC, 2021.
[27] L. Jiao, P. Wang, A. Alipour-Fanid, H. Zeng, K. Zeng, Enabling efficient blockage-aware handover in RIS-assisted mmWave cellular networks, IEEE Int. J. Adv. Comput. Sci. Appl. (2022).
[28] M.S. Mollel, S. Kaijage, M. Kisangiri, Deep reinforcement learning based handover management for millimeter wave communication, Int. J. Adv. Comput. Sci. Appl. (2021).
[29] Y.K. S., T. Ohtsuki, Influence and mitigation of pedestrian blockage at mmWave cellular networks, IEEE Trans. Veh. Technol. (2020).
[30] Z. Wang, L. Li, Y. Xu, H. Tian, S. Cui, Handover control in wireless systems via asynchronous multiuser deep reinforcement learning, IEEE Internet Things J. (2018).
[31] Y. Sun, G. Feng, S. Qin, Y.C. Liang, T.P. Yum, The SMART handoff policy for millimeter wave heterogeneous cellular networks, IEEE Trans. Mob. Comput. (2018).
[32] C.L. Vielhaus, J.V.S. Busch, P. Geuer, A. Palaios, J. Rischke, D.F. Kulzer, V. Latzko, F.H.P. Fitzek, Handover predictions as an enabler for anticipatory service adaptations in next-generation cellular networks, in: Proceedings of ACM MobiWac, 2022.
[33] M. Feng, B. Akgun, I. Aykin, M. Krunz, Beamwidth optimization for 5G NR millimeter wave cellular networks: A multi-armed bandit approach, in: Proceedings of IEEE ICC, 2021.
[34] H. Chung, J. Kang, H. Kim, Y.M. Park, S. Kim, Adaptive beamwidth control for mmWave beam tracking, IEEE Commun. Lett. (2021).
[35] C. Tunc, M.F. Ozkoc, S. Panwar, Millimeter wave coverage and blockage duration analysis for vehicular communications, in: Proceedings of IEEE VTC, 2019.
[36] D. Moltchanov, A. Ometov, On the fraction of LoS blockage time in mmWave systems with mobile users and blockers, in: Proceedings of IFIP WWIC, 2018.
[37] S. Wu, M. Alrabeiah, C. C, A. Alkhateeb, Blockage prediction using wireless signatures: Deep learning enables real-world demonstration, IEEE Open J. Commun. Soc. (2022).
[38] K. Cho, B. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, 2014, arXiv:1406.1078.
[39] A. Almutairi, S. Srinivasan, A. Keshavarz-Haddad, E. Aryafar, Deep transfer learning for cross-device channel classification in mmWave wireless, in: Proceedings of IEEE MSN, 2021.
[40] S. Li, Q. Ni, Y. Sun, G. Min, S. Al-Rubaye, Energy-efficient resource allocation for industrial cyber-physical IoT systems in 5G era, IEEE Trans. Ind. Inform. (2018).
[41] H. Ren, C. Pan, T. Deng, M. Elkashlan, A. Nallanathan, Joint pilot and payload power allocation for massive-MIMO-enabled URLLC iIoT networks, IEEE J. Sel. Areas Commun. (2020).
[42] S. Zeb, A. Mahmood, H. Pervaiz, S.A. Hassan, M.I. Ashraf, Z. Li, M. Gidlund, On TOA-based ranging over mmWave 5G for indoor industrial IoT networks, in: IEEE Globecom Workshops, 2020.
[43] Q. Chen, X. Xu, H. Jiang, X. Liu, An energy-aware approach for Industrial Internet of Things in 5G pervasive edge computing environment, IEEE Trans. Ind. Inform. (2021).
[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017, arXiv:1706.03762.

Ahmed Almutairi received the B.S. degree in Computer Science from the Department of Computer Science, Western Oregon University, Monmouth, OR, USA, in 2017 and the M.S. degree in Computer Science from Southern Illinois University, Carbondale, IL, USA, in 2018. He is currently pursuing the Ph.D. degree with the Maseeh College of Engineering and Computer Science, Portland State University, OR, USA. His research interests are in wireless networks, networked system and deep learning.

Alireza Keshavarz-Haddad received the B.S. degree in Electrical Engineering from Sharif University of Technology, Tehran, Iran, in 2001; and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from Rice University, Houston, Texas, USA, in 2003 and 2007, respectively. Currently, he is an Associate Professor at the School of Electrical and Computer Engineering at Shiraz University, Shiraz, Iran. His research interests are in the field of computer and communication networks with emphasis on fundamental mathematical models, architectures and protocols for wireless networks, and network security.

Ehsan Aryafar is an Associate Professor of Computer Science in the Maseeh College of Engineering and Computer Science at Portland State University. Prior to joining PSU, from 2013 to 2017, he was a Research Scientist at Intel Labs in Santa Clara, CA. He received his B.S. in Electrical Engineering from Sharif University of Technology, Iran, in 2005, and his M.S. and Ph.D. in Electrical and Computer Engineering from Rice University, Houston, Texas, in 2007 and 2011, respectively. From 2011 to 2013, he was a PostDoctoral Research Associate at Princeton University. His research interests are in the areas of wireless networks and networked systems, and span both algorithm design as well as system prototyping. He is a recipient of 2020 NSF CAREER award and 2023–2025 Maseeh college Wedge Vision Professorship award.