Analysis of Deep Learning Models Towards High Performance Digital Predistortion for RF Power Amplifiers

Rajesh Kudupudi, Fariborz Lohrabi Pour, Dong Sam Ha, Sook Shin Ha, and Keyvan Ramezanpour
Multifunctional Integrated Circuits and System (MICS) Group
Bradley Department of Electrical and Computer Engineering
Virginia Tech, Blacksburg, Virginia, 24061, USA
E-mail: {rajeshk19, fariborzlp, ha, sook, rkeyvan8} @vt.edu

***Abstract*—This paper investigates direct and indirect learning methods to develop deep learning digital predistortion (DL-DPD) models and apply the models to improve the linearity of a power amplifier (PA). The two methods are applied to class-AB and class-F^{-1} PAs designed with gallium nitride (GaN) on silicon carbide (SiC) high electron mobility transistors (HEMTs). The simulation results show that both direct and indirect DL-DPD methods improve the linearity of the class-AB PA by about 12 dB and the class-F^{-1} PA by 11 dB, while the indirect method offers marginally better performance. The paper shows the direct learning method leads to significant improvement of the DL-DPD method based over the memory polynomial. It also presents the advantages of a BiLSTM based on the neural network architecture to design direct/indirect DPDs. Finally, it demonstrates that the DL-DPD can improve the linearity of class-AB and class-F^{-1} PAs without architectural changes.**

***Index Terms*—ACPR, BiLSTM, deep learning, digital predistortion (DPD), GaN, HEMT, linearity, power amplifier (PA).**

I. Introduction

The general trend of advanced communication systems is achieving higher data rates to meet the ever-increasing data demands by the users. This requires the use of signals with higher-order modulation schemes that leads to signals with high peak to average power ratio (PAPR), which necessitates the use of highly linear transmitter chain. The performance of power amplifier (PA) in the transmitter chain is the bottleneck of the linearity. However, the linearity of the PA, is limited due to the non-idealities of the active devices (i.e., transistors). This non-linear behavior of PA becomes a severe issue when it operates with high PAPR signals, leading to significant decrease in adjacent channel to power ratio (ACPR) and degrading the quality of communications. One effective solution to this problem is designing a digital predistorter (DPD), which modifies the signal accounting for the non-linearity of the PA [1], [2]. There has been a wide range of interest by researchers to develop DPDs in-order to improve the linearity of PA. This is important for a plethora of applications, including but not limited to RF frontends with high-efficiency requirements, such as ad-hoc and unmanned aerial vehicles (UAVs) networks, as well as high-power base stations [3]–[5].

Traditionally, Volterra series-based memory polynomial models have been used to design the DPD [1], [2], [6]–[8]. These models use Volterra kernels with different non-linear and memory orders. However, Braithwaite et al. pointed out several limitations of these models and showed their inability to compensate for extremely non-linear PAs [9]. With the success of deep learning in many fields such as image processing, time series analysis, speech, and text analytics, motivated researchers to design DL-DPD [10]–[20]. From a deep learning standpoint, the function of the DPD can be interpreted as a sequence to sequence conversion task, where given an input signal in the digital domain, the DPD predistorts the signal and outputs a sequence such that it improves the linearity of the PA. However, majority of the works in designing DL-DPD are based on multilayer perceptron (MLP) and convolutional neural networks (CNN), which are not suitable for these sequence to sequence conversion tasks because of their inability to model memory effects of the PA [14]–[18], [21], [22]. We adopt a bi-directional long short-term memory (BiLSTM) architecture for the DPD, which can readily model the memory effects. Further in this paper, we explore the effectiveness of our DPD on PAs with different characteristics i.e., class-AB and class-F^{-1} PA. We believe this is important to gauge the effectiveness of DPD models.

The traditional deep learning algorithms are based on supervised learning, which requires the input and output sequences for training. However, the pre-distorted signal for training the deep learning model is not available. Hence, two different methods, indirect and direct learning, are used to train the DL-DPD models [14], [16], [18], [20]. In the indirect learning approach, the inverse function of the PA is learned and the coefficients of this inverse model are copied to be used as a pre-distorter. On the other hand, direct learning is a multi-step process. First, a behavioral model of the PA is learned, and it is parameters frozen. Then, the DPD model is learned by back-propagating the gradients of error function through the behavioral model of the PA.

