Machine Learning Algorithms on Single Feature Cable Data

Daniel Charron, s#6785712

November 15, 2023

1 Abstract

I work full-time for the Royal Canadian Mounted Police (RCMP) on their radio system. One of my duties is to predict radio coverage for the RCMP's radio system throughout Canada in order to be able to predict where RCMP officers will be able to use their mobile and/or portable radios. In my work, I have used a machine learning algorithm to improve the precision of values that are used to predict whether or not a radio is able to communicate through the radio system. I show in this document how machine learning can be used to optimally predict a certain important value based on accurate data provided.

2 Introduction

The radio communication network used by the RCMP is composed of the radio system and Subscriber Units (SU) which include officers' mobile and portable radios. The SUs communicate with the radio system primarily through radio repeaters as illustrated in Figure 1.

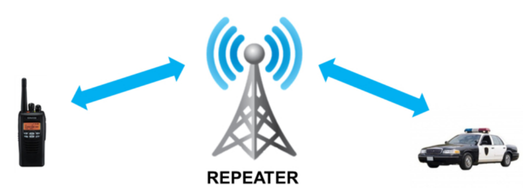

Figure 1: *Interfaces between a portable SU, a repeater and a mobile radio SU respectively [2].*

When predicting radio coverage, the goal is to generate a map showing where officers can use their SU, either mobile or portable, to communicate with the radio system. For a certain location away from the repeater, it is necessary to predict the received power at the repeater when the SU performs radio transmission and the received power at the SU when the repeater performs radio transmission. An SU is considered in-range of the radio system when bi-directional communication is predicted as being feasible.

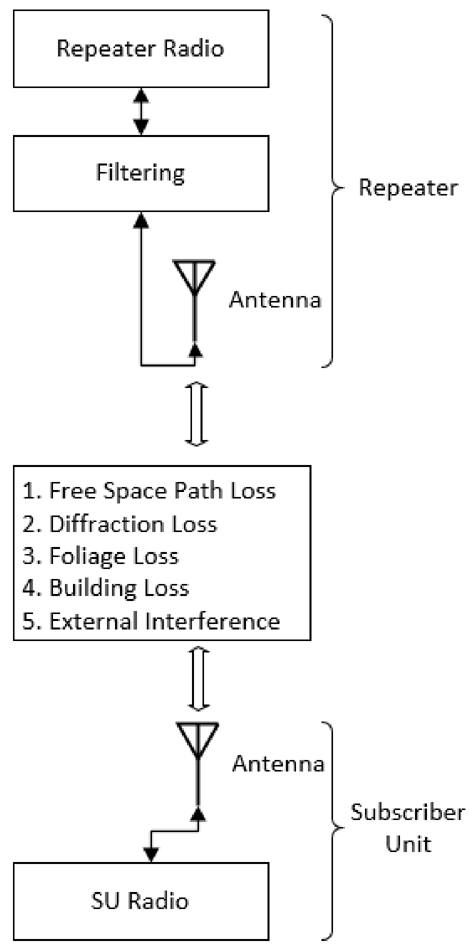

Figure 2: *Simplified view of the interface between a repeater and an SU [3].*

A simplified version of the interface between a repeater and an SU is shown in Figure 2. When the repeater transmits a signal, it goes through filtering followed by the repeater antenna. The signal output by the antenna is attenuated by the five of loss sources listed and reaches the SU antenna if and only if it its Signal-to-Noise Ratio (SNR) is above a certain threshold (e.g. 10 dB).

If the repeater transmits a signal with a power of P_{tx}, the received signal by the SU, P_{rx} can be expressed as

$$
\begin{aligned}
P_{rx} = P_{tx} - L_{filter} - L_{cable} \\
+ G_{atx} - \Sigma_{i=1}^{5} L_i + G_{arx}
\end{aligned}
\tag{1}
$$

where L indicates a loss due to different sources, G_{atx} represents the gain of the transmitting antenna and G_{arx} represents the gain of the receiving antenna. Losses and gains are expressed in dB and power levels are expressed in dBm. Equation 1 also holds for the transmission from the SU to the repeater as the losses and gains are equivalent in both direction, as they are expressed relative to the output power in dB.

