This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LWC.2021.3111874, IEEE Wireless Communications Letters

Channel Metamodeling for Explainable Data-Driven Channel Model

Hyun-Suk Lee

Abstract—**Machine learning can produce accurate data-driven channel models, but their black-box nature makes it harder to explain the models and to understand underlying channel characteristics. In this letter, we propose a channel metamodeling approach for such a black-box data-driven channel model. Our approach enables us to express the data-driven channel model in terms of transparent mathematical expressions based on symbolic function approximation methods. Through experiments with synthetic and real datasets, we demonstrate that our approach produces a channel metamodel of the data-driven channel model for each dataset that is highly accurate and allows us to easily explain the data-driven channel model and to understand the underlying channel characteristics.**

Index Terms—**Channel model, data-driven, deep learning, explainable AI, symbolic metamodeling**

I. Introduction

With the coming of future wireless communications, scenarios have become more diverse and complicated due to mmWave, massive antennas, and various vertical services [1]. In traditional channel modeling, channel characteristics are usually investigated through data processing, and then, a channel model is constructed based on the channel characteristics. However, new scenarios make data processing more challenging and expensive due to their increased diversity and complexity. To resolve this issue, recently, as in other areas in wireless communications [1], [2], machine learning (ML) techniques have been widely applied for channel modeling in the name of data-driven channel modeling [3]–[8]. In data-driven channel modeling, the channel model is directly learned from channel measurement data without any prior data processing.

A. Data-Driven Channel Model and Its Black-Box Nature

For data-driven channel models, channel measurement datasets and/or simulation datasets are usually used [9]. A dataset is composed of physical parameters, such as the Tx and Rx coordinates, the Tx-Rx distances, and the carrier frequency, labeled by the corresponding radio measurements. The data-driven channel model is trained using the dataset and ML techniques to predict channel statistical properties, such as the received power, root mean square (RMS) delay spread, and RMS angle spread, according to given physical parameters [3]. To describe this, we denote a vector of the input features of the channel model (i.e., physical parameters of the dataset) as $\mathbf{x} = (x_1, ..., x_K)^\top \in \mathcal{X}$, where K is the number of the input features and $\mathcal{X}$ is the feature space. We denote a vector of the outputs of the channel model (i.e., channel statistical properties) as $\mathbf{y} = (y_1, ...y_L)^\top \in \mathcal{Y} \subseteq \mathbb{R}^L$, where L is the number of the outputs that the channel model will predict and $\mathcal{Y}$ is the output space.[1] Without loss of generality, we assume that the feature space $\mathcal{X}$ is the unit hypercube (i.e, $\mathcal{X} = [0, 1]^K$). Then, the true relationship between the inputs and outputs can be expressed as $\mathbf{y} = H(\mathbf{x})$, where $H : \mathcal{X} \to \mathcal{Y}$ represents a physical wireless medium in the environment.

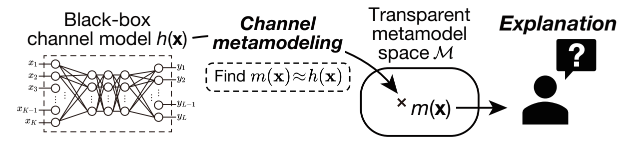


Fig. 1. An illustration of channel metamodeling for explainable data-driven channel model.

The data-driven channel modeling approach based on deep learning trains a DNN $h : \mathcal{X} \to \mathcal{Y}$ by using typical ML techniques in order to approximate the relationship (i.e., $h(\mathbf{x}) \approx H(\mathbf{x})$, $\forall \mathbf{x} \in \mathcal{X}$) as illustrated in Fig. 1.[2] Deep learning is typically used for data-driven channel modeling [3]–[7] because of the large representational capacity of the DNN and the great regularization and learning performance of state-of-the-art ML techniques. However, at the same time, it makes harder to explain the internal mechanism of the trained model, i.e., it is not clear why the DNN works well. Due to this nature of typical DNNs, they are usually called a *black-box* model. Hence, for such a channel model, we can access the outputs of the channel model $\mathbf{y} = h(\mathbf{x})$ for any given feature vector $\mathbf{x}$ only, and it is challenging to investigate underlying channel characteristics, such as the nonlinearity of channel and contributions of physical parameters, which are usually discovered via prior data processing in traditional channel modeling [11]. However, in general, such underlying channel characteristics are still required for further researches such as system design and network management. Thus, we need to interpret the data-driven channel model h into an understandable form to humans [11].

