2021 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS) | 978-1-6654-3557-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/COMCAS52219.2021.9629014

Adaptive Fuzzy-Based Models for Attenuation Time Series Forecasting

Dror Jacoby, Jonatan Ostrometzky *Senior Member, IEEE*, and Hagit Messer *Life Fellow, IEEE*
School of Electrical Engineering, Tel Aviv University, ISRAEL

***Abstract*—This work proposes an Adaptive Fuzzy Prediction (AFP) method for the attenuation time series in Commercial Microwave links (CMLs). Time-series forecasting models regularly rely on the assumption that the entire data set follows the same Data Generating Process (DGP). However, the signals in wireless microwave links are severely affected by the varying weather conditions in the channel. Consequently, the attenuation time series might change its characteristics significantly at different periods. We suggest an adaptive framework to better employ the training data by grouping sequences with related temporal patterns to consider the non-stationary nature of the signals. The focus in this work is two-folded. The first is to explore the integration of static data of the CMLs as exogenous variables for the attenuation time series models to adopt diverse link characteristics. This extension allows to include various attenuation datasets obtained from additional CMLs in the training process and dramatically increasing available training data. The second is to develop an adaptive framework for short-term attenuation forecasting by employing an unsupervised fuzzy clustering procedure and supervised learning models. We empirically analyzed our framework for model and data-driven approaches with Recurrent Neural Network (RNN) and Autoregressive Integrated Moving Average (ARIMA) variations. We evaluate the proposed extensions on real-world measurements collected from 4G backhaul networks, considering dataset availability and the accuracy for 60 seconds prediction. We show that our framework can significantly improve conventional models' accuracy and that incorporating data from various CMLs is essential to the AFP framework. The proposed methods have been shown to enhance the forecasting model's performance by $30-40\%$, depending on the specific model and the data availability.**

***Index Terms*—Machine Learning, Microwave Links, RNN Attenuation Forecasting, Time Series Forecasting,RNN, ARIMAX**

I. Introduction

Time-series is a sequence of observations $\{x_1, x_2, \ldots, x_t\}$ taken sequentially regularly in constant intervals. The time series forecasting task aims to predict the future development of an event based on the previous values. In practice, the time-series measurements can be generated by many separate sources, and the model structure might significantly change during the time series. Model-driven time series forecasting models typically assume the data follows an underlying model and often require statistical assumptions. Data storage tools and computation power development led to data-driven algorithms with Artificial Neural Networks (ANN) and deep learning algorithms that have shown remarkable performance in the time series forecasting area [1], [2], without assuming a specific model on the data. We focus on short-term forecasting of the attenuation in wireless communication links. The attenuation of a wireless communication link refers to the total loss in the signal's power density as the signal propagates through the channel from the transmitter to the receiver. This attenuation is a significant factor in designing and analyzing communication systems, and it's highly affected by environmental variations in the microwave link's channel. In particular, precipitation may cause significant attenuation at the frequencies on which modern wireless communication networks typically operate. Moreover, due to the growth in wireless traffic and high data rate transmission, higher frequency bandwidths are required, where the rain attenuation is more critical in higher frequencies bands and significantly reduce the link's performances. Therefore, the focus of this work is primarily on attenuation during rainfall events. Many rain attenuation models have been proposed based on the statistics of the rainfall rates, such as the ITU-R model [3]. However, we focus on models based only on the attenuation dataset available for the Network Management System (NMS). This way, the prediction method can be implemented without the need for external information. Model-driven such as variation of ARIMA models were suggested to predict the rain attenuation time series in terrestrial and satellite links, operating at the Ku and Ka-band frequencies [4]–[6]. Recently, learning-assisted prediction models have shown notable advancement in this area [7], and a recent work presented an RNN encoder-decoder model to predict a future time window of the attenuation values [8]. Yet, the DGP of the attenuation time series is not consistent, while it is composed of various processes that are not stationary. For example, while rain is present in the link's channel (wet periods), the measured attenuation rates significantly differ from the non-rain (dry periods). Therefore, without any additional pre-processing, a basic switching model is needed to separate the two different scenarios. Furthermore, rainfall is known as a non-stationary process [9]–[11] that change its characteristics during distinctive time intervals. With this motivation, a non-linear switching ARIMA/GARCH model [12] and [13], and adaptive In situ Learning Algorithm (ILA) has been suggested to enable tracking the non-stationary nature to predict rain attenuation levels. These models were implemented in Earth-To-Satelite links for specific frequencies bands. This work focuses on developing an adaptive framework for the attenuation time series terrestrial CMLs operating in a wide range of frequencies and lengths. We handle the variations between different attenuation patterns by incorporating the physical parameters that characterize the links in the forecasting models. The contributions of this paper are two-folded:

(i) The APS framework: Developing adaptive models to consider the non-stationary nature of the attenuation time-

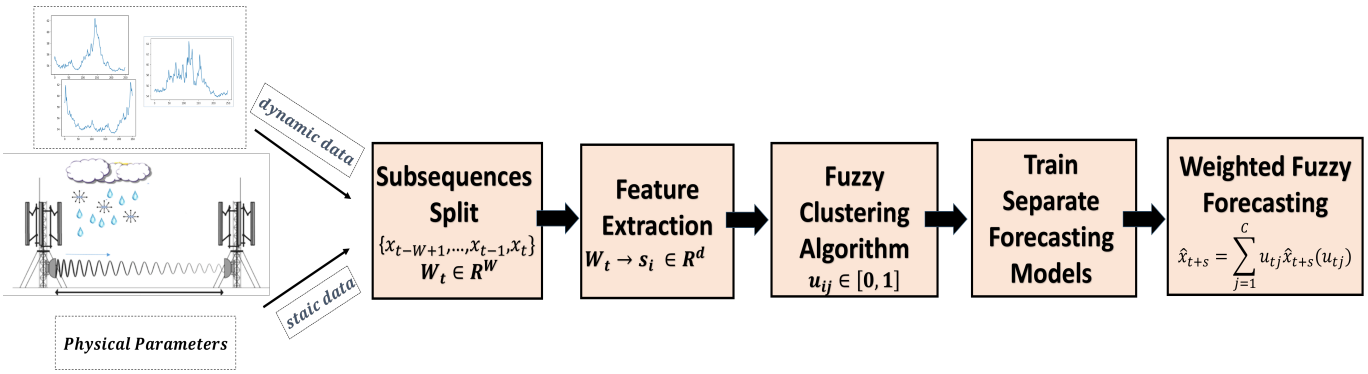

Figure 1: The adaptive fuzzy framework for the attenuation time series in CMLs.

series in CMLs. Our approach adopts the FCM algorithm for fuzzy clustering of subsequences based on similar temporal patterns. The related subsequences are grouped into clusters, representing the different states of the dynamic system. Then, the attenuation is predicted by a fuzzy mixture of separate forecasting models by the degree of membership of the current temporal sequence.

(ii) To study the integration of static data of CMLs as exogenous variables to time series forecasting models. This extension allows a considerably larger training dataset for training a time series foresting model. However, the complexity of such models is higher and may lead to lower performance.

Figure 1 describes the framework suggested in this work for the CML's static and dynamic measurements, which are detailed in Sec. II. The fuzzy partition methodology is detailed in Sec. III. The framework is evaluated for a linear forecasting model with ARIMAX and RNN for the data-driven method. The models and the extensions are detailed in Sec. IV, and Sec. VI provides the experiments and evaluates our proposed method. Finally, Sec. refsec.final conclude our paper and discuss future potential works.

II. Dataset Description

This section presents the dataset and the formulation of this work. We train, validate, and evaluate the suggested models using actual measurements from operational microwave networks in Sweden, provided by *Ericsson AB*. The dynamic data includes time series of the power levels measurements collected by the NMS. The CML measurements contain the logged periodically Transmitted Signal Level (TSL), $P_{Tx,t}$, and Received Signal Level (RSL), $P_{Rx,t}$ in a sampling time of 10 seconds. The total attenuation level for each time-step t of a the link link is calculated by the TSL and RSL measurements: $x_t = P_{Tx,t} - P_{Rx,t}$. The unviariate time series forecasting goal is to predict the future values of the attenuation sequence based on previous data $\hat{x}_{t+H} = f(x_t, x_{t-1}, \dots, x_1)$. However, we focus on **rain-induced attenuation**. The number of rain events occurrences on the specific CML's channel is limited. The problem of a limited dataset leads us to include the attenuation time series from different CMLs in the model's training method. The trade-off of this is clear when the measured attenuation is strongly affected by the physical parameters of different CMLs. Although we can increase the training dataset samples, the dissimilarity between the attenuation patterns might cause converging problems in the training procedure and lead to poor performance. Therefore, we suggest integrating the physical parameters fed as exogenous variables to the forecasting models to address the limited data issue. A feature vector $\mathbf{x_s}$ characterizes each CML contains the static data utilized on our work.

$$\mathbf{x_s} = (L, F, \mathbf{a^{(h)}}) \tag{1}$$

Where L is the effective distance in km between the transmitter to the receiver stations, F is the transmission frequency in GHz, and $\mathbf{a^{(h)}}$ is a vector containing the antennas heights above the sea level of the transmitter and receiver's antennas in meters. The polarization in all the CMLs in our dataset is identical and therefore is not included in $\mathbf{x_s}$. We focus only on data that is available for the NMS without side information. The training dataset containing 21 CMLs with a total of 1.1 million attenuation measurements, obtained between 2015 to 2017 in the area of the city Gothenburg. The length, L range of 0.6 km to 7.7 km, The frequency range of 12 GHz to 41 GHz, and heights between 10 to 60 meters. A detailed description of the measurements data set and the data collection process of the NMS for the dataset can be found in [14].

