VoglerNet: multiple knife-edge diffraction using deep neural network

Viet-Dung Nguyen*, Huy Phan†, Ali Mansour*, Arnaud Coatanhay*
*Lab-STICC, UMR6285 CNRS ENSTA Bretagne, Brest, France
†School of Computing, University of Kent, Chatham Maritime, United Kingdom
viet.nguyen, ali.mansour, arnaud.coatanhay@ensta-bretagne.fr, *h.phan@kent.ac.uk*

***Abstract*—Multiple knife-edge diffraction estimation is a fundamental problem in wireless communication. One of the most well-known algorithm for predicting diffraction is Vogler algorithm which has been shown to reach the state-of-the-art results in both simulation and measurement experiments. However, it can not be easily used in practice due to its high computational complexity. In this paper, we propose VoglerNet, a data-driven diffraction estimator, by converting the Vogler algorithm into a deep neural network based system. To train VoglerNet, we propose to minimize a regularized loss function using Levenberg-Marquardt backpropagation in conjunction with a Bayesian regularization. Our numerical experiments show that VoglerNet provides fast solution in order of milliseconds while its performance is very close to that of the classical Vogler algorithm.**

***Index Terms*—multiple knife edge diffraction, Vogler algorithm, deep neural network, deep learning, Levenberg-Marquardt backpropagation, wireless communication.**

I. Introduction

An accurate estimation of the diffraction attenuation is a fundamental problem in evaluating the propagation loss over irregular terrains [1], [2], aeronautical mobiles and ground station interactions [3], and channel modeling at cmWave or mmWave bands in indoor scenarios [4], to name a few. Consider predicting the propagation loss over irregular terrain as an example, a terrain model over the propagation path is essential and can be characterized or 'approximated' by knife-edges. In general, using a single knife-edge approximation is simple but unsatisfactory, thus requiring multiple knife-edges to obtain better accuracy.

In this paper, we consider the problem of estimating multiple knife-edge diffraction. So far, in the literature, there are many methods proposed to solve this problem. Due to limited space, we only present here some representatives. Generally, we can categorize existing methods into two groups by their computational complexity and characteristics: (i) The first group consists algorithms that provide precise results but suffer from high computational complexity. The well-known example of this group is the Vogler algorithm [5] which can be derived from the Fresnel-Kirchhoff theory. The obtained result is a multiple integral which can be computed practically in terms of series representation. The initial version of Vogler algorithm presented first method for computing more than two knife-edges. Moreover, it can yield accurate results up to ten knife-edges. Other important representatives with many variants [6]–[10] are based on the uniform theory of diffraction (UTD). The common characteristic of this class is the necessity to compute higher order UTD diffracted fields if a high precision is required. In contrast, the calculation of the algorithms in the second group are efficient but its accuracy is inadequate. Famous examples include the Epstein/Peterson method [11], the Deygout method [12], the Causebrook method [13] and the Giovanelli method [14]. Those algorithms are graphical-based methods and can be seen as an approximate multiple knife-edge diffraction. We note that the original works of those algorithms were limited for single or double knife-edges. However, it is possible to extend such methods to multiple knife-edge scenarios. The computation in their extended version is still relatively simple comparing to that of the algorithms in the first group. Thus, for time-sensitivity applications, it is important to provide a solution exploiting the benefits of both the groups.

In recent years, there has been an increasing interest in application of deep learning based methods because of empirical success on diverse fields such as computer vision, image processing, or natural language processing [15], [16]. This approach offers two attractive features: First, if the underlying process of model is complicated or it is hard to estimate parameters of that model, a deep learning approach can be an alternative solution or even better solution to the model based approach. Second, by experiments, deep learning methods often provide better results than the shallow ones due to their high-capacities. We note that this might be reached providing sufficient data.

