PESTnet - Pre-IFFT PAPR Estimation using Neural Networks for Improved OFDM Systems

Pavan Kumar Mangipudi† *Maneesh Merugu*† *Janise McNair*† *John Terry*§ *David Veney*‡
† Department of Electrical and Computer Engineering, University of Florida, Gainesville, USA
§Terry Consultants Inc ‡Renaissance Associates, Inc.
{pavan.mangipudi, maneesh.merugu}@ufl.edu, mcnair@ece.ufl.edu,
drjdterry@gmail.com, david.veney@renaissance-assoc.com

***Abstract*—For the next generation of wireless technologies, Orthogonal Frequency Division Multiplexing (OFDM) remains a key signaling technique. Peak-to-Average Power Ratio (PAPR) reduction must be included with OFDM to reduce the detrimental high PAPR exhibited by OFDM. The cost of PAPR reduction techniques stems from adding multiple IFFT iterations, which are computationally expensive and increase latency. We propose a novel PAPR Estimation Technique called PESTNet which reduces the necessary IFFT operations for PAPR reduction techniques by using deep learning to estimate the PAPR before the IFFT is applied. This paper gives a brief background on PAPR in OFDM systems and describes the PESTNet algorithm and the training methodologies. A case study of the estimation model is provided where results demonstrate PESTNet is able to give an accurate estimate of PAPR and can compute large batches of resource grids up to 10 times faster than IFFT based techniques.**

***Index Terms*—OFDM, 5G, LTE, PAPR, CNN**

I. Introduction

Orthogonal Frequency Division Multiplexing (OFDM) has been widely used in current wireless and cellular technologies, including 4G LTE(Long Term Evolution), Optical Fibres, IEEE 802.11 standards, and 5G NR, due to its increased spectral efficiency, robustness in the presence of multipath propagation, and reduced transceiver design. The challenging drawback of OFDM has been the high Peak-to-Average Power Ratio (PAPR). OFDM signals are generated using the Inverse Fast Fourier Transform (IFFT), which is a linear combination of Quadrature Amplitude Modulation (QAM) symbols. The combination of these signals can lead to a high PAPR [12]. Thus, OFDM must be used in combination with PAPR reduction techniques to reduce the detrimental effects of high PAPR. PAPR Reduction techniques include Clipping and Filtering, Partial Transmit Sequences, Active Constellation extensions, and related techniques. [21] provides a survey of these traditional methods, covering each and more in detail. Recently, PAPR reduction techniques use Deep Learning (DL)-based implementations of conventional PAPR reduction approaches [28], [14], [15], [25], while other unconventional algorithms rely on converting OFDM symbols using IFFTs to produce a result [16], [17], [24]. A limitation is that they are reliant on the nature of the training data, and they combine neural network computations with IFFT computations, which by itself is a very computationally expensive operation, and thereby increase the transceiver system's computational complexity significantly. To overcome these limitations, we propose to use DL to perform PAPR estimation of frequency domain symbols before IFFT is applied. Our technique, called PESTNet , does not depend on a given PAPR reduction strategy. PESTNet determines the PAPR estimate in the frequency domain. An accurate estimation would eliminate the need for IFFTs for PAPR measurement, and the subsequent IFFT-based PAPR calculations, and would reduce the computation load, complexity, and operational latency. Furthermore, convolutional neural networks (CNNs) are well-suited to predict correlated data, so PESTNet is capable of predicting PAPR with substantial accuracy for large batches of data with significantly less latency than the state-of-the-art. The estimation process is easily scalable for MIMO and Carrier Aggregation since it is independent of the traffic density, training circumstances, processing paths, and the number of antennas. Pre-trained models can easily be deployed, and a trained modified model can be deployed directly in base stations or transmitters without needing to process individual antenna chains.

This paper is organized as follows: Section II provides a brief background, discussing the challenges of PAPR in OFDM systems as well as the advantages of CNNs. Section III covers the related work. To our knowledge, there are only a few works that focus on PAPR estimation, and none that use DL to estimate PAPR. So, we also present related work using DL to reduce PAPR. Section IV describes the proposed model and training methodologies, while Section V gives an overview of the results of the estimation model. Section VI concludes the paper.

