A good approximation to power amplifier (PA) behavioral modeling requires precise baseband models to mitigate nonlinearities. Since digital predistortion (DPD) is used to provide the PA linearization, a framework is necessary to validate the modeling figures of merit support under signal conditioning and transmission restrictions. A field-programmable gate array (FPGA)-based testbed is developed to measure the wide-band PA behavior using a single-carrier 64-quadrature amplitude modulation (QAM) multiplexed by orthogonal frequency-division multiplexing (OFDM) based on long-term evolution (LTE) as a stimulus, with different bandwidths signals. In the search to provide a heuristic target approach modeling, this paper introduces a feature extraction concept to find an appropriate complexity solution considering the high sparse data issue in amplitude to amplitude (AM-AM) and amplitude to phase (AM-PM) models extraction, whose penalties are associated with overfitting and hardware complexity in resulting functions. Thus, experimental results highlight the model performance for a high sparse data regime and are compared with a regression tree (RT), random forest (RF), and cubic-spline (CS) model accuracy capabilities for the signal conditioning to show a reliable validation, low-complexity, according to the peak-to-average power ratio (PAPR), complementary cumulative distribution function (CCDF), coefficients extraction, normalized mean square error (NMSE), and execution time figures of merit. The presented models provide a comparison with original data that aid to compare the dimension and robustness for each surrogate model where (i) machine learning (ML)-based and (ii) CS interpolate-based where high sparse data are present, NMSE between the CS interpolated based are also compared to demonstrate the efficacy in the prediction methods with lower convergence times and complexities.

Keywords: behavioral modeling; linearization model; machine learning; nonlinear modeling; power amplifier

1. Introduction

Presently, given the continuous growth and development of modern wireless communication technologies, DPD is widely used for the linearization of power amplifiers. However, the introduction of base stations (i.e., femtocells, picocells, and microcells) limits the DPD hardware complexity and power consumption and requires faster adaption because the PA’s behavior may vary more often, which makes high-complexity methods unsuitable. In addition, Radio-frequency/Microwave transmission circuits with wide dynamic bandwidths represent a challenge for modeling and DPD linearization techniques. Behavioral modeling based on higher-order polynomial systems for long-term and short-term induced memory represents an efficient methodology for improving spectral efficiency [1,2]. The selection of the primary method for the modeling and linearization stages is essential if it is required to use multivariable data, gain precision, and not increase the involved coefficients during hardware implementation [3]. Various efforts to improve accuracy and reduce coefficients have been developed as alternative techniques
for out-of-band emission control. In this work, a training was performed to extract the inverse characteristics in terms of the normalized output using a small-signal gain of the PA-Tx system. In [4], a PAPR reduction is reached based on orthogonal frequency division multiplexing (OFDM); in this case, the error vector magnitude (EVM) for multiple-input multiple-output (MIMO) is evaluated as a metric. The developed system comprises a MIMO precoder, resource mapper, OFDM-PAPR reduction stage, and OFDM modulation scheme processed before the PA. In [5], a parallel structure of two boxes is developed for the modeling and DPD stage, the coefficient extraction method is based on applying an indirect learning structure and a least squares (LS) method; the main proposal is based on coefficient reduction up to 22% for a signal with 64-QAM. An example of feature selection techniques for DPD linearization with order reduction methods is presented as an alternative to reduce the high volume of coefficients of the Volterra Series [6], this work is applied to improve the predistortion performance for a class J-PA using a dual-band scenario. Consequently, in [7] spline-interpolated LUT methods are proposed for baseband applications built on special truncations of the Volterra Series such as Wiener, Hammerstein, and a combination of both; while the observation path uses a general scheme based on an indirect learning approach (ILA) structure for LTE-based systems 5G-new radio (NR). Despite existing works, the need for systems with accurate data analysis, ability to perform automated modeling/linearization processes, and big data management is still present, and in this regard, machine learning (ML) represents an efficient proposal. Artificial neural networks have been used in over-the-air (OTA) linearization for a highly nonlinear and wide-band mmWave active phased array (APA) transmitter, and results indicate that they are capable of learning distinctive nonlinear distortions, which may not easily fit to existing polynomial-based solutions [8].

In [9], an extreme learning approach is developed for the modeling stage compared to network learning algorithms in conventional neural networks, and it is applied to a Class-E PA. To balance computational efficiency and accuracy of the modeling stage, a low-complexity wireless method called simplified generalized memory polynomial, and a random forest process for classification are proposed [10]. In [11], a random forest application for DPD based on Volterra is developed; this approach is validated for a PA under 5G-NR generation signal, improving the coefficient extraction methodology. The divide-and-conquer paradigm can offer a solution while using calculation of multiple small-scale LS solutions which is easier and less complex than that of a large-scale LS problem. For addressing the latter case, probabilistic models that allow different submodels were proposed [12–14] to cover both PA modeling and DPD linearization with a double objective using a mixture of expert techniques for wide-band signals.

In this work, three model structures are presented to demonstrate rigorous approximations that can aid in the search for adaptive algorithms and are addressed in order to find specific functions that have been acquired from a training process either for a supervised or unsupervised learning conducted with a data-driven approach based on input and output samples. Thus, the prediction characteristics are explored for classification goals on different sparse feature data in order to extract and provide a decision model based on a feature inspection to provide an appropriate PA and/or PD model concerning this problem in a regression fashion technique considering two ML approaches. An application for determining digital predistorter model structures for wideband power amplifiers for 5G-NR signals through RF is proposed in [11]. Among different ML techniques, Decision Trees remain as one of the simplest learning approaches due to the method producing a comprehensive model from a given dataset [15]. More accurate results are obtainable with Decision Trees on nonlinear problems and tend to be mildly less accurate on problems with a linear structure [16]. RF and decision tree algorithms are widely popular for many prediction applications and are characterized by having few tuning parameters and working with high-dimensional data.

Sparse systems are a recurrent phenomenon that occurs while acquiring PA measurement curves for systems that operate with MIMO technologies and applications related to
the new 5G standard [17]. In recent research [18], an extension of a sparse-Bayesian treatment is developed to remove non-active regressors and re-estimate the model coefficient in a system with varying power levels. Additionally, in [19], a dynamic deflection reduction algorithm was developed due to the high performance of beneficial terms and reduced schemes compared to the substantial full Volterra model. However, alternative approaches, such as those derived from ML with a considerable reduction of coefficients, are possible. In this article, we extend our preliminary work in [20], where the basic ML modeling was considered for direct modeling of PAs compared against complex spline-interpolating models (b-splines and cubic-splines) in terms of explicit complexity, modeling accuracy, and linearization performance. Therefore, this paper is concentrated particularly in the search of linearization capabilities of the proposed models, where the high sparse data in the PA characteristics (AM-AM and AM-PM responses) yields to a substantial increase in the processing complexity.

The remainder of this paper is organized as follows: In Section 2, the theory and its extension to the modeling is presented to different basis structures and its associated algorithms. Section 3 describes the experimental setup and illustrates the proposed test and training setup for ML models with different bandwidths for 2.45 GHz band applications. Section 4 shows the developed RT, RF, and the cubic-spline algorithm for approximate data modeling and discusses the FPGA implementation complexity. Finally, Section 5 depicts the obtained conclusions.

