

Article

Path Loss Prediction Based on Machine Learning Techniques: Principal Component Analysis, Artificial Neural Network, and Gaussian Process

Han-Shin Jo ¹, Chanshin Park ², Eunhyoung Lee ³, Haing Kun Choi ⁴ and Jaedon Park ^{3,*}

¹ Department of Electronics and Control Engineering, Hanbat National University, Daejeon 34158, Korea; hsjo@hanbat.ac.kr

² Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA; chanship@usc.edu

³ Agency for Defense Development, Daejeon 34186, Korea; eunhyoung.lee@add.re.kr

⁴ TnB Radio Tech., Seoul 08511, Korea; radioeng@tnbrt.com

* Correspondence: elejdp@gmail.com

Received: 5 February 2020; Accepted: 27 March 2020; Published: 30 March 2020



Abstract: Although various linear log-distance path loss models have been developed for wireless sensor networks, advanced models are required to more accurately and flexibly represent the path loss for complex environments. This paper proposes a machine learning framework for modeling path loss using a combination of three key techniques: artificial neural network (ANN)-based multi-dimensional regression, Gaussian process-based variance analysis, and principle component analysis (PCA)-aided feature selection. In general, the measured path loss dataset comprises multiple features such as distance, antenna height, etc. First, PCA is adopted to reduce the number of features of the dataset and simplify the learning model accordingly. ANN then learns the path loss structure from the dataset with reduced dimension, and Gaussian process learns the shadowing effect. Path loss data measured in a suburban area in Korea are employed. We observe that the proposed combined path loss and shadowing model is more accurate and flexible compared to the conventional linear path loss plus log-normal shadowing model.

Keywords: wireless sensor network; path loss; machine learning; artificial neural network (ANN); principle component analysis (PCA); Gaussian process; multi-dimensional regression; shadowing; feature selection

1. Introduction

Wireless sensor network comprises typically multiple sensor nodes and central base stations, where the sensors measure physical status of the environment and report it to the central base station via radio signal. Path loss is a phenomenon between the transmitter and receiver in the strength of radio signal as it propagates through space. Since radio receivers require a certain minimum power (sensitivity) to be able to successfully decode information, path loss prediction is essential in mobile communications network design and planning such as link budget, coverage analysis, and locating base station. Many existing path loss models [1–14] adopt a linear log-distance model, which is empirically derived by assuming a linear proportionality between the path length and the path loss, and by determining a proportional factor through the adequate linear regression analysis of the measured data. The linear log-distance model is simple and tractable, but it does not guarantee accurate path loss prediction performance for all radio propagation environments. Advanced modeling methods are required to more accurately and flexibly represent the path loss for complex and various environments.

Path loss models are classified into empirical, deterministic, and semi-deterministic models. The empirical models are based on the measurements and use statistical characteristics. This model is a simple way of estimating path loss, not taking all of the many parameters that determine path loss into account. Distance is the most essential parameter of path loss model and its physical effect on the path loss is mathematically expressed using a linear log-distance function in literature. Thus, linear log-distance path loss plus shadowing model is used as a baseline for most empirical models and given by [2]

$$PL = PL_0 + 10n \log 10 \left(\frac{d}{d_0} \right) + X_\sigma, \text{ for } d \geq d_0 \quad (1)$$

where d is the length of the path and PL_0 is the path loss at the close-in reference distance d_0 , which is generally determined by the measuring points close to the transmitter. n is the path loss exponent representing the rate of change of path loss as the distance increases. The shadow fading denoted by X_σ is a Gaussian random variable with zero mean and σ standard deviation in dB. Most measurement based path loss models extend the baseline given in Equation (1) to incorporate the effect of other main radio parameters such as frequency [1,3,4,6], the height of the transmit and receive antennas [1,3,4,6], clutter and terrain [1,5,6], the percentage of the area covered by buildings in the built up area [7], and line-of-sight and non-line-of-sight [8–10]. Meanwhile, Jo et al. [14] improved the Modified Hata model to suit the higher frequency band of 3–6 GHz. This model is also an empirical model and adopts a linear log-distance function, which is labeled “(2) Proposed” in Figure 1. It is worth noting that the measured data are well presented by linear regression in Figure 1a, but not especially for the distances less than 200 m in Figure 1b. This is a typical example of the need for a model that can accurately represent both data distributions.

Machine learning is a set of methods based on a dataset and modeling algorithms to make predictions. These days, machine-learning-based techniques are utilized in image recognition, natural language processing, and many other fields. Machine learning tasks can be classified as supervised learning and unsupervised learning. The purpose of supervised learning is to learn a relation or function between inputs and outputs, which is for classification or regression problems. On the other hand, unsupervised learning is to derive hidden rules or relationships from unlabeled data.

The machine learning approach to path loss modeling is expected to provide a better model, which can generalize well the propagation environment since the model is being learned through training with the data collected from the environment. The prediction of propagation path loss is regarded as a regression problem, as stated in the literature [15–17]. In this context, path loss models have been developed by various supervised learning techniques such as support vector machine (SVM) [16,18], artificial neural network (ANN) [19–22], random forest [17], K-nearest neighbors (KNN) [17]. The authors of [19–22] provided path loss prediction using ANN models, which provide more precise estimation over the empirical models. Zhang et al. [21] developed a real-time channel prediction model, which could predict path loss (PL) and packet drop for dedicated short-range communications.

However, they did not deal with a multi-dimensional regression framework for path loss modeling with various features, called the radio environment factors, such as distance, frequency, antenna height, etc. In regression, many candidate functions are adopted because the related inputs are unknown. However, many of these features would be irrelevant or redundant. Furthermore, most of input features do not provide discrimination ability for prediction. To construct a good estimator, the input data need to be converted into a reduced representation set of features by using dimensionality reduction techniques [23] such as principal component analysis (PCA) or singular value decomposition (SVD). Very few articles (e.g., [16]) have applied PCA to learning path loss with a reduced input data. Meanwhile, no studies have been conducted on the development of shadowing models using machine learning although shadowing is a major factor in determining path loss.

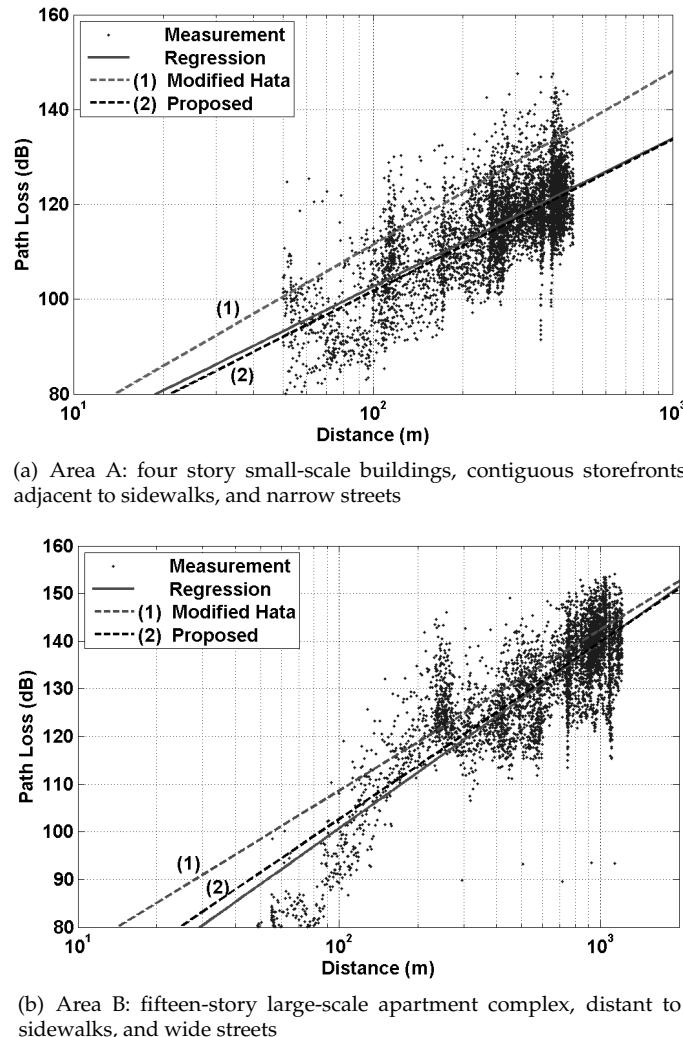


Figure 1. Measured path loss and linear-log distance models presented in [14].

In this work, we employ PCA for extracting key features. Dimensionality reduction is justified from the fact that the actual dimension might be larger than the intrinsic dimension. Then, the objectives of dimensionality reduction are to convert input data into a reduced representation set of features, while keeping as much relevant information as possible. We mainly develop a combined path loss and shadowing model, where ANN multilayer perceptron (ANN-MLP) learns path loss variations dependent on the input data with reduced features. For analyzing shadowing effect, we use Gaussian Process (GP) priors for extracting variance over training data. Based on generalized variance bounds, we can give more accurate confidence level over the path loss mean value. Finally, we attempt to combine ANN and Gaussian process models to build an optimal model in two key aspects: mean path loss value and shadowing effect variance.

The rest of this paper is organized as follows. Section 2 describes procedure of the proposed machine-learning-based path loss analysis. Section 3 presents the simulation results of feature extraction using PCA and its effect on path loss prediction accuracy of ANN-MLP learning. Sections 4 and 5 detail ANN-MLP and Gaussian process, respectively. Measurement system and scenarios are described in Section 6. Section 7 presents experimental results. We conclude in Section 8.

2. Machine Learning Based Path Loss Prediction

The path loss model maps the input features onto the output (path loss observation) values. It is not only important to model a predictor that can produce accurate predictions, but also how derive

more generalized model that does fluctuate due to specific conditional inputs. In this paper, as shown in Figure 2, we propose a three-step approach, dimensional reduction techniques, combining nonlinear regression models and variance analysis for predicting the path loss in a suburban environment. While the ANN-MLP-based nonlinear model focuses on how to accurately predict the path loss value, the PCA and the variance analysis balance it to produce a more generalized model.