II. Background

A. *Orthogonal Frequency Division Multiplexing (OFDM)*

In OFDM, symbols are modulated onto orthogonal sub-carriers with narrow bands of frequency which overlap with each other's nulls. The Inverse Discrete Fourier transform (IDFT) is used to generate an OFDM waveform; the output of an IDFT is a sum of orthogonal time domain signals.

This material is based upon work supported by the National Science Foundation under Grant Number 1738420.

IEEE WCNC April 2023 (ISBN and DOI TBD)/ ©2023 IEEE

Mathematically, let $X = \{X_k, k = 1, 2, ...N-1\}$ be the input data stream modulated onto a complex constellation map using any data modulation scheme (BPSK, QPSK, m-QAM, etc.). Applying the Inverse Fast Fourier Transform algorithm (IFFT) gives a discrete time domain signal:

$$x(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k e^{j2\pi k \frac{n}{N}}, \quad n = 0, 1, .., N-1, \qquad (1)$$

where X_k are the data symbols modulated onto orthogonal subcarriers represented by the complex exponential.

B. Peak-to-Average Power Ratio (PAPR)

PAPR is calculated [4]:

$$\text{PAPR}(x[n]) = \frac{\max\{|x[n]|^2\}}{E[|x[n]|^2]}, \qquad (2)$$

where $E[.]$ is the expectation operator and $x[n]$ is the signal generated from Eqn. (1).

High PAPR arises when subcarriers that are identical at a certain time instant add up and result in a higher peak than the average power of the signal [12]. High PAPR in communication systems is undesirable as it drives the power amplifier of the transmitter into the nonlinear region of operation, causing the frequency components of the OFDM signal to spread out and interfere with other subcarriers, resulting in out of band distortion [12]. Non-linear amplification also causes the output to be a distorted non-linear amplification of the input signal.

C. Convolutional Neural Networks (CNN)

Neural networks, in general, are information processing nodes inspired by the structure of the neurons within the human brain. The term Deep Learning (DL) refers to the use of multiple layers of such neurons to learn complex embedded mathematical functions in data. Convolutional Neural Networks (CNNs) are a DL domain, that use convolution operations on data to extract correlated information, before passing it on to fully connected layers (or the layers of neurons). There are multiple layers in a CNN, such as the Convolution Layer, Activation Layer, Pooling Layer, and the Fully Connected Layer [3]

The Convolution Layer convolves the input data with a variable kernel shape and moves it through a variable stride. The shape of the output of the Convolution Layer is:

$$O = 1 + \frac{N + 2P - F}{S} \qquad (3)$$

where N is the size of the input, P is the number of zeros padded, F is the size of the filter and S is the stride [3]. A wide range of activation functions are used in CNNs depending on the nature of the data, such as the ReLu, leaky ReLu, inverse tanh, sigmoid, and linear function [5]. An activation function is used to make sure the data is within a uniform range at all stages of the CNN. A Pooling operation, MaxPool or AvgPool, is performed to retain the important features obtained after convolution. Multiple Convolution layers are used in succession with varying kernels to extract features from the inputs. Finally, one or more Fully Connected (FC) Layers make up the deep neural network of the CNN. The neurons in each FC layer are connected to every neuron in the preceding and succeeding layer, with each connection assigned a weight. The output at the j^{th} neuron of the k^{th} layer is determined by:

$$y_j^{(k)} = \phi(\sum_{i=1}^{M}(w_{ji}^{(k)} * x_i^{(k-1)} + b_j^{(k)})), \qquad (4)$$

where $\phi(.)$ is the activation function, $w_{ji}^{(k)}$ is the weight associated with the connection between the i^{th} neuron in the $(k-1)^{th}$ layer and the j^{th} neuron of the k^{th} layer and $x_i^{(k-1)}$ is the input coming from this i^{th} neuron of the $(k-1)^{th}$ layer. There are M such neurons in the previous layer. $b_j^{(k)}$ is the bias associated with the j^{th} neuron of the k^{th} layer and $\phi(.)$ is the activation function [5]. At the output, Classification tasks require the number of neurons to match the number of classes, while Regression tasks have a single neuron with a linear activation.

