Electronic Design

HAMED M. SANOGO, Principal Engineer and End Market Specialist,
Analog Devices

A Neural-Network-Based Approach to Smarter DPD Engines

An AI-driven digital-predistortion (DPD) framework can help overcome the challenges of signal distortion and energy inefficiency in power amplifiers for next-generation wireless communication.

Launched by OpenAI in November 2022, ChatGPT became one of the fastest-adopted software products, showcasing the potential of artificial intelligence (AI). Machine learning (ML), a subset of AI, is transforming industries by enabling tasks such as decision-making and data analysis. In communications, AI and ML are advancing digital predistortion (DPD), a technique critical for reducing signal distortion and improving power-amplifier (PA) efficiency.

Traditional DPD models may struggle with nonlinearities and memory effects in modern communication systems like 5G. They assume that the PA's behavior is static and memoryless, relying on polynomial models that only account for instantaneous input-output relationships.

AI and ML, however, excel at learning complex patterns, offering more precise solutions. This article introduces an artificial neural-network-based DPD framework that leverages PA data to reduce gain/phase errors, enhance efficiency, and improve spectral performance, surpassing traditional methods.

Enhancing PA Efficiency: Digital Predistortion Meets AI Innovation

Digital predistortion is a critical technique enabling power amplifiers to operate efficiently near the saturation region without compromising linearity. By extending the PA's linear operating range, DPD allows radio-frequency (RF) designers to leverage the efficiency of a nonlinear PA while maintaining the transmit signal linearity required for complex modulation schemes like orthogonal frequency-division multiplexing (OFDM).

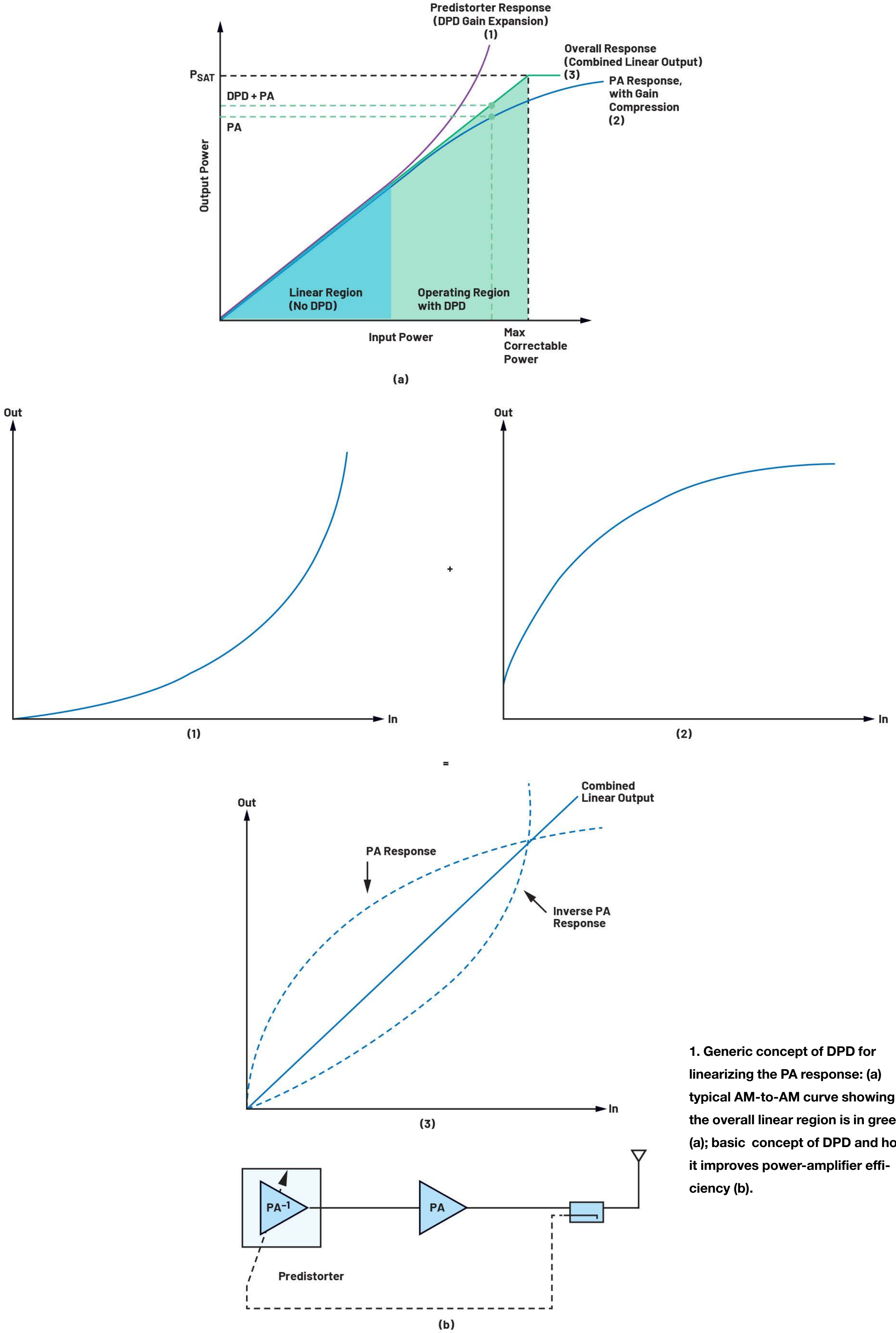
At its core, DPD works by introducing predistorter coefficients derived from modeling the inverse amplitude-to-amplitude (AM-to-AM) and amplitude-to-phase (AM-to-PM) characteristics of the PA. This process effectively compensates for the PA's nonlinearities by introducing precise anti-distortion into the input waveform. Consequently, DPD improves signal quality while enabling the PA to operate at peak efficiency.