B. Channel Metamodeling and Our Contributions

To address the issue raised due to the black-box nature of DNNs, in this letter, we propose a channel metamodeling approach for data-driven channel models illustrated in Fig. 1. In

This work was supported by the National Research Foundation of Korea (NRF) grant through the Korea Government (MSIT) under Grant 2021R1G1A1004796.

H.-S. Lee is with the School of Intelligent Mechatronics Engineering, Sejong University, Seoul, Korea (e-mail: hyunsuk@sejong.ac.kr).

[1]For generality, here we do not explicitly define the input features and the outputs of the channel model.

[2]In this letter, we do not specify the DNN since the channel metamodeling approach proposed in this letter is model-agnostic, which implies that it can be applied to data-driven channel models based on any types of black-box models such as DNN [3]–[7] and SVM [10].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LWC.2021.3111874, IEEE Wireless Communications Letters

the approach, a channel metamodeling problem is formulated to find a channel metamodel that approximates a given data-driven channel model. By solving the problem via symbolic representation methods, we can express the data-driven channel model in terms of transparent mathematical expressions. Accordingly, we can explain and understand the data-driven channel models. Through experiments with the DNN-based channel models for synthetic and real datasets, we demonstrate that our proposed channel metamodeling approach accurately expresses them in terms of a mathematical form which can be used to understand the channel models. To the best of our knowledge, this letter is the first attempt to metamodel data-driven channel models into an understandable mathematical expression.

II. Explainability via Channel Metamodeling

Deep learning provides a variety of learning mechanisms to address the bias and variance in datasets and to reflect the nature of datasets (e.g., time-series and causality). Since such learning mechanisms are not provided in most transparent models such as decision trees and mathematical expressions, deep learning is typically used in various applications including data-driven channel modeling despite its black-box nature and the transparent models cannot replace deep learning. With these backgrounds, recently, various methods to explain deep learning models have been widely studied under the name of *explainable artificial intelligence (XAI)*. In particular, the XAI methods such as LIME [12] and INVASE [13] focus on providing importance scores of each feature on the prediction of a given instance. Meanwhile, other methods such as SHAP [14] and GA^{2}M [15] focus on providing interactions between features. However, they typically focus on a local explanation of the model for the given instance, and thus, cannot capture *global nonlinearities* in a dataset. In aspects of data-driven channel models, this is a significant demerit of such local explanation methods since for further researches that address system-level issues, it is important to distill global nonlinearities in channel characteristics.

In this context, channel metamodeling should be designed to provide not only feature importance or interaction but also global nonlinearity in data-driven channel models. Finding a global mathematical expression of a data-driven channel is one of the appropriate methods. In a view of explainability, such a global mathematical expression itself naturally captures the global nonlinearity underlying data-driven channel models while providing transparency. Besides, it can be understood as a tabula rasa from which different forms of explanations in the literature can be derived [16]. Most existing XAI methods provide only a part of the explanations, such as instance-wise feature importance and feature interaction, using gradients or local approximations. On the other hand, the mathematical expression enables to easily derive the explanations taking into account both gradients and local approximations via Taylor expansion. We refer the reader to [16] for more details.

III. Channel Metamodeling of Data-Driven Channel Model

In this section, a channel metamodeling approach for data-driven channel models is proposed to find a mathematical expression, which approximates the channel characteristics, $H(\mathbf{x})$.