2. Modeling Framework

We introduce three model validations that are used to provide an algorithm construction, used to perform the PA modeling and obtain the PD linealization model. In the context of the algorithm construction to pursue modeling capabilities, a system is built based on an indirect learning architecture as shown in Figure 1.

![General diagram of the proposed behavioral model on indirect learning approach.](image)

Following these ideas, initially in this work an ensemble approach is presented to obtain the PA behavioral modeling with the Classification and Regression Trees (CART) supervised learning algorithm. This method performs repeated splits on a training dataset until a rule establishes a terminal condition. Each split becomes a node of a tree and terminal nodes contain the response of the modeling system [16]. Decision Trees are conceptually simple, have good enough learning performance, and are computationally cheap in comparison with other learning methods, with training routines available in a wide range of programming languages. Because of their rule-based nature, tree-based methods are also appealing in terms of interpretability which makes them popular in many applications [21]. Commonly, a good approximation is obtained to highly nonlinear models using a cost function. Figure 2 shows a general diagram of a RT.
Making a prediction requires traversing the tree from the root node to a leaf node. Since Decision Trees are approximately balanced, traversing through the tree to make a prediction requires roughly \( \log_2(n) \) nodes, where \( n \) is the number of training samples. For simplicity the model can be constructed with each sample of the PA input signal \( x_n \) which is transformed into a feature vector \( X_n \). The vector \( X_n \) in Equation (1) serves as the input of the model.

\[
X_n = \{ z_n, z_{n-1}, ..., z_{n-m} \} \in \mathbb{R}^{3m}
\]  

where \( z_n \) is a vector containing the real, imaginary, and absolute values of the PA input signal, and \( m \) is a memory parameter which includes lagged samples of the input signal. The \( z_n \) vector is described as follows

\[
z_n = \{ \Re(x_n), \Im(x_n), |x_n| \} \in \mathbb{R}^{3}
\]  

The model output \( \bar{y} \) is then computed in an ensemble fashion with a set of regression trees RTs, which is defined in Equation (3). Since standard CART is limited to real valued inputs and outputs, different models are trained to produce the complex output.

\[
\bar{y} = y_1(X) + y_2(X)i + \epsilon_1(X) + \epsilon_2(X)i
\]  

having the complex signal I and Q components as a set of RT \( y_1 \) and \( y_2 \), and an estimated error term \( \epsilon_1 \) and \( \epsilon_2 \) trained on the difference between \( y_1, y_2 \) and the PA output signal \( y \). The estimated error terms are given as

\[
\epsilon_1(X) = y_1(X) - \Re(y)
\]  

\[
\epsilon_2(X) = y_2(X) - \Im(y)
\]  

The calculated value \( \bar{y} \) of Equation (3) is the complex output of the model. The estimated error term is used to further increase prediction accuracy.

2.1. Model Identification with Regression Tree

RTs are a model-based supervised learning technique which requires building a tree out of examples in a training dataset, in order to use that model to make predictions. The construction process of a RT is depicted in Figure 3 as a block diagram. The model identification stage starts by obtaining a vector of delayed values of the PA input signal \( x_n \). The values of the vector are compared, if data are available, then the terminal condition is tested. For non-terminal nodes, the values are sorted and a split is computed, this results in a left and right subset of the vector which are used as new inputs to the process. As terminal nodes are added to the tree, the construction process continues recursively until no more data are available.
Figure 3. Block diagram of the construction process of a RT.

The process of constructing a Decision Tree using the CART algorithm can be estimated using computational complexity theory. In this kind of analysis, the time and space resources required to run an algorithm are calculated for any input of size \( n \). Big O notation is used as a mathematical description of the growth of a function, this approach is useful due to measurement inconsistencies that arise given different algorithm parameterization, initialization, hardware implementation, data representation, and other factors. Decision Trees follow a divide and conquer scheme with runtime of \( O(mn \log_2 n) \) where \( n \) is the number of samples and \( m \) is the number of features. However, since the algorithm grows the tree recursively and repeats the process many times, the runtime can exponentially scale up to \( O(m^k n^q) \) where \( k \) and \( q \) can be any constant \( c \leq 2 \) [15]. In this particular case, the number of features corresponds to the memory depth assigned to the modeling system.

We consider two cases to denote the modeling complexity:

Case 1: In the interest of validating the complexity estimation of the algorithm, a dataset of a previous work is used to evaluate resource use. The dataset was collected from a GaN HEMT Class-AB PA with carrier frequency 2.0-GHz using an LTE-A 10-MHz bandwidth signal comprising 65,531 samples [20].

Case 2: Although this work focuses on a dataset based on a ZX60-V63+ PA which uses a carrier frequency 2.45 GHz and LTE 10-MHz and LTE 15-MHz signals with 30,720 samples, the bigger dataset is deliberately chosen to extrapolate on the algorithms resource use for greater sample sizes. The convergence running times are performed on an Intel Core i7-4790 CPU @ 3.60GHz machine running MATLAB 2019b.

The algorithm is implemented using the MATLAB function fitrtree that trains the model depending on different parameters. The MinParentSize and MinLeafSize parameters are used as the stopping criteria, the former refers to the minimum number of observations for each branch in the tree and the latter refers to the least number of observations per leaf. Both of these parameters are set to 1. The MergeLeaves and Prune parameters are set to off in order to speed up tree construction. Figure 4a portrays measurements of the training process runtime using different input sample sizes and memory values. The measured execution time of the algorithm increases with the number of input samples and memory depth according to the estimated complexity \( O(mn \log_2 n) \). A star marker is used to denote the results of the Case 2 experiments, which are the main focus of these comparisons.
Figure 4. Complexity for the; (a) Construction process of a RT and (b) Allocated memory of the RT model (c) Prediction runtime of the model. The star denotes the Case 2 approximation results.
The model sizes are calculated in bytes of memory as a reference for comparison. For this, the default data structures are changed from double to single precision floating point numbers to optimize memory resources. Despite running a simulated framework on MATLAB, an FPGA hardware resource estimation is made by calculating the number of M10K memory blocks required to store the model using the conversion 1 block equal to 10 kilobits. Figure 4b outlines the allocated memory for each instance with a single floating point representation of the values. The memory usage increases as the number of input samples becomes greater, but the same does not necessarily apply for the memory parameter. This is explained on the grounds that the algorithm selects the best split based on the input features, in this case memory, and depending on that some trees end up with more nodes than others. For example, when analyzing the model with memory 0, the depths of trees $y_1$, $y_2$, $\epsilon_1$, and $\epsilon_2$ are 44, 37, 397, and 419. In contrast, the trees $y_1$, $y_2$, $\epsilon_1$, and $\epsilon_2$ of the model with memory 14 are 35, 37, 196, and 11.

The complexity of the prediction runtime mainly depends on the depth of the tree, and it is significantly smaller in comparison to the training procedure runtime. In order to accommodate the data structures changes from double to single precision floating point numbers, a custom C implementation of the predict procedure is used. Figure 4c presents the measured runtime of the algorithm in seconds for different sample sizes and memory values. Since making a prediction requires traversing the tree, and RTs are approximately balanced, the prediction process has an estimated complexity of $O(\log_2(n))$, where $n$ is the sample size. It is noted that the memory value does not appear in the complexity expression, this is why memory does not significantly impact the prediction runtime as observed in Figure 4c.