To address the above-mentioned problems and take advantage of deep learning approaches, the main contribution of this paper is to recasting the Vogler algorithm into "VoglerNet", a system based on deep neural network (DNN), for tackling multiple knife-edge diffraction problem. To the best of our knowledge, this is a pioneer approach to solve this fundamental problem. Our motivation stems from the fact that DNN is data-driven and suitable approach for complicated underlying process which is the case for the Vogler method. The main advantage of the proposed approach is that our solution is practical for time-sensitivity applications, while its accuracy is close to the Vogler method. We also show by simulations that DNN is essential since the performance of a shallow neural network (SNN) is unsatisfied. The superior of DNN to SNN is due to the fact that DNN is high-capacity model which permits representing more complicated processes than SNN.

II. Vogler algorithm

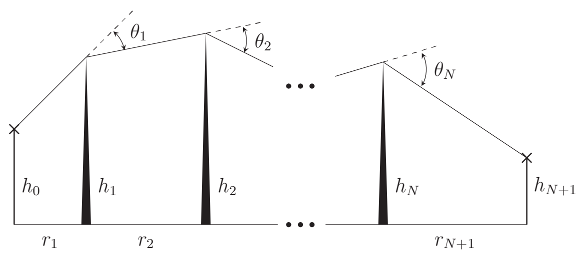

Fig. 1: Geometry of multiple knife edge.

We consider the geometry of N knife-edge diffraction ($N \leq 10$) in Fig.1 where $\{h_n\}_{n=1}^{N}$ are the knife-edge heights to a reference surface, $\{\theta_n\}_{n=1}^{N}$ are diffraction angles, and $\{r_n\}_{n=1}^{N+1}$ are $N+1$ separation distances between knife-edges. We use h_0 and h_{N+1} to denote the transmitter and receiver heights respectively. Then the diffraction attenuation, A, is given by [5]:

$$A_N = \frac{1}{2^N} C_N \exp(\sigma_N) \left(\frac{2}{\sqrt{\pi}}\right)^N \underbrace{\int_{\beta_1}^{\infty} \cdots \int_{\beta_N}^{\infty}}_{N-\text{fold}} \exp\left(2F\left(\{u_n\}_{n=1}^{N}, \{\beta_n\}_{n=1}^{N}\right)\right) \prod\nolimits_{n=1}^{N} \exp\left(-u_n^2\right) du_n \tag{1}$$

where

$$C_N = \begin{cases} 1 \quad \text{for } N = 1 \\ \sqrt{\frac{\left(\sum_{n=1}^{N+1} r_n\right) \prod_{n=1}^{N} r_n}{\prod_{n=1}^{N} (r_n + r_{n+1})}} \quad , \ N \geq 2 \end{cases}, \tag{2}$$

$$\sigma_N = \sum\nolimits_{n=1}^{N} \beta_n^2, \tag{3}$$

$$F = \begin{cases} 0 \quad \text{for } N = 1 \\ \sum_{n=1}^{N-1} \alpha_n (u_n - \beta_n)(u_{n+1} - \beta_{n+1}) \quad , \ N \geq 2, \end{cases} \tag{4}$$

$$\alpha_n = \left[\frac{r_n r_{n+1}}{(r_n + r_{n+1})(r_{n+1} + r_{n+2})}\right]^{1/2}, \ 1 \leq n \leq N-1, \tag{5}$$

$$\beta_n = \theta_n \left[\frac{jk r_n r_{n+1}}{2(r_n + r_{n+1})}\right]^{1/2}, \ 1 \leq n \leq N. \tag{6}$$

For computing β_n, the following angle approximation is used

$$\theta_n \approx \frac{h_n - h_{n-1}}{r_n} + \frac{h_n - h_{n+1}}{r_{n+1}}, \ n = 1, \cdots, N. \tag{7}$$

To evaluate (1), the key idea is that instead of computing the N-fold integral, we convert such task into computing N single integrals. To this end, Vogler proposed to express $\exp(2F)$ in terms of power series as:

$$\exp(2F) = \sum\nolimits_{m=0}^{\infty} \frac{(2F)^m}{m!}. \tag{8}$$

Then by exploiting the fact that

$$\frac{2}{\sqrt{\pi}} \int_{\beta}^{\infty} (u - \beta)^m \exp\left(-u^2\right) du = m! I(m, \beta) \tag{9}$$

where $I(m, \beta)$ refers to the repeated integrals of the complementary error function, we obtain a residual series form solution of A_N.