In this paper, we train the DL-DPD using both of these methods and compare their resulting performance in terms of improving the linearity of PA. Given the rising interest in DL-DPD, it is of paramount importance to know which way

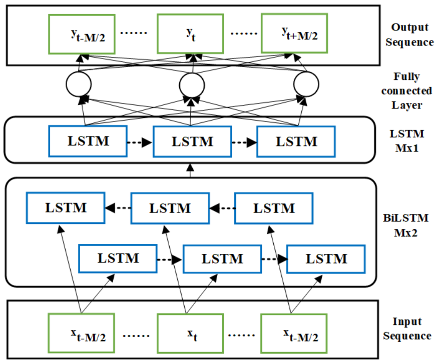

Fig. 1. BiLSTM model architecture.

of training them yields better performing DPD models. To our knowledge, similar comparisons have not been presented previously with DL-DPD. Further, we present a theoretical analysis highlighting the differences between these two training procedures.

The remainder of this paper is organized as follows. In Section II, we establish some preliminaries and discuss some related works. In Section III we present the theoretical differences between direct and in-direct learning training methodologies. In Section IV, we discuss our simulations. Finally, Section V concludes this work.

II. Preliminaries

A. Memory Polynomial based DPD models

The active devices contributing towards the non-linearity of the PA also exhibit memory effects. These effects should be accounted for in developing a DPD. Substantial research has been dedicated to developing DPDs for PAs [2], [6], [7], [23]. Dennis et al. have derived a generalized memory polynomial (GMP) using Volterra functions to design a DPD that achieved state-of-the-art performance in terms of improving the linearity of the PA [8]. This success of GMP can be attributed to its ability to model memory and self-heating effects of the PA via the leading and lagging envelope terms in (1). The GMP models were described by:

$$\begin{aligned}\hat{y}(n) &= \sum_{k=0}^{K_1}\sum_{l=0}^{L_1} a_{kl}x(n-l)|x(n-l)|^k \\ &+ \sum_{k=0}^{K_2}\sum_{l=0}^{L_2}\sum_{m=1}^{M_2}(b_{klm}x(n-l)|x(n-l-m)|^k \\ &+ \sum_{k=0}^{K_3}\sum_{l=0}^{L_3}\sum_{m=1}^{M_3} c_{klm}x(n-l)|x(n-l+m)|^k)\end{aligned} \tag{1}$$

However, the high correlation between polynomial bases makes it difficult to further improve the performance of these GMP models, thereby limiting their ability to model higher orders of non-linearity [9].

B. Deep Neural Network based DPD models

In the last decade, the exceptional ability of deep learning algorithms to represent complex non-linear functions and solve problems in computer vision, text analytics, etc. has made them popular [11]–[20], [22], [24]. This motivated researchers to apply deep learning algorithms to design DPDs. Most of these works are based on MLP CNN architectures, which require a lot of fine-tuning [14]–[18], [20]. However, as these neural network architectures cannot readily model the memory effects of the PA, a significant amount of time has to be spent on fine-tuning these architectures. On the other hand, BiLSTM architecture Fig. 1 has two LSTM networks, forward and backward networks which propagate hidden vectors between them and give the BiLSTM access to previous and future information. This gives BiLSTM access to leading and lagging input terms that are similar to the GMP. Hence, we adopt a BiLSTM for designing DPD. Although BiLSTM was used as a DPD in [19], we explore its performance on class-AB and class-F^{-1} and compare their results.

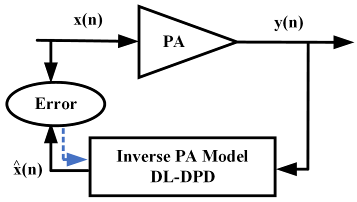

Fig. 2. Training DL-DPD Indirect Learning. Dotted lines show direction of the gradient.

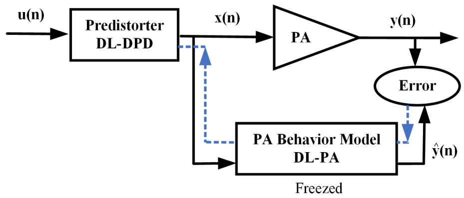

Fig. 3. Training DL-DPD Direct Learning. Dotted lines show the direction of the gradient.