### 2.2. Regression Tree Framework

Algorithm 1 describes the training procedure of the model, where a RT is constructed from a training dataset in a recursive manner. The algorithm tests whether or not the training dataset can be split into two subsets, if that is the case, then it computes the mean squared error (MSE) of the response values and their corresponding mean. At this point, the input and response values are sorted for each feature in the training dataset and different splits are proposed. The splits are evaluated by calculating the MSE of the left and right subsets, and the split that maximizes the MSE decrease is chosen for that node in the tree. This process is then repeated for the left and right subsets of the tree until the terminal condition is met. The MSE is defined with $y_i$ as the observed value and $\bar{y}_i$ as the corresponding predicted value as follows

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{N} (y_i - \bar{y}_i)^2$$

The model trains four RTs in order to make predictions. The first tree estimates the I component of the output signal, likewise the second tree outputs the Q component of the output signal. These two trees are evaluated and an error of the model is estimated. A second set of trees is trained to predict the estimated error. To summarize, this method is described by the following steps:

1. Obtain the input vector $X$ which contains $m$ delays of the input signal $x$.
2. Construct the RTs $y_1$ and $y_2$ for the real and imaginary components of the signal.
3. Evaluate $y_1$ and $y_2$ and compute the difference between their prediction and the PA output signal.
4. Construct the RTs $\epsilon_1$ and $\epsilon_2$ for the real and imaginary components of the estimated error.
5. Compute $\bar{y}$ using Equation (3).
Algorithm 1 Training procedure for Regression Tree.

1: procedure TRAIN_RT_TREE(T)
2:   if CAN_SPLIT(T) then
3:     $X_T, y_T \leftarrow T$
4:     $\epsilon_t \leftarrow \frac{1}{n} \sum_{i \in T} (y_i - \text{MEAN}(y_T))^2$
5:   for $f \in X_i$ do
6:     sort $X_T, y_T$ in ascending order by $X_T, f$
7:     $s^* \leftarrow \text{NULL}$
8:   for $s \in S$ do
9:     $T_L, T_R \leftarrow s$
10:    $\epsilon_{t_L} \leftarrow \frac{1}{n} \sum_{i \in T_L} (y_i - \text{MEAN}(y_{T_L}))^2$
11:    $\epsilon_{t_R} \leftarrow \frac{1}{n} \sum_{i \in T_R} (y_i - \text{MEAN}(y_{T_R}))^2$
12:    $\Delta \epsilon \leftarrow \epsilon_t - \epsilon_{t_L} - \epsilon_{t_R}$
13:    $s^* \leftarrow \text{MAX}(\Delta \epsilon)$
14:   end for
15: end for
16: $t_L \leftarrow \text{TRAIN_RT_TREE}(T_L)$
17: $t_R \leftarrow \text{TRAIN_RT_TREE}(T_R)$
18: return $X_{t,f}, y_i, t_L, t_R$
19: end if
20: return NULL, MEAN($y$), NULL, NULL
21: end procedure
22:
23: procedure TRAIN_RT(T)
24:   $X, y \leftarrow T$
25:   $y_1 \leftarrow \text{TRAIN_RT_TREE}(X, \mathbb{R}(y))$
26:   $y_2 \leftarrow \text{TRAIN_RT_TREE}(X, \mathbb{S}(y))$
27:   $\bar{y} \leftarrow y_1(X) + y_2(X)i$
28:   $\bar{\epsilon} \leftarrow y - \bar{y}$
29:   $\epsilon_1 \leftarrow \text{TRAIN_RT_TREE}(X, \mathbb{R}(\bar{\epsilon}))$
30:   $\epsilon_2 \leftarrow \text{TRAIN_RT_TREE}(X, \mathbb{S}(\bar{\epsilon}))$
31: return $y_1, y_2, \epsilon_1, \epsilon_2$
32: end procedure
33:

Notation: $T, T_L, T_R,$ denote the training dataset, the left subset of a split, and the right subset of a split. The input and response values $X_{t,f}$ and $y_i$ are contained within $T$. The $S$ values correspond to the proposed splits and $\epsilon_t, \epsilon_{t_L}, \epsilon_{t_R}, \Delta \epsilon$ represent the MSE of the response values, the MSE of the left subset of a split, the MSE of the right subset of a split, and the MSE decrease of a split. The $y_1, y_2, \epsilon_1,$ and $\epsilon_2$ are the RTs obtained after the training procedure.

2.3. Random Forest Framework

RF is a supervised learning algorithm for general purpose classification and regression tasks. RF consist of a collection of Decision Trees, where each tree is constructed using randomly selected subsets of predictor variables, and the output is computed by aggregating the predictions for each observation [22]. The method is recognized for its accuracy and ability to work on small sample sizes and high-dimensional feature spaces [23]. The algorithm grows $M$ randomized RTs. Before the construction of each tree, $a_n$ observations are randomly sampled from the original dataset and these observations are the only ones used in the construction process for that tree. The output value at input $x$ is denoted by $m_n(x, \Theta_1, D_n)$, having $D_n$ as the original dataset and $\Theta_1, \ldots, \Theta_M$ as an independent random variable which is used to resample the original dataset prior to training. The trees are combined to form the prediction as Equation (7).
\[ m_{M,n}(x_1, \Theta_1, \ldots, \Theta_m, D_n) = \frac{1}{M} \sum_{j=1}^{M} m_n(x_j; \Theta_j, D_n) \]  

(7)

The RF training procedure is described in Algorithm 2. The procedure trains \( M \) different trees using randomly sampled subsets \( a_i \) of the dataset and also a randomly sampled subset of input features \( M \). In this particular case, \( M \) refers to the memory values included as inputs. For PA modeling, a model has to be constructed for the real and imaginary values of the complex output signal, in a similar fashion to RTs. Since RF uses a combination of trees, the time and space complexity of the algorithm also depends on the construction process of the aforementioned trees. The MATLAB TreeBagger class is used as the implementation of RF, with the MinLeafSize parameter set to 1 and the MinParentSize parameter set to 10 as the stopping criteria. The total number of trees \( M \) is set to 12; therefore, an increase in resource usage is expected as a result of using six times as many trees in comparison to the RT approach. Figure 5 depicts (a) the training runtime for the RF algorithm with different memory values (b) the allocated memory for each instance with a double floating point representation of the values (c) the prediction runtime of the model. The training procedure executes faster than RT due to the different parametrization on MinParentSize, which reduces the tree size and consequently the number of iterations that are performed during training. The expected 6 times increase is reflected on the prediction runtime; however, the memory consumption of the algorithm is aggravated considering the use of more trees with double floating point representations instead of single floating point, which in turn results in more allocated memory.

![Graphs](https://via.placeholder.com/150)

(a) Random Forest Training Runtime  
(b) Random Forest Allocated Memory  
(c) Random Forest Prediction Runtime

**Figure 5.** Complexity for the; (a) Construction process of a RF and (b) Allocated memory of the RF model (c) Prediction runtime of the model. The star denotes the Case 2 approximation results.
Algorithm 2 Training procedure for Random Forest.