Another way to express equation 1 is to use the repeater site's *Effective Radiated Power* (ERP_r), which is defined as the power level immediately following the antenna at the repeater as follows:

$$P_{rx} = \mathrm{ERP}_r - \Sigma_{i=1}^{5} L_i + G_{arx}, \tag{2}$$

$$\mathrm{ERP}_r = P_{tx} - L_{filter} - L_{cable} + G_{arx}. \tag{3}$$

Note that equation 2 no longer holds for both directions of communication, only for the communication from the repeater to the SU, which is referred to as *downlink*. For the opposite direction of communication, known as *uplink*, where the transmission is performed by the SU, another equation is necessary using the ERP of the SU:

$$
\begin{aligned}
P_{rx} = \mathrm{ERP}_{SU} - \Sigma_{i=1}^{5} L_i + G_{arx} \\
- L_{filter} - L_{cable},
\end{aligned}
\tag{4}
$$

$$\mathrm{ERP}_{SU} = P_{tx} + G_{atx}. \tag{5}$$

In order to make an accurate prediction for these two values of received power, it is necessary to improve the precision of each of the components of equation 1 or to improve the accuracy of the ERP in both uplink and downlink directions. The components P_{tx}, L_{filter}, G_{atx} and G_{arx} are easily quantifiable given the transmission frequency, relative orientations of the antennas, as well as distances of both the transmitting and receiving antenna. The path loss components $\Sigma_{i=1}^{5} L_i$ are difficult to quantify and multiple algorithms exist to perform this quantification [4]. The component of the received power that is of interest in this report is the cable loss L_{cable}. In the following sections, data is taken from a specific cable manufacturer's datasheet for a specific cable model. Different machine learning strategies are applied and compared to learn the best model to quantify the loss of the cable given a certain frequency.

3 Cable Loss Models

Cable loss is dependent on the frequency of the radio transmission going through the cable. Cable manufacturers usually express cable loss as is presented in the following *attenuation* (which is equivalent to loss) table as a function the frequency of the radio signal, in MHz.

Attenuation

Frequency (MHz)	Attenuation (dB/100 m)	Attenuation (dB/100 ft)	Average Power (kW)
100.0	1.162	0.354	7.23
108.0	1.209	0.368	6.95
150.0	1.433	0.437	5.86
174.0	1.548	0.472	5.43
200.0	1.665	0.507	5.05
204.0	1.682	0.513	4.99
300.0	2.059	0.628	4.08

Figure 3: *AVA5-50 Cable Loss Data as in [1]. Note that only a subset of the table is shown.*

Let the set of frequencies shown in the table be $F = \{100.0, 108.0, ..., 300.0\}$. For a certain frequency $f \in F$, it is possible to use the attenuation data in the table to learn the total cable loss. That is, there exists a function $a : F \to \mathbb{R}$ providing accurate loss information for f in dB/100m. Using this function, the total cable loss L_{cable}, expressed in dB, can be obtained using the following equation:

$$L_{cable} = a(f)\frac{l}{100}, \tag{6}$$

where l is the total length of the cable in meters (or feet, depending on the feature).

The problem is that it is not possible to use the function a on a different frequency $g \notin F$. An approximation function $b : \mathbb{R} \to \mathbb{R}$ is therefore created using the information in a which attempts to accurately output a loss value for g.

Some solutions are shown and evaluated to show which is the best to approximate the data shown in Figure 4.