A detailed discussion about DPD algorithms, as well as how ADI's ADRV9040 RF transceiver provides a streamlined hardware platform for designing and implementing it, is presented in the article "Simplifying Your 5G Base Transceiver Station Transmitter Line-Up, Design, and Evaluation." *Figure 1* illustrates the DPD concept for linearizing a PA response.

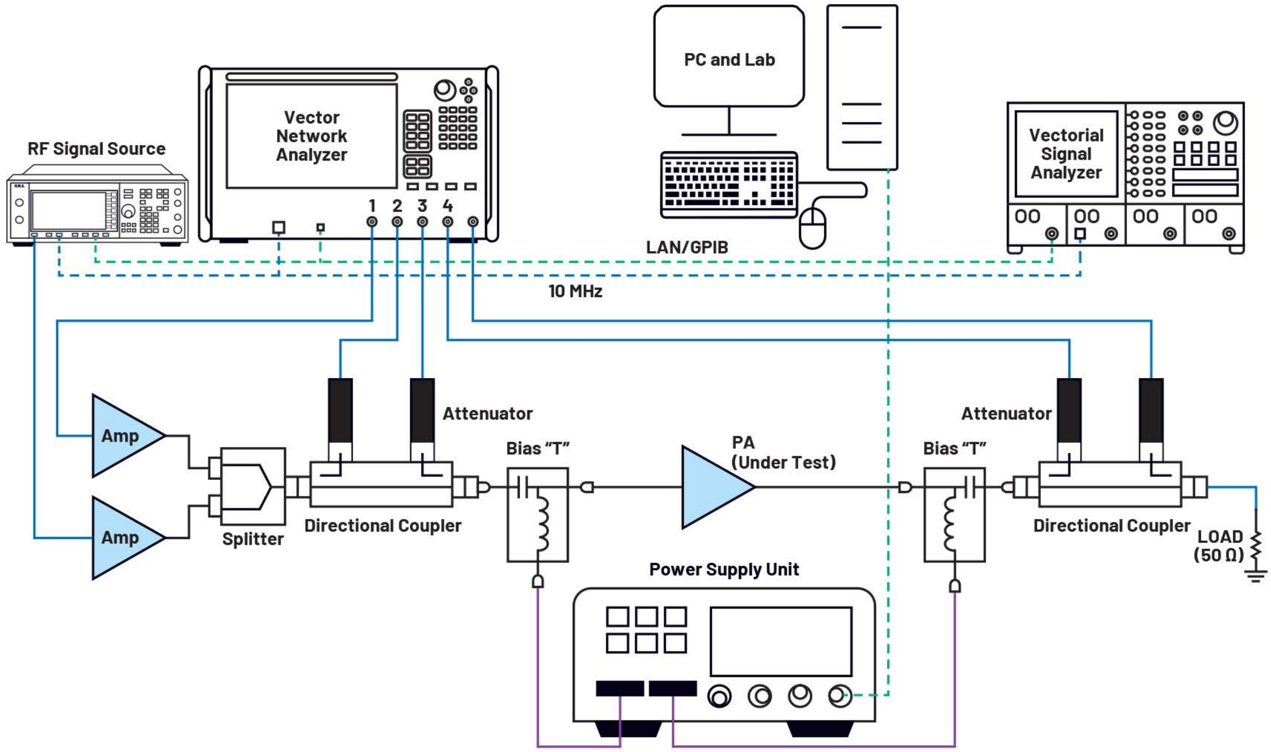
Power amplifiers exhibit nonlinearities near saturation, causing signal distortion, spectral regrowth, and reduced efficiency, especially in high-bandwidth systems with impairments like I/Q imbalance and memory effects. AI and ML, particularly neural networks (NNs), offer a transformative solution by modeling PA distortions and dynamically optimizing predistortion. This AI-driven approach improves efficiency and adaptability, surpassing traditional methods while balancing performance and computational complexity.