1: procedure TRAIN_RF_TREE(T)
2:     if CAN_SPLIT(T) then
3:         X_T, y_T ← T
4:         e_t ← \frac{1}{2} \sum_{i \in T} (y_i - MEAN(y_T))^2
5:         Sample subset \mathcal{M} uniformly in X_t
6:         for f ∈ \mathcal{M} do
7:             sort X_T, y_T in ascending order by X_T, f
8:             s^* ← NULL
9:             for s ∈ S do
10:                 T_L, T_R ← s
11:                 e_{tL} ← \frac{1}{2} \sum_{i \in T_L} (y_i - MEAN(y_{T_L}))^2
12:                 e_{tR} ← \frac{1}{2} \sum_{i \in T_R} (y_i - MEAN(y_{T_R}))^2
13:                 \Delta e ← e_t - e_{tL} - e_{tR}
14:                 s^* ← MAX(\Delta e)
15:             end for
16:         end for
17:         t_L ← TRAIN_RF_TREE(T_L)
18:         t_R ← TRAIN_RF_TREE(T_R)
19:         return X_{t,f}, y_{t}, t_L, t_R
20:     else
21:         return NULL, MEAN(y), NULL, NULL
22:     end if
23: end procedure

2: procedure TRAIN_RF(T, M)
3:     D_n ← T
4:     for m ∈ M do
5:         Sample points a_i uniformly in D_n
6:         m ← TRAIN_RF_TREE(a_i)
7:     end for
8:     return M
9: end procedure

2.4. Surrogate Cubic-Spline Precomputed Model Approach

For the first model capabilities demonstration of the CS, an AM-AM behavioral curve whose thickness with very smooth data can be extracted with a fast convergence speed with lower training samples for a polynomial complexity given as shown in Figure 6, while in Figure 7 we exemplify an AM-AM behavior that shows a thick high sparse data that increase the extraction complexity. Therefore, we formulate the mathematical models with the Bézier polynomial curves that can be described in terms of splines y(u) using also the combination of allied to control points \mathbf{c}_i, as described in the standard Formula (8), unconditional stability is guaranteed. Additionally, N^n_n(u) are piece-wise polynomial functions with finite support that satisfies certain continuity conditions. In the recurrence fashion relationship to define b-splines basis, it is considered for simplicity a sequence (a_i) doubly infinite of simple knots, such that a_i < a_{i+1} for all i. Therefore, the b-splines N^n_n are defined through the recurrence relationship as in (9). In the base model we use two parameters that are determined to not only maintain stability but also ensure the desired accuracy. Thus, this model is a simple step-by-step algorithm denoted by (8). Therefore, the stability and accuracy analysis of the b-spline algorithm exhibits an efficiency and computational cost viable for the proposed method and is demonstrated through two numerical simulations [24].
\[ y(u) = \sum c_i N_i^0(u) \] (8)

\[ N_i^0(u) = \begin{cases} 1 & \text{if } u \in [a_1, a_1 + 1) \\ 0 & \text{otherwise} \end{cases} \] (9)

\[ N_i^0 = a_i^{n-1} N_i^{n-1}(u) + (1 - a_i^{n-1}) N_i + 1^{n-1}(u) \]

where

\[ a_i^{n-1} = (u - a_i) / (a_{i+1} - a_i) \] (10)

is a local parameter with respect to the support of \( N_i^{n-1} \). Thus, for b-spline recursive as in (9), it is found by iterative determining the most relevant functions according to the reduction linearization coefficients, the model can be simply derived as a cubic-spline basis function in terms of gain expansion \( G_m \) described as

\[ y(n) = \sum_{m=0}^{M-1} G_m |z(n-m)|^2 z(n-m) \] (11)

where \( y(n) \) denotes the received RF signal associated to an \( m \) memory depth value. Following the notation in a recursive way, the containing gain factors \( G_m(x) \) are expanded in terms of 1D cubic-spline basis functions given by,

\[ G_m(x) = \sum_{j=1}^{l} a_j B_j(x) \] (12)

where \( a_j \) corresponds to the real-valued coefficients of the \( j \)-th spline expansion, and the \( B_j(x) \) function. LS solution is used to develop an error model with the goal of minimizing estimations, with a matrix definition given as,

\[ G_m(x) = \sum_{i=0}^{N_k-1} c_i(m) \phi_i(|x|) \]

\[ \phi_i(|x|) \triangleq \delta_{i,u} \]

\[ = \begin{cases} 1 & \forall \ i = u \\ 0 & \forall \ \text{other knots} \end{cases} \]

\[ \forall \ i \in [0, N_k], \forall \ u \in [0, N_k - 1] \]

where \( c_i(m) \) are the weight model coefficients and the gain function \( G_m(x) \) for each memory delay index \( m \), and \( N_k \) number of splines. In (13) the uniform \( \phi_i \) basis functions with an entire amplitude span decay swiftly from 1 at \( (u) \) to 0 at all other knots thus yield to evaluate the limits of a memoryless PA and deterministic memory mitigation. Thus, a smooth function degree \( N_k \), as the value and first \( N_k - 1 \) derivatives of the spline function will be continuous [25]. The one-dimensional cubic B-splines \( (N_k = 3) \) are for instance the most common form, and have continuous value, first, and second derivatives.
Figure 6. Approximation for a fitted model with polynomial basis.

Figure 7. Approximation model with b-spline basis.

3. Implementation and Measurement

Experimental Setup Testbed

The captured signals are then used for training and testing of different ML models. The PA testbed can perform signal playback and data capture from the PA device under test assuming different instantaneous bandwidths (IBWs). The experimental setup considers the acquisition of real-valued data from several run tests to calculate crest factor reduction (CFR), PAPR, and adjacent channel power ratio (ACPR) signal measurements considering the device under test (DUT).

Therefore, in the proposed system, a transceiver software-defined radio (SDR) base makes suitable the model’s implementation and the experimental results. The SDR platform consists of a system-on-chip (SoC) device to provide a software and hardware interface to develop an SDR system. Moreover, the transceiver board uses an interface (AXI) interconnect which provides very low latency with high throughput. The radio-frequency conversion system includes a direct conversion with $2 \times 2$ channels for the (I) in-phase and (Q) quadrature signals. The system also includes a filter stage with an interpolation 128-tap type finite impulse response (FIR) that can handle and control by an adjustable internal sampling rate to produce a 12-bit output signal. The testbed setup considers a ZX60-V63+ power amplifier DUT with 50 ohms output impedance, biased with a 5 V bias, and provides a typical gain of around 20 dB under a 2.45-GHz frequency operating condition. In [12], a DPD linearization is validated for LTE and 256-QAM modulated signals that can linearize the dynamic nonlinearity of the ZX60-V63+ PA.
One can observe from Figure 8 the block diagram of the experimental setup. For the digital part (A) it consists of software and hardware interfacing based on an ADI Linux libiio library runs on a Microblaze ARM (Zynq) architecture that uses an embedded Linux to provide a full-duplex communication from MATLAB and the FPGA hardware implementation. Therefore, the RF transceiver AD9361 transmits the baseband signal; thus, MATLAB casts and stores the data in the DDR memories of the FPGA. The data communication link is made over Ethernet protocol between the ADI Hardware connected to the FPGA-SoC platform. While, in the analog part (B), the RF transceiver generates a stimulus to the ZX60-V63+ PA. The radio-frequency up-conversion path uses a quadrature modulator driven by an internal local oscillator (LO) set at 2.45 GHz frequency to generate upconverted signal.

