ML-Assisted Resource Allocation Outage Probability: Simple, Closed-Form Approximations

Rashika Raina*, *Student Member, IEEE*, Nidhi Simmons*, *Senior Member, IEEE*, David E. Simmons†, and Michel Daoud Yacoub‡, *Member, IEEE*

*Centre for Wireless Innovation, School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, BT3 9DT, UK, (e-mail: rraina01@qub.ac.uk, nidhi.simmons@qub.ac.uk)
†Dhali Holdings Ltd., Belfast BT5 7HW, UK, (e-mail: dr.desimmons@gmail.com)
‡Wireless Technology Laboratory, School of Electrical and Computer Engineering, University of Campinas, Campinas 13083-970, Brazil, (e-mail: mdyacoub@unicamp.br)

***Abstract*—In this paper, we establish simple and efficient approximations for the outage probability of a single-user multi-resource allocation system that consists of a machine learning (ML) based outage predictor whose task is to assign resources to the user while minimizing outages. We begin by presenting the outage probability expressions for this system. We then propose the approximations to this system's outage probability using both the sinc function and the zeroth-order Bessel function of the first kind. These approximations are based on naive upper and lower bounds and stem from understanding how the channel samples de-correlate over time. Our results demonstrate that the outage probability indeed lies within the range defined by the bounds. Moreover, the effectiveness of the proposed outage probability approximations are evident as they exhibit a strong alignment with the trend of the outage probability curve. Finally, because of the simplicity of our approximations, they can be calculated in a computationally efficient manner.**

***Index Terms*—Approximations, Custom loss function, Greedy resource allocation, Loss functions, Machine learning, Outage prediction, Outage probability, Resource allocation.**

I. INTRODUCTION

Wireless communication channels exhibit dynamic characteristics, influenced by surrounding environmental conditions and mobility, often causing radio link fluctuation and consequent deterioration. Machine Learning (ML) has emerged as a valuable tool for predicting quality-of-service degradation and enhancing the reliability of future wireless systems like 6G [1]. In this context, ML techniques are capable of utilizing relevant available data to predict outages and identify link blockages [2], [3]. They can analyze beamforming vector observations, channel metrics, geographical and visual information, among others [4]. The integration of computer vision with deep learning (DL) tools, for example, has been effective for blockage prediction in millimetre wave (mmWave) and terahertz (THz) systems, as shown in [5], [6]. Additionally, [7] showed that a deep neural network (DNN) could predict mmWave blockages with 90% accuracy by correlating user positions and data traffic demands to blockage status and beam index. In [8], a recurrent neural network with meta-learning was employed to overcome the challenges in obtaining training data that represents dynamic blockage characteristics.

Similarly, numerous studies have explored ML techniques to optimize radio resource allocation in future wireless communication systems. For example, [9], [10] employed deep reinforcement learning (DRL) for resource allocation in ultra-reliable low-latency communication edge networks and vehicle-to-vehicle communications, respectively, with the latter addressing both unicast and broadcast scenarios. In [11], a DRL-based approach was used for power and resource allocation in an orthogonal frequency division multiple access downlink scenario, aiming to minimize power consumption while considering reliability, latency, and data rate constraints.

While these studies demonstrate the utility of ML techniques in enhancing wireless system reliability, it is worth noting that they all employed off-the-shelf loss functions, such as binary cross entropy (BCE) or mean squared error, to train their ML models. This approach may yield subpar performance gains, as illustrated in [12], and potentially fail to meet the stringent demands of 6G applications. Critically, the unprecedented challenges in 6G wireless network design will necessitate integrating domain specific knowledge into ML algorithms [13], including into their loss functions.

Given this, there has been a growing emphasis on developing custom loss functions tailored to the specialized demands of particular communication systems. For instance, in [14], the authors proposed a DNN-based resource allocation algorithm for a cell-free massive multiple-input multiple-output network with hardware impairments. They introduced a novel loss function to maximize the sum rate while considering user power and fronthaul capacity constraints. In [15], a custom loss function was introduced for hybrid beamforming under the constraints of finite-precision phase shifters and power. In [16], the authors used an unsupervised DNN to optimize reflection coefficients of intelligent reflecting surfaces for both single-input single-output and multiple-input single-output systems. The custom loss function used for training aimed at minimizing spectral efficiency discrepancies.

A notable contribution presented in [12] introduced a custom loss function, specifically tailored for training ML models

This work was supported by the Royal Academy of Engineering (grant ref RF\201920\19\191).

in a resource allocation context, placing a primary emphasis on minimizing the outage probability. Unlike existing loss or custom loss functions mentioned previously, [12] focuses on optimizing learning at the tails of the channel's distribution, crucial for instances where outages occur rarely. Furthermore, when ML models were trained using this custom loss function, the study showed improvements in wireless system reliability by multiple orders of magnitude compared to those trained with conventional off-the-shelf functions. Thus, learning accurately from rare events during training is important, as neglecting them could compromise ML solutions, thereby failing to meet the rigorous reliability demands of 6G applications.