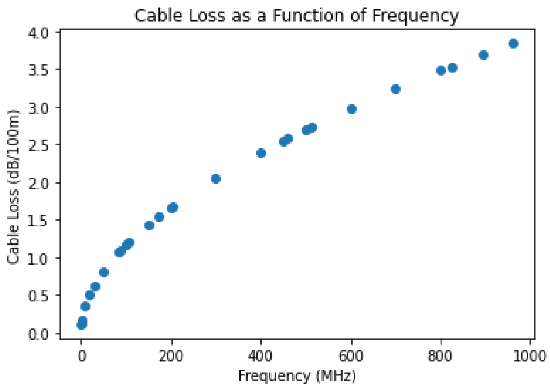

Figure 4: *Plotted AVA5-50 Cable Loss Data as in [1].*

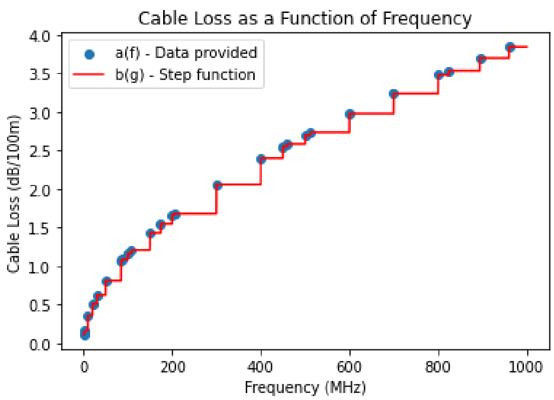

Figure 5: *Plotted AVA5-50 Cable Loss Data as in [1] along with the step function approximation.*

3.1 Simple Solutions

The simplest possible solution to construct an approximation function b is to use a step function which increases only when another data point in the dataset is presented.

$$b(g) = a(f_{idx}) \tag{7}$$

where f_{idx} is the frequency of the previous point and $g \in \mathbb{R}^+$. This model is shown in Figure 5.

Another simple solution is to approximate $b(g)$ using the previous and next points in $a(f)$; f_{idx} and f_{idx+1}. Using the two points, the equations for the local linear approximations are

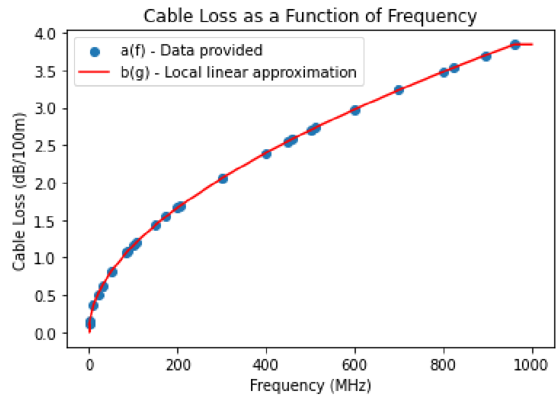

Figure 6: *Plotted AVA5-50 Cable Loss Data as in [1] along with the local linear approximation model.*

$$b(g) = wf_{idx} + w_0, \tag{8}$$

$$w = \frac{a(f_{idx}) - a(f_{idx+1})}{f_{idx} - f_{idx+1}}, \tag{9}$$

$$w_0 = a(f_{idx}) - wf_{idx}. \tag{10}$$

Using this model, the approximation is generated in Figure 6. It seems as though this model performs decently compared to the step function.

The weakness of this model is that the indices must be found each time the function is used, the function is not smooth and it will significantly **overfit** the data, as is shown in sections 4 and 5.

3.2 Linear Regression

To apply linear regression to this dataset, the function $b(g)$ is approximated using the linear function

$$b(g) = wg + w_0 \tag{11}$$

which minimizes the expected error on the dataset. The error used here is the squared error and is defined for N data points, using the i-th frequency in the dataset f_i, as

$$E(b) = \sum_{i=1}^{N} (b(f_i) - a(f_i))^2. \tag{12}$$

Using this error function, the optimized values of the weights are calculated at the minimum by taking the partial derivatives with respect to the two weights w and w_0 as in equations 13 and 14.

$$\frac{\partial E}{\partial w} = 2\sum_{i=1}^{N}(wf_i + w_0 - a(f_i))f_i \quad (13)$$

$$\frac{\partial E}{\partial w_0} = 2\sum_{i=1}^{N}(wf_i + w_0 - a(f_i)) \quad (14)$$

The partial derivatives are then set to zero to yield the following two equations for w and w_0 respectively

$$w = \frac{\bar{a}\sum_i f_i - \sum_i a(f_i)f_i}{\bar{f}\sum_i f_i - \sum_i f_i^2} \quad (15)$$

$$w_0 = \bar{a} - w\bar{f} \quad (16)$$

where $\bar{a}$ and $\bar{f}$ are the averages of $a(f_i)$ and f_i respectively $(N^{-1}\Sigma_i a(f_i)$ and $N^{-1}\Sigma_i f_i)$. This yields the following result in Figure 7.