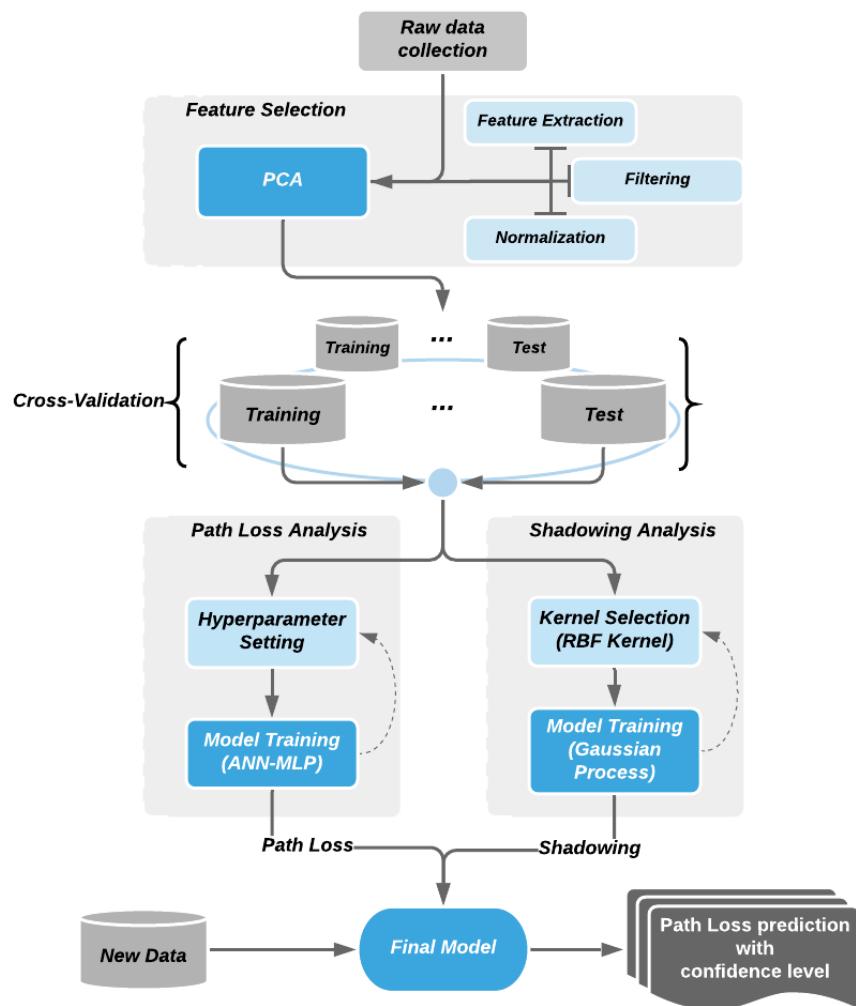


Figure 2. Procedure of machine-learning-based path loss analysis.

2.1. Feature Selection

Once measured data are collected, a dimensionality reduction strategy is applied to obtain simpler feature space in order to expedite the learning. Although there are various types of features in effect on the path loss variation, practical measurable features are the frequency, the height of the transmitting antenna (elevation altitude), the height of the receiving antenna (elevation altitude), and the difference between these two heights. We analyzed features using PCA for evaluating correlation of values between each features and the path loss. The result shows that log distance and log frequency determine over 70% of the path loss variation. In addition, based on the correlation distribution analysis, it can be inferred that various features are dependent on each other.

2.2. Data Preprocessing

ANN is a learning model and its accuracy depends on the training data introduced to it. Aside from its algorithmic and tuning options, well distributed, sufficient, and accurately measured set of data is the prerequisite for acquiring an accurate model. In this perspective, data preprocessing is an essential procedure toward obtaining an ANN learning model. Sampling and normalization are also conducted to reduce process time and prevent bias. The objective of learning is to find the optimal weights on given learning data which enables precise prediction. The key factor for obtaining the right weight is to normalize the magnitude of input values, which minimizes side effects from different scales. For instance, with the same increase with 0.0005, different magnitude of inputs with 0.001 and 0.1 can produce a quite dramatic results in gradient, 0.5 and 0.005. If the input features are not properly normalized, backpropagation with iterative partial derivatives throughout MLP-NN can risk deriving biased weights. Based on propagation characteristics of the input features and balancing the different scale of them, we applied logarithmic transformation on the frequency (MHz) and distance (m) values. Then, data are divided by train and test sets with running iterations by cross-validation for offsetting sampling bias. Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k , which indicates how many groups the given data sample are split into. As such, the procedure is often called k -fold cross-validation. For preparing learning data, all the measured data are divided into two sets, training (80 %) and testing (20 %), with uniform random sampling. The test set is for adjusting hyperparameters for model optimization.

2.3. Path Loss Model

ANN is a nonlinear regression system motivated by learning the weighted network of latent variables through backpropagation. The ANN model outperforms the polynomial regression model in prediction performance [24] and handles more dimensions than the look-up table method [25]. Considering the complex propagation due to the fluctuating heights and complex distribution of buildings in urban area, the nonlinear model can fit better to linear regression. We applied the logistic sigmoid function as an activation function within ANN networks.

2.4. Shadowing Model

Shadowing is attenuation on signal power, which is caused by obstacles between the transmitter and receiver through scattering, reflection, diffraction, and absorption. The receiver power variation due to path loss is observed over long distances, whereas variation due to shadowing occurs by the formation of obstructing object or the length of it. Traditionally, shadowing was treated as a coefficient with normal distribution since it is hard to generalize the characteristics of obstacles and, generally, its effects on path loss are trivial. However, in outdoor environment with various obstacles, shadowing effects play a significant role in analyzing path loss prediction with certain confidence level.

3. PCA

PCA is a dimension reduction technique that linearly transforms the original space into a new space of smaller dimensions while simultaneously describing the variability of the data as much as possible. In fact, the PCA projects along the eigenvectors of the covariance matrix corresponding to the largest eigenvalues, where the eigenvectors points in the direction with the highest amount of data variation. In addition, the magnitude of eigenvalues can be used to estimate the intrinsic dimension of the data. Indeed, if x eigenvalues have a magnitude much larger than the remaining ones, the number x can be regarded as the true dimension of the data. Clearly, a linear technique is able to only extract linear correlations between variables, whereas it usually overestimates the intrinsic dimension for the data having complex relations.

Figures 3 and 4 show the path loss prediction results of learning with four features (transmitting antenna height, receiving antenna height, transmitting/receiving antennas heights ratio, and distance) and PCA-applied one feature (distance), respectively. We can easily notice that the performance of PCA-applied simple model is similar to the original multi-featured training model, as shown in the summarized prediction loss error comparison in Table 1. The data analysis with major variables from PCA requires more effort in experimental data gathering and learning time cost compared to multiple variables model. PCA can filter out unnecessary noise or less-related independent variables, so that the major variables derived from PCA reflect the more inherent path loss properties and result in higher accuracy. The training cost for applying the learning method is crucial since it can be a feasible factor in a real environment. The one variable model in ANN-MLP and GP model takes 10% and 1%, respectively, of the training time compared to the four-variable model. Therefore, we can focus on a distance feature along the same frequency for ANN model training in the following sections.

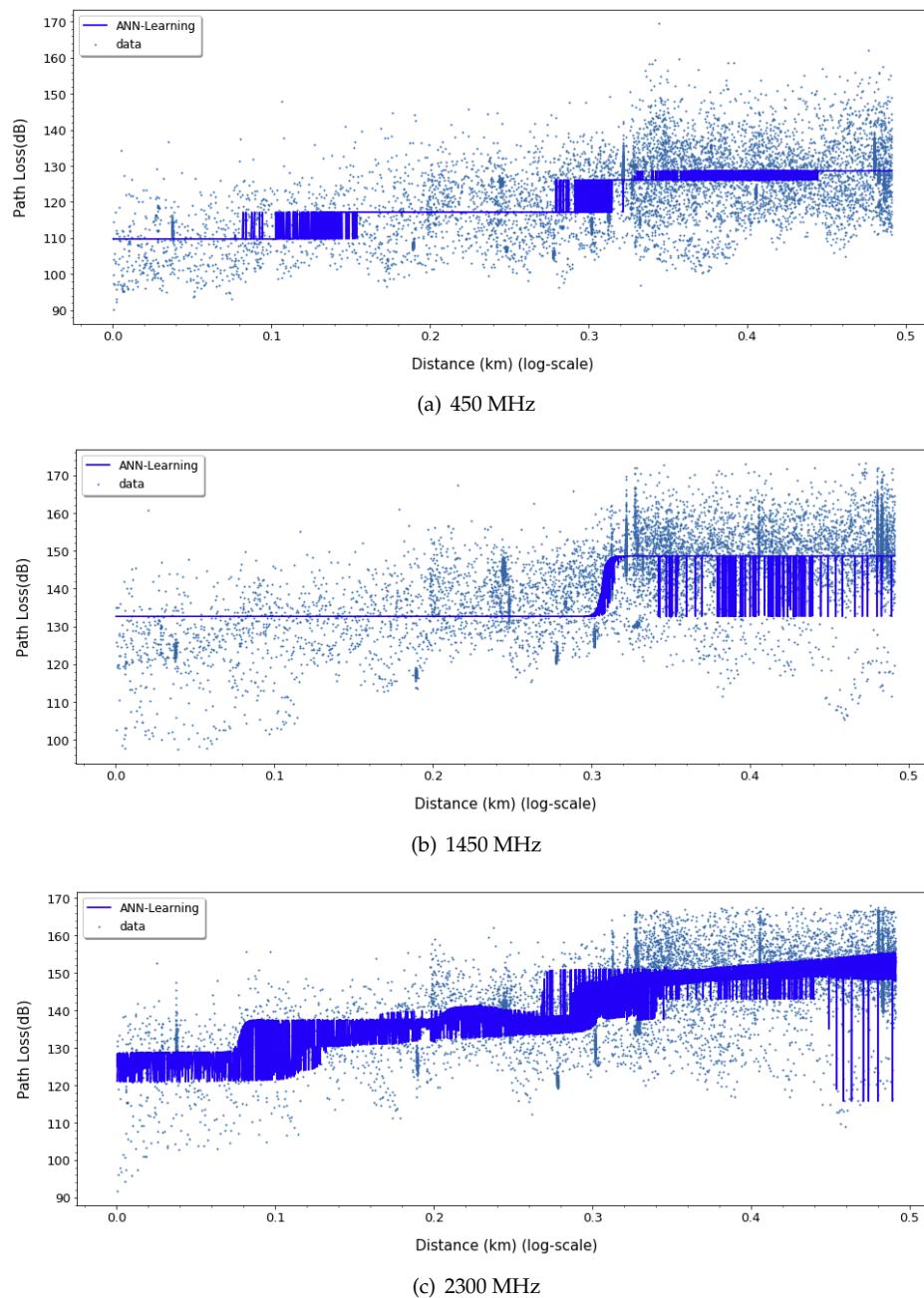


Figure 3. Four-feature-based ANN-MLP model.