III. Dynamic Fuzzy Clustering

Clustering time-series data is different from general classification algorithms, where the dynamic changes over the time dimension have to be considered during the clustering process. Our approach includes converting a univariate time-series $\{x_1, x_2, \dots, x_n\}$ into a raw feature vectors $\{\mathbf{s_1}, \mathbf{s_2}, \dots, \mathbf{s_{N-W}}\}$, where $\mathbf{s_i} \in R^d$ and applying a fuzzy clustering algorithm over the extracted features. Then, the most related temporal patterns are grouped into clusters, and a separate model is fitted for each cluster of similar patterns. The last step includes forecasting by a fuzzy mixture of the above prediction models weighted by the degree of membership of the current feature vector. The procedure is described in Figure 1. Consider a time series $\{x_1, x_2, \dots, x_n\}$ that describes an interval of attenuation measurements of a CML, the following steps detail the adaptive methodology:

1. Sliding-Window: Constructing subsequences using the sliding window method over the time series. Each window contains the W previous measurements W_t =

$\{x_{t-W+1}, \ldots, x_{t-1}, x_t\}$. The sliding window over attenuation time series interval is demonstrated in Figure 3.

2. Feature Extraction: Each window is converted to a feature vector $\mathbf{s_t} \in R^d$ that characterizes the recent samples. The purpose is to describe the current samples by an informative representation, which serves as the clustering algorithm inputs. We use a feature vector based on the weighted second-order statistics of the current window:

$$\hat{\mu}_{W_t} = \sum_{i=t-W+1}^{t} w_i x_i \tag{2a}$$

$$\hat{\sigma}_{W_t} = \sqrt{\frac{\sum_{i=t-W+1}^{t} w_i (x_i - \hat{\mu}_{W_t})^2}{1 - \sum_{i=t-W+1}^{t} w_i^2}} \tag{2b}$$

Where $\hat{\mu}_{W_t}$ and $\hat{\sigma}_{W_t}$ represents the emperical weighted mean and standard devation for normalized weights $\sum_{i=t-W+1}^{t} w_i = 1.$, $\forall t \in \{W, W+1, \ldots, N\}$. The weights allow a greater influence on recent data points than the previous samples by assigning a decreasing weighting factor to each sample. We use exponentially decreasing weights, where $\tilde{w}_{t-k} = (1-\alpha)^k$. Each observation in the window W_t is multiplied by $w_k = (1-\alpha)^k / \sum_{k=t-W+1}^{t} \tilde{w}_k$, where the denominator is the normalization factor.

3. Fuzzy C-Means (FCM): We use a dynamic fuzzy clustering model for the feature vectors in the training dataset $\mathbf{s_i} = (\hat{\mu}_{W_i}, \hat{\sigma}_{W_i})$, $\forall i \in D$. D represent all available the attenuation time series dataset. The FCM [15] is an unsupervised clustering algorithm that assigns groups to related elements in respect to some distance criteria to the cluster centers' point. The algorithm attempts to find a fuzzy partition of the dataset by minimizing a distance metric with respect to fuzzy memberships and centers. Formally, for a given a finite collection of N data points $\{\mathbf{s_1}, \mathbf{s_2}, \ldots, \mathbf{s_N}\}$, the objective is the partition into C fuzzy clusters. The FCM generates C cluster centers $\{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_C}\} \in R^{CXd}$ and a partition matrix $U \in R^{NXC}$, such that $u_{ij} \in [0,1]$ such that within the cluster the distance between elements to the cluster center is minimized with respect to a given criterion The FCM aims to minimize an objective function:

$$J^* = \arg\min_{U,V} \sum_{i=1}^{N} \sum_{j=1}^{C} (u_{ij})^m ||\mathbf{s_i} - \mathbf{v_j}||^2 \tag{3a}$$

under the constrains:

$$\sum_{j=1}^{C} u_{ij} = 1, \quad u_{ij} \in [0,1], \quad 1 \le j \le C, 1 \le i \le N \tag{3b}$$