C. Training Deep Neural Networks

As we mentioned earlier, the pre-distorted signal is not available to train the deep learning models. Hence, direct learning and indirect learning methodologies are used in training DL-DPD. In indirect learning, the inverse PA model is learned using the output of the PA as the input of the DL-DPD and the input of the PA as output Fig. 2. This inverse model is then used as a DPD for the PA. As the first step in the direct learning approach, a deep learning model is trained to mimic the behavior of the PA (DL-PA model). Then, an another deep learning model is trained to learn the DPD (DL-DPD model) using the PA-DL. This is done by cascading the DL-DPD and DL-PA, and backpropagating the error function through the DL-PA to train the DL-DPD Fig. 3. During training the DL-DPD, the weights of DL-PA are frozen so that only DL-DPD weights can change. This is necessary to keep DL-PA unchanged while learning the DL-DPD. A comparison between direct and indirect learning methodologies was made by [25] using memory polynomial/LUT-based DPD's using self-tuning controllers to optimize for the DPD coefficients. However, this comparison is not applicable for DL-DPDs since

DL algorithms use gradient descent for arriving at the optimal model coefficients.

III. Theoretical Analysis

In this section we discuss theoretical formulation of direct learning and contrast it with indirect learning methodology. In Fig. 3, $u(n)$ is the input to the DL-DPD model whose output will then be passed to DL-PA model as $x(n)$. The goal of direct learning approach is to learn the DL-DPD from the DL-PA. DL-PA model is a representation of PA behavior function $f : x \rightarrow y$ which accounts for the memory effects. For the purpose of this analysis, we use a linear function approximation for the deep learning model. We assume that we have learned the DL-PA model given by:

$$\hat{y}_{pa}(n) = X^T(n)w_{pa} + e(n) \tag{2}$$

$$X^T(n) = [x(n-J), x(n-J+1), ..., x(n))] \tag{3}$$

where $\hat{y}_{pa}(n)$ is the response of the DL-PA at time instant n and w_{pa} are the parameters of DL-PA model. We also represent the modeling error as the Gaussian random variable, $e(n) \in \mathcal{N}(\mu_e, \sigma_e^2)$ which is the difference between the real PA output and the output of DL-PA model. $X(n)$ is a vector containing J lagging samples of the input, which help in effectively modeling memory effects of the PA. Hence, the vector w_{pa} also has dimensions of Jx1. It is straight forward to learn w_{pa} by minimizing the mean squared error between $\hat{y}_{pa}(n)$ and $y(n)$ via gradient descent. After learning the PA model (DL-PA), we can learn the, DL-DPD model is learned by back-propagating the error (difference between the DL-PA output and desired reference signal) through the DL-PA model in (2). Since, the DL-PA expects all the J inputs lagging terms, the equation of the combined system is given by

$$\hat{y}(n) = (\overline{U}^T(n)w_{dpd})^T w_{pa} + e(n) \tag{4}$$

$$\overline{U}(n) = [U(n-J), U(n-J+1), ...U(n)] \tag{5}$$

where each column in the matrix $\overline{U}(n)$ is a column vector similar to $X(n)$, but including L1 lagging and L2 leading samples of the signal u(n). As a result, U(n) in the above equation is a (L1+L2+1)×J matrix. The vector w_{dpd} represents the DL-DPD model which is to be learned. The inner product of $\overline{U}^T(n)w_{dpd}$ would give us a vector with dimensions of $X(n)$, and is an estimate for the desired input to the PA (after pre-distortion). To solve for w_{dpd} the error function, to be minimized is :

$$C = \min_{w_{dpd}} \frac{1}{N} \sum_{n \in N} ((\overline{U}^T(n)w_{dpd})^T w_{pa} + e(n) - u(n))^2 \tag{6}$$

The predominent method used in DL based algorithms to obtain the optimal coefficients is gradient descent. Gradient descent updates the coefficients iteratively by moving in the negative direction of the gradient of C. The gradient of C w.r.t w_{dpd} as

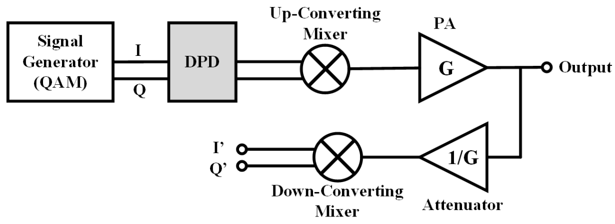

Fig. 4. Test bench used to generate data and test the DPD.

$$\begin{aligned}
&\nabla_{w_{dpd}} C \\
&= \frac{2}{N} \sum_{n \in N} ((\overline{U}^T(n)w_{dpd})^T w_{pa} + e(n) - u(n)) \\
&(\nabla_{w_{dpd}}(w_{dpd}^T \overline{U} w_{pa})) \\
&= \frac{2}{N} \sum_{n \in N} ((\overline{U}^T(n)w_{dpd})^T w_{pa} + e(n) - u(n))(\overline{U} w_{pa})
\end{aligned} \tag{7}$$