This contribution builds upon the work in [12], where the outage probability of an ML-assisted resource allocation system was derived. Specifically, we introduce simple approximations to this outage probability using the sinc function and the zeroth-order Bessel function of the first kind. These approximations are grounded in understanding how the channel samples de-correlate with time. Employing a Deep Q-Network (DQN) methodology, a class of ML networks not implemented in [12], we first show that the custom loss function in [12] provides superior outage probability performance when compared to BCE. We then demonstrate that the approximations formulated sit within the established bounds, closely track the trend of the outage probability curves, and can be obtained in a computationally efficient manner.

The remainder of the paper is organized as follows: Section II revisits the resource allocation strategy from [12] that incorporates an ML model. Section III presents the outage probability expressions and the custom loss function introduced in [12] for training the ML model. Section IV presents simple approximations to the system's outage probability. Section V showcases Monte Carlo simulations obtained by implementing a DQN and discusses the results of this paper. Section VI concludes the paper with some finishing remarks.

II. System Model

We revisit the system model from [12], which considered a single-user multi-resource system. Here, resource $j \in \mathsf{R}$ is assumed to have a fluctuating channel state that can be represented as a time series, $h_j(t) \in \mathbb{C}$. It is assumed that the channel states $h_j(t)$ and $h_{j'}(t)$ are independent and identically distributed (i.i.d.) for $j \neq j'$. The correlation between $h_j(t)$ and $h_j(t+l)$, with $l \in \mathbb{N}$, decreases as l increases, and approaches zero as $l \to \infty$. To model this de-correlation, it is assumed that the channel states $h_j(t)$ for all resources are obtained from the fast Fourier Transform (FFT) of a continually evolving tapped channel response. At each point in time t, this channel response is expressed as a vector of independent zero-mean complex Gaussian variables: $G(t) = [g_1(t), \ldots, g_{|\mathsf{R}|}(t)]$ where $g_i \sim \mathcal{CN}(0, \sigma^2)$. With $\theta_p \sim \text{Unif}(-\xi, \xi)$, the evolution of $G(t)$ is governed by

$$g_j(t) \prod_{p=1}^{l} e^{i\theta_p} = g_j(t+l). \tag{1}$$

Using this model, each element of the time-domain response undergoes a small and random phase shift with each time step. Thus, the channel coefficients for each resource, $h_j(t)$, gradually de-correlate with time. When $\xi = 0$, the channel remains static. As described in [12], this channel model is in-line with Clarke's 3D model [17], where the autocorrelation between successive channel samples is given by the sinc function.

The user possesses an ML model that learns these correlations to assist in the selection of a suitable resource. Let

$$H_j(t, \mathtt{k}) \triangleq [h_j(t-\mathtt{k}+1), \cdots, h_j(t)]^T \in \mathbb{C}^{\mathtt{k}} \tag{2}$$

represent the window of past samples of channel states for resource j. The ability to support the user's communication from time $t-\mathtt{k}+1$ to t of resource j can be determined by the capacity, $C(H_j(t, \mathtt{k})) \in \mathbb{R}^+$. For instance, for a quasi-static Gaussian channel with a unit average signal-to-noise ratio per sample, the capacity for resource j is given by [18, eq. (5.80)]

$$C(H_j(t, \mathtt{k})) = \sum_{i=1}^{\mathtt{k}} \log_2 \left(1 + |h_j(t-\mathtt{k}+i)|^2\right) \text{ bits/s/Hz}. \tag{3}$$

If the user's required communication rate is below this capacity, the resource is considered satisfactory; otherwise, it will be in outage. Furthermore, it is assumed that the user's required communication rate is determined by a threshold $\gamma_{\mathtt{th}}$. The outage probability of a single resource j is given as

$$P_1(\gamma_{\mathtt{th}}) \triangleq \mathbb{P}[C(H_j(t, \mathtt{k})) < \gamma_{\mathtt{th}}] \quad \forall j \in \mathsf{R}. \tag{4}$$

The user relies on an ML classification model for resource allocation. However, a major limitation is that the ML model lacks access to future channel state information and can only rely on historical data to predict resource outages. Here, the resource indexing starts at $j = 1$, and in the event of an outage prediction by the ML model, j is incremented by 1. The model takes the input vector $H_j(t, \mathtt{k})$ for each resource j, yielding an output in the range $[0, 1]$. This output classifies whether the subsequent l channel samples (i.e., $H_j(t+l, l)$) support outage-free communication for each resource $j \in \mathsf{R}$. Formally, $\mathsf{Q}(H_j(t, \mathtt{k}); \Theta) \in [0, 1]$, where Θ represents the model's parameters. When $\mathsf{Q}(H_j(t, \mathtt{k}); \Theta) > \mathtt{q}_{\mathtt{th}}$, where $\mathtt{q}_{\mathtt{th}}$ denotes the model's classification threshold, an outage is predicted for the next l samples.