A. Channel Metamodeling Problem

We use a metamodel $m \in \mathcal{M}$ that approximates the data-driven channel model $h(\mathbf{x}) \in \mathcal{Y}$ for all $\mathbf{x} \in \mathcal{X}$, where $\mathcal{M}$ is a given class of concise mathematical expressions for channel metamodeling that are analytically manipulable and understandable. It is worth emphasizing that the model $h(\mathbf{x})$ is a vector-valued function with L outputs. Then, with given $\mathcal{M}$, we need to find a function m in the class $\mathcal{M}$ that approximates the channel model h most accurately. To formally formulate this, we define a channel metamodeling loss l as the mean squared error (MSE) between the channel model h and the channel metamodel m as

$$l(m,h) = \|m-h\|_2^2 = \int_{\mathcal{X}} \|m(\mathbf{x}) - h(\mathbf{x})\|_2^2 dF(\mathbf{x}), \qquad (1)$$

where $F(\mathbf{x})$ is the cumulative distribution function on the feature space. We then define a channel metamodeling problem as

$$(\mathbf{MP}) \ \ \underset{m\in\mathcal{M}}{\operatorname{argmin}}\, l(m,h). \qquad (2)$$

B. Decomposition of Metamodeling Problem

To solve the metamodeling problem, all the possible mathematical expressions should be considered, but it is intractable because of its infinite number of very diverse functional forms due to the multi-dimensional vector-valued output of the channel model. Hence, to efficiently address the metamodeling problem, we should reduce the complexity of the metamodel class. To this end, we first decompose the channel model into each component of the output as $h(\mathbf{x}) = (h_1(\mathbf{x}), ..., h_L(\mathbf{x}))$, where $h_l(\mathbf{x})$ is the channel model of the l-th component of the output y_l. With this decomposed channel model, we can metamodel each component in a form of real-valued functions, $m^r \in \mathcal{M}^r : \mathcal{X} \to \mathbb{R}$, as $m(\mathbf{x}) = (m_1(\mathbf{x}), ..., m_L(\mathbf{x}))$, where $\mathcal{M}^r$ is a class of real-valued mathematical expressions. We then rewrite the channel metamodeling loss in (1) as the following lemma.

Lemma 1: If the channel metamodel is decomposed into each component channel model, then the following holds:

$$l(m,h) = \sum_{l\in\{1,...,L\}} l(m_l, h_l).$$

Proof: With the decomposed channel metamodel, $l(m,h)$ can be rewritten as

$$\int_{\mathcal{X}} (m_1(\mathbf{x}) - h_1(\mathbf{x}))^2 + \cdots + (m_L(\mathbf{x}) - h_L(\mathbf{x}))^2 dF(\mathbf{x})$$

$$= \sum_{l\in\{1,...,L\}} \int_{\mathcal{X}} (m_l(\mathbf{x}) - h_l(\mathbf{x}))^2 dF(\mathbf{x}) = \sum_{l\in\{1,...,L\}} l(m_l, h_l). \ \blacksquare$$

Using this objective function, the channel metamodeling problem in (2) is reformulated as

$$(\mathbf{DMP}) \quad \underset{m_1,...,m_L\in\mathcal{M}^r}{\operatorname{argmin}} \sum_{l\in\{1,...,L\}} l(m_l, h_l). \qquad (3)$$

This problem finds the channel metamodel $m(\mathbf{x})$ that belongs to the L-ary power of the class of real-valued mathematical expressions, $\{\mathcal{M}^r\}^L$. From this fact and Lemma 1, we can have the following proposition.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LWC.2021.3111874, IEEE Wireless Communications Letters

Algorithm 1 Channel Metamodeling Approach

1: **Input**: Data-driven channel model $h(\mathbf{x})$
2: **for** $l = 1, 2, \ldots L$ **do**
3: Initialize $X_i \sim \mathcal{U}([0,1]^K),\ i = \{1, \ldots, N_s\}$
4: Obtain $m_l(\mathbf{x})$ solving problem (4) via symbolic methods using X_i's
5: **end for**
6: **Output**: Channel metamodel $m(\mathbf{x}) = (m_1(\mathbf{x}), \ldots, m_L(\mathbf{x}))$

Proposition 1: If $\mathcal{M} \subseteq \{\mathcal{M}^r\}^L$, we can find the channel metamodel $m^*(\mathbf{x})$ by solving the problem in (3), which satisfies

$$l(m^*, h) \leq l(m^*_{\mathrm{MP}}, h),$$

where $m^*_{\mathrm{MP}}(\mathbf{x})$ is the optimal solution to the problem in (2).