III. Proposed VoglerNet approach

A. VoglerNet system

Consider the proposed approach as shown in Fig. 2 where we want to estimate diffraction attenuation values of several path profiles from a specific terrain. For simplicity, we further assume that queried parameters belong to the range of terrain parameters.

We can divide the proposed approach into two parts. In the first part, a real-life profile of interest is first approximated[1] by N knife-edges to obtain the preferred parameters such as knife-edge heights and their separation distances. Those parameters as well as antenna heights and operation frequency are then fed to a processing center (the second part) to obtain the corresponding predicted diffraction values.

In the second part, given terrain parameters of interest such as the minimum and maximum heights of the terrain, antenna parameters, the minimum and maximum distance between knife-edges and the intended frequency, we will generate synthetic path profiles to cover the terrain. Then, those profiles are used for training the proposed deep neural network. We note that the second part is implemented offline and can be prepared to cover in advance the terrain of interest. When a new query with parameters is arrived, the process immediately returns the predicted result, thus preserving the efficiency of the system. We now describe the key component of the second part, how to obtain optimal weights $\mathbf{W}$ and bias $\mathbf{b}$ of the proposed deep neural network architecture.

B. Deep neural network

Consider a deep neural network with n-input, and 1-output and L layers, we can represent the architecture (see Fig. 4) in terms of the mathematical formula as follows

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)} \mathbf{y}^{(l-1)} + \mathbf{b}^{(l)}, \tag{10}$$

$$\mathbf{y}^{(l)} = g\left(\mathbf{z}^{(l)}\right), \tag{11}$$

where $l \in [1, L]$ and g is an activation function. Generally, different activation functions, such as signmoid, can be used for each layer. Following this notation, $\mathbf{y}^{(0)} = \mathbf{x}$ for the input layer. Thus, all parameters of DNN can be summarized as

$$\boldsymbol{\theta}_{\text{DNN}} = \left\{\mathbf{W}^{(1)}, \cdots, \mathbf{W}^{(L)}, \mathbf{b}^{(1)}, \cdots, \mathbf{b}^{(L)}\right\} \tag{12}$$

with $\mathbf{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}, \mathbf{b}^{(l)} \in \mathbb{R}^{d_l}$. Now, the objective is to train the network to minimize the loss function $\mathcal{L}$ over N_t training data pairs $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_t}$

$$\widehat{\boldsymbol{\theta}}_{\text{DNN}} = \underset{\boldsymbol{\theta}_{\text{DNN}}}{\arg\min} \sum\nolimits_{i=1}^{N_t} \mathcal{L}\left(f_{\theta}(\mathbf{x}_i), y_i\right) \tag{13}$$

[1]The approximation is based on finding N highest local maxima (Fig. 3).