**Figure 8.** Proposed testbed block diagram of the characterization and linearization for the PA. (A) denotes the FPGA digital to RF stage, and (B) analog part with devices and measurement equipment.

Figure 9 shows a real-time acquisition system; in this process, the DUT is ZX60-V63+ PA for the behavior acquisition. The developed system in the SDR board with a dual channel transceiver operating at 2.45 GHz. The programming of the AD9361 transceiver system provides a wideband capacity ideal for radiofrequency applications; the SDR system has flexibility for mixed-signal baseband selection and internal synthesizers to operate on 2.45 GHz as a carrier frequency. Systems of this type allow using current communication standards, such as frequency-division duplexing (FDD) and interfacing with baseband processors using the 12-bit parallel data port for dual channels. Additionally, an FIR filter is used for internal adjustment of LTE signals with bandwidths of 10 and 15-MHz. The ZX60-V63+ operates at a gain of 18.79 dB for LTE-10 and 18.45 dB for LTE-15, under a bias of 5 V, a 30.72 MSPS is used, with a received signal strength indicator (RSSI) of 103 dB, and an automatic gain control (AGC) to keep amplitude variations and gain for a waveform preamble phase. In this test, the gain control among the blocks achieved by the internal AGC is enabled, such as slow mode gain control. In order to perform the tests power amplifier DUT output measurement is initially measured in the frequency domain with a GW INSTEK GSP-830 as shown in Figure 8. Therefore, the setup for NMSE, PAPR, and ACPR is calculated in MATLAB through the LTE toolbox and the RF Blockset models for Analog Devices RF Transceivers. The data input is cast from the host computer, and the ARRadio card setting tests operates with an AD 9361 RF transceiver using a carrier frequency 2.45-GHz and an LTE10-MHz with a sampling rate of 245.76 MHz and 122.88 for the LTE 15-MHz and performed for an IFFT window size of 1048 points. For the particular validation tests every iteration, the PAPR is obtained. Furthermore, the CCDF is calculated to represent the probability that the envelope actual signal exceeds a defined threshold. The LTE signal data are measured during a given interval to determine the probability under
OFDM multiplexing for the PAPR and CCDF are calculated with the real-valued samples data as:

\[
PAPR(n) = \frac{|x(n)|^2}{\frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2}
\]

(14)

CCDF is related to the PAPR calculation based on the relation:

\[
CCDF(\%) = P_{\text{ove}}(\text{PAPR} > P_{\text{tar}})
\]

(15)

where \(P_{\text{ove}}\) provides information on the overall time of the signal duration at or above a specific power level, and \(P_{\text{tar}}\) is denoted as the PAPR target. Previous to this step, the setup system is adequately calibrated so that a synchronized signal is applied properly to the PA input. Please note that we set a calibration power for power peak and average levels at the PA output after each iteration to ensure adequate CFR amplitude levels in the overall system [1]. Figure 9 (1) (2) shows the transceiver set at 2.45 GHz center frequency using the ARRadio card integrates an AD9361 RF transceiver. Once the sample rate has been established, and radio-frequency bandwidth, the signal implemented in MATLAB is designed, as shown in Figure 9 (6), performing with 64-QAM modulation. Moreover, the processes of LTE parameter store can be observed for a specific LTE mode selected, while the transmission and reception path using the RF transceiver board (AD-FMCOMMS3-EBZ) and the MATLAB library to pre- and post-processing captured real-valued data.

Figure 9. Experimental testbed photograph. Relevant instruments depicted as follows. 1: Altera Cyclone V FPGA SoC-Kit. 2: AD9361 RF Agile Transceiver working at 2.45 GHz center frequency. 3: Power amplifier Mini-Circuits ZX60-V63+. 4: Power Supply GW INSTEK GPS-3303. 5: Spectrum Analyzer GW INSTEK GSP-830. 6: Display HOST PC-MATLAB.

4. Experimental Design

4.1. Pre-Processing and Post-Processing in System Data-Driven Modeling

The procedure for the algorithms extraction is performed to ensure a real-time FPGA implementation. Thus, it is based on the measured outputs which are post-processed in MATLAB with subsampled synchronization to compensate for delays in time and phase; thus, a minimum delay is given by a correlation index \((i \times 5.00 \, \text{ns})\) according to the clock running at 200 MHz for direct memory access (DMA) between the LPC bus interface of
transmitter-receiver or input/output data of the PA using samples of \( y(n - i) \) in the AD9361 core for the FPGA data acquisition. In this manner, and as we explained the desirable model for the performance evaluation, considers the model prediction ability that should be tested to verify its robustness compared to CS, RF, and RT models. In the experiments, two cases are considered with datasets of 65,651 and 30,720 samples length, respectively. Both are used to compare the PA model extraction, PD model construction and DPD coefficients estimation to achieve a balance between the modeling and linearization performance, runtime, and computational complexity, which increases with the samples depth. To obtain the PA behavioral models and accuracy for the NMSE of the AM-AM, AM-PM prediction is used to provide the differences between estimated and real-time measurement data, the NMSE is expressed in dB as follows:

\[
\text{NMSE} (n)_{\text{dB}} = 10 \log_{10} \left( \frac{\sum_{n=1}^{N} |y(n) - \bar{y}(n)|^2}{\sum_{n=1}^{N} |y(n)|^2} \right)
\]

(16)

where \( y(n) \) is the actual data and \( \bar{y}(n) \) the predicted models in time convergence and accuracy of the estimated PA output.

4.2. Framework Development and Application of the PA Models Study Case

In this work, the performance of LTE signals of 10 and 15-MHz bandwidths are evaluated for commercial LTE applications. For the evaluation a single band LTE signal is used as the input stimulus to drive a wideband power amplifier ZX60-5916MA+ from Mini-Circuits. According Figure 10 a PA model extraction is performed based on the ILA architecture, while procedure (A) can select the parameters for the PA output model \( y(n) \) extraction under the spline parameters (i.e., memory \( m \), knots, coefficients \( c_{i}^{(m)} \), breakpoints, etc.) and/or RT parameters (i.e., trees, regression trees \( \epsilon_{1}, \epsilon_{2} \), etc.) prior the algorithm iteration in order to set the training framework adequately for each AM-AM conversion model. Thus, the procedure (B) is copying the coefficients for the PD model construction. The NMSE performance and runtime for the RT model, model extraction, and validation consider a dataset with 30,720 samples acquired from the experimental testbed. RT gives a best performance for ML algorithm in terms in NMSE with a value of \(-53.0346 \text{ dB} \) NMSE in modeling is obtained for the PA RT model output.

Figure 10. An extract feature block model approach for the PA (connections model referred to a feature surrogate dependent on a complex input envelope with an indirect learning approach). (A) denotes the PA model extraction and, (B) the PD construction process.