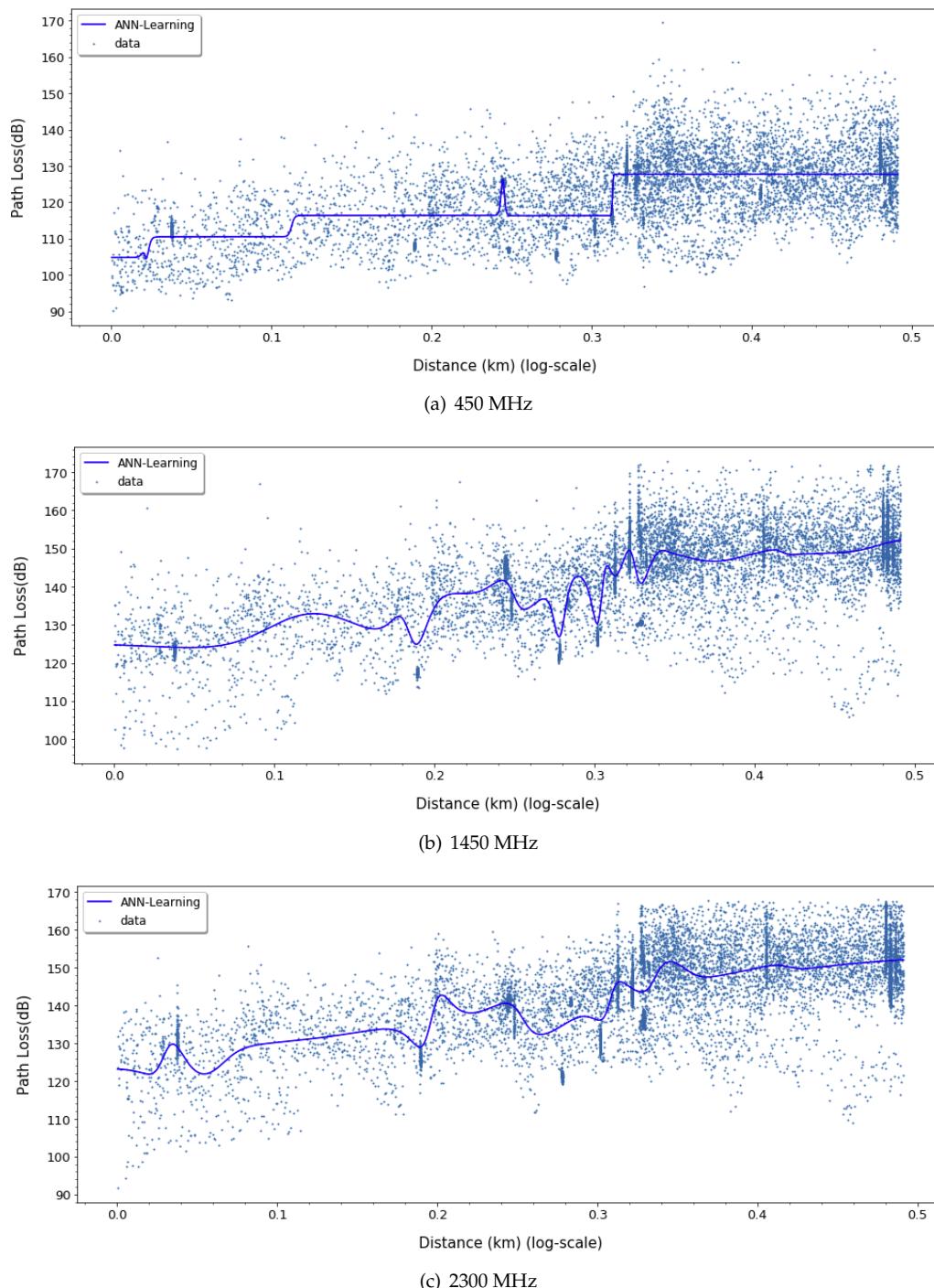


Figure 4. Single-feature-based ANN-MLP model.

Table 1. Prediction loss comparison for PCA-applied and original 4-featured data.

Model/Error (dB)	RMSE	MAE	MAPE	MSLE	R2
4-feature-based Model	8.40197	6.47590	4.85529	0.00393	0.67463
Single-feature-based Model	8.61307	6.58675	4.94076	0.00413	0.66113

4. ANN-MLP

The most common type of ANN is the MLP neural network in which multiple neurons are arranged in layers, starting from an input layer, followed by hidden layers, and ending with an output layer. The outputs from each node at the layer are the weighted sum of its inputs over an activation function. We design the fully connected ANN-MLP, which constitutes several hidden layers of nodes and a single hidden layer of the network structure is depicted in Figure 5. More specifically, $\mathbf{z}_n^{(l)} = [z_{1,n}^{(l)} z_{2,n}^{(l)} \cdots z_{M,n}^{(l)}]^T$ is the n th column vector of the $M \times N$ matrix $\mathbf{Z}^{(l)}$ in the l th layers ($l = 1, 2, \dots, L - 1$), given by

$$\mathbf{z}_n^{(l)} = H(\mathbf{a}_n^{(l)}) = [H(a_{1,n}^{(l)}) H(a_{2,n}^{(l)}) \cdots H(a_{M,n}^{(l)})]^T, \quad (2)$$

$$\mathbf{a}_n^{(l)} = \begin{cases} \mathbf{W}^{(l,n)} \cdot \mathbf{x}_n & \text{for } l = 1 \\ \mathbf{W}^{(l,n)} \cdot \mathbf{z}_n^{(l-1)} & \text{for } l = 2 \dots L - 1, \end{cases} \quad (3)$$

where $\mathbf{x}_n = [x_{1,n} \cdots x_{D,n}]^T$ is the n th column vector of the $D \times N$ matrix \mathbf{X} in input layer. Here, we assume D features with each N data. The weight matrices for the layer $l = 1$ and for the layers $l = 2, 3, \dots, L - 1$ associated with \mathbf{x}_n are given by

$$\mathbf{W}^{(1,n)} = \begin{bmatrix} w_{1,1}^{(1,n)} & w_{1,2}^{(1,n)} & \cdots & w_{1,D}^{(1,n)} \\ w_{2,1}^{(1,n)} & w_{2,2}^{(1,n)} & \cdots & w_{2,D}^{(1,n)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M,1}^{(1,n)} & w_{M,2}^{(1,n)} & \cdots & w_{M,D}^{(1,n)} \end{bmatrix} \text{ and} \quad (4)$$

$$\mathbf{W}^{(l,n)} = \begin{bmatrix} w_{1,1}^{(l,n)} & w_{1,2}^{(l,n)} & \cdots & w_{1,M}^{(l,n)} \\ w_{2,1}^{(l,n)} & w_{2,2}^{(l,n)} & \cdots & w_{2,M}^{(l,n)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M,1}^{(l,n)} & w_{M,2}^{(l,n)} & \cdots & w_{M,M}^{(l,n)} \end{bmatrix}, \quad (5)$$

respectively. In this study, three types of the commonly used activation functions are evaluated, namely rectifier, logistic sigmoid, and hyperbolic tangent functions. The Rectified Linear Unit (Relu) [26] function, given by Equation (6), is known for ramp function that allows the model to easily obtain sparse representation.

$$H(a) = \max(0, a). \quad (6)$$

The logistic sigmoid function is a nonlinear activation function that derives smooth thresholding curve for artificial neural network. The output range of the standard logistic sigmoid function is from 0 to 1, from the following equation:

$$H(a) = \frac{1}{1 + e^{-a}}. \quad (7)$$

The drawback of standard logistic sigmoid function is that strongly negative inputs are mapped to near zero, which can cause a neural network to become stuck during training. The hyperbolic tangent function, given by Equation (8), is a differential nonlinear activation function such the negative inputs are mapped to large negative values and the zero inputs are mapped near zero.

$$H(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}. \quad (8)$$

All these activation functions are bounded, nonlinear, monotonic, and continuously differentiable. The universal approximation theorem [27] shows that a feedforward neural network with three layers and finite number of nodes can approximate any continuous function under moderate assumptions on the activation function in any desired accuracy. However, some highly nonlinear problems need more hidden layers and nodes, since the degree of nonlinearity depends on the number of layers and nodes.

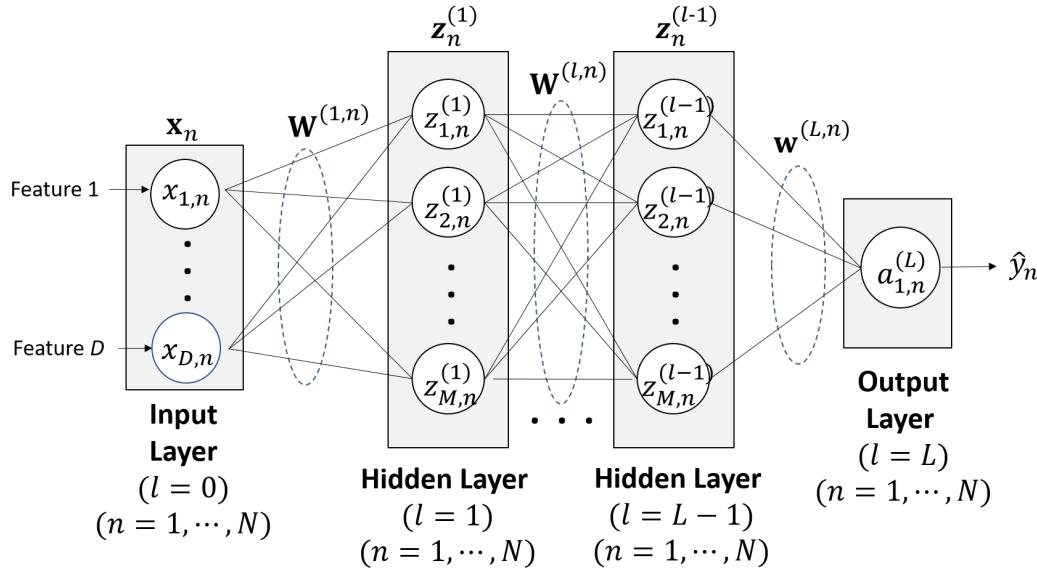


Figure 5. Block diagram of ANN-MLP with D features.