The training of the CNN is done in two stages, the forward propagation and the backward propagation [23]. Initially, all the layers' weights are initialized. In forward propagation, the inputs are fed across the layers of the CNN described above, resulting in an output. The loss function or cost function, compares these outputs with the labels, or the ground truths, to calculate the measure of the deviation or loss. The loss obtained, L is used in the backpropagation step to update the weights through the Gradient Descent algorithm [22]. Using the set of initial weights and the corresponding loss, L, the new set of weights for each layer is calculated using Eqn.(5).

$$w_{ij} := w_{ij} - \alpha * \frac{\delta L}{\delta w_{ij}}, \qquad (5)$$

where α is the learning rate hyper-parameter which controls how much the weights change in response to the calculated loss.

The design of the CNN includes the process of selecting and tuning the best hyper-parameters, i.e., the convolution layer kernel shapes, number of filters, stride, pooling, learning rate, loss function, and optimization algorithm, that cause the model to learn the underlying patterns to the best extent possible. The design is followed by training using back-propagation and the gradient descent algorithm [6], where the weights of each layer in the CNN are optimized such that the loss function is minimized.

III. Related Work

Reduction of PAPR in OFDM systems has been widely researched using traditional techniques, such as Clipping and Filtering, Selective mapping (SLM, Partial Transmit Sequences(PTS), Active Constellation Extension (ACE), and Tone reservation (TR). Each of these is defined in the survey in [21]. On the other hand, there is very limited research on the estimation of PAPR in OFDM systems. [26], [10] estimate PAPR for use in selective mapping to reduce complexity, but at the expense of accuracy. The related estimation techniques are specifically tailored to work with the associated selective mapping approach for PAPR reduction, while our technique does not depend on the given technique for PAPR reduction.

DL is being increasingly applied in communication systems for various applications such as constellation design [18], channel encoding and decoding [27] and channel estimation [27], etc. This is driven by the advantages that DL offers, such as energy efficiency, and the potential to provide better solutions for complex and resource intense processes [20]. Similarly, DL has been applied to the PAPR reduction problem in OFDM systems.

A. Deep-learning-based PAPR Reduction Solutions

[16] proposes an Autoencoder to learn the best constellation mapping, that reduces PAPR while retaining bit error rate (BER), by taking into account the PAPR and BER into the loss function. Once trained, the encoder and decoder are separated and deployed in the transmitter and receiver respectively. In a similar approach, [17] incorporates a significantly less complex neural network, than the one used in [16] into the OFDM chain to reduce the PAPR. Instead of using data symbols in the frequency domain, the neural network is trained on time domain OFDM signals and the loss function is a combination of 3 objective functions to reduce PAPR, out of band distortion BER. By dividing the neural network into multiple modules, [17] achieves better results than [16].

[24] proposes to use a loss function that requires a trade-off between the BER and PAPR, in addition to a modified tanh-based activation layer, which forces the power to become Gaussian. The activation layer controls the PAPR while the autoencoder (AE) loss can focus only on the BER. This method requires modifications at both transmitter and receiver and does not take into account the effects of multipath propagation.

[14] proposes SeqNet which reduces the PAPR of a DFT-s-OFDM system by finding the optimal phase sequence indices in an approach inspired by SLM. The information about the best phase indices that reduce the PAPR of the DFT signal is required to be sent along with the message such that the receiver can reconstruct the original signals.

[25] proposes a DNN based tone reservation scheme, where the DNN is trained to obtain a set of peak canceling symbols, that can be used to reduce the PAPR in the time domain equivalent. No additional signals or distortion are needed to be transmitted since the receiver can simply remove the reserved subcarriers and recover the data from the remainder of subcarriers, but it results in induces spectral inefficiency

While these studies successfully reduce the PAPR using DL with various algorithms, they are reliant on the nature of training data. In addition, the IFFT operations of PAPR reduction significantly increase the transceiver system's computational complexity. Our approach can supplement the PAPR reduction techniques by eliminating the need for these IFFT operations. PESTNet uses DL to estimate the PAPR in the frequency domain symbols before IFFT is applied. Our technique enables the PAPR estimate of the data symbols to serve as a basis for PAPR reduction algorithms.

B. Estimation Models

The approach used is based on the ResNet model, which is one of several popular architectures that have been introduced for image classification. Others include LeNet, AlexNet, and GoogLeNet. ResNet is made up of multiple residual connections between layers that help in learning the features by solving the issue of vanishing gradient [6]. It has been regarded as one of the leading architectures in image classification. The work in [8] replaces the convolutional layers with fully connected layers to adapt ResNet to non-linear regression. Since our task employs non-linear regression, we use a similar version of ResNet as the one used in [8] to compare with our PAPR estimation results.