Where u_{ij} is the membership degree of object $\mathbf{s_i}$ to cluster c_j. $||\cdot||$ is the euclidean distance between the elements and the cluster's centers (3a). $1 \le m < \infty$ is the parameter that controls the fuzziness of the partition. Large values of m correspond to fuzzier clusters, where for $m = 1$, the fuzzy memberships, u_{ij}, converge to $\{0, 1\}$, lead to hard partitioning where each element belongs to one cluster. The minimization problem is solved by iteratively calculating cluster centers $\mathbf{v_j}$ and fuzzy membership matrix elements $u_{ij} \in U$. The optimal choice is received by differentiating the objective function (3a) with respect to $\mathbf{v_j}$ and u_{ij} under the constraints in (3b), for $1 \le j \le C$, and $1 \le i \le N$:

$$\mathbf{v_j} = \frac{\sum_{i=1}^{N} (u_{ij})^m \mathbf{s_j}}{\sum_{i=1}^{N} (u_{ij})^m} \tag{4a}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{||\mathbf{s_i} - \mathbf{v_j}||}{||\mathbf{s_i} - \mathbf{v_k}||}\right)^{\frac{2}{m-1}}} \tag{4b}$$

The general framework of the FCM includes the setting of the algorithm's parameters: The number of clusters C, stopping criteria for the iterative procedure, and m. The initialization of the coefficients matrix U, that randomly assigns a membership degree to each data point $\mathbf{s_i}$ for being in the clusters. Then, we repeat (4) until the change between two iterations is less than the stopping criteria.

4. Training Procedure : A forecasting model is fitted for the maximal members according to the fuzzy clusters, detailed in IV. The training dataset for each model,I_j, is composed of the measurements associated with the maximal membership elements of each cluster. First we include x_i in the $I_j = \{x_i | \quad j = \text{argmax}_{j'}(u_{ij'})\}$. Then, inputs and outputs pairs are formed by the previous and future values associated with x_i according to the forecasting model formulation.

5. Weighted Forecasting: The fitted trained forecasting models are used to predict x_{t+H}. The FCM model is trained over all the attenuation time series in the training data. For new measurement x_t, the existing FCM model is used to assign the membership degrees $\{u_{t1}, \ldots, u_{tC}\}$ of the current window according to steps (1-2) and (4b). The final step is the forecasting of x_{t+H} using the membership degrees as weights: $\hat{x}_{t+H} = \sum_{j=1}^{C} u_{tj} \hat{x}_{t+H}(u_{tj})$, where $\hat{x}_{t+H}(u_{tj})$ represent the prediction of the fitted model associated with cluster j.

The dynamic fuzzy clustering steps are a general framework for the attenuation time series data. Note that steps 1-2 describe the framework for a specific attenuation time-series interval, $\{x_1, x_2, \ldots, x_n\}$, where the clustering process is applied for x_i in steps 3-5 refer to the available training dataset for all the CMLs, after the pre-processing steps detailed in Section VI.

IV. Attenuation Forecasting Models

This section presents the attenuation forecasting methods, the implementation details, and the required pre-processing steps. The ARIMA model as a model-driven and an RNN as a data-driven model are state-of-art forecasting models. Both are popular methods in the time series forecasting area, and were proved their efficiency for the short-term rain attenuation forecasting task in particular [7], [8], [16].

A. Model-Driven: ARIMA(X) Models

Autoregressive models express the following values using the previous steps of variables to predict their future values. ARIMA refers to a class of models that aim to capture a

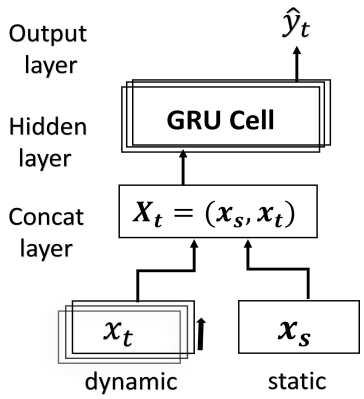

Figure 2: Integrating static data and dynamic inputs in RNN to forecast sequential attenuation outputs