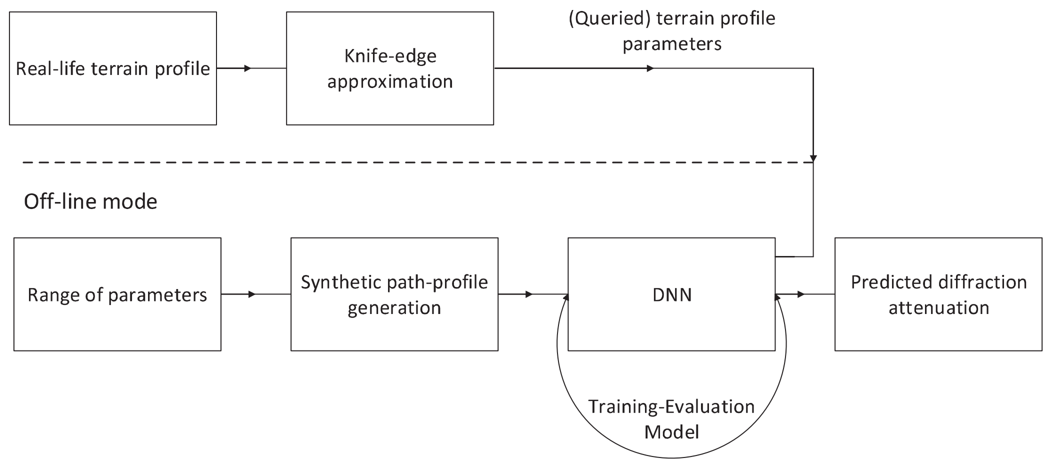

Fig. 2: Illustration of our proposed approach "VoglerNet". In off-line mode, given range of parameters, synthetic path-profiles are generated randomly to cover an irregular terrain of interest. These profiles are used to train and evaluate a deep neural network (DNN). When new queried profile parameters are sent to the DNN, the DNN replies by the predicted diffraction attenuation.

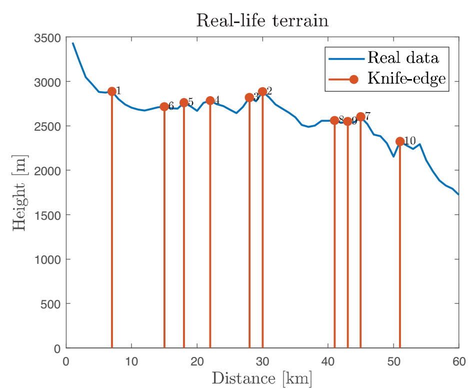

Fig. 3: Multiple knife-edge approximation of a real-life terrain. A knife-edge is chosen as local maxima. Here, ten knife-edges are numbered in descending order.

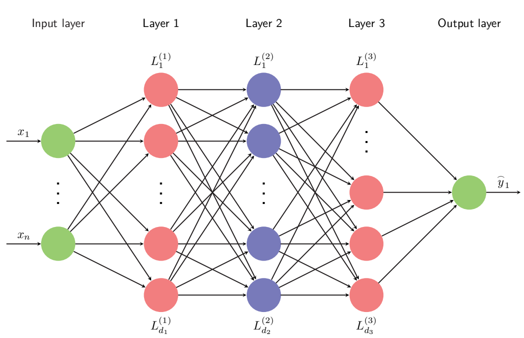

Fig. 4: Illustration of a deep neural network architecture with n-input $\mathbf{x}$, 1-output $\widehat{y}$ and $L = 3$ hidden layers.

where $f_\theta\left(\mathbf{x}_i\right)$ represents to DNN response to the input $\mathbf{x}_i$. To this end, let $f(\mathbf{x})$, $f : \mathbb{R}^n \to \mathbb{R}$, be a function we want to approximate. Our purpose is to use the DNN estimator defined by $\boldsymbol{\theta}_{\text{DNN}}$, $\tilde{f}_\theta\left(\mathbf{x}\right) : \mathbb{R}^n \to \mathbb{R}$ so that

$$\min_{\theta_{\text{DNN}}} \left\| f - \tilde{f} \right\| \leq \varepsilon \tag{14}$$

where ε is a desired precision. Thus, given a set of training data pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N_t}$, we propose to optimize the parameters on training set as follows

$$\theta = \arg\min_{\theta} \lambda \mathcal{E}_D + \gamma \mathcal{E}_\theta, \tag{15}$$

where

$$\mathcal{E}_D = \frac{1}{2} \sum\nolimits_{i=1}^{N_t} \left(y_i - \widehat{y}_i\right)^2, \tag{16}$$

$$\mathcal{E}_{\boldsymbol{\theta}} = \sum\nolimits_{i=1}^{k} \theta_i^2, \tag{17}$$

which define the empirical risk and regularization terms respectively with $\widehat{y}_i$ being the response of DNN to the input $\mathbf{x}_i$, for $1 \leq i \leq N_t$; and λ and γ are parameters of the loss function. The empirical risk aims to obtain the neural network parameters which are optimal to the given training dataset. Meanwhile, the goal of the regularization term is to avoid overfiting problem. In our case, the Tikhonov regularization is used so that the small weights and biases are preferred [16], thus 'smoothing' the DNN response. Moreover, adding the loss function parameters allows to balance between the empirical risk and regularization terms, and improves generalization.