IV. System Description

The proposed model for PAPR estimation is shown in Fig. 1. We create an OFDM case study based on LTE, according to the 3GPP standards [11]. Raw symbols are modulated with 64-QAM, followed by mapping them to the LTE resource grid. Fig.1(a) shows the traditional LTE system, where the resource grid is sent through the IFFT block to generate the OFDM signal in the time domain, at which point the PAPR is calculated. Fig.1(b) shows the architecture of our CNN model, with a single output node to estimate the PAPR. The LTEgrid is the 2-dimensional LTE resource grid data, which serves as the input to the CNN. PAPR measurements of the time domain signals serve as the PAPR labels or output for the CNN Regression task. Fig.1(c) shows the convolution layers used. A combination of two convolution layers is used to facilitate sufficient extraction of features from the inputs. The final MaxPool layer is replaced with a GlobalMaxPooling layer, which extracts only the maximum values among all the available filters [9]. Each Fully Connected Layer (Dense) is designed with 32 neurons followed by the ReLu activation [5]. The use of four fully connected layers provides sufficient depth to the CNN, which is important to get the best estimation performance. 32 neurons in each fully connected layer, provided the best tradeoff between the performance and the complexity of the neural network. ReLu activation function is preferred over other activation functions because it provides better performance without introducing vanishing gradients. [19]. Additionally, all the layers of the CNN except the first are preceded by a dropout layer, with a 0.5 dropout factor [6]. The use of the dropout following each layer improves the performance of the neural network by randomly turning off a percentage of neurons in each epoch, forcing the neural network to learn the underlying relationship between the inputs and the output, thereby reducing overfitting. Finally, the output layer has a single neuron with a linear activation function, to make Regression possible.

A. Dataset generation and CNN training

Using the MATLAB LTE toolbox, two distinct datasets are generated with R.6 and R.9 configurations as described in Table I. 100,000 samples of R.6 and 50,000 samples of R.9 are generated, with each of these having sizes of 6.26GB

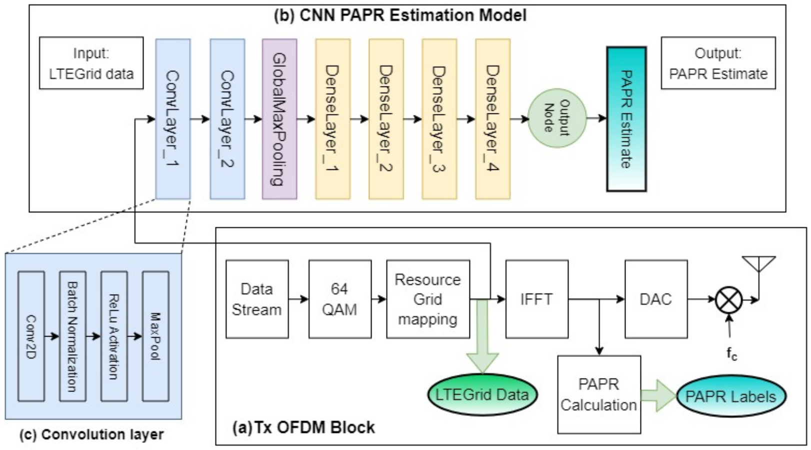


Fig. 1. Block diagram of the proposed PESTNet

TABLE I
LTE DATASET CONFIGURATIONS

Config.	Modulation	Subcarriers	Code Rate
R.6	64QAM	300	0.75
R.9	64QAM	1200	0.75

TABLE II
OPTIMAL HYPERPARAMETERS

Parameter	R.6	R.9
Batch Size	64	64
Conv2D-1 Shape, Filter size	(3,5), 32	(1,3), 32
MaxPool Shape, Stride	(2,6),(2,6)	(2,6),(2,6)
Conv2D-2 Shape, Filter size	(3,5), 16	(1,3), 16
Dropout factor	0.5	0.5
Learning Rate	0.0005	0.0005