sequence of different standard temporal structures in time series data. The time series model using the Box-Jenkins approach [17] has been widely used in the literature because of its simplicity and high performance. In Autoregressive Moving Average (ARMA) modeling, a stationary time series is described as a combination of an Auto-Regressive (AR) and a Moving Average (MA) model, which uses past forecast errors in a regression model rather than using past values of the forecast variable in a regression. The ARMA modeling assumes the data was created from a stationary process and requires that the statistical properties not change over time. The **'I'** corresponds to the **Integrated** component, which implies the time series values may have been replaced with the differenced series, applying a differencing operator $\nabla^{(d)} x_t$. In first order differencing ($d = 1$), the time series is replaced by $\nabla x_t = x_t - x_{t-1}$, and d refers the number of times that we perform the operator. The outcome is stabilizing the mean and the variance of the time series and other non-stationary components such as trend and seasonality. While ARIMA assumes that previous values carry all the information to predict the future, **ARIMAX** extends the model to exogenous input through the addition of variables that are not explained by the model. Let $\mathbf{x_s}$ be the a vector of exogenous input and a time series $\{x_1, x_2, \ldots, x_t\}$, the ARIMAX(p,d,q) process is described by the following equation:

$$\nabla^{(d)} x_t = \sum_{k=1}^{p} \phi_i \nabla^{(d)} x_{t-k} + \sum_{m=1}^{q} \theta_m \epsilon_{t-m} + \sum_{l=1}^{S} \beta_i x_{s,l} + \epsilon_t \quad (5)$$

ϵ_t represents a white noise error term. The parameters p and q are non-negative integers indicating the order of the AR and MA terms, which are the maximal previous measurements and error terms included in the model, respectively. The differencing (d) order is the minimum required to get near-stationary time-series data (also called difference-stationary). In the general model form, the exogenous variables may vary along with the time series values. However, in our case, the static data aims to capture the differences between the CMLs and is modeled by a linear combination with the vector β. The ARIMA formulation do not include the static data inputs.

B. Data Driven: RNN

Recurrent Neural Networks (RNNs) are a class of artificial neural networks for processing sequential data. They can handle the time dependence of the input variables, unlike other learning regression predictive models. It is done by performing a hidden state that allows storing and memorizing information and non-linear dynamics to update their hidden state. The purpose RNN cell is to update state h_t using previous state h_{t-1} and current input x_t, i.e, $h_t = f_W(h_{t-1}, x_t; \Theta)$. The computation process of updating the hidden state depends on the RNN cell architecture. The simple RNN architecture suffers from a vanishing gradient problem [18] when trying to handle long sequences. The information from the beginning of the sequence has a negligible influence in updating the model's weights by performing Back Propagation Through Time (BPTT). Therefore, various variants of RNN architectures were suggested, including LSTM [18] and GRU [19] units to address the vanishing gradient problem. Both include internal mechanisms called gates that can regulate the flow of input information and control the memorization process. In this work, we implement an RNN with a gating mechanism of GRU cells to reduce the model's complexity and yield comparable performance to LSTM. The main distinction between vanilla RNNs and GRUs is that the latter supports a gating mechanism of the hidden state. It includes controlling the flow of information by employing two internal gates: An update gate and a rest gate. The update gate, z_t, would allow us to control how much of the current state changes with new information. It determines which and the previous time steps are passed to the next state. The reset gate, r_t, allows controlling how much of the previous state we might still want to remember. The following equations describe the update of the GRUs:

$$z_t = \sigma_g(A_{zx} x_t + A_{zh} h_{t-1} + b_z) \quad (6)$$

$$r_t = \sigma_g(A_{rx} x_t + A_{rh} h_{t-1} + b_r) \quad (7)$$

$$\tilde{h}_t = \tanh(A_{xg} x_t + A_{gh}(r_t \odot h_{t-1}) + b_h) \quad (8)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (9)$$

where $\mathbf{A}$ are weight parameters matrices and b_r, b_z are bias vectors. We use the sigmoid activation function, $\sigma_g(x) = \frac{1}{1+\exp^{-x}}$ for the internal gates to transform input values to the interval $(0, 1)$, and the hyperbolic tangent, $tanh(x) = \frac{e^x - e^{-x}}{e^x - e^{-x}}$ to ensure that the candidate hidden state result, $\tilde{h}_t$, is in the interval $(-1, 1)$. The output hidden state updates by incorporating the outcome of $\tilde{h}_t$ and the update gate z_t, described in equation (9), where $\odot$ denotes the Hadamard product operator. To extract the static information from the CMLs, we concatenated the dynamic attenuation time series and the static features and fed them into the RNN. The modified architecture is depicted in Figure 2. Notice, we process the static and dynamic data as an input vector $\mathbf{X_t} = (x_t, \mathbf{x_s})$ to the GRU cell to train the recurrent network to adopt the change between the CMLs. This Integrated RNN (I-RNN) architecture got better performance than an architecture that the concatenate layer appears after the RNN, and a fully connected layer

calculates the output, such as suggested in previous works [20], [21]. The RNN is trained to generate an output vector $\{\hat{x}_{t+1}, \hat{x}_{t+2}, \ldots, \hat{x}_{t+H}\}$, with l_2 loss between the outputs and the following time steps.