4.3. Cubic-Spline Algorithm for Coarse-Data Modeling

A behavioral model is used in the first case to develop a memoryless predistor, while the ILA technique is used to correct the residual spectral regrowth ensuring the memory trade-off between complexity and NMSE performance. In this work, they are performed for the LTE digital multiplexing OFDM FDD under 64-QAM modulation. For the residual ACPR spectral regrowth and their CFR metric associated its measured for the transmitter chain distortion, the characteristics can be seen in the AM-AM curve of the
DUT PA, modeling, and linearization stage is developed based on the cubic-spline model as shown in Figure 11 and is defined as follows:

\[ y(n) = \sum_{m=0}^{ML-1} (|x(n-m)|^2)x(n-m) \]

\[ G_m(x) = \sum_{i=0}^{N_k} c_i^{(m)} \varphi_i(|x|^2) \]  

(17)

Based on the peak gain of the system, the \( y_d(n) \) is determined, which is expressed by the following:

\[ y(n) = \frac{z(n)}{G_m(n)} \]  

(18)

where \( G_m(n) \) is based on the maximum values of \( x(y) \) and \( y(n) \), as follows

\[ G_m(n) = \frac{y(n) \max |x(n)|}{\max |y(n)|} \]  

(19)

It can be noticed that for even nonlinear behavior effect and parameter estimation can be obtained by different basis functions, such as conventional polynomials, according to Equation (17), assuming a gain factor that can be expanded as in (13) in terms of a one-dimensional CS basis function to the gain \( G_m(|x(n)|^2) \).

![Figure 11. Modeling extraction and evaluation for the LTE 10-MHz and construction process of the AM-AM PA model for the DPD output linearization with CS model basis for a z(n) PD model normalization ≤ 0.9999 after 5 iterations reach an NMSE −48.7995 @ 2.45 GHz.](image)

4.4. Model Complexity Analysis Assessments and Comparison

The model inversion considers the computation of the \( 8 \times 8, 16 \times 16, 32 \times 32, 64 \times 64 \) matrix inversion associated with the hardware resources targeted for each model at the FPGA device Cyclone V SX SoC—5CSXFC6D6F31C6N. For the complexity calculation, the corresponding amount for each matrix dimension considers the following available resources, that is 110K LEs, 41509 ALMs, 5140 M10K memory blocks, which also corresponds with the complexity limits in Figure 12.
Figure 12. FPGA hardware resource use for the AM-AM PA model with RT implementation (in terms of ALM, DSP, M10K, and Memory Bits).

It is noted that for simplicity, we assume in this work that the matrix form in Equation (11) is expressed as follows

\[ Y = H_{N \times L} C_{L \times 1} \]  

(20)

where \( Y \) is the vector of the output signal, \( C \) is the coefficient vector, and \( H \) is the matrix of the model composed of the terms in (13). \( N \) is the number of samples. \( L \) represents the number of coefficients and \( L = 2K(M + 1) \). After using the LS, the coefficient vector \( C \) can be expressed as

\[ \bar{C} = (H^H H)^{-1} H^H Y \]

(21)

As regards the RT model, it is found that the RT prediction closely resembles the PA output in terms of the gain of the normalized PA output signal and the CCDF of the PA output signal. Figure 13 depicts the aforementioned RT modeling performance for the (a) normalized gain (b) CCDF of the PA output signal and the RT model.

Figure 13. (a) Gain for the normalized output signal using RT prediction model (b) CCDF for the PA output signal and the RT prediction model.
The evaluation of the RT model is performed on the AM-AM characteristic of the PA with noticeable similarity seen in Figure 14a. For further assessments, the normalized power spectrum of the PA output signal is also compared against the RT model prediction in Figure 14b, the model exhibits when a notable gain expansion occur for low power levels, and the corresponding optimum model is desirable for a narrow bandwidth.

![Figure 14. Modeling extraction and evaluation for the; (a) Construction process of the AM-AM PA model with RT and (b) PSD of the output PA spectral regrowth with RT model basis.](image)

For the linearization of the power amplifier (CS model) ILA is used; the spectral regrowth of the received signal is observed as a result of the use of the amplifier where DPD is not used as before the PA, while in Figure 15 we observe the spectrum of the received spectra (PA output), while its power values decrease in the bands after parametrization of the CS model (PD CS model) under 5 iterations, that is, required only 90 number of coefficients for the CS model.

![Figure 15. PSD for the LTE 10-MHz PA output signal and the CS prediction model with the actual predistorter z(n) model normalization ≲ 0.9999 after 5 iterations reach an NMSE = −48.7995 @ 2.45 GHz.](image)

The predistorter training is performed in an iterative way; as noted in Figure 15, the algorithm is able to execute a condition to fit the best convergence in terms of NMSE. In this case, the signal is measured in the Tx under a 64-QAM modulation scheme for both bandwidth cases. For the LTE-10 MHz an average output power of −21.11 dBm is obtained with a gain of 18.79 dB while for the LTE-15 MHz a gain of 18.45 dB is achieved. Thus, the validation of the DPD coefficients after training and feeding as the new input to the linearization performance algorithm given its ability to the output signal at the PA then exhibits an adequate spectral regrowth decrease. Figure 15 shows the model iteration of the linearization figures of merit with several selected coefficients for the PA = 22 and
PD = 68 models, respectively, that can obtain the best NMSE with only 90 coefficients used for the cubic-spline algorithm. It is noted that the overall MATLAB procedure acquires data using the PA model extraction, PD model, AM-AM, AM-PM, and linear output model without any iterations.

For these particular validation tests, we iterate the model five times in order to show the runtime convergence for the defined parameters listed in Table 1. For the PA/PD behavioral model extraction using the algorithms of RF, RT, and CS after 5 iterations the results are listed in Table 2. As observed in PSD metrics of Figures 15 and 16b, an effective linearization is achieved after 5 times of the CS algorithm iteration. The validation demonstrates that it is possible to achieve up to 1.52 dB from the first to the fifth iteration for the LTE 10-MHz, and 0.52 dB for the LTE 15-MHz case with a lower runtime, as shown for each bandwidth signal, respectively, before and after the iterations (see CS† (1st iter) and CS‡ (5th iter) in Table 2).

As proof of the CS modeling capabilities with the LTE 15-MHz bandwidth signal, Figure 16a,b shows the AM-PM characteristic and the PSD, respectively, for both the measured and modeled data from the PA DUT. The ACPR of the PA output and the PD CS model achieves up to 22.57 dB for the best case of −43.90 dBc ACPR reduction. It can be noticed that a linear DPD output model is achievable, and Table 1 collects and summarizes the parameters for the proposed spline-based models employed in the construction of the PA model extraction and PD prediction, respectively.