and 12.51GB respectively. The input to the CNN is the 2-dimensional LTE resource grid and the PAPR measurements of the time domain signals serve as the PAPR labels for the CNN Regression task. The data pre-processing steps include normalization of the data to a [0,1] scale, an 80-20 train-test split, with 20% of the training data reserved for validation. Early stopping is used to make sure the model doesn't overfit. Over the course of training, the model is modified in accordance with each configuration's input shapes and optimized. Specifically, the hyper-parameters we use to optimize training are batch size, kernel size and the number of filters in each convolution layer, shape and stride of MaxPool layer, learning rate, and dropout factor. Table II lists each of these hyperparameters. The choice of the Conv2d shape, filter size, MaxPool shape and stride, for each configuration, is made such that the dimensionality of the inputs is reduced uniformly while extracting as much information as possible. Each configuration, R.6 and R.9 is trained using the Adam optimizer and the Huber loss, which is less sensitive to outliers compared to the Mean squared error (MSE) and combines the benefits from the Mean Absolute Error and the Mean Squared Error functions [7]. Furthermore, a learning rate of 0.0005 is selected as it provided a steady loss performance when compared to other learning rates. This ensures that the model learns the data accurately, without underfitting or overfitting. Training and testing are done using the TensorFlow2.6 libraries [2] on the University of Florida's HiperGator systems, equipped with two NVIDIA A100-SXM4-80GB GPU cores, 8 AMD EPYC 7742 processors and 64GB of RAM [1].

B. *Performance metrics*

The objective of the proposed approach is to accurately estimate the PAPR while reducing computational load and latency. The time taken to compute IFFT is calculated by implementing the IFFT algorithm on a single frame of the LTE resource grid in Matlab. The hardware used is the HiperGator system [1] with just the AMD EPYC 7742 processors and 64GB of RAM, without the use of the GPU. As is the convention, Cosine Similarity(CS) and Normalized MSE (NMSE) [7] are used together to evaluate the regression model, where the NMSE is the MSE normalized by the signal power of ground truth data and ranges between [0,1], with 0 being the least erroneous and 1 being the most. Combined, a lower NMSE and a higher CS mean that the error between the prediction and label is small and both vectors are similar to each other. We also measure the frequency at which prediction errors of a given magnitude are made by the model. Finally, we consider the computational latency, i.e., the time taken to process a large number of resource grid samples. A large number of samples is used because existing IFFT-based PAPR reduction algorithms are designed to first observe and analyze the PAPR trend across a large number of samples, followed by tuning of the front end for PAPR reduction.

The proposed model is not dependent on the traffic density or the training circumstances. Because it is based only on the nature and volume of the training data, pre-trained models can easily be deployed in a variety of locations as long as the dataset configuration remains the same. Additionally, the model is independent of processing paths or the number of antennas so the resource grid's dimensionality can be increased to allocate the number of antennas, to accommodate MIMO. Such a trained modified model can be deployed directly in base stations or transmitters to estimate the PAPR, without the need of having to process individual antenna chains. In existing PAPR reduction methods, each such chain is fed to IFFTs. Thus the proposed approach is easily scalable to meet MIMO and Carrier Aggregation requirements.

C. Comparison with Deep ResNet

In addition to PESTNet , we use a Deep ResNet based architecture for non-linear regression to estimate the PAPR. The work in [8] used a similar technique, but modified the popular ResNet50 architecture [13] used for classification problems, such that it is applicable to nonlinear regression. In our analysis, the LTEGrid is converted to a 1D array to match the input configuration of the Deep ResNet model proposed in [8]. The Deep ResNet is used on both the R.6 and R.9 configurations. Due to a large number of residual connections in the ResNet architecture, the number of parameters is significantly higher than the proposed PESTNet . Table III shows the difference in complexity of the two models. For a given model, the different number of parameters for each configuration is due to the different number of subcarriers. A similar data preprocessing and preparation method is followed, as PESTNet . The hyperparameters used in training the Deep ResNet are the same as the ones used in [8], except for the loss function. Huber loss is used instead of the MSE, as it is better suited for regression, as explained in Section IV-A.

TABLE III
MODEL COMPLEXITY

Model	Config.	Number of parameters
PESTNet	R.6	12,695
Deep ResNet [8]	R.6	278,449
PESTNet	R.9	5713
Deep ResNet [8]	R.9	1,084,849

V. RESULTS

To evaluate PESTNet, we implemented PESTNet using the training procedure and CNN architecture described in the previous sections. We gathered the performance metrics discussed in Section IV-B and used them to determine the accuracy and computational complexity, indicated by latency. We also used the state-of-the-art DL architecture, ResNet [8], to compare the results.