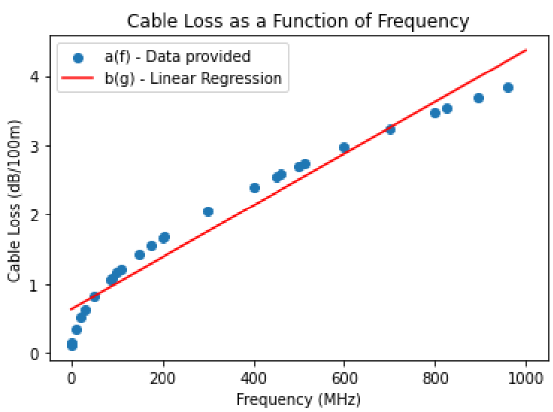

Figure 7: *Plotted AVA5-50 Cable Loss Data as in [1] along with the linear regression result.*

A piece of information that can be used to improve the accuracy of the linear approximation function b is that, based on RCMP data, the frequencies used are most likely ($> 99.99\%$) going to be between 100 and 900 MHz. Using this information, data outside this range can be removed from the dataset. When the exact same algorithm is applied to the modified dataset, this results in Figure 8. Notice that the approximation function more closely fits the data.

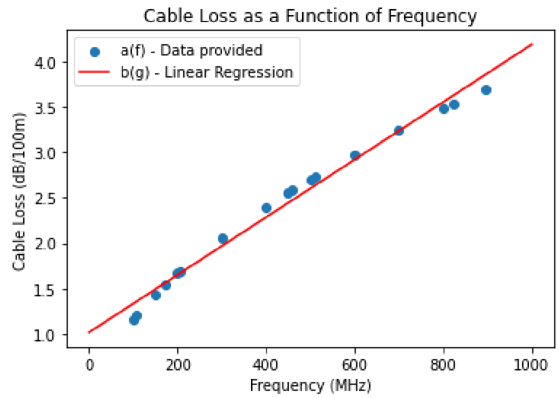

Figure 8: *Plotted AVA5-50 Cable Loss Data between 100 and 900 MHz as in [1] along with the linear regression result.*

Another thing to notice from the plot of the data as in Figure 4 is that it looks like a logarithm function and *"...instead of fitting a nonlinear function, one trick is to map the problem to a new space by using nonlinear basis functions"* [5]. Using this prior knowledge and the knowledge that $\log_c(c^x) = x$ where $x \in \mathbb{R}$, the input data can be transformed so that it more closely takes the shape of a line to then apply the linear regression algorithm.

However, the base of the exponent that is to be used to linearize the data is unknown. More precisely, $\exists c^* \in \mathbb{R}^+$ where the error function $E(b,c) = \Sigma_{i=1}^{N}(b(f_i) - c^{a(f_i)})^2$ is minimized;

$$c^* = \arg\min_c \sum_{i=1}^{N}(b(f_i) - c^{a(f_i)})^2. \quad (17)$$

Using this error function, analytic solutions can be found for optimal w and w_0 values as in equations 15 and 16. Specifically,

$$w = \frac{\overline{c^a}\sum_i f_i - \sum_i c^{a_i}f_i}{\bar{f}\sum_i f_i - \sum_i f_i^2}, \quad (18)$$

$$w_0 = \overline{c^a} - w\bar{f} \quad (19)$$

where $a_i = a(f_i)$ and $\overline{c^a} = N^{-1}\Sigma_{i=1}^{N}c^{a_i}$. However, the partial derivate of the error function with respect to c is

$$\frac{\partial E}{\partial c} = -\frac{2}{c}\sum_{i=1}^{N}(wf_i + w_0 - c^{a_i})(a_i c^{a_i}). \quad (20)$$

This equation *may* have an analytic solution but would be incredibly complicated to perform the algebra. Therefore, the gradient descent algorithm is used where the value of c is updated at each iteration. Using this strategy, at iteration number $t+1$ where $t \in \mathbb{N} \cup \{0\}$, c can be updated by

$$c_{t+1} = c_t - \eta \frac{\partial E}{\partial c_t} \tag{21}$$

for some $0 < \eta < 1$. Using gradient descent with 100 iterations, an optimal value is found for c which gives the following approximation function.