Optimizing DPD Engines with Neural Network Models: A Proposed Game-Changer Framework

Artificial NNs are a cornerstone of artificial intelligence, particularly in deep learning, designed to overcome the limitations of traditional ML algorithms. Inspired by the human brain's ability to process information, NNs excel at identifying patterns, learning, and making decisions, making them ideal for addressing complex, nonlinear problems.



1. Generic concept of DPD for linearizing the PA response: (a) typical AM-to-AM curve showing the overall linear region is in green (a); basic concept of DPD and how it improves power-amplifier efficiency (b).



2. Measurement setup for wideband PA characterization.

For example, in 5G LTE systems, challenges such as I/Q imbalance, phase shifts, DC offsets, crosstalk, and PA nonlinearity can be addressed effectively using an NN-based DPD approach.

Unlike polynomial-based DPD solutions, which require extensive knowledge of system mechanics and struggle with scalability, NN models excel at handling complex nonlinear behaviors with fewer constraints. This section introduces a proposed NN DPD framework to mitigate nonlinearities and transmitter impairments.

The process involves three key steps: characterizing the PA and collecting extensive data, training a postdistorter neural-network model, and deploying the model with performance monitoring and adjustments. By leveraging machine learning, this approach transforms large datasets into actionable insights, enabling robust, scalable solutions for modern communication challenges.

Step 1: PA Characterization Data Collection

To design and implement AI/ML models for optimizing wireless power amplifiers, it's crucial to collect comprehensive, high-quality characterization data that accurately reflects the PA's real-world performance under diverse conditions. *Figure 2* shows an example setup that one might use for a PA characterization data-collection effort.

The bench configuration presented in *Figure 2* allows for complete characterization by extracting parameters like S-parameters, delivered power, power-added efficiency (PAE), input impedance, input return loss, power gain, AM-to-PM conversion, and others. *Table 1* presents a comprehensive list of data points for input into the model. However, it's important to note that the model's dimensionality impacts its response time. In addition, the captured data must be digitized before it can be utilized in the training process.

This rigorous and systematic approach to data collection forms the foundation for developing AI/ML models capable of accurately predicting and optimizing PA performance. By leveraging this comprehensive dataset, designers can achieve reliable and efficient wireless communication systems.

Step 2: Model Training

The model training process involves feeding (a few or all) the signals collected in *Table 1* into this system and optimizing the DPD model to minimize error via a loss function. The neural-network architecture consists of interconnected layers of nodes (e.g., artificial neurons), organized into the main core components *(Figure 3 and Table 2)*.