Proof: Since $m^*_{\mathrm{MP}} \in \mathcal{M}$ and $\mathcal{M} \subseteq \{\mathcal{M}^r\}^L$, the solution to the problem in (3), $m^*(\mathbf{x})$, satisfies

$$\sum_{l \in \{1,\ldots,L\}} l(m^*_l, h_l) \leq \sum_{l \in \{1,\ldots,L\}} l(m^*_{\mathrm{MP},l}, h_l).$$

Then, from Lemma 1, the proposition holds. ■

This proposition implies that if the class of real-valued mathematical expressions used in channel metamodeling, $\mathcal{M}^r$, is diverse enough, we can find a channel metamodel that could be better than that from the problem in (2) by solving the problem in (3). Reversely, it also implies that if we want to find the best channel metamodel in $\mathcal{M}$, instead of directly finding it, we can find one by solving the problem in (3) with $\mathcal{M}^r$ satisfying $\mathcal{M} \subseteq \{\mathcal{M}^r\}^L$. The diversity of $\mathcal{M}^r$ depends on symbolic representation methods and will be compared in Section III-E. To solve the problem in (3), we first decompose it into the problem for l-th component as

$$(\mathbf{DMP})_l \quad \operatorname*{argmin}_{m_l \in \mathcal{M}^r} l(m_l, h_l). \tag{4}$$

This decomposition of the metamodeling problem allows us to consider each component of the output independently and to obtain the channel metamodel of each component by using symbolic methods for function approximation.

To obtain the channel metamodel $m_l(\mathbf{x})$, we first uniformly draw N_s feature vectors. We then find the channel metamodel by solving the problem in (4). For this, we apply the symbolic function approximation methods in which the feature vectors are used to estimate the metamodeling loss in (4). The channel metamodeling approach is summarized in Algorithm 1. In the following sections, we introduce two symbolic methods that can solve the problem in (4) (line 4 of Algorithm 1). They construct a metamodel structure that induces the mathematical expressions for each component $m_l(\mathbf{x})$. By exploiting the constructed structure, we can efficiently solve the channel metamodeling problem. For brevity, we omit the component index l in the following subsections.

C. Channel Metamodeling via Meijer G-Function

We first introduce a channel metamodeling approach using projection pursuit which is a well-known method in the statistics literature [17], [18]. It approximates a given function ($h(\mathbf{x})$ in this subsection) in a form of

$$m(\mathbf{x}) = \sum_{n=1}^{N} w_n g_n\left(\frac{[\mathbf{v}_n^\top \mathbf{x}]^+}{\|\mathbf{v}_n\|\sqrt{K}}\right),$$

where $[\cdot]^+ = \max(0, \cdot)$, $\mathbf{v}_n \in \mathbb{R}^K$ is a vector onto which $\mathbf{x}$ is projected, and each g_n is a function which belongs to a specified class $\mathcal{F}$ of univariate functions. To build this approximation, each term is iteratively obtained in stages. At stage j, the vector $\mathbf{v}_j$ and the function g_j are determined to improve the current approximation $M_{j-1}(\mathbf{x})$ as follows:

$$\operatorname*{argmin}_{(g_j, w_j, \mathbf{v}_j) \in \mathcal{G} \times \mathbb{R}^{K+1}} \int_{\mathcal{X}} \left(r_j(\mathbf{x}) - w_j g_j\left(\frac{[\mathbf{v}_n^\top \mathbf{x}]^+}{\|\mathbf{v}_n\|\sqrt{K}}\right)\right)^2 dF(\mathbf{x}), \tag{5}$$

where $r_j = h - m_{j-1}$ is the residual of the current approximation. It is worth noting that this process eventually solves the problem in (4) achieving a predefined precision.