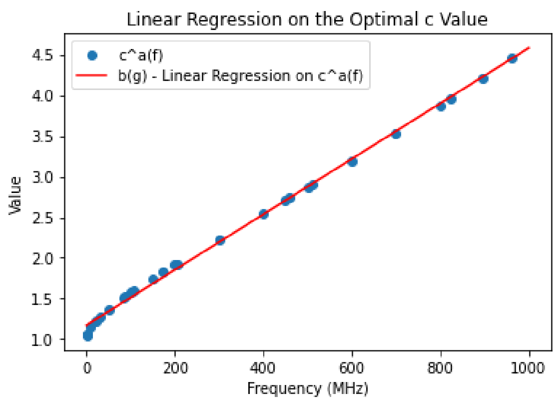

Figure 9: *c to the value of the AVA5-50 Cable Loss Data as in [1] along with the linear regression result on the updated data.*

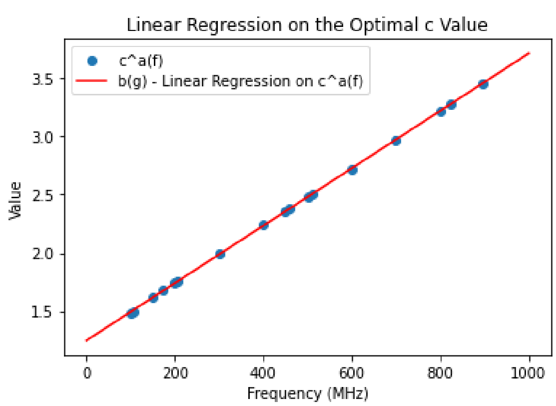

Figure 10: *c to the value of the AVA5-50 Cable Loss Data as in [1] along with the result on the updated data with extremities removed.*

Notice that the linear function seems to better approximate the data after the data modification from Figure 7 to Figure 9. By removing data outside the range of frequency between 100 and 900 MHz as well as using the data modification by exponentiation, the following approximation is generated in Figure 10.

Returning the data back to standard form results from taking the logarithm with base c of the modified data since $\log_c c^{a(f)} = a(f)$. Thus, the base c logarithm of of the approximation function $b(g)$ transforms the approximation function back to cable data. This is illustrated in Figure 11.

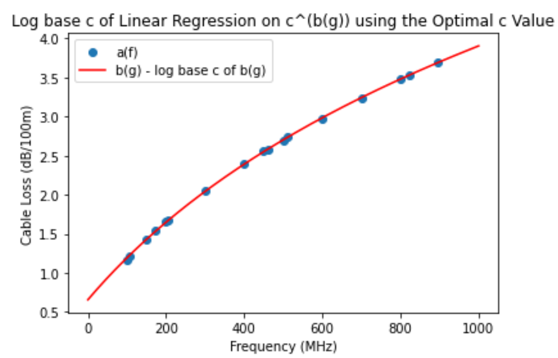

Figure 11: *AVA5-50 Cable Loss Data as in [1] along with the base c logarithm of the linear regression result on the updated data with extremities removed.*

4 Model Comparison

The best way to evaluate the models is to split the data into training and testing sets and see how well the models perform on the test data after having been trained, in order to evaluate the expected error to use as a comparison metric. However, there are $\binom{N}{n}$ possible partitions of the training and test set, where n is the number of entries in the training set.

The expected error for each model m_j on the test set is then

$$E[E(m_j)] = \sum_{i=1}^{\binom{N}{n}} p(p_i) E_{test}(m_j | p_i) \tag{22}$$

where p_i is the i-th partition of the training and test set. Since the partitions are uniformly distributed, the expected error can be rewritten as

$$E[E(m_j)] = \binom{N}{n}^{-1} \sum_{i=1}^{\binom{N}{n}} E_{test}(m_j|p_i). \quad (23)$$