Table 1. Parameters summary for the PA extraction and PD prediction with CS model.

| Signal          | Parameters | Coefficients |
|-----------------|------------|--------------|
|                 | Memory PA  | Memory PD    |
| LTE 10-MHz      | 2          | 5            |
| LTE 15-MHz      | 3          | 6            |

Table 2. NMSE and runtime convergence for the LTE 10-MHz and LTE 15-MHz signals.

| Model          | LTE 10-MHz NMSE (dB) | LTE 10-MHz Runtime (s) | LTE 15-MHz NMSE (dB) | LTE 15-MHz Runtime (s) |
|----------------|----------------------|------------------------|----------------------|------------------------|
|                | Training             | Prediction             | Training             | Prediction             |
| Random Forest  | −25.0470             | 10.7984                | 0.631920             | 12.5964                |
| Regression Tree| −53.0346             | 2.3700                 | 0.048308             | 2.84020                |
| CS† (1st iter) | −47.2767             | 0.1925                 | −41.6136             | 0.3586                 |
| CS‡ (5th iter) | −48.7995             | 1.16                   | −42.1412             | 1.8889                 |

CS† and CS‡ denote the prediction results shown in Figures 15 and 16a,b.
4.5. Discussion

Further studies of state-of-the-art machine learning algorithms applied to PA modeling are compared in Table 3. The present work is devoted to providing a surrogate alternative to models based on RT, RF, and CS algorithms that can be adopted to obtain low complexity PA and PD models for power amplifier characteristics. Additionally, a development SDR platform successfully aids in the search for adequate matrix inversion calculations to validate the proposed models based on an FPGA hardware architecture, whose performance shows an adequate complexity reduction and runtime. Furthermore, we emphasize the high dispersion of AM-AM curves that become high complexity in matrix inversion, but it is demonstrated that the proposed CS and RT can adequately model the PA and PD behavior while having precise NMSE extraction-prediction approximation and learning complexity. The developed models for the PA extraction and update coefficients for the PD model provide a good time convergence and NMSE; this gives an advantage in approximation to very thickness AM-AM and AM-PM behaviors as is in the case of Figures 11 and 16, respectively. With the same principle of obtaining a lower model complexity that yields an effective linearization while using the CS model, we demonstrate the model extraction validation with an affordable runtime when it runs five times iteratively as revealed in Figure 15. The RT model is quite simple yet widely recognized as an effective learning technique for nonlinear problems that establishes the grounds for more elaborated ensemble methods such as RF [26]. These algorithms are able to operate on high-dimensional datasets and select the most biased feature, their comprehensibility nature makes it easier to interpret the model with an enhanced view of performance outcomes [27].

Table 3. Performance comparison with some Machine Learning-based algorithms.

| Model                      | Freq (MHz) | Signal/BW (MHz) | NMSE (dB) | PAPR (dB) | ACPR (dBC) | Ref.         |
|----------------------------|------------|-----------------|-----------|-----------|------------|-------------|
| Regression Tree *          | 2450       | LTE/10 & 15-MHz | −53.03 *  | 9.3       | −43.5  | This Work   |
| Random Forest              | 3600       | LTE/30-MHz      | −48.40    | 9.6       | −49, −56 | [11]        |
| Gradient Boosting          | 1900       | 5G NR/60-MHz    | −38.00    | –         | −43.46 | [28]        |
| Neural Network             | 1900       | 5G NR/60-MHz    | −31.58    | –         | −39.97 | [28]        |
| Chebyshev Polynomial LSTM †| 2800       | 5G NR/100-MHz   | −36.12    | 9.6       | –      | [29]        |
| AVDTDNN                    | 3750       | OFDM/40-MHz     | −38.50    | 6.9       | −48.95/ −49.09 | [30]        |

* After PA model extraction for the proposed RT model without any iteration. † A long short-term memory (LSTM) network is considered.

This work expands upon the modeling stage of the PA linearization process and compares the performance of RT, RF, and CS models. It presents an ensemble approach to the RT algorithm that estimates an NMSE term and corrects the model output, this allows the model to better fit the data and achieve greater accuracy at the cost of higher memory consumption. A computational complexity breakdown is realised in order to quantify the performance improvements over previous methods, the analysis is sustained by various measurements on resource use, such as the execution runtime of the algorithm, and the size of the resulting model. An estimation of FPGA memory consumption is calculated for the aforementioned tree-based methods that run on MATLAB, and it is then compared to the CS model with 1 and 5 iterations. The results are presented in terms of NMSE and execution runtime in seconds, while previous works on tree-based models scored an error of −30.93 dB with an execution runtime of 16.82 seconds [20], the improved approach is able to obtain an error of −53.03 dB in 2.41 seconds in the case of the RT algorithm. The addition of an error term, and optimizations to parameters and memory consumption contribute to an overall enhancement of the model. The strong precision on RT is corroborated by the NMSE results of Table 2, having a sufficient runtime compared to other approaches. The RF
method introduces different models with more diverse predictions that, when combined reduces variance on the model which in turn leads to a more general output.

5. Conclusions and Future Work

In this work, a feature comparison for algorithm performance is developed to provide a fair comparison of ML methods including RT, RF, and a cubic-spline basis for PA modeling. The proposed validation is carried out with 64-QAM modulation LTE−10 and 15-MHz signal bandwidths to develop the resulting algorithm to exploit feature characteristics of DPD solution as surrogate models in terms of extraction and evaluation, while also accurate enough to enhance convergence and steer the iterations target to the global optimum following the same approach by using the AM-AM/AM-PM real-valued data characteristics. The measurement examples for DUT performance based on the ILA architecture offer real-time convergence on results for each algorithm developed in order to obtain a precise approximation for the PA (e.g., extraction, prediction, linearized model) under test while with LTE signals are used as stimulus. Finally, for the data conditioning aspects from three models compared the NMSE is provided with a good accuracy when the complexity is preserved according to the smoothness-straightness dynamic dependencies for the AM-AM characteristics considered at each developed algorithm while preserving model with a few samples, such as the ML based and feature basis splines construction to give a fair comparison. Further development of the present models considers an engine based on the FPGA-based system present in order to adequate precision in terms of floating point.

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

Funding: This research was funded by the National Technological Institute of Mexico (TECNM) research project under grant 14284.22-P, and by Program for Teacher Professional Development (PRODEP) under the project ITTIJ-CA-9.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to acknowledge the support and collaboration of the National Technological Institute of Mexico (TECNM), and the School of Engineering and Sciences at Tecnologico de Monterrey.

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

References

1. Cárdenas-Valdez, J.R.; Galaviz-Aguilar, J.A.; Vargas-Rosales, C.; Inzunza-González, E.; Flores-Hernández, L. A Crest Factor Reduction Technique for LTE Signals with Target Relaxation in Power Amplifier Linearization. Sensors 2022, 22, 1176. [CrossRef]

2. Galaviz-Aguilar, J.A.; Vargas-Rosales, C.; Cárdenas-Valdez, J.R.; Martínez-Reyes, Y.; Inzunza-González, E.; Sandoval-Ibarra, Y.; Nuñez-Pérez, J.C. A Weighted Linearization Method for Highly RF-PA Nonlinear Behavior Based on the Compression Region Identification. Appl. Sci. 2021, 11, 2942. [CrossRef]

3. Khusro, A.; Hashmi, M.S.; Ansari, A.Q.; Auyenur, M. A new and Reliable Decision Tree Based Small-Signal Behavioral Modeling of GaN HEMT. In Proceedings of the IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), Dallas, TX, USA, 4–7 August 2019; pp. 303–306.