During training, hidden layers propagate data forward while weights and biases are optimized via backpropagation using gradient descent. The network structure can be adjusted to include more neurons for highly nonlinear components or fewer neurons for smoother elements.

While an in-depth discussion of the best AI hardware, software, and tools for creating an effective and scalable AI model training environment is beyond the scope of this article, we recommend that AI engineers explore KNIME, a no-code platform for data analytics and machine learning. KNIME features a graphical user interface (GUI) that enables users to design workflows by simply dragging and dropping nodes, eliminating the need for extensive coding knowledge. These workflows are highly visual and easy to understand, making the platform accessible to a broad audience.

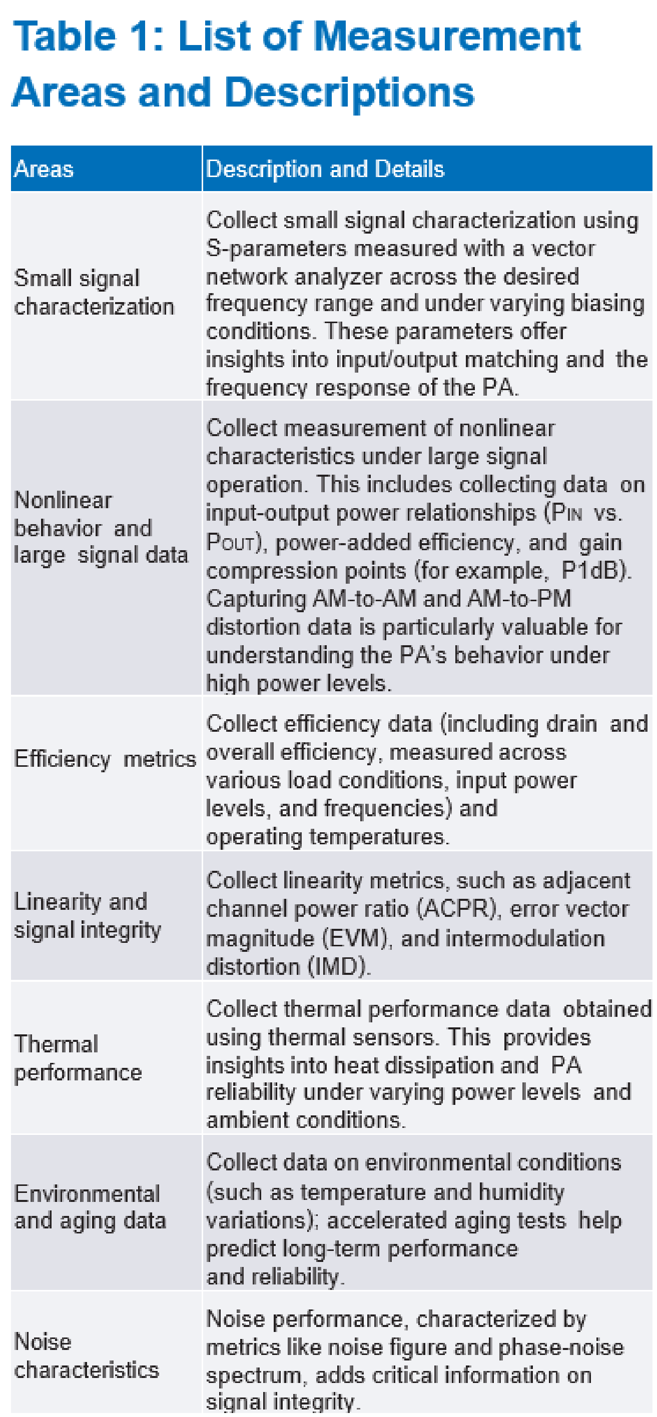


Table 1: List of Measurement Areas and Descriptions

Areas	Description and Details
Small signal characterization	Collect small signal characterization using S-parameters measured with a vector network analyzer across the desired frequency range and under varying biasing conditions. These parameters offer insights into input/output matching and the frequency response of the PA.
Nonlinear behavior and large signal data	Collect measurement of nonlinear characteristics under large signal operation. This includes collecting data on input-output power relationships (PIN vs. POUT), power-added efficiency, and gain compression points (for example, P1dB). Capturing AM-to-AM and AM-to-PM distortion data is particularly valuable for understanding the PA's behavior under high power levels.
Efficiency metrics	Collect efficiency data (including drain and overall efficiency, measured across various load conditions, input power levels, and frequencies) and operating temperatures.
Linearity and signal integrity	Collect linearity metrics, such as adjacent channel power ratio (ACPR), error vector magnitude (EVM), and intermodulation distortion (IMD).