Here the gradient $\nabla_{w_{dpd}} C(n)$ is off from the true gradient by $e(n)\overline{U}w_{pa}$, whose variance is

$$= \sigma_e^2 \sum_{k=L_1}^{-L_2} \sum_{j=0}^{J} u(n-k+j) w_{pa}(j) \tag{8}$$

In direct learning, we have to back-propagate through a DL-PA model (w_{pa} in the above formulation), which introduces a source of error with a variance given in (8). This error cannot be removed as it arises from the modeling of the DL-PA model. Our simulations results showed a mean square error (MSE) $4{\times}10^{-4}$, in modeling the PA after the completion of training (DL-PA model error). On the other hand, the indirect learning method does not require backpropagation through a PA model, thereby making this source of error non-existent.

IV. Simulation Details & Results

Fig. 4 shows the simulation setup, which consists of a transistor-level power amplifier. Two classes of PAs are designed in Keysight Advanced Design Systems (ADS) using Wolfspeed 6W CGH40006P galium nitride (GaN) on silicon carbide (SiC) transistor model. A quadrature amplitude modulator was used to generate baseband signals of order 64 (I, Q). The Up-mixer is used to convert this signal to the carrier frequency of 1 GHz, and the resulting signal is delivered to the PA. In order to sample the output of the PA, it is down-converted through the DC-mixer and demodulated in the baseband domain ($I^{'}$, $Q^{'}$). It should be noted that the attenuator is used to adjust the output signal of the PA before delivering it to the mixer.

A. Dataset collection and DL model details

The baseband signals (I, Q) and ($I^{'}$, $Q^{'}$) were sampled from the test bench. These sampled signals were split into training and test sets, which were used for training and evaluating the DPDs, respectively. As a baseline, a memory polynomial model from MATLAB's communication toolbox has been used. The memory and polynomial orders of this model are set to 5.

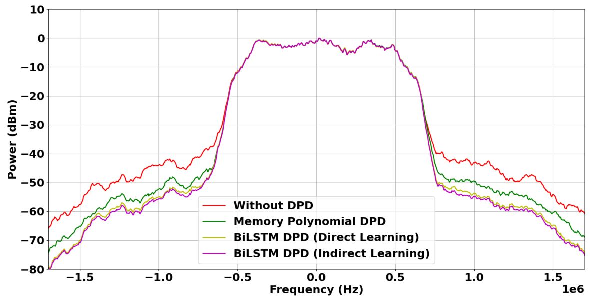

Fig. 5. Power Spectral Density of class-AB PA, driven by pre-distorted signals from different models

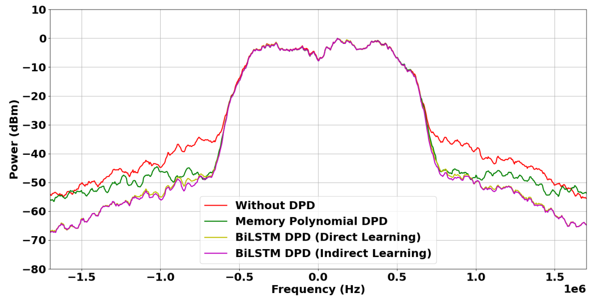

Fig. 6. Power Spectral Density of class-F^{-1} PA, driven by pre-distorted signals from different models

We adopted a BiLSTM for our DL-DPD because at any time instant BiLSTM network has access to past and future data like the GMP. This access to past future information enables DL-DPD to model the memory, self-heating effects of the PA. The DL-DPD model architecture which is used in this work consists of 2 LSTM layers, with a hidden vector of size 256. The results of the 2 LSTM layers are fed into a single LSTM layer which is in turn connected to a fully connected layer. This fully connected layer outputs the pre-distorted signals Fig. 1. Normalized mean squared error function (NMSE) was chosen as the loss function. Adam optimizer [26] is used for updating the DL model's parameters with a learning rate of 10^{-4}. The dependency of BiLSTM on future samples necessitates the use of a time delay block that is used to gather some future I/Q baseband inputs. This would result in increased but acceptable latency. The DL models were created in Python using the Pytorch framework.