The cumulative distribution function of the model's output is denoted as

$$F_{\mathsf{Q}}(x) \triangleq \mathbb{P}[\mathsf{Q}(H_j(t, \mathtt{k}); \Theta) \leq x] \quad \forall j \in \mathsf{R}. \tag{5}$$

The model's resource acceptance probability, denoted as $F_{\mathsf{Q}}(\mathtt{q}_{\mathtt{th}})$, is obtained by setting x equal to the classification threshold $\mathtt{q}_{\mathtt{th}}$, while the model's resource rejection probability is given by $1 - F_{\mathsf{Q}}(\mathtt{q}_{\mathtt{th}})$. The model sequentially scans the available resources $\mathsf{R} = \{1, 2, \cdots, |\mathsf{R}|\}$, stopping when it predicts successful communication for a resource. If none of the resources satisfy this condition, the model selects the final resource. In summary, the described resource allocation procedure can be outlined as follows. Let

$$\mathsf{R}^\star(\mathtt{q}_{\mathtt{th}}) = \{j \in \mathsf{R} \text{ s.t. } \mathsf{Q}(H_j(t, \mathtt{k}); \Theta) \leq \mathtt{q}_{\mathtt{th}}\}, \tag{6}$$

be the subset of resources for which the ML model predicts no outages, then the greedy scheme selects

$$j^\star(\mathtt{q_{th}}) = \begin{cases} \min\limits_{j \in \mathsf{R}^\star(\mathtt{q_{th}})} j & \text{if } \mathsf{R}^\star(\mathtt{q_{th}}) \neq \emptyset \\ |\mathsf{R}| & \text{otherwise.} \end{cases} \tag{7}$$

The outage probability for a system with $|\mathsf{R}|$ resources naturally depends on the model's classification threshold, rate threshold, and ξ (the parameter of θ_p in (1)). We denote this as $P_{|\mathsf{R}|}\left(\gamma_{\mathtt{th}}, \mathtt{q_{th}}, \xi\right)$. This allows us to write the outage probability in the infinite resource limit as

$$P_\infty\left(\gamma_{\mathtt{th}}, \mathtt{q_{th}}, \xi\right) \triangleq \lim_{|\mathsf{R}|\to\infty} P_{|\mathsf{R}|}\left(\gamma_{\mathtt{th}}, q_{th}, \xi\right). \tag{8}$$

For conciseness in subsequent notation, we may omit the model's dependency on $H_j(t, \mathtt{k})$ and Θ where suitable, and similarly, the dependency of H_j on t and $\mathtt{k}$.

III. Expressions for the Outage Probability and Custom Loss Function

In this section, we first provide the outage probability expressions for the system discussed earlier. We then present a custom loss function that approximates the outage probability while maintaining essential differentiable properties. This custom loss function is used to train the ML model to generate the results obtained in Section V. Detailed derivations of these expressions can be found in [12, Theorems 1 and 2].

A. Expressions

The outage probability of the resource allocation system (discussed in section II) with $|\mathsf{R}|$ resources can be expressed as [12, eq. (13) and (15)]

$$\begin{aligned} P_{|\mathsf{R}|}(\gamma_{\mathtt{th}}, \mathtt{q_{th}}, \xi) = P_1(\gamma_{\mathtt{th}})\left(1 - F_{\mathsf{Q}}(\mathtt{q_{th}})\right)^{|\mathsf{R}|-1} \\ + P_\infty(\gamma_{\mathtt{th}}, \mathtt{q_{th}}, \xi)\Big(1 - (1 - F_{\mathsf{Q}}(\mathtt{q_{th}}))^{|\mathsf{R}|-1}\Big), \end{aligned} \tag{9}$$

where

$$P_\infty\left(\gamma_{\mathtt{th}}, \mathtt{q_{th}}, \xi\right) = \mathbb{P}\left[C(H_j) < \gamma_{\mathtt{th}} | \mathsf{Q}\left(H_j\right) \leq \mathtt{q_{th}}\right], \tag{10}$$

while $P_1\left(\gamma_{\mathtt{th}}\right)$, $F_{\mathsf{Q}}\left(\mathtt{q_{th}}\right)$, $\mathsf{Q}\left(H_j\right)$, $C(H_j)$, $\gamma_{\mathtt{th}}, \mathtt{q_{th}}$ and ξ are defined previously. See [12, Theorem 1] for proof. As per [12, Theorem 2] the custom loss function for the considered resource allocation system can be formulated as follows:

$$\begin{aligned} \widehat{P}_{|\mathsf{R}|}\left(W_n\right) \triangleq\ & \widehat{P}_1\left(W_n\right)\Big(1 - \widehat{F}_{\mathsf{Q}}\left(W_n\right)\Big)^{|\mathsf{R}|-1} \\ & + \widehat{P}_\infty\left(W_n\right)\left(1 - \Big(1 - \widehat{F}_{\mathsf{Q}}\left(W_n\right)\Big)^{|\mathsf{R}|-1}\right), \end{aligned} \tag{11}$$

where $W_n = \{(H_j, b_j)\}$, $b_j \in \{0, 1\}$, represents a collection of n labelled samples drawn uniformly and independently from the channel model. A label of $b_j = 1$ indicates that an outage event will follow H_j, whereas $b_j = 0$ indicates that no outage event will follow H_j. The constituents of (11) are given by