Thermal performance	Collect thermal performance data obtained using thermal sensors. This provides insights into heat dissipation and PA reliability under varying power levels and ambient conditions.
Environmental and aging data	Collect data on environmental conditions (such as temperature and humidity variations); accelerated aging tests help predict long-term performance and reliability.
Noise characteristics	Noise performance, characterized by metrics like noise figure and phase-noise spectrum, adds critical information on signal integrity.

For those who prefer a Python-based approach, Keras with TensorFlow offers significant advantages. This combination merges the simplicity of Keras with the robustness and scalability of TensorFlow, making it an excellent choice for projects ranging from experimentation to production-grade deep-learning applications.

In the PA characterization effort, millions of samples will be collected—70% will be used for training and 30% reserved for testing and validation to assess the model's ability to mimic the PA's behavior. Model performance will be evaluated using metrics such as accuracy, precision, recall, F1 score, and ROC-AUC.

Step 3: Neural Network Model Validation and Deployment

The deployment process begins with validating the model to ensure robustness and accuracy, using validation data to monitor quality during training and stopping criteria, while test data independently evaluates accuracy and generalization. Addressing overfitting and underfitting is crucial to ensure the model generalizes well to new data.

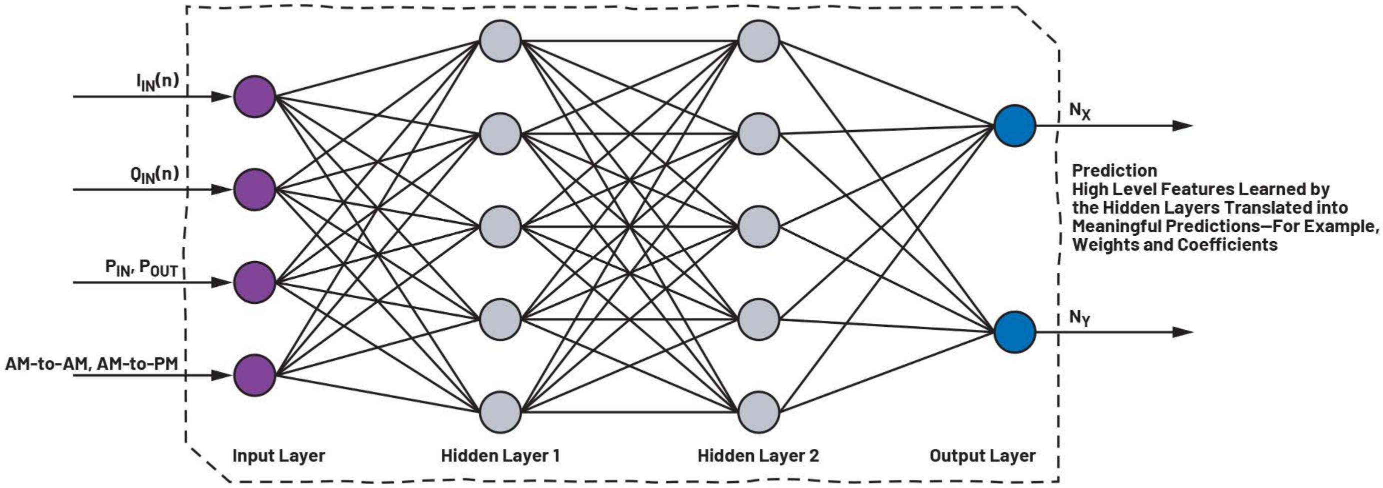
Overfitting is mitigated by limiting the number of layers, hidden neurons, or parameters to simplify the model, by expanding the training dataset, or even by pruning (for example, removing redundant neurons that don't contribute significantly to performance) to enhance generalization. On the other hand, mitigating underfitting is addressed by increasing hidden neurons to boost model complexity, and/or by adjusting hyperparameters such as learning rate, batch size, or regularization strength to improve performance.