The ANN learning is obtained by updating the weights along the MLP neural network in consecutive iterations of feedforward and backpropagation procedures. Using Equation (2), the feedforward computation is performed on the following equation:

$$\mathbf{z}_n^{(L-1)} = H(\mathbf{W}^{(L-1,n)} H(\mathbf{W}^{(L-2,n)} H(\cdots \mathbf{W}^{(2,n)} H(\mathbf{W}^{(1,n)} \mathbf{x}_n))). \quad (9)$$

The prediction value \$\hat{y}_n\$ from the final output of feedforward procedure is \$a_{1,n}^{(L)}\$, which is linear output of \$\mathbf{z}_n^{L-1}\$ and \$\mathbf{w}^{(L)} = [w_{1,1}^{(L)} w_{1,2}^{(L)} \cdots w_{1,M}^{(L)}]\$ at the last layer without applying activation function as given by

$$\hat{y}_n = a_{1,n}^{(L)} = \mathbf{w}^{(L,n)} \cdot \mathbf{z}_n^{(L-1)}. \quad (10)$$

The objective of the training is minimize the loss function by adjusting the ANN weights \$\mathcal{W} = \{\mathbf{W}^{(l,n)}\}_{l=1,\dots,L}\$ and \$n=1,\dots,N\$. The loss function is given by

$$J(\mathcal{W}) = \frac{1}{N} \sum_{n=1}^N |\hat{y}_n - y_n|^2 + \frac{\alpha}{2} \sum_{n=1}^N \sum_{l=1}^L \|\mathbf{W}^{(l,n)}\|_2^2, \quad (11)$$

where \$\hat{y}_n\$ is prediction value for given weight \$\mathcal{W}\$, and \$y_n\$ is measured path loss values. \$\frac{1}{N} \sum_{n=1}^N |\hat{y}_n - y_n|^2\$ is the mean square error (MSE), \$\frac{\alpha}{2} \sum_{n=1}^N \sum_{l=1}^L \|\mathbf{W}^{(l,n)}\|_2^2\$ is an L2-regularization term that penalizes the ANN model from overfitting, and \$\alpha\$ is the magnitude of the invoked penalty.

After feedforward phase, adaptive updates for the weight on each connections are conducted by backpropagation [28]. Starting from initial random weights, the backpropagation is repeatedly updating these weights based on gradient descent of loss function with respect to the weights.

$$\begin{aligned}\frac{\partial J}{\partial w_{j,i}^{(l,n)}} &= \frac{\partial J}{\partial a_{j,n}^{(l)}}(z_{i,n}^{(l-1)}) \\ \frac{\partial J}{\partial a_{j,n}^{(l)}} &= \begin{cases} H'(a_{j,n}^{(l)}) \sum_{k=1}^M w_{k,j}^{(l+1,n)} \frac{\partial J}{\partial a_{k,n}^{(l+1)}} & \text{for } l = 1 \dots L-1 \\ \frac{\partial J}{\partial a_{1,n}^{(l)}} & \text{for } l = L, \end{cases}\end{aligned}\quad (12)$$

where $H'(a_{j,n}^{(l)}) = dH/d a_{j,n}^{(l)}$ is the derivative for the corresponding activation function. Finally, the weights are updated as follows.

$$w_{j,i}^{(l,n)} \leftarrow w_{j,i}^{(l,n)} - \lambda \frac{\partial J}{\partial w_{j,i}^{(l,n)}} = w_{j,i}^{(l,n)} - \lambda \frac{\partial J}{\partial a_{j,n}^{(l)}}(z_{i,n}^{(l-1)}), \quad (13)$$

where λ is the learning rate, the hyperparameter for controlling the step-size in parameter updates. This backward pass propagates from the output layer to previous layers with updating weights for minimizing the loss, as shown in Equation (13). After finishing backpropagation up to the first layer's weights, it continues to the next iteration of another feedforward and backpropagation process until the weight values are converged certain tolerance level, which is another hyperparameter determining the model. For backpropagation optimization, the Quasi-Newton method, which iteratively approximates the inverse Hessian with $O(N^2)$ time complexity, is applied. The Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) [29–31] is most practical batch method of the Quasi-Newton algorithms and we use the Scipy version of it.

5. Gaussian Process

The Gaussian process is kernel-based fully Bayesian regression algorithm that computes a posterior predictive distributions for new data. The Gaussian process is the extension of multivariate Gaussians, which is useful in modeling collections of real-valued variables by its analytical properties. A Gaussian process is a stochastic process based on a sub-collection of random variables that has a multivariate Gaussian distribution. A collection of random variables $\{h(x) : x \in \mathbf{X}\}$ is drawn from a Gaussian process with mean function $m(\cdot)$ and covariance function $k(\cdot, \cdot)$ if any finite set of elements $x_1, \dots, x_m \in \mathbf{X}$, the associated finite set of random variables $h(x_1), \dots, h(x_m)$ have distribution,

$$\begin{bmatrix} h(x_1) \\ \vdots \\ h(x_m) \end{bmatrix} \sim N \left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_m) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_m) \\ \vdots & \ddots & \vdots \\ k(x_m, x_1) & \cdots & k(x_m, x_m) \end{bmatrix} \right). \quad (14)$$

The mean expression of the Gaussian process model consists of the mean function $m(x)$ and the covariance function, i.e., kernel $k(x, x')$, in the following equation. The final expression is expressed as follow.

$$h(x) \sim GP(m(\cdot), k(\cdot, \cdot)) \quad (15)$$

The mean function and covariance function are aptly named as follows,

$$m(x) = E(h(x)) \quad (16)$$

$$k(x, x') = E[(h(x) - m(x))(h(x') - m(x'))] \quad (17)$$

for any $x, x' \in \mathbf{X}$. A function $h(\cdot)$ drawn from a Gaussian process prior is an extremely high-dimensional vector from a multivariate Gaussian. In other words, the Gaussian process provides a method for modeling probability distributions over functions.

Let $\mathbf{S} = \{(x^{(i)}, y^{(i)})_{i=1}^m\}$ be a training set with unknown distribution. In the Gaussian process regression model,

$$y^{(i)} = h(x^{(i)}) + \epsilon^{(i)}, i = 1, \dots, m \quad (18)$$

where the $\epsilon^{(i)}$ are noise variables with independent $\mathbf{N}(0, \sigma^2)$ distributions. Similar to Bayesian linear regression, a prior distribution over functions $h(\cdot)$ is assumed that it is regarded as a zero-mean Gaussian process prior,

$$h(\cdot) \sim GP(0, k(\cdot, \cdot)) \quad (19)$$

for some valid covariance function $k(\cdot, \cdot)$.

Let X_* be testing points drawn from the same unknown distribution. The marginal distribution over any set of input points belonging to X must have a joint multivariate Gaussian distribution

$$\begin{bmatrix} h(X) \\ h(X_*) \end{bmatrix} \sim N \left(0, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix} \right). \quad (20)$$

Based on noise assumption ($\epsilon^{(i)} \sim N(0, 1)$), the noise term can be derived as

$$\begin{bmatrix} \epsilon \\ \epsilon_* \end{bmatrix} \sim N \left(0, \begin{bmatrix} \epsilon^2 \mathbf{I} & 0 \\ 0 & \epsilon^2 \mathbf{I} \end{bmatrix} \right) \quad (21)$$

As a result, the sum of independent Gaussian random variables is also Gaussian, thus

$$\begin{bmatrix} y(X) \\ y(X_*) \end{bmatrix} = \begin{bmatrix} h(X) \\ h(X_*) \end{bmatrix} + \begin{bmatrix} \epsilon \\ \epsilon_* \end{bmatrix} \sim N \left(0, \begin{bmatrix} K(X, X) + \epsilon^2 \mathbf{I} & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) + \epsilon^2 \mathbf{I} \end{bmatrix} \right) \quad (22)$$

Finally, using the rules for conditioning Gaussians, we have

$$y_* | y, X, X_* \sim \mathbf{N}(\mu^*, \Sigma^*), \quad (23)$$

where μ^* is predicted mean values and Σ^* is standard deviation values for given distribution, which are given by

$$\mu^* = K(X_*, X)(K(X, X) + \sigma^2 \mathbf{I})^{-1}y \quad (24)$$

$$\Sigma^* = K(X_*, X_*) + \sigma^2 \mathbf{I} - K(X_*, X)(K(X, X) + \sigma^2 \mathbf{I})^{-1}K(X, X_*) \quad (25)$$

Gaussian process regression derives the model based on the relationship between input (\mathbf{X}) and output (\mathbf{Y}) being given by the unknown function h . The input data \mathbf{X} are the feature values such as distance, frequency, etc. The output data \mathbf{Y} are the measured path loss value for given input data. Through the Gaussian process, we learn the relationship $h(\cdot)$ between input and output.

We can estimate the posterior distribution using Bayesian inference, which consists of a prior distribution and likelihood function. Gaussian process is an attractive model for use in regression problems since it allows quantifying uncertainty of predictions due to the errors in the parameter estimation as well as inherent noise in the problem. Furthermore, the model selection and hyperparameter selection methods used in the Bayesian method are directly applicable to Gaussian

processes. Similar to locally-weighted linear regression, Gaussian process regression is non-parametric and consequently can model intrinsically arbitrary functions of the input points.

The Gaussian process uses the kernel as a function of covariance that can model various distributions in terms of mean square derivatives for a variety of different data. That is, when x and x' are close values, the covariance value between those points becomes large, and, inversely, when x and x' are distant values, the covariance value becomes small. In other words, the closer the value is, the larger the weight it gets, and vice versa. In addition, the distribution of the near and far values is smoothly derived during Gaussian process learning.

$$k(x_i, x_j) = \exp\left(-\frac{1}{2d(\frac{x_i}{length_scale}, \frac{x_j}{length_scale})^2}\right) \quad (26)$$

Kernels (also called *covariance functions* in the context of Gaussian processes) are a crucial ingredient of Gaussian processes which determine the shape of the prior and posterior of the Gaussian process. They express the function being learned by defining the *similarity* of two data points combined with the assumption that similar data points have similar target values. The kernel can be divided into stationary and non-stationary kernels. Stationary kernels only depend on the distance of two data points and not on their absolute values, therefore they do not change in the conversion of input space. On the other hand, non-stationary kernels depend on the specific value of the data point. Stationary kernels can further be classified into isotropic and anisotropic kernels, where isotropic kernels are also unaltered by a rotation of input space. Common types of kernel function are Constant kernel, Radial-basis function (RBF) kernel, Matern kernel, Rational quadratic kernel, Exp-Sine-Squared kernel, and Dot-Product kernel. Alpha value is a value to adjust the noise level, and it is added to the diagonal values of the kernel matrix. The larger is the noise, the more training data should be applied.