Let R be a matrix where each row i contains one of the $\binom{N}{n}$ partitions of the training and test data. That is, $R_{ik} = 1$ when the data point belongs to the training set and $R_{ik} = 0$ when the data point belongs to the testing set $\forall i, k$. In order to train the model on the i-th partition of the data set, the error function used previously can be modified to include the i-th row of the R matrix. For example, the linear regression error function can be modified to become

$$E_{train}(m_j|p_i) = \sum_{k=1}^{N} R_{ik}(b_j(f_k) - a(f_k))^2 \quad (24)$$

on the training set and

$$E_{test}(m_j|p_i) = \sum_{k=1}^{N} (1 - R_{ik})(b_j(f_k) - a(f_k))^2 \quad (25)$$

on the test set. For a different dataset, it may be unfeasible to iterate through each of the $\binom{N}{n}$ combinations of training and testing data, however for this instance where an 80%/20% split in training and test data is used, $\binom{N}{n}$ is only 2380 so iterating the possibilities to calculate the expected error is possible for each model to compare. The expected error values were found in Table 1 where 500 iterations of gradient descent were used for the log input transform model.

Model	**Test Error** $E[E(m_j)]$
Step function	0.21219
Local Linear	0.00216
Linear Regression	0.05481
Log Input Transform	0.00112

Table 1: *Expected error for the models given the data provided.*

5 Conclusion

Given the results found in section 4, it is logical to choose the model which was analyzed which has the least expected error. The models' performances show that the step function model is significantly worse than the rest of the models as expected. It also shows that the best model is the model which transforms the input using an exponential constant c to attempt to *linearize* the data before performing linear regression on the modified data. The exponential constant c was found using gradient descent as it appeared too complicated to solve analytically.

In order to maximize the effectiveness of the input-modifying model, it is reasonable to include all the data from the cable's datasheet within the range that the RCMP will use, as there is little noise in these measurements, to obtain optimal parameters w, w_0 and c.

The local linear and step function models match the training data the best out of all of the models as training error is always zero. However, as is shown in section 4 based on the expected error, the models are not the best for predicting new inputs as the models are overfit to the data in comparison to the more general log input transform model. The bias of the two simple models is low but their variance high, while the linear regression models have their bias high but variance low.

All that remains to best make predictions for new frequencies is to use the best model to calculate the cable loss by using the following equation using parameters learned during training,

$$L_{cable}(g) = \log_c(wg + w_0)\frac{l}{100}. \quad (26)$$

Received power P_{rx} can then be predicted more accurately than without using machine learning as the model used provides the lowest expected error and thus the highest accuracy. In this way, machine learning has been applied to construct an accurate approximation model; to infer a logical extension of information contained in the dataset. Additional examples of the models are shown in the Appendix below with 80%/20% training/testing data split.

6 Appendix - Examples

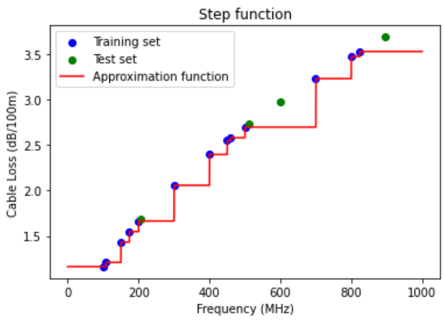

Figure 12: *Step function separated with* $R_{i,:} = [1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0]$

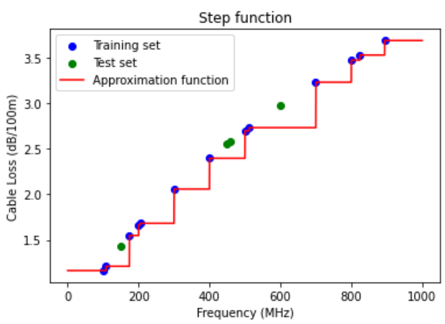

Figure 13: *Step function separated with* $R_{i,:} = [1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1]$

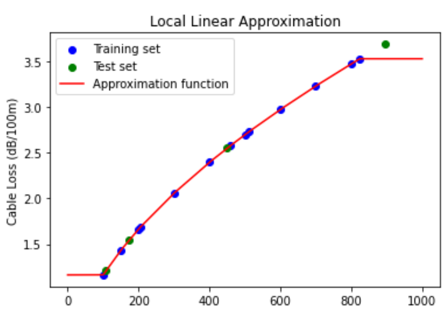

Figure 14: *Local linear model separated with* $R_{i,:} = [1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0]$