The ML engineer must balance these strategies and iteratively evaluate the DPD model's performance to achieve a robust and generalizable model while keeping an eye on the execution speed of the model. *Figure 4* illustrates the high-level block diagram of the neural-network DPD model evaluation system's architecture.

In any event, determining the optimal number of hidden neurons requires empirical studies, trial and error, or adaptive methods during training. These adjustments ensure the NN achieves an appropriate balance between complexity and performance, enabling efficient and effective model deployment. Deployment of the model could be facilitated by an edge-AI embedded MCU such as ADI's MAX78000 convolutional-neural-network (CNN) accelerator chip.

Integrating AI/ML with DPD Systems: Challenges and Opportunities

Integrating artificial intelligence and machine learning into DPD systems offers significant potential for improvement, but it also introduces practical chal-



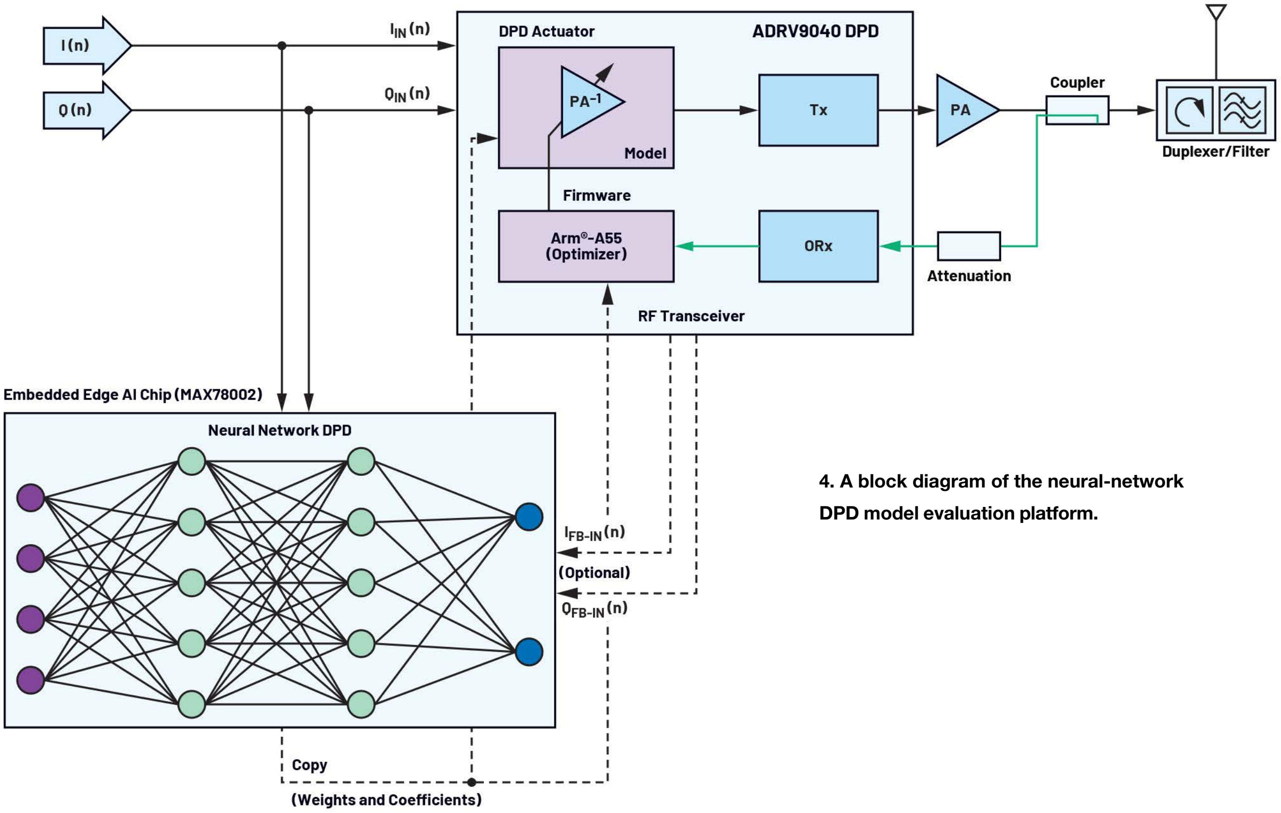
3. Architecture of a multiclass prediction neural network in deep learning.