6. Measurement System and Scenarios

Path loss was measured for three frequencies 450, 1450, and 2300 MHz. The measurement system is composed of a transmitter and a receiver, as shown in Figure 6. The transmitter consisted of cw-generator, power amp, transmitting antenna, and RF cable for connecting each system component. The receiver was composed of the continuous wave (CW) signal receiving part and the GPS signal receiving part. The CW signal receiving part was composed of a receiving antenna, a low noise amplifier (LNA), a spectrum analyzer, and a laptop. The receiving antenna, LNA, and spectrum analyzer were connected by RF cables, and the spectrum analyzer and the laptop were connected by GPIB cable. Both transmitting and receiving antennas had omni-directional radiation patterns. The GPS signal processing system was equipped to obtain accurate distance data between the transmitter and receiver. The GPS signal receiving part consisted of a GPS receiving antenna, a GPS signal processing board, and a laptop. Signals from the GPS signal processing board are transferred to the laptop through a serial port using the RS-232 protocol. The GPS data and CW signal power levels were synchronized using the measured time information, and GPS data were converted into position data.

Measurements were conducted in the suburbs of a small town called Nonsan, which consists of residential areas with 1–4-story buildings, agricultural land, and very low hills. As shown in Figure 7, the transmitting system was installed on the roof of a four-story building, and the transmitting antenna height was 15 m above ground level. The receiving system was mounted on a vehicle and the receiving antenna height was 2 m above ground level. Figure 8 depicts the measurement path and received power level using a color bar. The distance between the transmitter and receiver was within about 3.5 km and the vehicle was moved to receive the power at various distances. The moving speed of the vehicle was maintained so that the separation distance between the measuring points ranged from 25 to 30 cm.

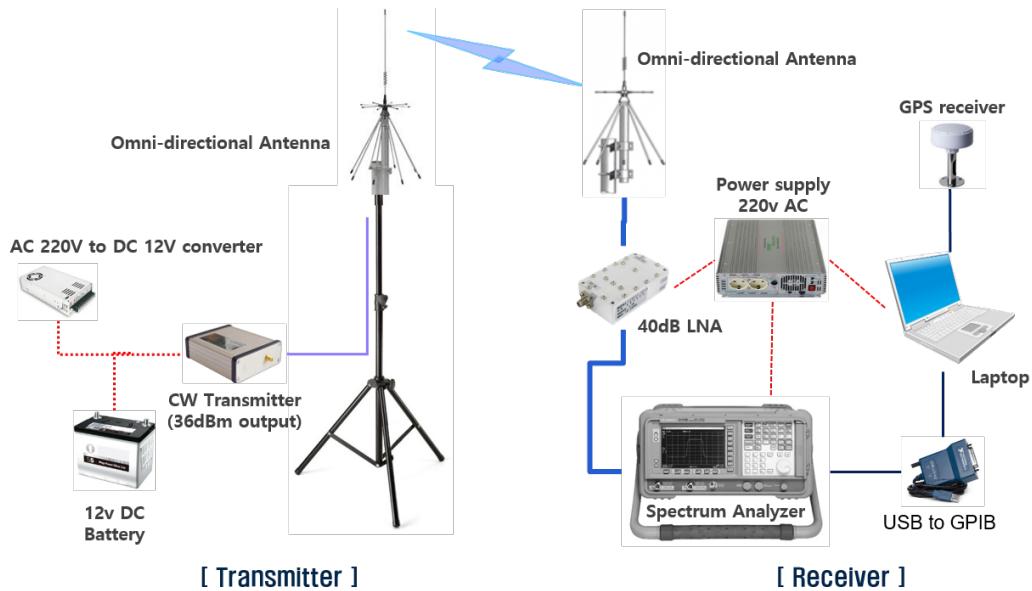


Figure 6. Block diagram of the measurement system.



Figure 7. Installation environment of the measurement system.

To reduce the Doppler effect due to vehicle movement, the vehicle speed was kept at 30 km/h or less. In the data refining process, the power level measure at 30 km/h or more was filtered out, and then the data were time-averaged to remove the effect of short-term fading. Since the bandwidth used for the measurement was 10 kHz, received power was ideally -134 dBm; however, the noise level observed during field measurements was around -110 dBm. Therefore, -100 dBm, which is above 10 dB of noise signal, was defined as an effective reception level for stable data refinement.

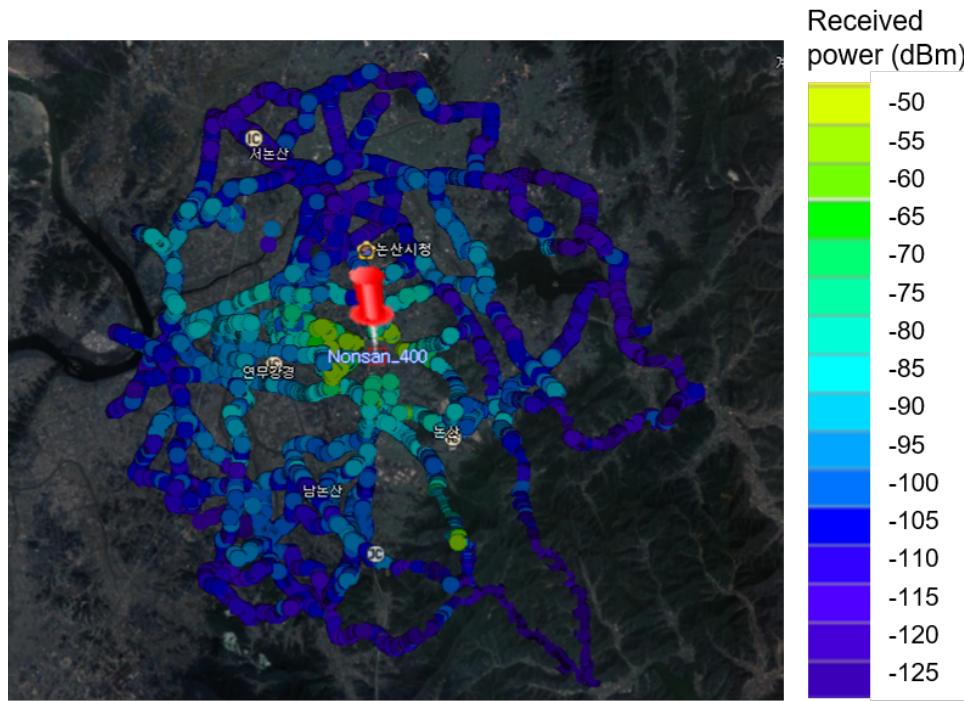


Figure 8. Measurement path and received power level.

7. Results and Discussion

7.1. Evaluation Metrics

This section describes experimental results for the different models in three frequencies. The path loss values were predicted with distance from 1 to 3.1 km (in logarithmic distance 0.0–0.49). The predicted path loss values were calculated with the measured data for evaluating prediction error. We applied various prediction error metrics such as root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), mean squared logarithmic error (MSLE), and squared R (R^2) with different perspective. RMSE is defined by

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^N |\hat{\mathbf{y}}_n - \mathbf{y}_n|^2}. \quad (27)$$

RMSE represents the square root of the differences between predicted values and observed values or the quadratic mean of these differences. Although RMSE and MSE are similar in terms of models scoring, they are not always immediately interchangeable for gradient based methods.

MAE is defined by

$$\text{MAE} = \frac{1}{N} \sum_{n=1}^N |\hat{\mathbf{y}}_n - \mathbf{y}_n|. \quad (28)$$

MAE is a linear index, which means that all the individual differences are weighted equally in the average. It tends to be less sensitive to outliers than MSE.

MAPE is defined by

$$\text{MAPE} = \frac{100\%}{N} \sum_{n=1}^N \left| \frac{\hat{\mathbf{y}}_n - \mathbf{y}_n}{\mathbf{y}_n} \right|. \quad (29)$$

MAPE is considered as the weighted version of MAE and therefore the optimal constant predictions for MAPE were found to be the weighted median of the target values [32]. MAPE is

commonly used as a loss function for regression problems and in model evaluation, because of its very intuitive interpretation in terms of relative error.

MSLE is just MSE calculated in logarithmic scale as

$$\text{MSLE} = \frac{1}{N} \sum_{n=1}^N (\log(\mathbf{y}_n + 1) - \log(\hat{\mathbf{y}}_n + 1))^2. \quad (30)$$

$$= \text{MSE}(\log(\mathbf{y}_n + 1), \log(\hat{\mathbf{y}}_n + 1)). \quad (31)$$

From the perspective of logarithm, it is always better to predict more than the same amount less than target in that RMSLE penalizes an under-predicted estimate greater than an over-predicted estimate.

R^2 is closely related to MSE, but has the advantage of no scale. Thus, R^2 has a value between $-\infty$ and 1, regardless of the output value. To make it more clear, this baseline MSE can be regarded as the MSE that the simplest possible model would get. The simplest possible model would be to always predict the average of all samples. A value close to one indicates a model with little error, and a value close to zero represents a model very close to the baseline. Thus, R^2 represents how good our model is against the naive mean model.

$$R^2 = 1 - \frac{\sum_{n=1}^n (\mathbf{y}_n - \hat{\mathbf{y}}_n)^2}{\sum_{n=1}^n (\mathbf{y}_n - \bar{\mathbf{y}}_n)^2}, \text{ where } \bar{\mathbf{y}} = \frac{1}{n} \sum_{n=1}^n \mathbf{y}_n. \quad (32)$$

7.2. Activation Functions for ANN-MLP Model

A key element in the ANN configuration is the activation function that determines the nonlinear transformation for the given learning data. To find the optimal activation function, we examined the prediction loss values (RMSE, MAE, MAPE, MSLE, and R2) which are processed with the validation set, which was initially sampled separately from learning data. To minimize the variance from hyperparameters in learning ANN-MLP models, L-BFGS algorithm was mainly used, which is a batch type of computational optimization method, different from other stochastic mini-batch approaches. For the reference, the fixed hyperparameter of learning rate, epoch, and tolerance rate were set to 0.001, 1000, and 0.00001, respectively, throughout the course of experiments.

For ANN models, to derive a better fit for various distribution and express nonlinearity of path loss characteristics, choosing the right activation function is crucial. Based on the results in Figures 9–11, we can find quite diverse figures from alternating activation function network. In Table 2, we evaluate three different activation functions for ANN-MLP network architecture. Comparing hyperbolic tangent and ReLU activation function, the performance of the sigmoid shows better performance in overall metrics. Even though many different types of activation functions exist, we designated the sigmoid function for simplifying network architecture and further performance comparison.