$$\widehat{P}_1(W_n) \triangleq \frac{\mathrm{TP}(W_n) + \mathrm{FN}(W_n)}{\mathrm{TN}(W_n) + \mathrm{FN}(W_n) + \mathrm{TP}(W_n) + \mathrm{FP}(W_n)}, \tag{12}$$

$$\widehat{F}_{\mathsf{Q}}(W_n) \triangleq \frac{\mathrm{TN}(W_n) + \mathrm{FN}(W_n)}{\mathrm{TN}(W_n) + \mathrm{FN}(W_n) + \mathrm{TP}(W_n) + \mathrm{FP}(W_n)} \tag{13}$$

and

$$\widehat{P}_\infty(W_n) \triangleq \frac{\mathrm{FN}(W_n)}{\mathrm{TN}(W_n) + \mathrm{FN}(W_n)}, \tag{14}$$

where the differentiable approximations to TN (true negative), FN (false negative), TP (true positive), and FP (false positive) are defined as follows:

$$\mathrm{TN}\left(W_n\right) \triangleq \sum_{(H_j, b_j) \in W_n} \phi_\alpha\left(\mathtt{q_{th}} - \mathsf{Q}\left(H_j\right)\right)\left(1 - b_j\right) \tag{15}$$

$$\mathrm{FN}\left(W_n\right) \triangleq \sum_{(H_j, b_j) \in W_n} \phi_\alpha\left(\mathtt{q_{th}} - \mathsf{Q}\left(H_j\right)\right) b_j \tag{16}$$

$$\mathrm{TP}\left(W_n\right) \triangleq \sum_{(H_j, b_j) \in W_n} \phi_\alpha\left(\mathsf{Q}\left(H_j\right) - \mathtt{q_{th}}\right) b_j \tag{17}$$

$$\mathrm{FP}\left(W_n\right) \triangleq \sum_{(H_j, b_j) \in W_n} \phi_\alpha\left(\mathsf{Q}\left(H_j\right) - \mathtt{q_{th}}\right)\left(1 - b_j\right). \tag{18}$$

Here, $\phi_\alpha(x) = 1/(1 + e^{-\alpha x})$ is a logistic function.

IV. Outage Probability Approximations

This section proposes simple approximations to the outage probability of the resource allocation system presented in section III, which are based on a theoretical understanding of how channel samples de-correlate with time.

Recall from (1) that $g_j(t)$ and $g_j(t + l)$, with $l \in \mathbb{N}$, relate to each other as

$$g_j\left(t\right) \prod_{p=1}^{l} e^{i\theta_p} = g_j\left(t + l\right), \tag{19}$$

where $\theta_p \sim \mathrm{Unif}(-\xi, \xi)$. In the absence of phase shift (i.e., when ξ = 0) the channel does not de-correlate and the outage event is predicted with certainty. Following a trivial strategy, the ML algorithm assigns an outage prediction only when the previous samples are in outage. In this case,

$$P_{|\mathsf{R}|}(\gamma_{\mathtt{th}}, \mathtt{q_{th}}, \xi)|_{\xi=0} = P_1(\gamma_{\mathtt{th}})^{|\mathsf{R}|}. \tag{20}$$

As $\xi \to \infty$, the ML model's performance degrades to random guessing. Excessive and noisy phase shifts hinder the model's predictive ability, leading to performance comparable to the outage probability of a single resource system i.e.,

$$P_{|\mathsf{R}|}(\gamma_{\mathtt{th}}, \mathtt{q_{th}}, \xi)|_{\xi\to\infty} = P_1(\gamma_{\mathtt{th}}). \tag{21}$$

From (20) and (21), it is clear that the outage probability lies between $P_1(\gamma_{\mathtt{th}})$ and $P_1(\gamma_{\mathtt{th}})^{|\mathsf{R}|}$. Thus, we represent it as a linear combination of these probabilities:

$$\begin{aligned} P_{|\mathsf{R}|}(\gamma_{\mathtt{th}}, \mathtt{q_{th}}, \xi) = P_1(\gamma_{\mathtt{th}})^{|\mathsf{R}|} f(\gamma_{\mathtt{th}}, \mathtt{q_{th}}, \xi) \\ + P_1(\gamma_{\mathtt{th}})(1 - f(\gamma_{\mathtt{th}}, \mathtt{q_{th}}, \xi)), \end{aligned} \tag{22}$$

where $f(\gamma_{\mathtt{th}}, \mathtt{q_{th}}, \xi) \in [0, 1]$ is some unknown function satisfying $f(\cdot, \cdot, 0) = 1$ and $f(\cdot, \cdot, \infty) = 0$. This ensures that (20) and (21) are satisfied.