To train the DNN to minimize the loss function, we propose to use the Levenberg-Marquardt backpropagation [17] with a Bayesian regularization [18] as presented in [19]. This algorithm, named here Bayesian regualrized Levenberg-Marquardt backpropagation (BRLMB), is suitable for medium

TABLE I: RBLMB algorithm parameters used for training.

Parameters	Value
Maximum number of epochs to train	1000
Performance objective ε	0
Dumping factor μ	0.005
Decrease factor for μ	0.1
Increase factor for μ	10
Minimum performance gradient	10^{-7}

size datasets as in our case (i.e., up to several hundred thousand data points). We summarize here only the main ideas[2]. To obtain the weights and biases, the backpropagation is combined with Levenberg-Marquardt update (i.e., a Gaussian-Newton type method). By using a damping factor μ, the update step flexibly corresponds to that of either steepest descent or Gauss-Newton algorithm. Thus, its convergence rate is faster than the steepest descent while keeping a lower computational complexity than the Gauss-Newton. To properly select the parameters of the loss function, γ and λ, Bayesian regularization framework [18] is used. Particularly, in the first level of Bayesian interpretation, it is shown that maximizing the posterior corresponds to minimize the loss function. It is assumed that the noise distribution in the training set is Gaussian. Then in the second level, the parameters are obtained by expanding the loss function in terms of second-order around a minimum point and solving normalization factor. We achieve the result provided that the regularization parameters γ and λ follows a uniform distribution. In fact, the rationale behind the selection of the Levenberg-Marquardt in conjunction with the Bayesian framework is to minimize additional computational complexity (i.e., exploiting available calculation of Hessian matrix of the Levenberg-Marquardt algorithm).

IV. Numerical results

In this section, we assess the effectiveness of the proposed approach by comparing its performance with the state-of-the-arts. Particularly, we first describe the setup and compare the results of VoglerNet, SNN and the Giovanelli method. The Giovanelli method is chosen since it provides the most accurate results among graphical based methods as presented in [20]. In this case, the result from the Volger method is used as reference to other methods. Then, we analyze the effect of training data size on performance of VoglerNet.

Performance comparison: We use a number of knife-edges $N = 3$ for illustration. We randomly generate the path profiles for terrain of interest as follows. The heights of three knife-edges are in the range $(0, 1)$ km. The separation distances between two knife-edges are of $(1, 10)$ km. The operating frequency of antenna is at 100 MHz. We note that the terrain parameters and the operating frequency are chosen so that they cover the first example of [5]. This standard example is presented in many publications due to its well-understanding behavior. We consider the case of $N = 3$ of a 30 km

[2] We refer the reader to [19] for further technical details.

TABLE II: MSE-based error comparison of three algorithms (The smaller value is, the better result reaches).

Methods	VoglerNet	SNN	Giovanelli
MSE (dB)	**0.2003**	3.7594	1.8098
Min. (dB)	0.0187	0.0125	**0.0076**
Max. (dB)	**1.2360**	4.3198	4.7136

propagation path where the transmitter and the receiver are in the reference plane (i.e., $h(0) = h(4) = 0$). There are two fixed knife-edges at distances of 10 km and 20 km respectively. Their heights are at 100 m above the reference plane. A third knife-edge with variable height is located at the distance of 15 km. When the height h_2 increases, the attenuation curve converges toward the single knife-edge one. The oscillations appear because of the effect of two other knife-edges. Thus, we can further evaluate the result by visualizing as Fig. 5. We emphasize that the probability of generating exact path profile, as the first example in [5] of training test, is zero. Thus the evaluation is fair.