With this approach, the class for metamodels $\mathcal{M}^r$ is constructed based on the class $\mathcal{F}$ of g_n. We use a Meijer G-function, which is a general function from specified by the parameters $\mathbf{a}_p = (a_1, \ldots, a_p)$, $\mathbf{b}_q = (b_1, \ldots, b_q)$, n, and m, as the basis function for the class.

Definition 1 (Meijer G-functions [19]): A Meijer G-function is defined by an integral along a path $\mathcal{L}$ in the complex plane,

$$G^{m,n}_{p,q}\left(\begin{matrix} a_1,\ldots,a_p \\ b_1,\ldots,b_q \end{matrix} \middle| x\right) = \tag{6}$$
$$\frac{1}{2\pi i}\int_{\mathcal{L}} \frac{\prod_{j=1}^{m}\Gamma(b_j - s)\prod_{j=1}^{n}\Gamma(1 - a_j + s)}{\prod_{j=m+1}^{q}\Gamma(1 - b_j + s)\prod_{j=n+1}^{p}\Gamma(a_j - s)} x^s\, ds,$$

where $a_i, b_j \in \mathbb{R}$ for all $i = 1, \ldots, p$ and $j = 1, \ldots, q$, $m, n, p, q \in \mathbb{N}$ with $m \leq q$, $n \leq p$, and $\Gamma(\cdot)$ is the Gamma function. According to the parameters, the Meijer G-function represents a very large class of functions including most of the known special functions, such as polynomials, exponential, trigonometric, and hypergeometric functions, as particular cases. For example, $e^{-x} = G^{1,0}_{0,1}\left(\begin{matrix} - \\ 0 \end{matrix} \middle| x\right)$ and $\log(1+x) = G^{1,2}_{2,2}\left(\begin{matrix} 1,1 \\ 1,0 \end{matrix} \middle| x\right)$. This high generality of the Meijer G-function allows us to construct the class for metamodels $\mathcal{M}^r$ large enough for accurate metamodeling. Another advantage of the Meijer G-function is that a numerical gradient of these Meijer G-functions with respect to the parameters can be easily obtained [19]. This allows us to easily solve the optimization problem in (5) by optimizing their parameters a_i's and b_j's by using standard gradient-based optimization techniques.

We solve the channel metamodeling problem in (4) by the projection pursuit method with the Meijer G-function. In the problem (5), we can find g_j by optimizing the hyperparameters of the Meijer G-function, $\mathbf{a}_p$ and $\mathbf{b}_q$, with a pre-configuration set for (m, n, p, q), which can be done via the standard gradient-based optimization techniques. As a result, the channel metamodel $m(\mathbf{x})$ is constructed by finding the projection pursuit model iteratively while minimizing the metamodeling loss in (1). We refer the readers to [18] for more details.

D. Channel Metamodeling via Symbolic Regression

We describe a channel metamodeling approach using symbolic regression. Symbolic regression finds the best mathematical expression to fit a given dataset within the space of mathematical expressions, $\mathcal{M}^r$, which is composed of the combinations of predefined mathematical elements such as operators (e.g., addition, subtraction, multiplication, and division), functions (e.g., $\log(\cdot)$ and $\sin(\cdot)$), and constants. To

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LWC.2021.3111874, IEEE Wireless Communications Letters

this end, a genetic programming approach is usually used [20] with a fitness function to evaluate how good the expression is to fit. This approach starts with an initial random mathematical expression and makes changes over and over to the best mathematical expression according to the fitness function. In symbolic regression, considering more various mathematical elements may improve its metamodeling performance since it enlarges the space $\mathcal{M}^r$. However, at the same time, it increases the computational complexity of symbolic regression. Hence, the mathematical elements are usually predefined carefully considering the trade-off.

For channel metamodeling, we solve the problem in (4) by using symbolic regression. We consider a symbolic regressor $SR(\mathbf{x})$ which represent a mathematical expression for $m(\mathbf{x})$ and then evolve $SR(\mathbf{x})$ via genetic programming as in typical symbolic regression. Here, we use $l(SR, h)$ in (1) as the fitness function since we try to solve the channel metamodel problem in (4). Then, the channel metamodel $m(\mathbf{x})$ is evolved via genetic programming to minimize the metamodeling loss in (1). We refer the readers to [20] for more details.