For $\beta > 0$, possible approximations for f satisfying these are

$$f(\cdot, \cdot, \xi) \approx |\mathrm{sinc}(\xi)^{\beta l}| \tag{23}$$

and

$$f(\cdot,\cdot,\xi) \approx \left|\left(J_0\left(\xi\right)\right)^{\beta l}\right|, \tag{24}$$

the zeroth-order Bessel function of the first kind [19, Section 8.402]. Indeed, these choices for $f(\cdot,\cdot,\cdot)$ are not arbitrary. By observing Lemma 1 below we see that the covariance between time shifted channel coefficients decays with $\mathrm{sinc}(\xi)^l$. As previously stated, this observation demonstrates that our data generation aligns with Clarke's 3D model [17]. Similarly, the zeroth-order Bessel function of the first kind exhibits comparable autocorrelation behaviour in Clarke's 2D model [20], prompting us to explore its suitability for the 3D scenario.

Lemma 1. *With $g_j(t)$ satisfying* (1)*, the covariance of $g_j(t)$ and $g_j(t+l)$ is given by*

$$\mathrm{Cov}\left(g_j(t), g_j(t+l)\right) = \mathbb{E}\left[\left|g_j\left(t\right)\right|^2\right] \mathrm{sinc}\left(\xi\right)^l. \tag{25}$$

Proof: See Appendix A. ∎

V. Monte Carlo Simulations

This section demonstrates the outage performance of the resource allocation system (described in section II) as the channel samples de-correlate with time. The results illustrate that our approximations closely mirror the trends of the system's actual outage probability.

A. Training

A DQN [21] incorporating a long short-term memory (LSTM) layer is trained on a simulated wireless channel dataset using

1) the custom loss function (when $\mathtt{q_{th}} = 0.5$ and $\alpha = 10$) and a traditional loss function, BCE;
2) TensorFlow's Keras API and an ADAM optimiser.

To ensure a balance between exploitation and exploration during training, we employed an epsilon decaying strategy. Table I presents the values of the hyperparameters used in training the DQN-LSTM model, which were chosen based on empirical experiments. Additionally, we have integrated the early stopping technique into our training process to prevent overfitting. This technique concludes the training if there is no improvement in the performance over consecutive epochs. The subsequent results are obtained from independently generated test datasets, which helps mitigate potential model overfitting and its impact on outcomes. On a desktop computer equipped with a 12th Gen Intel(R) Core(TM) i9-12900HK, 2500 MHz, 14 Core(s) and 20 Logical processor(s), the training of a single DQN-LSTM model takes approximately 3 minutes. Each data point in the figures represents the average performance obtained after retraining the ML model 10 times. Furthermore, for each individual run, the model evaluates its performance on 9000 separate instances (i.e., tests), to generate results. To allow the reader to reproduce these results, the supporting software is made available here [22]. As per Section II, the training/testing dataset for the channel is synthetically obtained as follows:

TABLE I
Parameters of the DQN-LSTM model

Parameter	Value
Learning rate	0.001
Discount factor	0.9
Hidden units	32
Epochs	20
Epoch size	150
Input sequence length	100
Output sequence length	10
Output dimension	1

1) Time-domain response generation: we generate a channel impulse response vector of $\nu = 1024$ zero-mean complex Gaussian i.i.d. variables at $t = 0$.
2) Introduce per-element phase shifts: each time-domain response element has a uniformly distributed phase shift in the range $\pm\,\xi$ radians applied (for figures in this paper ξ was varied from 0.01 to 0.9).
3) Frequency response generation: we perform an FFT on the impulse response vector, giving a set of 1024 frequency domain samples.
4) Extract channel coefficients: for each frequency domain vector, obtain $|\mathtt{R}| \leq 1024$ equally spaced samples for each resource at a fixed point in time.
5) Repeat 1) to 4) over $\mathtt{k} + l$ time increments, obtaining a sequence of $\mathtt{k} + l$ channel coefficients for each resource.

In our simulations, $\mathtt{k}$ and l have been specified as 100 and 10, respectively. The initial $\mathtt{k}$ samples serve as input to the DQN-LSTM, while the last l samples create a label. The label is determined by comparing the corresponding capacity $C(\cdot)$ (3) with the rate threshold, $\gamma_{\mathtt{th}}$. If $C(\cdot) \geq \gamma_{\mathtt{th}}$, a label of 0 is assigned and if $C(\cdot) < \gamma_{\mathtt{th}}$, a label of 1 is assigned.

The LSTM component captures sequential dependencies in the input data, and the DQN's action mechanism is utilized to predict class labels for the input sequences. The key elements of the DQN-LSTM are defined below:

1) State: Refers to sequential segments of input data, namely the input vector $H_j(t, \mathtt{k})$ associated with resource j.
2) Action: Determined by the exploration rate, the action is either chosen randomly (exploration) or based on the model's prediction of the next state value (exploitation). Each action corresponds to a predicted value for the given input sequence.
3) Rewards: A value of 1 is allocated if the predicted next state matches the true next state, and a reward of -1 is assigned otherwise.

It is worth highlighting that implementing the above procedure using LSTM yields similar results to DQN-LSTM. This is attributed to their similar architectures, predictive capabilities, and input representations.