B. *Performance evaluations*

To compare the performance of the two approaches, a BiLSTM DL model is trained using direct and indirect learning methodologies. The effectiveness of DPD models is presented in the Power Spectral Densities (PSD) plots [Fig. 5, 6]. The ACPR from these PSD plots is listed in Table 1. It can be noticed that the class-AB PA offers better linearity performance compared to class-F^{-1}. It is found that the BiLSTM DPD trained using direct/indirect learning achieves almost the same performance, with indirect learning being marginally better in terms of ACPR. We present that this small performance variance in direct learning is because of backpropagating gradients through the PA behavioral model, which introduces a source of error. This is in line with our theoretical analysis of direct learning, which gives the variance of these gradients as a function of this error in equation (8). This is true for both class-AB and class-F^{-1} PA. Further, training the DL-DPD using direct learning needs more resources as two DL models need to be trained, increasing the time and computational resources required. These results imply that indirect learning is computationally efficient and yields marginally better DPDs than direct learning.

The same results hold for both class-AB and class-F^{-1} PAs. This is an interesting result since it was achieved without tuning the BiLSTM model architecture parameters. This is advantageous because this shows that the DL-DPD model's architecture can be used for any PA, without tuning the architecture. This drastically reduces the design and exploration time while designing a DPD. It can be seen from Table I, that BiLSTM always performs better than memory polynomial DPDs, irrespective of the training method used. Moreover, to explore the generalization of DL-DPD instead of QAM, QPSK modulated signals are fed into the DL-DPD without retraining. The simulation results for QPSK signals showed a 10 dB improvement in ACPR. This indicates that the DL-DPD achieves good generalization as it is not trained with QPSK signals.

TABLE I
COMPARISON OF ACPR FOR DIFFERENT DPD MODELS ON CLASS-AB AND CLASS-F PA FOR QAM SIGNALS.

DPD Model Details	Class-AB Lower/Upper ACPR(dB)	Class-F^{-1} Lower/Upper ACPR(dB)
Without DPD	45.19/44.53	36.40/30.4
Memory polynomial DPD	55.02/51.08	37.11/40.73
BiLSTM DPD (DL)	56.77/55.67	46.13/41.95
BiLSTM DPD (IL)	57.65/56.41	46.85/41.75

V. CONCLUSION

In this paper, we compared direct and indirect learning methodologies for training the deep neural network-based DPDs. It is found that DL-DPD trained with indirect learning method is better. We discover that back propagating through a PA behavioral model in direct learning adds another source of error. This is in agreement with our theoretical analysis. This analysis can be used as a baseline to extend the comparison between DL-DPDs with different neural network architectures. Further, we demonstrated that DL-DPD, irrespective of the training methodology used, outperform memory polynomial-based DPDs. Finally, we highlighted the advantages of BiLSTM models for designing DPDs and show how without architecture tuning, an improvement in ACPR can be achieved for different PAs.

ACKNOWLEDGEMENT

This research was supported in part by the National Science Foundation Award no. 2106987.

References

[1] J. Kim and K. Konstantinou, "Digital predistortion of wideband signals based on power amplifier model with memory," *Electronics Letters*, vol. 37, no. 23, pp. 1417–1418, 2001.

[2] L. Ding, G. Zhou, D. Morgan, Z. Ma, J. Kenney, J. Kim, and C. Giardina, "A robust digital baseband predistorter constructed using memory polynomials," *IEEE Transactions on Communications*, vol. 52, no. 1, pp. 159–165, 2004.

[3] J. Sabzehali, V. K. Shah, Q. Fan, B. Choudhury, L. Liu, and J. H. Reed, "Optimizing number, placement, and backhaul connectivity of multi-UAV networks," *arXiv preprint arXiv:2111.05457*, 2021.

[4] J. Sabzehali, V. K. Shah, H. S. Dhillon, and J. H. Reed, "3D placement and orientation of mmWave-based UAVs for guaranteed LoS coverage," *IEEE Wireless Communications Letters*, vol. 10, no. 8, pp. 1662–1666, 2021.

[5] Z. Zhou, K. Bai, N. Mohammadi, Y. Yi, and L. Liu, "Making intelligent reflecting surfaces more intelligent: A roadmap through reservoir computing," *arXiv preprint arXiv:2102.03688*, 2021.

[6] C. Eun and E. Powers, "A new Volterra predistorter based on the indirect learning architecture," *IEEE Transactions on Signal Processing*, vol. 45, no. 1, pp. 223–227, 1997.

[7] S. Boumaiza, F. Mkadem, and M. Ben Ayed, "Digital predistortion challenges in the context of software defined transmitters," in *2011 XXXth URSI General Assembly and Scientific Symposium*, 2011, pp. 1–4.