V. Experimental Analysis

This section details the experiments in this work. It includes the experimental setup, implementation details of the suggested models, and the empirical evaluation of the proposed and the baseline methods. The analysis includes the two suggested extensions for the models: the forecasting based on the fuzzy partition and the static information integration of the CMLs. The test data contains 6 hours of measurements from an operating CML with $\mathbf{x_s} = (2.9, 37.2, 46, 35)$. The train-test split for this CML is according to $70 - 30$ division. While this split is used for the benchmark models, the models described in Sec. IV allow us to integrate time series from separate CMLs, and by that, to increase the available training data dramatically. Furthermore, it demonstrates a real scenario when only a limited amount of data that includes rain events is available from a specific CML.

A. Implementation Details

For both the model-driven and data-driven analysis, we use the baseline models as the ARIMA model and the data-driven an RNN without incorporating the static information. The AFP is proposed and performed for the baseline models with and without the static information extensions. For the model-driven, the model minimized the AIC criterion [22] over the training data was selected. Notice that we perform multi-step forecasting of the time series, where $H = 6$, concerning the data sampling time. However, we apply a smoothing moving-average filter followed by downsampling in a factor of 2 to remove noise, and better fit the models to the multi-step forecasting task. In this case, the ARIMA modeling is fitted to the sampled time series and performs iterative forecasting to generate the predictions for $\hat{x}_{t+3}$. The values for $\hat{x}_{t+2}$ and $\hat{x}_{t+3}$ are computed using the fitted model with the former predictions as input to the model equation (5). For the RNN models, we use one hidden layer with 128 units in each GRU cell and a batch size of 120. We use a l_2 weight regularization with constant $\lambda = 1e{-}4$ to improve the model performance by reduce overfitting. All the hyperparameters were tuned using a grid search over a validation set. We assigned 20% from the training dataset of each CML as a validation data set for the RNN. We use Adam [23] as an optimization algorithm of stochastic gradient descent. All models were implemented using Keras 2.5.0 with the Tensorflow backend in Python 3.7.1.

Pre-Processing: Utilizing datasets obtained from different CMLs requires scaling steps to the inputs for the AFP framework. The first pre-processing step is to subtract the baseline attenuation from each time series data of the CML. This constant is commonly different for distinct CMLs and was set to the median of the minimal 200 samples. This step assigns the observed attenuation that will be affected by the environmental changes in the channel. Then, the dynamic and static datasets were scaled separately according to the z-score standardization $\tilde{x} = \frac{x-\hat{\mu}}{\hat{\sigma}}$, where $\hat{\mu}$ and $\hat{\sigma}$ represents the empirical mean and standard deviation of all the available training dataset of each CML. For the static dataset, the standardization performed for each feature separately: $\tilde{\mathbf{x}}_\mathbf{s} = \frac{\mathbf{x^{(s)}} - \hat{\mu}^{(s)}}{\hat{\sigma}^{(s)}}$. In addition, for the ARIMA models, we use a logarithmic transformation after the fuzzy clusters were determined. Several parameters influence the adaptive forecasting algorithm performance described in III. The feature vector values are determined according to the sliding window length W, and the α, the rate of the exponential decrease of the weights within the window. The number of fuzzy clusters $C = 3$ for both methods, $\alpha = 0.3$ and the window lengths are 4 and 6 for the model and data-driven models correspondingly. The values were obtained by optimizing over the validation set.

Table I: The datasets balance and RMSE values for model and data-driven methods according to each cluster

Cluster	Data Percentage		RMSE	
	Training	Test	ARIMAX	I-RNN
1	45.5 %	27.1%	**0.27**	0.31
2	38.2%	42.3 %	0.64	**0.55**
3	16.3%	31.6 %	0.96	**0.83**

B. Results

We conduct an empirical evaluation of the result by comparing the two proposed methods (AFP framework and static data integration) for the baseline methods. The adaptive fuzzy clustering algorithm is demonstrated in Figure 3 over attenuation time-series interval. We observe periods with increased attenuation levels on the LHS figure due to heavy and weak rainfall patterns in different time intervals. On RHS, a data point represents the features of the current window, where $(\tilde{\mu}_{W_t}, \tilde{\sigma}_{W_t})$ refers to the normalized attenuation values after the pre-processing step for $W = 6$. The different clusters refer to the maximal membership degrees of each data point. As can be seen, high standard deviation values are often associated with high attenuation values. Table I presents the split of the total training and test data set for clusters and the $RMSE$ values obtained for each set for the **test data** measurement for each model. The imbalance in training and test data is due to choosing the test as intervals that contain rainfall events. The purpose is to evaluate our method primarily for rain-induced attenuation. However, the training set may also include longer dry periods within the interval of measurements. The FCM algorithm will classify those dry periods to cluster 1 with high probability. Low volatility levels characterize the signal in cluster 1. Therefore, the adaptive forecasting model achieves the lowest $RMSE$ values for this cluster. Moreover, the ARIMAX as the model-driven method outperforms only in this cluster, suggest that ARIMAX can achieve comparable and more beneficial results for moderate attenuation values. However, the I-RNN performs better for the other clusters. The total $RMSE$ values are 0.63 and 0.54 correspondingly, which signifies that the adaptive I-RNN outperforms the ARIMAX model in the attenuation time series forecasting task. To evaluate the adaptive fuzzy mechanism and integration of the