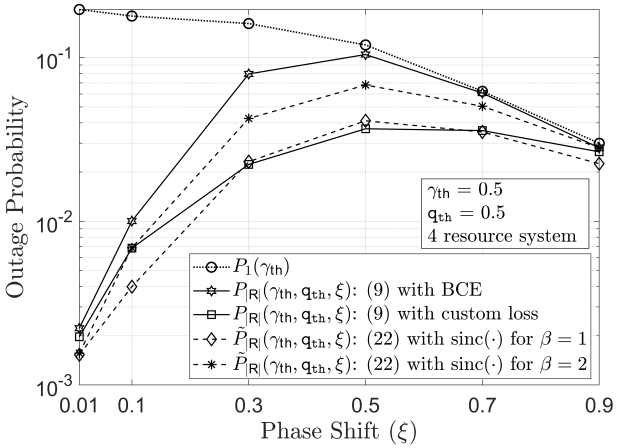

Fig. 1. Monte Carlo simulations for the outage probability of a four resource system $P_{|\mathsf{R}|}(\gamma_{\mathtt{th}}, \mathtt{q}_{\mathtt{th}}, \xi)$ using (9), versus the approximation $\tilde{P}_{|\mathsf{R}|}(\gamma_{\mathtt{th}}, \mathtt{q}_{\mathtt{th}}, \xi)$ for a range of phase shifts using (22) with sinc(·) (23), for $\beta = 1$ and 2 while employing the custom loss function and BCE.

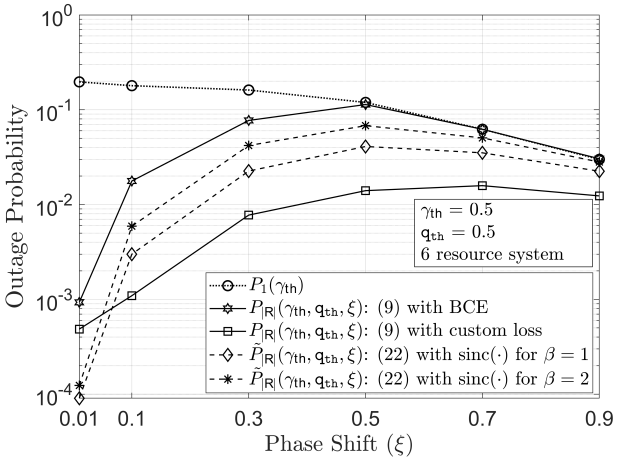

Fig. 2. Monte Carlo simulations for the outage probability of a six resource system $P_{|\mathsf{R}|}(\gamma_{\mathtt{th}}, \mathtt{q}_{\mathtt{th}}, \xi)$ using (9), versus the approximation $\tilde{P}_{|\mathsf{R}|}(\gamma_{\mathtt{th}}, \mathtt{q}_{\mathtt{th}}, \xi)$ for a range of phase shifts using (22) with sinc(·) (23), for $\beta = 1$ and 2 while employing the custom loss function and BCE.

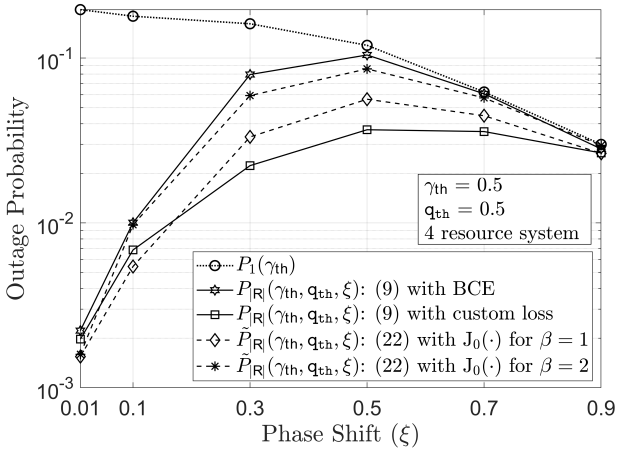

Fig. 3. Monte Carlo simulations for the outage probability of a four resource system $P_{|\mathsf{R}|}(\gamma_{\mathtt{th}}, \mathtt{q}_{\mathtt{th}}, \xi)$ using (9), versus the approximation $\tilde{P}_{|\mathsf{R}|}(\gamma_{\mathtt{th}}, \mathtt{q}_{\mathtt{th}}, \xi)$ for a range of phase shifts using (22) with $\mathrm{J}_0(\cdot)$ (24) for $\beta = 1$ and 2 while employing the custom loss function and BCE.

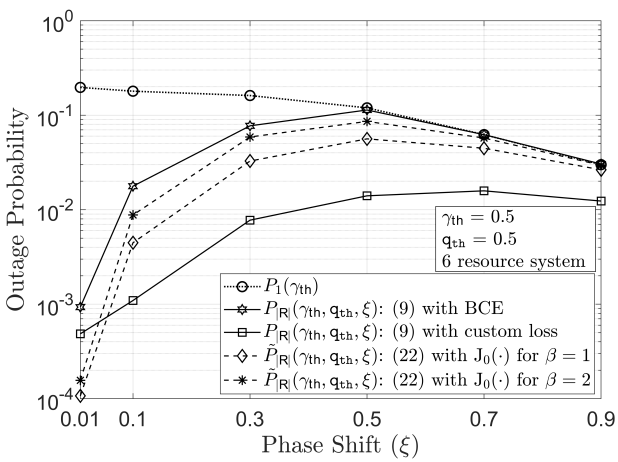

Fig. 4. Monte Carlo simulations for the outage probability of a six resource system $P_{|\mathsf{R}|}(\gamma_{\mathtt{th}}, \mathtt{q}_{\mathtt{th}}, \xi)$ using (9), versus the approximation $\tilde{P}_{|\mathsf{R}|}(\gamma_{\mathtt{th}}, \mathtt{q}_{\mathtt{th}}, \xi)$ for a range of phase shifts using (22) with $\mathrm{J}_0(\cdot)$ (24) for $\beta = 1$ and 2 while employing the custom loss function and BCE.