E. Comparison of Two Channel Metamodeling Approaches

Here, we compare two different channel metamodeling approaches in the previous sections. The main difference between them is the channel metamodel space $\mathcal{M}^r$ used to find the metamodel. In the approach via Meijer G-function, we can consider diverse classes of mathematical expressions thanks to the generality of Meijer G-function. On the other hand, in the approach via symbolic regression, the metamodel space is constructed by the combinations of predefined mathematical elements. This allows us to find the channel metamodel composed of only intended mathematical elements with the relatively lower computational complexity compared with the approach via Meijer G-function, but at the same time, due to such a limitation on the metamodel space, its metamodeling precision may be degraded. Hence, we should appropriately choose them according to the purpose of channel metamodeling.

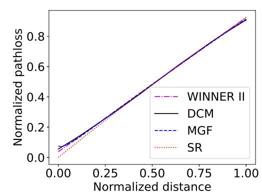
IV. Experimental Results

We demonstrate our channel metamodeling approaches through experiments with a synthetic dataset based on the WINNER II pathloss model and a *real* dataset provided in [4]. For each dataset, we consider a data-driven channel model (DCM) based on deep neural networks. For channel metamodeling, we implement our channel metamodel approaches via Meijer G-function (MGF) and symbolic regression (SR) based on the Python implementation of symbolic metamodels in [18] and `gplearn` library [20], respectively. For MFG and SR, we use the default settings provided in [18] and [20]. We call a metamodel constructed by MGF an MGF metamodel and call one constructed by SR an SR metamodel.

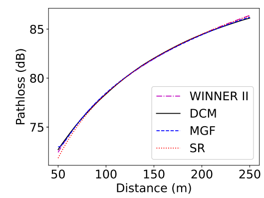
We first consider a synthetic dataset generated by the WINNER II free-space pathloss model with 2 GHz system frequency given by $PL_{\text{free}} = 20\log_{10}(d) + 46.4 + 20\log_{10}(2/5)$, where d is the distance between the transmitter and the receiver in meters. This dataset is a simple explanatory one composed of a single feature. To generate it, we draw a distance

TABLE I
Results of RMSEs on synthetic dataset

	Normalized pathloss			Restored pathloss (dB)		
	DCM	MGF	SR	DCM	MGF	SR
vs. DCM	-	0.005	0.017	-	0.074	0.271
vs. WINNER II	0.006	0.006	0.013	0.095	0.095	0.217



(a) Normalized pathloss.



(b) Restored pathloss.

Fig. 2. Pathloss of channel (meta)models on the synthetic dataset.

$d \in [50, 250]$ from a uniform distribution and add a Gaussian noise $\mathcal{N} \sim (0, 0.1)$ to the corresponding pathloss. We build a DCM by a DNN composed of 2 hidden layers with 100 units and train it by using 1000 samples in the synthetic dataset. We then fit the metamodels to the DCM through our channel metamodeling approaches. We obtain the root mean squared error (RMSE) among the WINNER II model, DCM, and the metamodels using another 1000 random distances.

To demonstrate how accurately the channel metamodels (i.e., MGF and SR) express the DCM, we provide the RMSEs among the WINNER II model, DCM, and metamodels in Table I. The RMSEs of the metamodels against the DCM (vs. DCM) show that both MGF and SR metamodels accurately represent the DCM. However, the MGF metamodel achieves a slightly lower error than the SR metamodel. Accordingly, as shown in the RMSEs against the WINNER II model (vs. WINNER II), the MGF metamodel predicts the WINNER II model precisely as much as the DCM does while the SR metamodel has the larger RMSE compared with the DCM or the MGF metamodel.

In Fig. 2, the pathloss of the WINNER II model, DCM, and channel metamodels is illustrated. From the figure, we can see that the DCM represents the original WINNER II pathloss model well. Moreover, both MGF and SR metamodels closely follow the DCM as well. As described in Table I, the MGF metamodel is more precise than the SR metamodel in terms of the accuracy of the restored channel model. Consequently, in the figure, the MGF metamodel represents the original WINNER II model more closely than the SR metamodel as well.