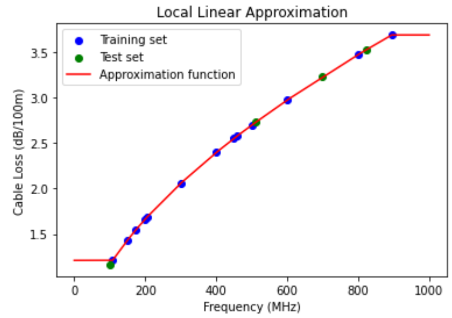

Figure 15: *Local linear model separated with* $R_{i,:} = [0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1]$

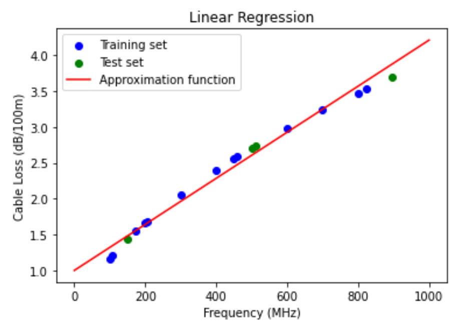

Figure 16: *Linear regression model separated with* $R_{i,:} = [1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0]$

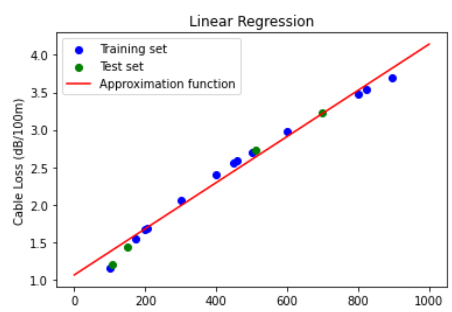

Figure 17: *Linear regression model separated with* $R_{i,:} = [1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1]$

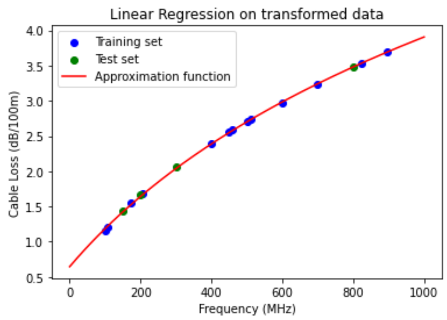

Figure 18: *Log input transform linear regression model separated with* $R_{i,:} = [1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1]$

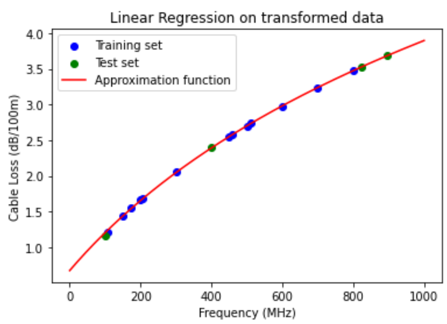

Figure 19: *Log input transform linear regression model separated with* $R_{i,:} = [0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0]$

References

[1] Commscope, AVA50 Datasheet, 2023-11-03, https://www.commscope.com/global-assets/digizuite/115359-p360-ava5-50-external.pdf

[2] Scott Karcich, Understand how repeaters work, when you need one, what to expect in the buying process, and selecting the right vendor for the job., 2016-10-14, https://commandradio.com/two-way-radio-repeater-system-buyers-guide/

[3] Urgent Communications Administrator, Close race: If you had a choice, which frequency band would you choose: VHF or 800 MHz?, 2009-11-01, https://urgentcomm.com/2009/11/01/close-race/

[4] Noman Shabbir, Muhammad T. Sadiq, Hasnain Kashif and Rizwan Ullah, Comparison of radio propagation models for Long Term Evolution (LTE) network, International Journal of Next-Generation Networks (IJNGN) Vol.3, No.3, 2011-09, https://arxiv.org/ftp/arxiv/papers/1110/1110.1519.pdf

[5] Ethem Alpaydin, Introduction to Machine Learning, Thomas Dietterich, Editor Christopher Bishop, David Heckerman, Michael Jordan, and Michael Kearns, Associate Editors, Third Edition, The MIT Press, 2014, Cambridge, Massachusetts, London, England (page 352)