4. Kant, S.; Bengtsson, M.; Fodor, G.; Goransson, B.; Fischione, C. EVM Mitigation with PAPR and ACLR Constraints in Large-Scale MIMO-OFDM Using TOP-ADMM. IEEE Trans. Wirel. Commun. 2022. [CrossRef]

5. Belabad, A.R.; Motamedi, S.A.; Sharifian, S. A Novel Generalized Parallel Two-Box Structure for Behavior Modeling and Digital Predistortion of RF Power Amplifiers at LTE Applications. Circuits Syst. Signal Process. 2018, 37, 2714–2735. [CrossRef]
6. Becerra, J.A.; Madero-Ayora, M.J.; Marqués-Valderrama, E.; Nogales, M.; Crespo-Cadenas, C. Preconditioning the Regression of Power Amplifier Behavioral Models and Digital Predistorters. In Proceedings of the IEEE Topical Conference on RF/Microwave Power Amplifiers for Radio and Wireless Applications (PAWR), Las Vegas, NV, USA, 16–19 January 2022; pp. 58–61.

7. Campo, P.P.; Brihuega, A.; Anttila, L.; Turunen, M.; Korpi, D.; Allen, M.; Valkama, M. Gradient-Adaptive Spline-Interpolated LUT Methods for Low-Complexity Digital Predistortion. IEEE Trans. Circuits Syst. I Regul. Pap. 2021, 68, 336–349. [CrossRef]

8. Jalili F.; Zhang, Y.; Hintsala, M., Jensen, O.K.; Chen, Q.; Shen, M.; Pedersen, G.F. Highly non-linear and wide-band mmWave active array OTA linearisation using neural network. IET Microwaves Antennas Propag. 2022, 16, 62–72. [CrossRef]

9. Zhang, C.Y.; Zhu, Y.Y.; Cheng, Q.F.; Fu, H.P.; Ma, J.G.; Zhang, Q.J. Extreme learning machine for the behavioral modeling of RF power amplifiers. In Proceedings of the IEEE MTT-S International Microwave Symposium (IMS), Honolulu, HI, USA, 4–9 June 2017; pp. 558–561.

10. Lu, G.; Huang, S.; Ying, S.; Zeng, Y.; Yao, Y.; Ning, F.; Feng, Z. Low-Complexity Power Amplifier Model Driven Wireless Emitter Identification Using Random Forest. In Proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.

11. Álvarez-López, L.; Becerra, J.A.; Madero-Ayora, M.J.; Crespo-Cadenas, C. Determining a Digital Predistorter Model Structure for Wideband Power Amplifiers through Random Forest. In Proceedings of the IEEE Topical Conference on RF/Microwave Power Amplifiers for Radio and Wireless Applications (PAWR), San Antonio, TX, USA, 26–29 January 2020; pp. 50–62.

12. Bhatt, M.; Tripathi, G.C.; Rawat, M.; Mathur, S. Measurement results for compensation of impairments of LTE stimulated direct-conversion transmitter through augmented MoE. Int. J. RF Microw. Comput.-Aided Eng. 2021, 31, e22825. [CrossRef]

13. Brihuega, A.; Abdelaziz, M.; Anttila, L.; Li, Y.; Zhu, A.; Valkama, M. Mixture of Experts Approach for Piecewise Modeling and Linearization of RF Power Amplifiers. IEEE Trans. Microw. Theory Tech. 2022, 70, 380–391. [CrossRef]

14. Barry, A.; Li, W.; Becerra, J.A.; Gilabert, P.L. Comparison of feature selection techniques for power amplifier behavioral modeling and digital predistortion linearization. Sensors 2021, 21, 5772. [CrossRef] [PubMed]

15. Sani, H.M.; Lei, C.; Neagu, D. Computational Complexity Analysis of Decision Tree Algorithms. In Proceedings of the Artificial Intelligence XXXV—SGAI 2018, Cambridge, UK, 11–13 December 2018; p. 11311.

16. Breiman, L.; Friedman, J.; Stone, C.J.; Olshen, R.A. Classification and Regression Trees; Chapman and Hall/CRC: New York, NY, USA, 2017.

17. Becerra, J.A.; Ayora, M.J.M.; Reina-Tosina, J.; Crespo-Cadenas, C. Sparse Identification of Volterra Models for Power Amplifiers Without Pseudoinverse Computation. IEEE Microw. Theory Tech. 2020, 68, 4570–4578. [CrossRef]

18. Crespo-Cadenas, C.; Madero-Ayora, M.A.J.; Becerra, M.A.J.; Cruces, S. A Sparse-Bayesian Approach for the Design of Robust Digital Predistorters Under Power-Varying Operation. IEEE Trans. Microw. Theory Tech. 2022, 70, 4218–4230. [CrossRef]

19. Zhu, A.; Pedro, J.C.; Brazil, T.J. Dynamic Deviation Reduction-Based Volterra Behavioral Modeling of RF Power Amplifiers. IEEE Trans. Microw. Theory Tech. 2006, 54, 4323–4332. [CrossRef]

20. Aguila-Torres, D.S.; Galaviz-Aguilar J.A.; Cárdenas-Valdez, J.R. Reliable comparison for power amplifiers nonlinear behavioral modeling based on regression trees and random forest. In Proceedings of the 2022 IEEE International Symposium on Circuits and Systems (ISCAS), Austin, TX, USA, 29 May–1 June 2022; pp. 1527–1530.

21. Carriozza, E.; Molero-Rio, C.; Romero Morales, D. Mathematical optimization in classification and regression trees. TOP 2021, 29, 5–33. [CrossRef]

22. Steiper, J.L.; Miller, M.E.; Tooze, J.; Ip, E. A comparison of random forest variable selection methods for classification prediction modeling. Expert Syst. Appl. 2019, 134, 93–101. [CrossRef]

23. Biau, G.; Scornet, E. A random forest guided tour. TEST 2016, 25, 197–227. [CrossRef]

24. Rostami, S.; Shojaee, S. A Family of Cubic B-Spline Direct Integration Algorithms with Controllable Numerical Dissipation and Dispersion for Structural Dynamics. Iran J. Sci. Technol. Trans. Civ. Eng. 2018, 42, 17–32. [CrossRef]

25. De Boor, C. A Practical Guide to Splines, 3rd ed.; Applied Mathematical Sciences: Wisconsin, DC, USA, 1978; p. 1.

26. Amanatidis, D.; Dossis, M. High Level Synthesis of CART. In Proceedings of the 2019 4th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Piraeus, Greece, 20–22 September 2019; pp. 1–5.

27. Patel, H.H.; Prajapati, P. Study and analysis of decision tree based classification algorithms. Int. J. Comput. Sci. Eng. 2018, 10, 74–78. [CrossRef]

28. Dikmese, S.; Anttila, L.; Campo, P.P.; Valkama, M.; Renfors, M. Behavioral modeling of power amplifiers with modern machine learning techniques. In Proceedings of the 2019 IEEE MTT-S International Microwave Conference on Hardware and Systems for 5G and Beyond (IMC-5G), Atlanta, GA, USA, 15–16 August 2019; pp. 1–3.

29. Xu, G.; Yu, H.; Hua, C.; Liu, T. Chebyshev Polynomial-LSTM Model for 5G Millimeter-Wave Power Amplifier Linearization. IEEE Microw. Wirel. Components Lett. 2022, 32, 611. [CrossRef]

30. Zhang, Y.; Li, Y.; Liu, F.; Zhu, A. Vector decomposition based time-delay neural network behavioral model for digital predistortion of RF power amplifiers. IEEE Access 2019, 7, 91559–91568. [CrossRef]