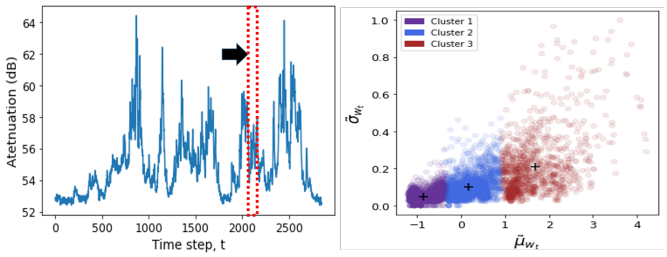

Figure 3: The FCM partition of the attenuation time series measurements. An increase attenuation due to rain is evident during an interval rain-event.

static data of the CMLs in forecasting models, we compare our two proposed methods to the baseline methods. The results are summarized in in Table II. The baseline methods are the ARIMA and RNN models without incorporating the static data of the links. The training dataset in the first two rows is restricted for the tested CML without utilizing the static and AFP framework, containing $35k$ measurements. The last two rows combine datasets from all the CMLs. The $RMSE$ values obtained are presented in the first row as the baselines for each model. The suggested extensions' performances are evaluated in the following rows, compared to the baseline results as a reference. In the second raw, we apply the AFP method with the same training data as the baseline. The AFP led to undesirable effects with a reduction in performance for the RNN. As can be seen, the RNN is sensitive to the available dataset for model training. The AFP reduces the training dataset for each model and therefore leads to poor results. However, the AFP improves the ARIMA performance by 20%. The last two rows (Multi-Link training) signify the suggested improvements of this work and includes all the available training datasets from all the CMLs in the training process. The third row evaluates the integration of the link's physical features in the models, where the last raw uses the AFP method. The results in the third raw imply that the extensions for the models (ARIMAX and I-RNN) are essential when using time series data obtained from other CMLs, where the most significant improvement is for the RNN with 39% better from the benchmark. The last raw indicates that the AFP method improved all model's performance when using a sufficient training dataset, where the most notable (15%) is for the model-driven.

VI. Conclusions and Future Work

This work proposes a framework to predict the attenuation time series in CMLs, considering the different characteristics of the data generating process. Our approach includes two distinct extensions to traditional short-term attenuation forecasting models and is implemented and evaluated for model and data-driven methods. The motivation is to consider the non-stationarity of the signals and the rainfall patterns, which might cause poor performance in training forecasting models. The clustering allows increasing the similarity between sequences in the training set. Moreover, we present the integration of the physical parameters that characterize the CML in the forecasting models. We show that incorporating the static data

Table II: Performance of the Adaptive Fuzzy Prediction method and the static-data

Training Method	**Model-Driven**		**Data-Driven**	
	ARIMA	ARIMAX	RNN	I-RNN
One-Link[1]	0.93	-	0.98	-
One-Link AFP	19.3%	-	-27.5 %	-
Multi-Link	-17.4%	13.1%	9.1 %	39.0%
Multi-Link AFP	-5.1%	28.4%	12.9%	41.2%

improves the model's performance by including attenuation time series from multiple CMLs and increasing the available training dataset. This extension is essential for the proposed AFP method, which split the total training dataset. The results indicate that the two proposed approaches, the AFP and the static data extensions, outperform the baseline in both model and data-driven models. The data-driven model includes an RNN with GRU units using the architecture that receives dynamic and static features as an input. The result shows that the RNN is more sensitive to the training dataset. The more significant gains than the baseline were achieved by incorporating the static dataset and increasing the total training dataset. Deep learning algorithms can learn and adopt the non-stationary temporal changes between different regions in the time series with sufficient training data. The model-driven models include ARIMA variations, which require learning specific model parameters. The non-stationary variations in the time series, leading to unsatisfactory performances. Therefore, applying the AFP method is essential for the attenuation time series to achieve better performance. We suggested the fuzzy framework in this work and proved its efficiency on the attenuation time-series datasets. In future studies, we will evaluate the robustness of our method to different dynamic clustering algorithms by applying recent time series clustering methods. Moreover, the evaluation can be extended to include more CMLs in the test dataset to more link lengths and frequencies. Another concern with the AFP framework is the reduction of the dataset for each forecasting model. We will address this matter by extending the algorithm to include fuzzy partition of the training datasets, allowing the models to employ all the available data.