We now consider a real channel measurement dataset that consists of the channel measurements from three base stations (BSs). The BSs serve different sectors and use different carrier frequencies (811 MHz or 2630 MHz). We use a DCM proposed in [4] based on the dataset. Compared with traditional channel model in 3GPP 38.901 and ray-tracing models, the DCM predicts reference signal received power (RSRP) much more precisely according to the following features: longitude, latitude, distances in latitude and longitude directions, distance straight as the crow flies, and BS indicators (carrier frequency).[3] This implies that the nonlinear characteristics of the dataset is too complex to be captured by the traditional channel model.

[3]It is worth noting that any other feature can be used according to the attributes of the dataset.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LWC.2021.3111874, IEEE Wireless Communications Letters

TABLE II
RESULTS OF RMSES ON REAL RADIO MEASUREMENT DATASET

	Normalized RSRP			Restored RSRP (dB)		
	DCM	MGF	SR	DCM	MGF	SR
vs. DCM	-	0.135	0.221	-	2.448	4.017
vs. dataset	0.297	0.297	0.337	5.399	5.401	6.116

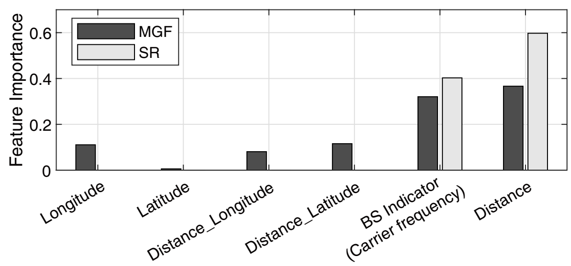


Fig. 3. Feature importance of the real dataset analyzed from the metamodels.

The dataset is divided into 60014 training samples and 6000 test samples. We train the DCM by using the training samples and fit metamodels to the DCM. We then obtain RMSE among the dataset, DCM, and metamodels by using the test samples.

To evaluate the accuracy of the DCM and metamodels, we provide the RMSEs among the dataset, DCM, and metamodels in Table II. Similarly to the results on the synthetic dataset, from the RMSE of metamodels against the DCM (vs. DCM), we can see that the MGF metamodel achieves a lower RMSE than the SR metamodel. This shows that the MGF metamodel describes the DCM more precisely than the SR metamodel. The RMSEs of the channel (meta)models against the true labels of the dataset (vs. dataset) show that the MGF metamodel achieves the close RMSE to that of the DCM. Also, the SR metamodel has a slightly higher RMSE than the others, but its RMSE is still reasonable compared with the others in terms of the restored RSRP.

From the channel metamodels, we can extract various information explaining the DCM to understand the underlying channel characteristics via analytic methods based on Taylor expansion [16]. As a representative example, we extract the instance-wise feature importances of the real dataset from both MGF and SR metamodels and here provide the overall feature importance by averaging them. For the comparison, we normalize the feature importance in each metamodel as their sum to be 1.

In Fig. 3, the feature importance in each metamodel on the real dataset are provided. As shown in the figure, the feature importance from both metamodels indicates the distance and the BS indicator (carrier frequency) as the important features to predict the pathloss. This is reasonable because the distance is a typically significant factor of the pathloss and the carrier frequency is another critical factor, which is determined according to the BS in the real dataset. In the figure, the MGF metamodel takes into account the other features, such as longitude and distances in longitude and latitude, as well while the SR metamodel ignores them. From this, we can speculate that the MGF metamodel achieves better performance than the SR metamodel thanks to considering those features, and the precision improvement from it is shown in Table II. This example clearly shows the advantages of channel metamodeling to explain the black-box data-driven channel model and to understand the underlying channel characteristics.

V. CONCLUSION

In this letter, we proposed the channel metamodeling approach for a black-box data-driven channel model. The proposed approach finds a transparent mathematical expression that accurately represents the data-driven channel model via symbolic representation. Through the experiments for synthetic and real datasets, we demonstrated that the channel metamodel produced by our approach is highly accurate and can be used to generate transparent information which is helpful to understand the underlying channel characteristics. As a future work on this subject, channel metamodeling tailored to the data-driven channel models focusing on specific channel environments can be considered.