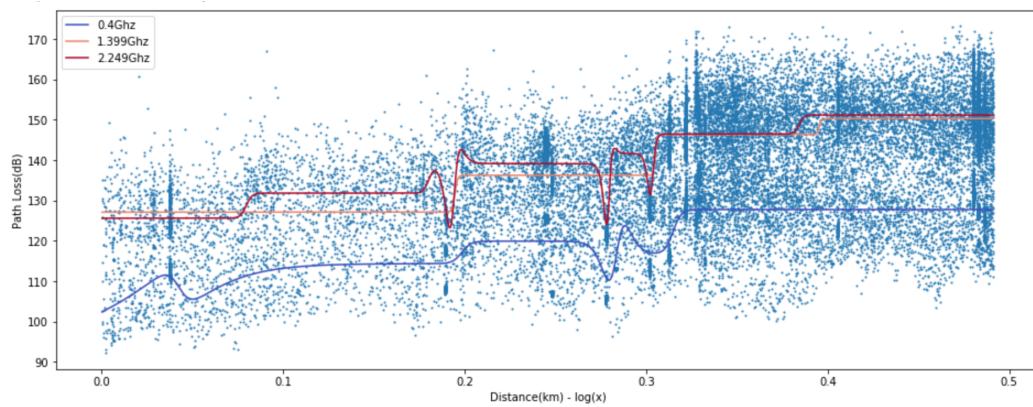


Figure 9. Path loss prediction based on ANN-MLP with Sigmoid activation function for 450, 1450, and 2300 MHz.

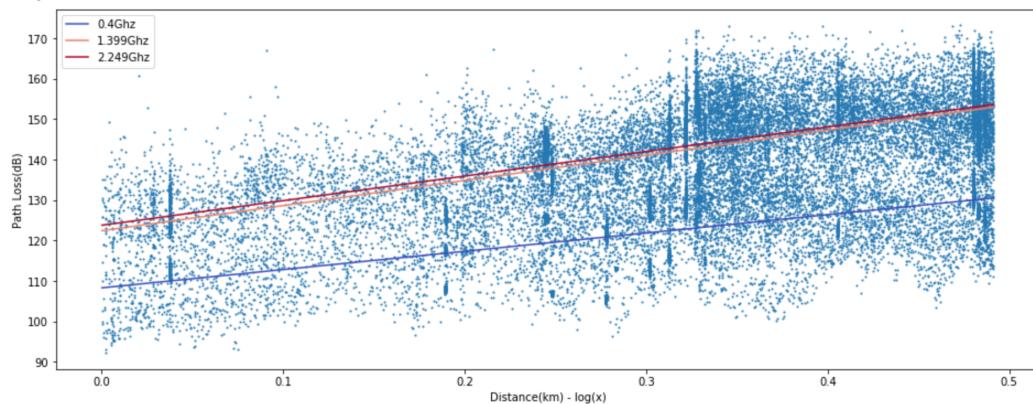


Figure 10. Path Loss prediction based on ANN-MLP with ReLU activation function for 450, 1450, and 2300 MHz.

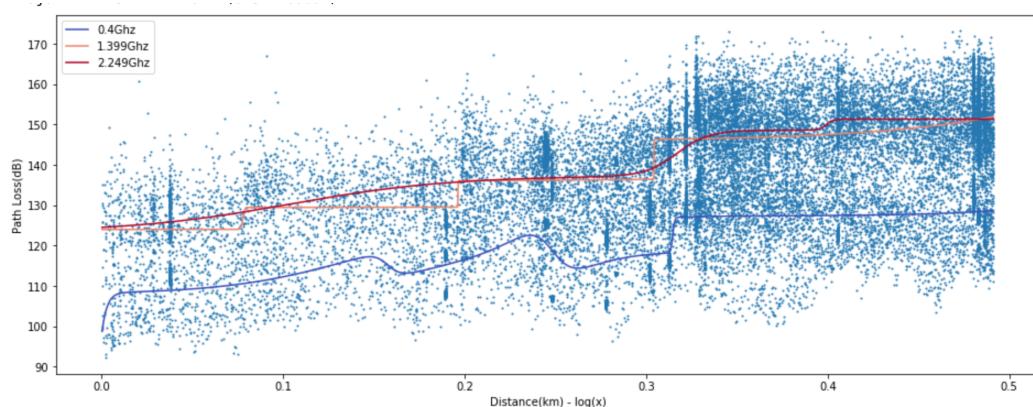


Figure 11. Path Loss prediction based on ANN-MLP with tangent hyperbolic activation function for 450, 1450, and 2300 MHz.

Table 2. Comparison of prediction accuracy of different metrics for Gaussian process model.

Model	Error (dB)	RMSE	MAE	MAPE	MSLE	R2
(ANN-MLP) (Sigmoid)	450 MHz	7.87647	5.89603	4.84501	0.00407	0.39747
	1450 MHz	8.96238	6.80976	4.93310	0.00419	0.46041
	2300 MHz	8.23471	6.20676	4.41919	0.00343	0.48938
	Overall	8.52169	6.48413	4.87674	0.00405	0.66402
(ANN-MLP) (ReLU)	450 MHz	8.42747	6.84639	5.63144	0.00468	0.31022
	1450 MHz	9.32876	7.22502	5.22711	0.00453	0.41540
	2300 MHz	8.79632	6.75497	4.83320	0.00394	0.41735
	Overall	9.17173	7.23891	5.43902	0.00466	0.61081
(ANN-MLP) (tanh)	450 MHz	7.93312	6.02763	4.95792	0.00414	0.38877
	1450 MHz	8.90661	6.77473	4.90238	0.00412	0.46711
	2300 MHz	8.54053	6.58882	4.72276	0.00374	0.45075
	Overall	8.54226	6.54968	4.93007	0.00407	0.66239

7.3. Prediction Accuracy of Path Loss Model

We compared the prediction loss values (RMSE, MAE, MAPE, MSLE, and R2) of the proposed ANN-MLP with other models. The ANN-MLP model is presented in Figure 12, and the popular log linear regression model and two-ray model [33] are presented in Figures 13 and 14. Received signal consists of the line-of-sight path and the ground reflected path in the two-ray model and the resulting path loss is given by

$$PL = \begin{cases} PL_c + 40 \log_{10}(d/d_c) & \text{for } d \geq d_c \\ PL_c + 20 \log_{10}(d/d_c) & \text{for } d < d_c \end{cases}, \quad (33)$$

where break point distance $d_c = 4h_t h_r / \lambda$ and the path loss at the distance $PL_c = 20 \log_{10} (16\pi h_t h_r / \lambda^2)$ are given as a function of transmitting antenna height h_t , receiving antenna height h_r , and wavelength λ . Given that the break point distances are 180, 580, and 920 m for frequencies 450, 1450, and 2300 MHz, respectively, and $h_t = 15$ m and $h_r = 2$ m, the path loss values beyond the break point distance are shown in Figure 14.

The following sections consider the results of modeling with Gaussian processes shown in Figure 15. Gaussian processes can be modeled for mean values and standard deviations. The mean value is a model of the path loss prediction value, and the standard deviation predicts the range of the expected path loss values. The standard deviation can be used to predict the minimum and maximum path loss that can be distributed at the given input. The Gaussian process was trained using RBF kernel function as

$$k(x_i, x_j) = \exp\left(-\frac{1}{2} d\left(\frac{x_i}{l}, \frac{x_j}{l}\right)^2\right), \quad (34)$$

The RBF kernel is a stationary kernel and also known as the *squared exponential kernel*. It is characterized by a length-scale parameter l , which is either a scalar or a vector with the same number of dimensions as the inputs. This kernel is infinitely differentiable, which implies that Gaussian processes with this kernel as a covariance function have mean square derivatives of all orders, and are thus very smooth.

In Table 3, we evaluate the results obtained for linear, ANN-MLP, and Gaussian process model. The prediction loss values (RMSE, MAE, MAPE, and MSLE) from Gaussian process and linear model are higher than ANN-MLP model and the lower R^2 value shows that its prediction values are not as good fits as ANN-MLP model, which means ANN-MLP slightly outperforms Gaussian process in the path loss prediction. Based on the results in Figures 12–15, we can easily find that ANN-MLP model is more sensitive and tweak well on complex data distribution. On the other hand, the straight line is

too simple to represent the change in pathloss over distance, and Gaussian process model also shows smooth shapes reluctant to tweak on localized distribution. Unlike other models, the two-ray model for $d \geq d_c$ are independent of frequency, as shown in Equation (33). The two-ray model produces much prediction error compared with other models since its assumption of line-of-sight sight and flat terrain are not guaranteed.

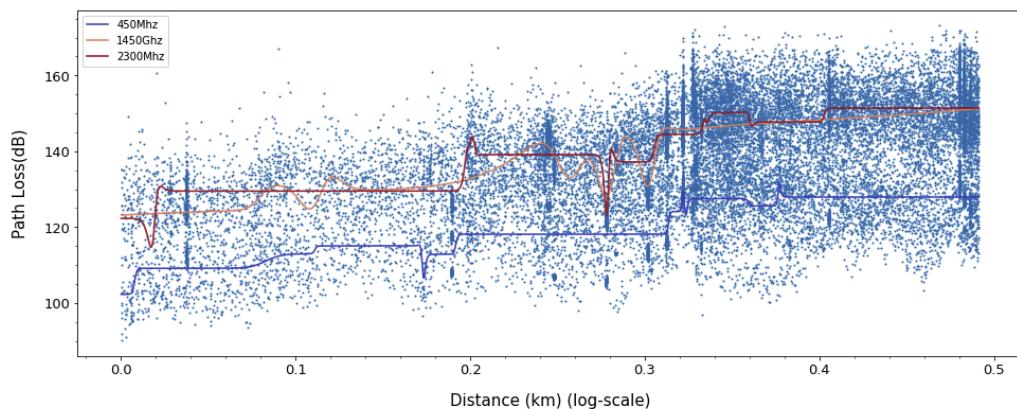


Figure 12. Path loss prediction based on ANN-MLP for 450, 1450, and 2300 MHz.