[8] D. Morgan, Z. Ma, J. Kim, M. Zierdt, and J. Pastalan, "A generalized memory polynomial model for digital predistortion of rf power amplifiers," *IEEE Transactions on Signal Processing*, vol. 54, no. 10, pp. 3852–3860, 2006.

[9] R. N. Braithwaite, "Digital predistortion of an RF power amplifier using a reduced Volterra series model with a memory polynomial estimator," *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 10, pp. 3613–3623, 2017.

[10] N. Mohammadi, L. Liu, and Y. Yi, "Policy-based fully spiking reservoir computing for Multi-Agent distributed dynamic spectrum access," in *2022 IEEE International Conference on Communications (ICC): Green Communication Systems and Networks Symposium (IEEE ICC'22 - GCSN Symposium)*, Seoul, Korea (South), May 2022.

[11] R. Hongyo, Y. Egashira, T. M. Hone, and K. Yamaguchi, "Deep neural network-based digital predistorter for doherty power amplifiers," *IEEE Microwave and Wireless Components Letters*, vol. 29, no. 2, pp. 146–148, 2019.

[12] A. Ahmed, E. Srinidhi, and G. Kompa, "Neural network and memory polynomial methodologies for PA modeling," in *TELSIKS 2005 - 2005 uth International Conference on Telecommunication in ModernSatellite, Cable and Broadcasting Services*, vol. 2, 2005, pp. 393–396 vol. 2.

[13] F. Mkadem and S. Boumaiza, "Physically inspired neural network model for RF power amplifier behavioral modeling and digital predistortion," *IEEE Transactions on Microwave Theory and Techniques*, vol. 59, no. 4, pp. 913–923, 2011.

[14] Y. Wu, U. Gustavsson, A. G. i. Amat, and H. Wymeersch, "Residual neural networks for digital predistortion," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 01–06.

[15] S. Boumaiza and F. Mkadem, "Wideband rf power amplifier predistortion using real-valued time-delay neural networks," in *2009 European Microwave Conference (EuMC)*, 2009, pp. 1449–1452.

[16] X. Hu, Z. Liu, X. Yu, Y. Zhao, W. Chen, B. Hu, X. Du, X. Li, M. Helaoui, W. Wang, and F. M. Ghannouchi, "Convolutional neural network for behavioral modeling and predistortion of wideband power amplifiers," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.

[17] P. Jaraut, A. Abdelhafiz, H. Chenini, X. Hu, M. Helaoui, M. Rawat, W. Chen, N. Boulejfen, and F. M. Ghannouchi, "Augmented convolutional neural network for behavioral modeling and digital predistortion of concurrent multiband power amplifiers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 69, no. 9, pp. 4142–4156, 2021.

[18] C. Tarver, L. Jiang, A. Sefidi, and J. R. Cavallaro, "Neural network DPD via backpropagation through a neural network model of the PA," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, 2019, pp. 358–362.

[19] J. Sun, W. Shi, Z. Yang, J. Yang, and G. Gui, "Behavioral modeling and linearization of wideband RF power amplifiers using BiLSTM networks for 5G wireless systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 10 348–10 356, 2019.

[20] Y. Zhang, Y. Li, F. Liu, and A. Zhu, "Vector decomposition based time-delay neural network behavioral model for digital predistortion of RF power amplifiers," *IEEE Access*, vol. 7, pp. 91 559–91 568, 2019.

[21] G. Singh and M. Sachan, "Multi-layer perceptron (mlp) neural network technique for offline handwritten gurmukhi character recognition," in *2014 IEEE International Conference on Computational Intelligence and Computing Research*, 2014, pp. 1–5.

[22] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.

[23] J. Kim and K. Konstantinou, "Digital predistortion of wideband signals based on power amplifier model with memory," *Electronics Letters*, vol. 37, no. 23, pp. 1–2, Nov 08 2001, copyright - Copyright The Institution of Engineering Technology Nov 8, 2001; Document feature - Equations; Graphs; ; Last updated - 2014-11-10; CODEN - ELLEAK.

[24] N. Mohammadi, J. Bai, Q. Fan, Y. Song, Y. Yi, and L. Liu, "Differential privacy meets federated learning under communication constraints," *IEEE Internet of Things Journal*, 2021.

[25] H. Paaso and A. Mammela, "Comparison of direct learning and indirect learning predistortion architectures," in *2008 IEEE International Symposium on Wireless Communication Systems*, 2008, pp. 309–313.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980