In all experiments, we randomly divide data into two sets, 80% over total data for training and 20% over total data for testing. We use the mean squared error (MSE) as a performance index

$$\mathrm{MSE} = \frac{1}{N_{\text{test}}} \sum\nolimits_{i=1}^{N_{\text{test}}} \left(y_i^{\text{test}} - \widehat{y}_i^{\text{test}} \right) \qquad (18)$$

where N_{test} is number of of test data samples. y^{test} is the test data which is diffraction attenuation result of the Vogler method; $\widehat{y}^{\text{test}}$ is response of the DNN to test input data. Moreover, we also use two other indices, maximum value (Max.) and minimum value (Min.), which are the worst and best predicted diffraction differences respectively when comparing to the result of Vogler method. We design the DNN to have 10 hidden layers (i.e., $L = 10$) and each layer has 20 neurons (i.e., $d_1 = \cdots = d_{10} = 20$). The activation functions are hyperbolic tangent sigmoid used for all hidden layers. In the output layer, a linear transfer function is chosen. We notice that SNN (two-layer feedforward network) is a special case of DNN which has one hidden layer (i.e., $L = 1$) with 200 neurons. Moreover, we use BRLMB for training both DNN and SNN. Parameters of BRLMB are summarized in Table I. The results are reported using the dataset including 500,000 points.

It can be observed from Table. II that VoglerNet obtains the best results in terms of accuracy (MSE and Max. categories) among three employed algorithms while keeping its running time in order of millisecond (using Matlab). The running time is the same order as that of the Giovanelli method in this example. The Giovanelli method, however, yields the best result in Minimum category (Min.).

While Table II serves as quantitative assessment, we also investigate the qualitative one which provides insight of the proposed approach (see Fig. 5). When the height h_2 increases from 0.35 to 0.8 km, the attenuation curve moves toward the single knife-edge one. The 'oscillations' appear because of the effect of two other knife-edges. We can see that

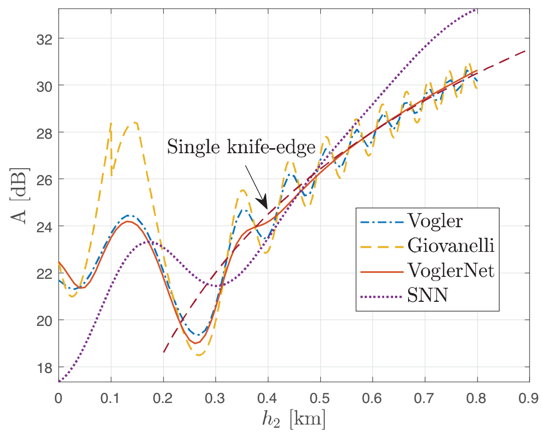

Fig. 5: Qualitative comparison of four methods.

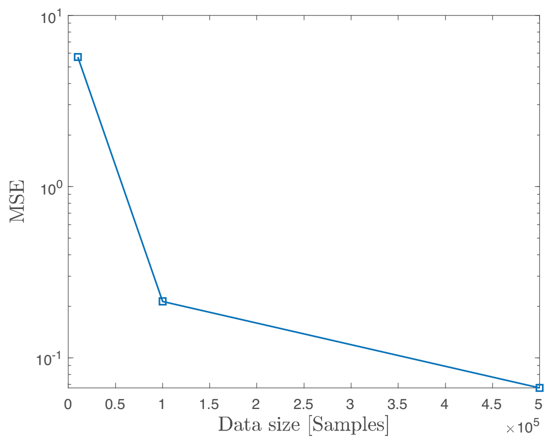

Fig. 6: Effect of training data size on performance of VoglerNet.

the result of VoglerNet can preserve better the main trend of the curve comparing to the the Giovanelli method and SNN. The Giovanelli method overestimates the diffraction value while SNN result is inadequate. We can observe that VoglerNet underestimates the diffraction result in the range $(0.35, 0.8)$ since the DNN considers the oscillations as noises and 'denoises' this effect.

Effect of training data size: One of the main advantages of VoglerNet is that we can exploit as much data as we want to train the DNN. This is due to the benefit of synthetic path-profile generation. It means that more accurate results can be obtained when increasing the dataset size. This effect can be observed in Fig. 6. In this experiment, the total size of dataset is increased from 10,000 to 500,000 data points. It is showed that the difference between results of VoglerNet and Vogler method can reach to below 0.1 dB.