lenges. DPD systems require low latency and high processing speeds, which can be difficult to achieve with computationally intensive ML models. Moreover, dynamic operating conditions, such as temperature fluctuations and hardware aging, necessitate adaptive techniques like real-time learning or transfer learning to maintain optimal performance.

Energy efficiency is another critical factor, as AI/ML models, particularly deep-learning architectures, often consume more power than traditional DPD methods, making them less suitable for energy-sensitive environments. Future experimentations should be conducted with lightweight neural networks, which are optimized versions of standard neural networks.

These lightweight NNs are designed to have fewer parameters, require less computation, and are memory efficient. They're particularly useful for applications where computational resources are limited, such as mobile and Internet of Things (IoT) devices or other resource-limited systems.

The lack of interpretability in many ML models, especially deep neural networks, further complicates their integration with DPD systems. Debugging and optimization are challenging when decision-making processes are opaque, as these models reduce complex operations to weights, biases, and activation functions.



4. A block diagram of the neural-network DPD model evaluation platform.

Conclusion: DPD Evolution via AI/ML

As 5G technologies like massive MIMO demand lower power and greater precision, DPD systems must evolve to address new complexities. AI/ML will be instrumental in enabling scalable, energy-efficient solutions through innovations such as adaptive learning and hybrid modeling. Neural networks, with their ability to model complex nonlinearities and memory effects, simplify DPD system design by approximating nonlinear functions without explicit mathematical formulations.

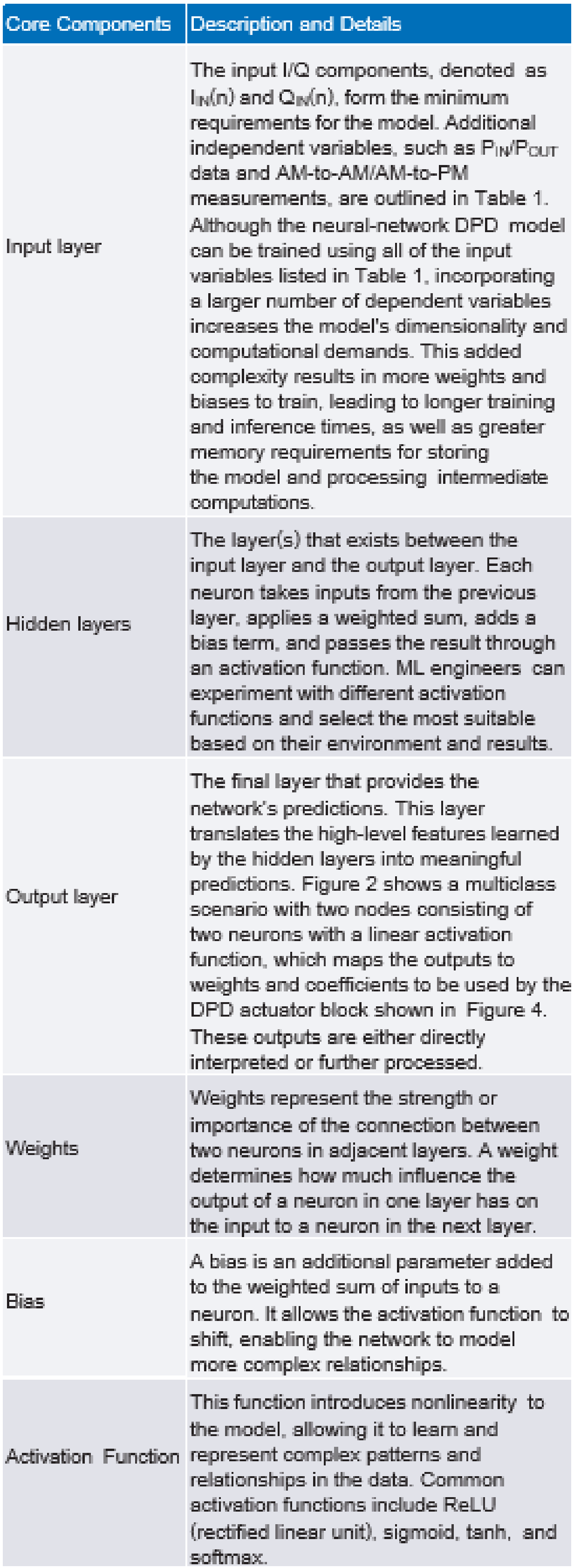
The integration of AI/ML enhances power efficiency, allowing PAs to operate closer to saturation while reducing costs with nonlinear PAs. Despite the challenges, AI/ML-driven systems hold great promise for enhancing the accuracy, adaptability, and scalability of DPD systems.