A. Accuracy

Table II shows the optimal hyperparameters for configurations R.6 and R.9 as revealed by monitoring the PAPR estimation performance metrics. For Deep ResNet, the default parameters suggested by [8] are used in its implementation. Table IV shows the performance results for both DL models and both configurations. The NMSE and Cosine Similarity are very close to ideal conditions, i.e. 0 and 1 respectively. The MSE for R.6 is higher than the case of R.9. This can be attributed to the higher number of subcarriers, 1200 in the R.9 compared to just 300 in case of R.6, allowing the CNN to extract more information.

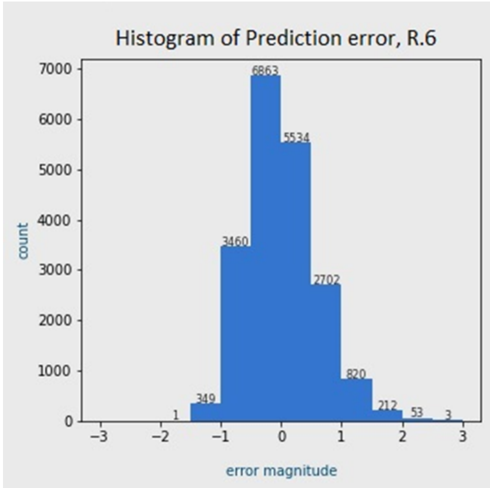
TABLE IV
PAPR ESTIMATION ERROR: PESTNET AND DEEP RESNET [8]

Model	Dataset	NMSE	CS	MSE
PESTNet	R.6	0.0031	0.9984	0.3319
	R.9	0.0021	0.999	0.2446
Deep ResNet [8]	R.6	0.0031	0.9985	0.3274
	R.9	0.0021	0.999	0.2446

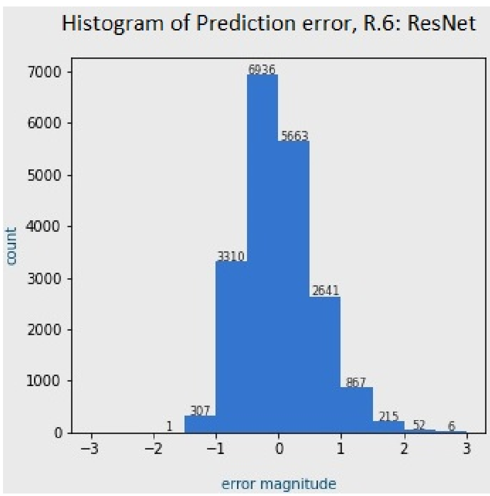
Histograms are provided in Fig. 2 and Fig. 3 to show the deviations between the actual PAPR (ground truth labels) and the estimated PAPR (predicted output). Samples that have 0dB deviation have no difference between the actual PAPR and the predicted or estimated PAPR. Furthermore, if the frequency of the estimation values at 0dB in the histogram are equal to the test sample size, then the model has accurately estimated the PAPR for all the test symbols. The PESTNet deviation error distributions shown in Fig. 2 are concentrated at the respective deviation means for R.6 and R.9, which are at 0.011dB and 0.018dB respectively. For reference, the maximum, minimum, and average PAPR values are 14.25dB, 8.71dB, and 10.32dB, respectively for R.6, and 13.76dB, 9.46dB, and 10.78dB for R.9. The results are similar for ResNet, shown in Fig. 3. Here, the deviation error is concentrated at the respective means, 0.019 and 0.030 for the R.6 and R.9 dataset configurations respectively. For all cases, PESTNet and Deep ResNET; R.6 and R.9, approximately 95% of the errors lie within 2 standard deviations of the mean deviation error. This narrow spread of the errors from the mean 95% of the estimated values are clustered around a very low deviation error. The model predictions are close to the ideal predictions.