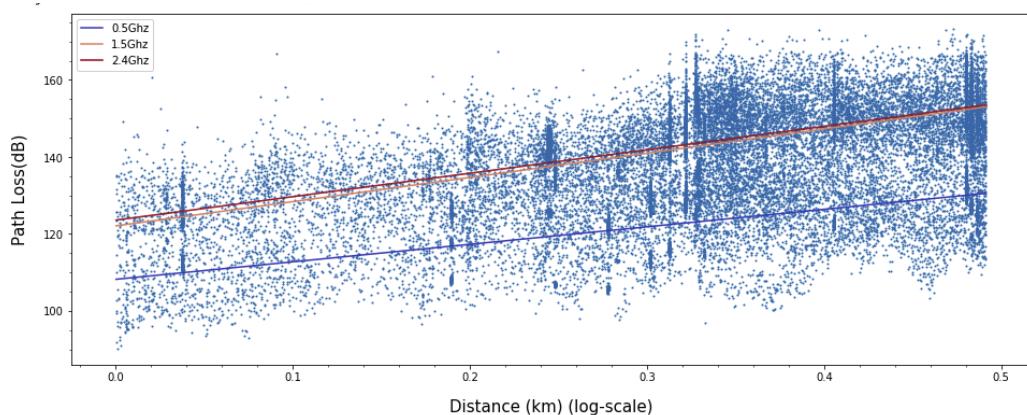


Figure 13. Path loss prediction based on linear model.

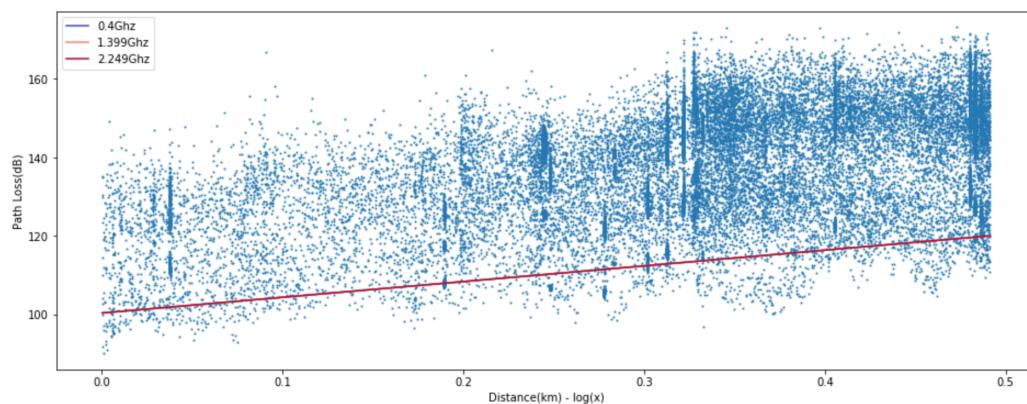


Figure 14. Path loss prediction based on two-ray model.

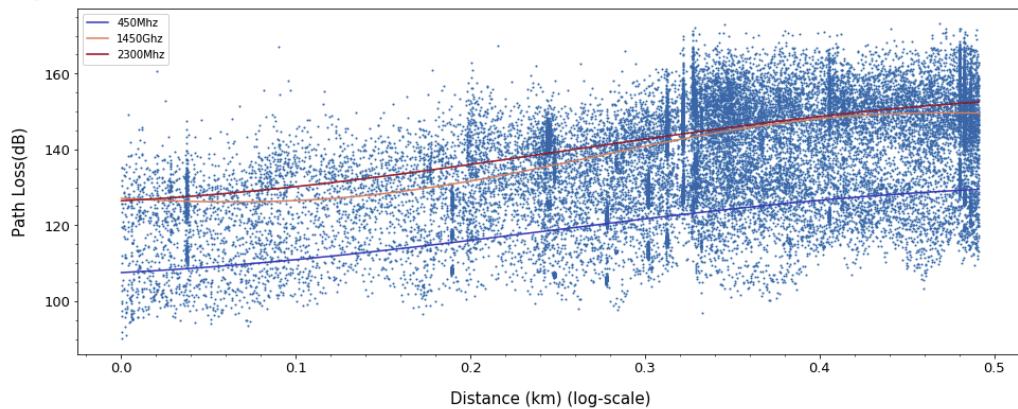


Figure 15. Path loss prediction based on Gaussian process for 450, 1450, and 2300 MHz.

The first Fresnel radius at the midpoint between the transceiver is given by $r = \sqrt{(\frac{\lambda d}{4})}$, which range from 6 to 24 m. Given the antenna heights $h_t = 15$ m and $h_r = 2$ m, the terrain and objects are likely to be inside of the first Fresnel zone. However, the places for measurement campaign is not ideal flat terrain, but they have different ground height for each receiver position. Accordingly, the clearances of the first Fresnel zone are highly variable. As a result, ANN is more practical in such environments.

Table 3. Comparison of prediction accuracy of different metrics for Gaussian process model.

Model	Error (dB)	RMSE	MAE	MAPE	MSLE	R2
ANN-MLP	450 MHz	7.87647	5.89603	4.84501	0.00407	0.39747
	1450 MHz	8.96238	6.80976	4.93310	0.00419	0.46041
	2300 MHz	8.23471	6.20676	4.41919	0.00343	0.48938
	Overall	8.52169	6.48413	4.87674	0.00405	0.66402
Linear Model	450 MHz	8.52682	6.93022	5.68143	0.00476	0.31199
	1450 MHz	9.27597	7.19697	5.17684	0.00442	0.41033
	2300 MHz	8.98500	6.96090	4.95213	0.00406	0.38408
	Overall	9.22838	7.30487	5.46315	0.00467	0.60944
Two-ray Model	450 MHz	12.88614	10.64310	8.31157	0.01084	-0.58098
	1450 MHz	30.69274	29.15810	20.03675	0.05465	-5.45023
	2300 MHz	31.50410	30.23772	20.66583	0.05734	-6.86500
	Overall	26.46132	23.34631	16.33805	0.04094	-2.24508
Gaussian Process	450 MHz	8.53235	6.82668	5.60452	0.00479	0.29261
	1450 MHz	9.30737	7.05678	5.11159	0.00447	0.40428
	2300 MHz	8.73691	6.83405	4.85415	0.00387	0.44244
	Overall	8.94361	6.97418	5.23819	0.00446	0.63462

7.4. Prediction Accuracy of Combined Path Loss and Shadowing Model

As shown in the following equation, a simple log-distance model based on existing deterministic regression analysis has to rely on predetermined variables PL_0 and n , and empirically measured log-normal shadowing X_σ .

$$PL = PL_0 + 10n \log_{10}(d) + X_\sigma, \text{ for } d \geq 1 \text{ km}, \quad (35)$$

where PL_0 is the path loss at 1 km distance. On the other hand, the learning-based model can predict the targeted values based on actual distribution of training data, which is able to build more accurate forms of regression model with nonlinear characteristics. In particular, steep slope or complex shaped data distribution can be accurately estimated in this model, since nonlinear model can fit better to

those cases than linear model. In addition, the key variables are determined by PCA by evaluating impact level upon co-distribution of input features. By analyzing two variables, the proposed model comprises of ANN-MLP based path loss model and Gaussian process based shadowing model as

$$PL = PL_ANN(d) + GP_{\sigma}, \text{ for } d \geq 1 \text{ km} \quad (36)$$

In Figure 16 and Table 4, we assess *prediction coverage* that quantifies how closely the predicted values from a model match the actual test data within the standard deviation range. More specifically, the prediction coverage is a ratio of the number of measured data in the range of α to β to total number of measured data, where α and β are given by

$$\alpha = \begin{cases} PL_0 + 10n \log_{10}(d) - r\sigma_1 & \text{for the linear model in Equation (35)} \\ PL_ANN(d) - r\sigma_2 & \text{for the proposed model in Equation (36)} \end{cases} \quad (37)$$

$$\beta = \begin{cases} PL_0 + 10n \log_{10}(d) + r\sigma_1 & \text{for the linear model in Equation (35)} \\ PL_ANN(d) + r\sigma_2 & \text{for the proposed model in Equation (36)} \end{cases} \quad (38)$$

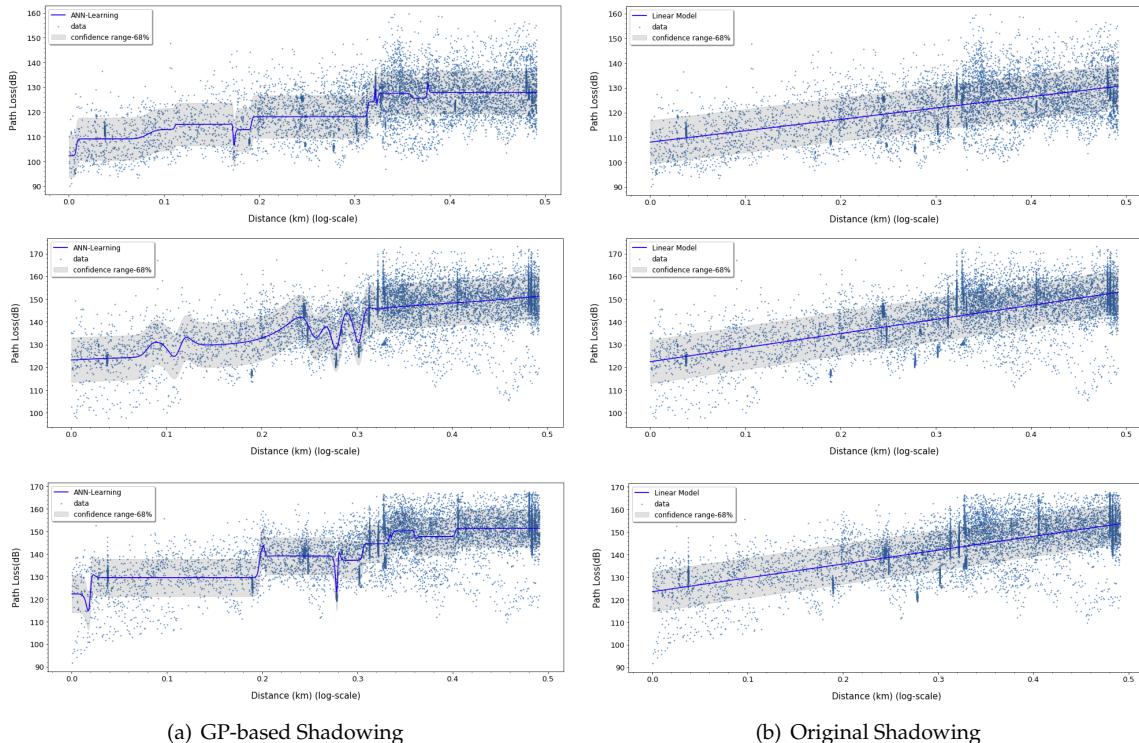


Figure 16. Training graph for the proposed and the linear model with $1\times$ standard deviation.