Hybrid approaches that combine traditional polynomial-based methods with AI/ML techniques offer a balanced solution, merging the interpretability of classical models with the advanced capabilities of AI/ML. By addressing these challenges through innovative strategies, AI/ML can drive transformative advances in DPD systems, supporting the evolution of modern communication technologies.

Hamed M. Sanogo is a seasoned principal engineer in the Customer Solutions Group (CSG) at Analog Devices Inc. (ADI), specializing in aerospace, defense, communications, and data centers. With nearly two decades at ADI, he has held diverse roles, including FAE/FAE manager, product line manager for Secure RFID & Authenticators, and his current position as an end market specialist driving cutting-edge solutions in critical industries.

Hamed's expertise is built on a strong foundation—earning his MSEE from the University of Michigan-Dearborn and an MBA from the University of Dallas. Before joining ADI, he was a senior design engineer at General Motors and a senior staff electrical engineer at Motorola Solutions, where he led the design of Node-B and RRU baseband cards.

Table 2: Neural-Network Core Components and Descriptions



Core Components	Description and Details
Input layer	The input I/Q components, denoted as $I_{IN}(n)$ and $Q_{IN}(n)$, form the minimum requirements for the model. Additional independent variables, such as P_{IN}/P_{OUT} data and AM-to-AM/AM-to-PM measurements, are outlined in Table 1. Although the neural-network DPD model can be trained using all of the input variables listed in Table 1, incorporating a larger number of dependent variables increases the model's dimensionality and computational demands. This added complexity results in more weights and biases to train, leading to longer training and inference times, as well as greater memory requirements for storing the model and processing intermediate computations.
Hidden layers	The layer(s) that exists between the input layer and the output layer. Each neuron takes inputs from the previous layer, applies a weighted sum, adds a bias term, and passes the result through an activation function. ML engineers can experiment with different activation functions and select the most suitable based on their environment and results.
Output layer	The final layer that provides the network's predictions. This layer translates the high-level features learned by the hidden layers into meaningful predictions. Figure 2 shows a multiclass scenario with two nodes consisting of two neurons with a linear activation function, which maps the outputs to weights and coefficients to be used by the DPD actuator block shown in Figure 4. These outputs are either directly interpreted or further processed.
Weights	Weights represent the strength or importance of the connection between two neurons in adjacent layers. A weight determines how much influence the output of a neuron in one layer has on the input to a neuron in the next layer.
Bias	A bias is an additional parameter added to the weighted sum of inputs to a neuron. It allows the activation function to shift, enabling the network to model more complex relationships.
Activation Function	This function introduces nonlinearity to the model, allowing it to learn and represent complex patterns and relationships in the data. Common activation functions include ReLU (rectified linear unit), sigmoid, tanh, and softmax.