V. Conclusion

In this paper, we has proposed a new algorithm, VoglerNet, based on deep neural network, to solve multiple knife-edge diffraction. To the best of our knowledge, this is a pioneer approach to handle this problem. Our approach benefits from the advantages of both the Vogler method and deep learning approach, where our fast solution is in order of milliseconds while the performance is very close to that of the Vogler method. In a near future work, further investigations in terms of DNN analysis and training time improvement will be conducted.

Acknowledgment

We would like to thank the Direction générale de l'armement (DGA) and specially the Agence de l'Innovation de Défense (AID) for the financial support of our project "LINASAAF". The authors are also grateful to Mr. Thierry Marsault, research engineer at DGA-MI, for supporting our project.

References

[1] A. Goldsmith, *Wireless communications*. Cambridge University Press, 2005.
[2] S. R. Saunders and A. Aragón-Zavala, *Antennas and propagation for wireless communication systems*. John Wiley & Sons, 2007.
[3] ITU, *Propagation par diffraction*. Recommendation ITU-R P.526-14, 2008.
[4] T. S. Rappaport, G. R. MacCartney, S. Sun, H. Yan, and S. Deng, "Small-scale, local area, and transitional millimeter wave propagation for 5g communications," *IEEE Trans. Antennas Propag.*, vol. 65, no. 12, pp. 6474–6490, Dec 2017.
[5] L. E. Vogler, "An attenuation function for multiple knife-edge diffraction," *Radio Science*, vol. 17, no. 06, pp. 1541–1546, Nov 1982.
[6] J. B. Andersen, "Transition zone diffraction by multiple edges," *IEE Proceedings - Microwaves, Antennas and Propagation*, vol. 141, no. 5, pp. 382–384, Oct 1994.
[7] P. D. Holm, "UTD-diffraction coefficients for higher order wedge diffracted fields," *IEEE Trans. Antennas Propag.*, vol. 44, no. 6, pp. 879–888, June 1996.
[8] J. B. Andersen, "UTD multiple-edge transition zone diffraction," *IEEE Trans. Antennas Propag.*, vol. 45, no. 7, pp. 1093–1097, July 1997.
[9] C. Tzaras and S. R. Saunders, "An improved heuristic UTD solution for multiple-edge transition zone diffraction," *IEEE Trans. Antennas Propag.*, vol. 49, no. 12, pp. 1678–1682, Dec 2001.
[10] P. Valtr, P. Pechac, and M. Grabner, "Inclusion of higher order diffracted fields in the Epstein–Peterson method," *IEEE Trans. Antennas Propag.*, vol. 63, no. 7, pp. 3240–3244, July 2015.
[11] J. Epstein and D. W. Peterson, "An experimental study of wave propagation at 850 MC," *Proc. of the IRE*, no. 5, pp. 595–611, May 1953.
[12] J. Deygout, "Multiple knife-edge diffraction of microwaves," *IEEE Trans. Antennas Propag.*, vol. 14, no. 4, pp. 480–489, July 1966.
[13] J. Causebrook and B. Davis, *Tropospheric radio wave propagation over irregular terrain: The computation of field strength for UHF broadcasting*. Research Department, Engineering Division, BBC, 1971.
[14] C. L. Giovaneli, "An analysis of simplified solutions for multiple knife-edge diffraction," *IEEE Trans. Antennas Propag.*, vol. 32, no. 3, pp. 297–301, March 1984.
[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
[17] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, Nov 1994.
[18] D. J. MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.
[19] F. Dan Foresee and M. T. Hagan, "Gauss-Newton approximation to Bayesian learning," in *Proc. of Int. Conf. on Neural Networks (ICNN'97)*, vol. 3, June 1997, pp. 1930–1935 vol.3.
[20] N. DeMinco and P. McKenna, "A comparative analysis of multiple knife-edge diffraction methods," *Proc. ISART/ClimDiff*, pp. 65–69, 2008.