TABLE V
OPERATION LATENCY COMPARISON

Method	Run Time(sec)	# of Frames
IFFT: R.6	12 (0.00012×10^4)	10,000
IFFT: R.9	9.7 (0.000097×10^4)	10,000
PESTNet : R.6	1.21	10,000
PESTNet : R.9	1.85	10,000
Deep ResNet [8]: R.6	3.01	10,000
Deep ResNet [8]: R.9	3.37	10,000

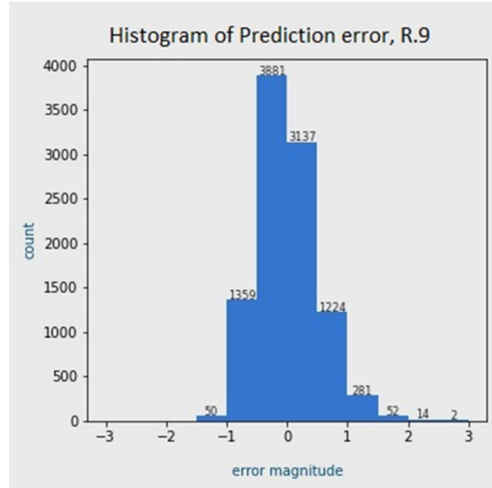


(a) PESTNet

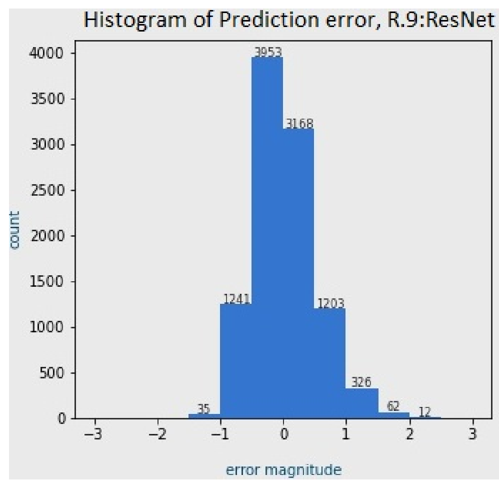


(b) ResNet

Fig. 2. Histogram of Prediction (Estimate) Deviation Error for R.6 Dataset Configuration for (a) PESTNet and (b) ResNet



(a) PESTNet



(b) ResNet

Fig. 3. Histogram of Prediction (Estimate) Deviation Error for R.9 Dataset Configuration for (a) PESTNet and (b) ResNet

B. Computational Complexity

Another claim of this paper is that PESTNet is less computationally complex compared to existing PAPR reduction algorithms, which perform the IFFT before the reduction algorithm is applied to the time domain signals. Table V compares the time taken to execute a single IFFT operation (for a single frame) versus the time for 10,000 frames for the IFFT, the trained PESTNet , and the trained ResNet. To calculate the value for IFFT for 10,000 frames, we multiplied the value for 1 frame by 10,000. However, this is a conservative value, since in current techniques, for each of the 10,000 frames, the IFFT computation would be followed by a PAPR computation, followed by a waiting time to process the next frame. Considering only the IFFT computation, the latency for an IFFT-based PAPR estimate for 10,000 frames is 12 seconds for R.6 and 9.7 seconds for R.9. Our DL comparison, Deep ResNet, is faster, at 3.01 seconds and 3.37 seconds for R.6 and R.9, respectively. The latency for PESTNet is fastest, with 1.21 seconds and 1.85 seconds for R.6 and R.9, respectively, which outperforms the IFFT latency by at least a factor of 10 for the R.6 dataset configuration and at least a factor of 5 for R.9. PESTNet is faster than Deep Resnet, 2.5 times faster for R.6 and almost 2 times faster for R.9. Furthermore, PESTNet provides similar estimation accuracy while having significantly fewer parameters and computation time than Deep ResNet. It shows that a simple CNN architecture is sufficient to estimate the PAPR, as the introduction of the Residual connections adds complexity to the model.

A limitation of the proposed method is that the models are only applicable in scenarios that mimic the training conditions, i.e. R.6 and R.9 configurations of the LTE resource grid respectively. Thus, the results discussed are applicable to all scenarios/environments where the same R.6 or R.9 configurations are used. Additionally, the model is trained offline and the training data includes all the different possible combinations of the 64QAM symbols. This ensures that the trained model has seen all the possible combinations of symbols of the LTE grid, guaranteeing similar performance obtained in testing, as long as the same configuration is maintained. Due to space constraints, tests of additional configurations are beyond the scope of this work. Future experiments can also test the robustness of these specific models in different environments/scenarios.