The conventional shadowing effect is a simple assumption that the mean difference along the predicted value forms the Gaussian distribution, whereas the deviation value derived by the Gaussian Process is based on the correlation analysis. It is a mathematical model for correlation analysis so that errors can be accurately derived. As can be seen from the results in Table 3, the shadowing effect of the existing linear model (1450 MHz, σ : 81.5%, 2σ : 99.9%, 3σ : 100%) shows different coverage than the theoretical value of Gaussian distribution. On the other hand, the predictive model coverage derived from the GP model reflects the actual Gaussian distribution (based on 1450 MHz, σ : 68.4%, 2σ : 95.0%, 3σ : 99.2%). The two models have ideal coverage values of 68%, 95%, and 99.7% for $r = 1, 2, 3$ when they perfectly match the measure data. For overall frequencies, the proposed model shows around 69.6%, 94.8%, and 99.2% of coverage, whereas the linear model marks 80%, 98%, and 99.8% of coverage

for $r = 1, 2, 3$, respectively. These results mean the proposed model predicts shadowed path loss more accurately than the linear model.

Table 4. Prediction coverage by different models.

Frequency	450 MHz	1450 MHz	2300 MHz	Overall
Linear Model ($r = 1 / 68\%$)	76.5%	81.5%	82.1%	80%
Linear Model ($r = 2 / 95\%$)	95.4%	99.9%	100.0%	98%
Linear Model ($r = 3 / 99.7\%$)	99.6%	100.0%	100.0%	99.8%
Proposed Model ($r = 1 / 68\%$)	68.8%	68.4%	71.7%	69.6%
Proposed Model ($r = 2 / 95\%$)	94.2%	95.0%	95.2%	94.8%
Proposed Model ($r = 3 / 99.7\%$)	99.4%	99.2%	99.2%	99.2%

8. Conclusions

In this paper, we develop a new approach for the prediction of the path loss based on machine learning techniques: dimensionality reduction, ANN-MLP, and Gaussian process. Tests were designed to evaluate whether dimensionality reduction could support the same path loss prediction accuracy as well as provide confidence level in regression problem based on combining ANN-MLP and Gaussian Process. Using PCA method, dimensionality reduction of the four feature model led to more generalized model with one feature and reduced significant amount of time in training model. The one variable model in ANN-MLP and GP model takes 10% and 1%, respectively, of the training time compared to the four-variable model. By deriving variance from GP, we can design a data-driven model for shadowing effect. The combined path loss and shadowing model, the final outcome of this study, accurately predicts the measured path loss with coverage error less than 1.6 %, whereas the conventional linear log-distance plus log-normal shadowing model has coverage error less than 12%. The new approach would be beneficial for the site-specific design of wireless sensor network with high reliability.

We could extend this study for various ANN structures with different numbers of hidden layers and nodes. Future extension could also develop the ANN-based model that is able to predict only the extra path loss above the reference distance. This makes it possible to develop a model for any reference distance, which will further improve the practicality of the model. It could be possible to apply the proposed method to the development of predictive models with the influence of other variables such as antenna height and building height.

Author Contributions: Conceptualization and methodology, H.-S.J. and J.P.; software, C.P.; validation, formal analysis, and investigation, H.-S.J. and C.P.; resources and data curation, C.P. and H.K.C.; writing—original draft preparation, H.-S.J. and C.P.; writing—review and editing, H.-S.J. and E.L. and J.P.; visualization, C.P.; supervision, J.P.; project administration, J.P.; funding acquisition, J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Agency for Defense Development (ADD).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hata, M. Empirical formula for propagation loss in land mobile radio services. *IEEE Trans. Veh. Technol.* **1980**, *29*, 317–325. [[CrossRef](#)]
2. Seidel Y.S.; Rappaport S.T.; Jain, S.; Lord, L.M; Singh, R. Path Loss, Scattering, and Multipath Delay Statistics in Four European Cities for Digital Cellular and Microcellular Radiotelephone. *IEEE Trans. Veh. Technol.* **1991**, *40*, 721–730. [[CrossRef](#)]
3. Okumura, Y.; Ohmori, E.; Kawano, T.; Fukuda, K. Field strength and its variability in VHF and UHF land mobile radio service. *Rev. Electr. Commun. Lab.* **1991**, *16*, 825–873.
4. COST Action 231. *Digital Mobile Radio Towards Future Generation Systems, Final Report*; Technical Report, EUR 18957; European Communities: Bruxelles, Belgium, 1999.

5. McGeehan, J.P.; Griffithes, J. Normalized Prediction Chart for Mobile Radio Reception. In Proceedings of the 4th International Conference on Antennas and Propagation 1985, Coventry, UK, 16–19 April 1985; pp. 395–399.
6. Lee, W.C.Y. *Mobile Communications Design Fundamentals*; John Wiley & Sons Inc: Hoboken, NJ, USA, 1993.
7. Akeyama, A.; Nagatsu, T.; Ebine, Y. Mobile radio propagation characteristics and radio zone design method in local cities. *Rev. Electr. Commun. Lab.* **1982**, *30*, 309–317.
8. Erceg, V.; Ghassemzadeh, S.; Taylor, M.; Li, D.; Schilling, D.L. Urban/suburban out-of-sight propagation modeling. *IEEE Commun. Mag.* **1992**, *30*, 56–61. [[CrossRef](#)]
9. Masui, H.; Ishii, M.; Sakawa, K.; Shimizu, H.; Kobayashi, T.; Akaike, M. Microwave path-loss characteristics in urban LOS and NLOS environments. In Proceedings of the 53rd IEEE VTC 2001 Spring, Rhodes, Greece, 6–9 May 2001; pp. 395–398.
10. Van Brandt, S.; Van Thielen, R.; Verhaevert, J.; Van Hecke, T.; Rogier, H. Characterization of Path Loss and Large-Scale Fading for Rapid Intervention Team Communication in Underground Parking Garages. *Sensors* **2019**, *19*, 2431. [[CrossRef](#)]
11. Tang, W.; Ma, X.; Wei, J.; Wang, Z. Measurement and Analysis of Near-Ground Propagation Models under Different Terrains for Wireless Sensor Networks. *Sensors* **2019**, *19*, 1901. [[CrossRef](#)]
12. Nilsson, M.G.; Gustafson, C.; Abbas, T.; Tufvesson, F. A Path Loss and Shadowing Model for Multilink Vehicle-to-Vehicle Channels in Urban Intersections. *Sensors* **2018**, *18*, 4433. [[CrossRef](#)]
13. Bi, J.; Wang, Y.; Li, Z.; Xu, S.; Zhou, J.; Sun, M.; Si, M. Fast Radio Map Construction by using Adaptive Path Loss Model Interpolation in Large-Scale Building. *Sensors* **2019**, *19*, 712. [[CrossRef](#)]
14. Jo, H.-S.; Yook, J. Path Loss Characteristics for IMT-Advanced Systems in Residential and Street Environments. *IEEE Antennas Wirel. Propag. Lett.* **2010**, *9*, 867–871. [[CrossRef](#)]
15. Zhang, Y.; Wen, J.; Yang, G.; He, Z.; Wang, J. Path loss prediction based on machine learning: Principle, method, and data expansion. *Appl. Sci.* **2019**, *9*, 1908. [[CrossRef](#)]
16. Piacentini, M.; Rinaldi, F. Path loss prediction in urban environment using learning machines and dimensionality reduction techniques. *Comput. Manag. Sci.* **2011**, *8*, 371. [[CrossRef](#)]
17. Oroza, C.A.; Zhang, Z.; Watteyne, T.; Glaser, S.D. A Machine-Learning-Based Connectivity Model for Complex Terrain Large-Scale Low-Power Wireless Deployments. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 576–584. [[CrossRef](#)]
18. Uccellari, M.; Facchini, F.; Sola, M.; Sirignano, E.; Vitetta, G.M.; Barbieri, A.; Tondelli, S. On the use of support vector machines for the prediction of propagation losses in smart metering systems. In Proceedings of the 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), Vietri sul Mare, Italy, 13–16 September 2016; pp. 1–6.
19. Ostlin, E.; Zepernick, H.; Suzuki, H. Macrocell Path-Loss Prediction Using Artificial Neural Networks. *IEEE Trans. Veh. Technol.* **2010**, *59*, 2735–2747. [[CrossRef](#)]
20. Popescu, I.; Nikitopoulos, D.; Constantinou, P.; Naforita, I. ANN Prediction Models for Outdoor Environment. In Proceedings of the 2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications, Helsinki, Finland, 11–14 September 2006; pp. 1–5.
21. Zhang, T.; Liu, S.; Xiang, W.; Xu, L.; Qin, K.; Yan, X. A Real-Time Channel Prediction Model Based on Neural Networks for Dedicated Short-Range Communications. *Sensors* **2019**, *19*, 3541. [[CrossRef](#)]
22. Mom, J.M.; Mgbe, C.O.; Igwue, G.A. Application of artificial neural network for path loss prediction in urban macro cellular environment. *Am. J. Eng. Res.* **2014**, *3*, 270–275.
23. Bishop, C.M. *Neural Networks for Pattern Recognition*; Oxford University Press, Inc.: New York, NY, USA, 1995.
24. Biernacki, R.M.; Bandler, J.W.; Song, J.; Zhang, Q. Efficient quadratic approximation for statistical design. *IEEE Trans. Biomed. Circuits Syst.* **1989**, *36*, 1449–1454. [[CrossRef](#)]
25. Meijer, P.B.L. Fast and smooth highly nonlinear multidimensional table models for device modeling. *IEEE Trans. Biomed. Circuits Syst.* **1990**, *37*, 335–346. [[CrossRef](#)]
26. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
27. Barron, A.R. Approximation and estimation bounds for artificial neural networks. *Mach. Learn.* **1994**, *14*, 115–133. [[CrossRef](#)]

28. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 323–323. [[CrossRef](#)]
29. Nocedal, J. Updating quasi-Newton matrices with limited storage. *Math. Comput.* **1980**, *35*, 773–782. [[CrossRef](#)]
30. Byrd, R.; Nocedal, J.; Schnabel, R. Representations of quasi-Newton matrices and their use in limited memory methods. *Math. Program.* **1994**, *63*, 129–156. [[CrossRef](#)]
31. Morales, J.; Nocedal, J. Remark on algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization. *ACM Trans. Math. Software* **2011**, *38*, 1–4. [[CrossRef](#)]
32. De Myttenaere, A.; Golden, B.; Le Grand, B.; Rossi, F. Mean absolute percentage error for regression models. *Neurocomputing* **2016**, *192*, 38–48. [[CrossRef](#)]
33. Jakes, W.C. *Microwave Mobile Communications*; Wiley-IEEE Press: New York, NY, USA, 1974.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).