REFERENCES

[1] G. Gui *et al.*, “6G: Opening new horizons for integration of comfort, security, and intelligence,” *IEEE Wireless Commun.*, vol. 27, no. 5, pp. 126–132, Oct. 2020.

[2] A. M. Elbir, “CNN-based precoder and combiner design in mmWave MIMO systems,” *IEEE Commun. Lett.*, vol. 23, no. 7, pp. 1240–1243, Jul. 2019.

[3] J. Huang *et al.*, “A big data enabled channel model for 5G wireless communication systems,” *IEEE Trans. Big Data*, vol. 6, no. 2, pp. 211–222, June 2020.

[4] J. Thrane, D. Zibar, and H. L. Christiansen, “Model-aided deep learning method for path loss prediction in mobile communication systems at 2.6 GHz,” *IEEE Access*, vol. 8, pp. 7925–7936, 2020.

[5] Y. Zhang *et al.*, “CV-3DCNN: Complex-valued deep learning for CSI prediction in FDD massive MIMO systems,” *IEEE Wireless Commun. Lett.*, vol. 10, Feb. 2021.

[6] G. P. Ferreira, L. J. Matos, and J. M. Silva, “Improvement of outdoor signal strength prediction in UHF band by artificial neural network,” *IEEE Trans. Antennas Propag.*, vol. 64, no. 12, pp. 5404–5410, Dec. 2016.

[7] H. Li, Y. Li, S. Zhou, and J. Wang, “Wireless channel feature extraction via GMM and CNN in the tomographic channel model,” *J. Commun. Inf. Netw*, vol. 2, no. 1, pp. 41–51, Mar. 2017.

[8] R. He *et al.*, “Clustering enabled wireless channel modeling using big data algorithms,” *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 177–183, May 2018.

[9] J. Huang *et al.*, “Multi-frequency mmwave massive MIMO channel measurements and characterization for 5G wireless communication systems,” *IEEE J. Sel. Areas Commun.*, vol. 35, no. 7, pp. 1591–1605, July 2017.

[10] M. Uccellari *et al.*, “On the application of support vector machines to the prediction of propagation losses at 169 MHz for smart metering applications,” *IET Microw. Antennas & Propag.*, vol. 12, no. 3, pp. 302–312, Feb. 2018.

[11] W. Guo, “Explainable artificial intelligence for 6G: Improving trust between human and machine,” *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 39–45, June 2020.

[12] M. T. Ribeiro, S. Singh, and C. Guestrin, “"why should i trust you?" explaining the predictions of any classifier,” in *Proc. ACM SIGKDD Intl. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1135–1144.

[13] J. Yoon, J. Jordon, and M. van der Schaar, “INVASE: Instance-wise variable selection using neural networks,” in *Proc. ICLR*, 2018.

[14] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” *Proc. NeurIPS*, 2017.

[15] Y. Lou, R. Caruana, J. Gehrke, and G. Hooker, “Accurate intelligible models with pairwise interactions,” in *Proc. ACM SIGKDD Intl. Conf. Knowl. Discovery Data Mining*, 2013.

[16] A. Alaa and M. van der Schaar, “Demystifying black-box models with symbolic metamodels,” in *Proc. NeurIPS*, 2019.

[17] J. H. Friedman and W. Stuetzle, “Projection pursuit regression,” *J. American Stat. Association*, vol. 76, no. 376, pp. 817–823, 1981.

[18] J. Crabbe, Y. Zhang, W. Zame, and M. van der Schaar, “Learning outside the black-box: The pursuit of interpretable models,” in *Proc. NeurIPS*, 2020.

[19] R. Beals and J. Szmigielski, “Meijer G-functions: a gentle introduction,” *Notices of the AMS*, vol. 60, no. 7, pp. 866–872, 2013.

[20] “Genetic programming in python with a scikit-learn inspired API: gplearn,” 2020. [Online]. Available: https://github.com/trevorstephens/gplearn