B. Results

Figs. 1 and 2 illustrate the outage performance of a four and six resource system, respectively, when using the BCE and custom loss functions, for a classification threshold $\mathtt{q}_{\mathtt{th}}$ of 0.5 and a rate threshold $\gamma_{\mathtt{th}}$ of 0.5 across varying phase shifts ξ. It can be seen that, as the phase shift increases (uncertainty intensifies), the outage performance approaches its upper bound, i.e., (21), which is expected because it becomes more difficult for the ML model to make valid predictions. Figs. 1 and 2 also show the outage probability of the system using (22) with the proposed sinc function approximation (23) with $\beta = 1$ and 2. We observe that these approximations perform favourably, accurately tracing the trend of the outage probability curves.

Figs. 3 and 4 show the outage probability of the system using the proposed approximation involving $\mathrm{J}_0(\cdot)$ (24) instead of $\mathrm{sinc}(\cdot)$, with $\beta = 1$ and 2. Again, these approximations follow well the trend of the outage probability curves, offering a foundation for further analytical exploration into their computational efficacy and applicability. Notably, these mathematical approximations are not only analytically tractable but also can be effortlessly computed utilizing the parameters intrinsic to the channel. Particularly, formulating estimates for the efficacy of an ML resource assigner does not require the training of an ML model, thereby facilitating a streamlined analysis. Also, we do not draw any definite conclusions regarding the superiority of one approximation over another. This is because, different approximations perform well under different phase shift values in both four and six resource systems. Crucially, however, it remains important to highlight that all explored

approximations conform to the trajectory of the underlying outage probability.

VI. CONCLUSIONS

This paper introduced simple outage probability approximations for an ML-assisted resource allocation system using the sinc and zeroth-order Bessel function of the first kind. These approximations were based on a theoretical understanding of how channel samples de-correlated over time. A DQN-LSTM approach was implemented to show that the system's outage probability provided superior performance when using the custom loss function compared to BCE. Furthermore, our experimental findings demonstrated that the proposed approximations closely followed the actual outage probability and were also computationally straightforward to calculate since no ML models were needed to be trained for this.

APPENDIX

A. Proof of Lemma 1

We have $g_j \sim \mathcal{CN}\left(0, \sigma^2\right)$ and $g_j(t+l) = g_j\left(t\right) \prod_{p=1}^{l} e^{i\theta_p}$, where $\theta_p \sim \text{Unif}(-\xi, \xi)$. Then,

$$\begin{aligned}
\text{Cov}(g_j(t), g_j(t+l)) &= \mathbb{E}\left[g_j\left(t\right)\overline{g_j\left(t+l\right)}\right] \\
&\quad - \mathbb{E}\left[g_j\left(t\right)\right]\mathbb{E}\left[\overline{g_j\left(t+l\right)}\right] && (26)\\
&= \mathbb{E}\left[g_j\left(t\right)\overline{g_j\left(t\right)}\prod_{p=1}^{l} e^{i\theta_p}\right] && (27)\\
&= \mathbb{E}\left[\left|g_j\left(t\right)\right|^2\right]\mathbb{E}\left[\prod_{p=1}^{l} e^{i\theta_p}\right] && (28)\\
&= \mathbb{E}\left[\left|g_j\left(t\right)\right|^2\right]\prod_{p=1}^{l}\mathbb{E}\left[e^{i\theta_p}\right] && (29)\\
&= \mathbb{E}\left[\left|g_j\left(t\right)\right|^2\right]\prod_{p=1}^{l}\int_{-\xi}^{\xi} e^{i\theta_p}\frac{1}{2\xi}d\theta_p && (30)\\
&= \mathbb{E}\left[\left|g_j\left(t\right)\right|^2\right]\prod_{p=1}^{l}\frac{1}{\xi}\left[\frac{e^{i\xi}-e^{-i\xi}}{2i}\right] && (31)\\
&= \mathbb{E}\left[\left|g_j\left(t\right)\right|^2\right]\prod_{p=1}^{l}\text{sinc}\left(\xi\right) && (32)\\
&= \mathbb{E}\left[\left|g_j\left(t\right)\right|^2\right]\text{sinc}\left(\xi\right)^l && (33)
\end{aligned}$$

where (32) follows from the definition of the characteristic function. This concludes the proof.