VII. Acknowledgment

This work was supported in part by the NSF-BSF grant number CNS-1910757. We would like to thank Ericsson for giving access to the data for this research.

References

[1] G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

[2] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1394–1401. IEEE, 2018.

[3] P Series. Propagation data and prediction methods required for the design of earth-space telecommunication systems. *Recommendation ITU-R*, pages 618–12, 2015.

[4] A. Mauludiyanto, G. Hendrantoro, M. H. Purnomo, T. Ramadhany, and A. Matsushima. Arima modeling of tropical rain attenuation on a short 28-ghz terrestrial link. *IEEE Antennas and Wireless Propagation Letters*, 9:223–227, 2010.

[5] Dhaval P Patel, Mitul M Patel, and Devendra R Patel. Implementation of arima model to predict rain attenuation for ku-band 12 ghz frequency. *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)*, 9(1):83–87, 2014.

[6] J. Sosa, C. Sosa, and B. Paz. Arima models in the rain attenuation prediction in a mexican tropical area. In *IEEE Antennas and Propagation Society International Symposium. Transmitting Waves of Progress to the Next Millennium. 2000 Digest. Held in conjunction with: USNC/URSI National Radio Science Meeting (C*, volume 2, pages 546–549 vol.2, 2000.

[7] Md Abdus Samad and Dong-You Choi. Learning-assisted rain attenuation prediction models. *Applied Sciences*, 10(17):6017, 2020.

[8] Dror Jacoby, Jonatan Ostrometzky, and Hagit Messer. Short-term prediction of the attenuation in a commercial microwave link using LSTM-based RNN. *EUSIPCO'20*, 2020 - Submitted.

[9] Shigetoshi Sugahara, Rosmeri Porfírio Da Rocha, and Reinaldo Silveira. Non-stationary frequency analysis of extreme daily rainfall in sao paulo, brazil. *International Journal of Climatology: A Journal of the Royal Meteorological Society*, 29(9):1339–1349, 2009.

[10] D Schertzer, S Lovejoy, F Schmitt, Y Chigirinskaya, and D Marsan. Multifractal cascade dynamics and turbulent intermittency. *Fractals*, 5(03):427–471, 1997.

[11] J Lavergnat and P Golé. A stochastic raindrop time distribution model. *Journal of Applied Meteorology*, 37(8):805–818, 1998.

[12] L de Montera, Cécile Mallet, Laurent Barthès, and Peter Golé. Short-term prediction of rain attenuation level and volatility in earth-to-satellite links at ehf band. *Nonlinear Processes in Geophysics*, 15(4):631–643, 2008.

[13] Bijoy Roy, Rajat Acharya, and MR Sivaraman. Attenuation prediction for fade mitigation using neural network with in situ learning algorithm. *Advances in space research*, 49(2):336–350, 2012.

[14] L Bao, C Larsson, M Mustafa, J Selin, JCM Andersson, J Hansryd, M Riedel, and H Andersson. A brief description on measurement data from an operational microwave network in gothenburg, sweden. In *15th International Conference on Environmental Science and Technology, Rhodes, Greece*, volume 31.

[15] James C Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & geosciences*, 10(2-3):191–203, 1984.

[16] MN Ahuna, TJ Afullo, and AA Alonge. Rain attenuation prediction using artificial neural network for dynamic rain fade mitigation. *SAIEE Africa Research Journal*, 110(1):11–18, 2019.

[17] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[19] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[20] Te-Cheng Hsu, Shing-Tzuo Liou, Yun-Ping Wang, Yung-Shun Huang, et al. Enhanced recurrent neural network for combining static and dynamic features for credit card default prediction. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1572–1576. IEEE, 2019.

[21] Cristóbal Esteban, Oliver Staeck, Stephan Baier, Yinchong Yang, and Volker Tresp. Predicting clinical events by combining static and dynamic information using recurrent neural networks. In *2016 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 93–101. IEEE, 2016.

[22] Hamparsum Bozdogan. Model selection and akaike's information criterion (aic): The general theory and its analytical extensions. *Psychometrika*, 52(3):345–370, 1987.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.