REFERENCES

[1] H. Yang, A. Alphones, Z. Xiong, D. Niyato, J. Zhao, and K. Wu, "Artificial-Intelligence-Enabled Intelligent 6G Networks," *IEEE Network*, vol. 34, no. 6, pp. 272–280, Oct. 2020.

[2] S. Wu, M. Alrabeiah, A. Hredzak, C. Chakrabarti, and A. Alkhateeb, "Deep Learning for Moving Blockage Prediction using Real mmWave Measurements," in *IEEE International Conference on Communications (ICC)*, Seoul, South Korea, May 2022, pp. 3753–3758.

[3] H. Sarieddeen, N. Saeed, T. Y. Al-Naffouri, and M.-S. Alouini, "Next Generation Terahertz Communications: A Rendezvous of Sensing, Imaging, and Localization," *IEEE Communications Magazine*, vol. 58, no. 5, pp. 69–75, May 2020.

[4] A. Alkhateeb, I. Beltagy, and S. Alex, "Machine Learning for Reliable mmWave Systems: Blockage Prediction and Proactive Handoff," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Anaheim, CA, Feb. 2018, pp. 1055–1059.

[5] M. Alrabeiah, A. Hredzak, and A. Alkhateeb, "Millimeter Wave Base Stations with Cameras: Vision-Aided Beam and Blockage Prediction," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, May. 2020, pp. 1–5.

[6] G. Charan, M. Alrabeiah, and A. Alkhateeb, "Vision-Aided 6G Wireless Communications: Blockage Prediction and Proactive Handoff," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 10 193–10 208, Oct. 2021.

[7] S. Moon, H. Kim, Y.-H. You, C. H. Kim, and I. Hwang, "Online Learning-Based Beam and Blockage Prediction for Indoor Millimeter-Wave Communications," *ICT Express*, vol. 8, no. 1, pp. 1–6, Mar. 2022.

[8] A. E. Kalør, O. Simeone, and P. Popovski, "Prediction of mmWave/THz Link Blockages Through Meta-Learning and Recurrent Neural Networks," *IEEE Wireless Communications Letters*, vol. 10, no. 12, pp. 2815–2819, Dec. 2021.

[9] T. Yang, Y. Hu, M. C. Gursoy, A. Schmeink, and R. Mathar, "Deep Reinforcement Learning based Resource Allocation in Low Latency Edge Computing Networks," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, Lisbon, Portugal, Aug 2018, pp. 1–5.

[10] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep Reinforcement Learning based Resource Allocation for V2V Communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3163–3173, April 2019.

[11] A. T. Z. Kasgari and W. Saad, "Model-Free Ultra Reliable Low Latency Communication (URLLC): A Deep Reinforcement Learning Framework," in *IEEE International Conference on Communications (ICC)*, Shanghai, China, May 2019, pp. 1–6.

[12] N. Simmons, D. E. Simmons, and M. D. Yacoub, "Outage Performance and Novel Loss Function for an ML-Assisted Resource Allocation: An Exact Analytical Framework," *arXiv 2305.09739v2 (eess.SP)*, Nov. 2023.

[13] C. She, C. Sun, Z. Gu, Y. Li, C. Yang, H. V. Poor, and B. Vucetic, "A Tutorial on Ultrareliable and Low-Latency Communications in 6G: Integrating Domain Knowledge Into Deep Learning," *Proceedings of the IEEE*, vol. 109, no. 3, pp. 204–246, Mar. 2021.

[14] N. Rajapaksha, K. B. S. Manosha, N. Rajatheva, and M. Latva-aho, "Unsupervised Learning-Based Joint Power Control and Fronthaul Capacity Allocation in Cell-Free Massive MIMO With Hardware Impairments," *IEEE Wireless Communications Letters*, vol. 12, no. 7, pp. 1159–1163, April 2023.

[15] T. Peken, S. Adiga, R. Tandon, and T. Bose, "Deep Learning for SVD and Hybrid Beamforming," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6621–6642, June 2020.

[16] K. Utkarsh, Ashish, and P. Kumar, "Transmit Power Reduction in an IRS Aided Wireless Communication System using DNN," in *2023 International Conference on Microwave, Optical, and Communication Engineering (ICMOCE)*, Bhubaneswar, India, May 2023, pp. 1–5.

[17] R. Clarke and W. L. Khoo, "3-D Mobile Radio Channel Statistics," *IEEE Transactions on Vehicular Technology*, vol. 46, no. 3, pp. 798–799, 1997.

[18] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. USA: Cambridge University Press, 2005.

[19] A. Jeffrey, D. Zwillinger, I. Gradshteyn, and I. Ryzhik, *Table of Integrals, Series, and Products*, 7th ed. Academic Press, 2007.

[20] R. H. Clarke, "A Statistical Theory of Mobile-Radio Reception," *Bell system technical journal*, vol. 47, no. 6, pp. 957–1000, 1968.

[21] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-Level Control through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[22] "Code for ML-Assisted Resource Allocation Outage Probability: Simple, Closed-Form Approximations," Nov. 2023, https://github.com/ML4Comms/greedy-resource-allocation-outage-classification.