VI. Conclusion and Future Work

In this work, we presented PESTNet , a novel Pre-IFFT PAPR estimation Technique for OFDM systems using DL. PESTNet provides accurate estimation while reducing the computational latency of current IFFT based techniques. This was achieved by using a CNN model to directly estimate the PAPR of OFDM symbols before IFFT is applied. A case study is performed for LTE, where the results show that PESTNet gives an accurate estimate of PAPR for two configurations, and is able to compute the PAPR of large batches of resource grids faster compared to existing IFFT based techniques. Note that the PAPR estimation technique, i.e., estimating the PAPR of the LTE symbols, is supplemental to PAPR reduction techniques. Thus, we do not compare our approach to PAPR reduction techniques, but we use a version of ResNet to validate our PAPR estimation model.

Furthermore, due to the accurate estimation it provides, PESTNet can serve as a first step in multi-level PAPR reduction algorithms where first the PAPR estimate of a given set of symbols is calculated and then an appropriate PAPR reduction algorithm is implemented. Additionally, it can be extended to MIMO and Carrier Aggregation scenarios by pre-training the CNN models on such datasets. This is a better alternative to IFFT based solutions, which need multiple processing blocks and IFFT blocks for each antenna chain.

References

[1] Research Computing at the University of Florida, May 2022. https://www.rc.ufl.edu/ Last accessed May 22, 2022.

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, et al. TensorFlow: Large-scale machine learning on heterogeneous systems. White Paper, 2015. Posted at https://www.tensorflow.org/about/bib Last accessed May 22, 2022.

[3] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, 2017.

[4] P. Preenu Ann and Renu Jose. Comparison of papr reduction techniques in ofdm systems. In *2016 International Conference on Communication and Electronics Systems (ICCES)*, pages 1–5, 2016.

[5] Andrea Apicella, Francesco Donnarumma, Francesco Isgrò, and Roberto Prevete. A survey on modern trainable activation functions. *Neural Networks*, 138:14–32, 2021.

[6] A. Bhardwaj, W. Di, and J. Wei. *Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling*. Packt Publishing, 2018.

[7] Alexei Botchkarev. A new typology design of performance metrics to measure errors in machine learning regression algorithms. *Interdisciplinary Journal of Information, Knowledge, and Management*, 14:045–076, 2019.

[8] Dongwei Chen, Fei Hu, Guokui Nian, and Tiantian Yang. Deep residual learning for nonlinear regression. *Entropy*, 22(2), 2020.

[9] François Chollet et al. Globalmaxpooling2d layer. Web article, 2015. https://keras.io/api/layers/pooling_layers/global_max_pooling2d/ Last accessed May 22, 2022.

[10] Lilin Dan, Peng Cheng, Yue Xiao, and Shaoqian Li. A low complexity peak estimation scheme for papr reduction in ofdm systems. In *2009 Pacific-Asia Conference on Circuits, Communications and Systems*, pages 66–69, 2009.

[11] ETSI. *ETSI TS 136 101 V14.3.0 (2017-04) LTE. Evolved Universal Terrestrial Radio Access (E-UTRA). User Equipment (UE) radio transmission and reception (3GPP TS 36.101 version 14.3.0 Release 14)*, April 2017. https://www.etsi.org/deliver/etsi_ts/136100_136199/136101/14.03.00_60/ts_136101v140300p.pdf Last accessed May 22, 2022.

[12] Arun Gangwar and Manushree Bhardwaj. An overview: Peak to average power ratio in ofdm system & its effect. *International Journal of Communication and Computer Technologies*, 1(2):22–25, 2012.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[14] Hyeondeok Jang, Seowoo Jang, Yosub Park, Jungsoo Jung, Juho Lee, and Sunghyun Choi. Seqnet: Data-driven papr reduction via sequence classification. In *2021 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, 2021.

[15] Aleksei Kalinov, Roman Bychkov, Andrey Ivanov, Alexander Osinsky, and Dmitry Yarotsky. Machine learning-assisted papr reduction in massive mimo. *IEEE Wireless Communications Letters*, 10(3):537–541, 2021.

[16] Minhoe Kim, Woongsup Lee, and Dong-Ho Cho. A novel papr reduction scheme for ofdm system based on deep learning. *IEEE Communications Letters*, 22(3):510–513, 2018.

[17] Zhijun Liu, Xin Hu, Kang Han, Sun Zhang, Linlin Sun, Lexi Xu, Weidong Wang, and Fadhel M. Ghannouchi. Low-complexity papr reduction method for ofdm systems based on real-valued neural networks. *IEEE Wireless Communications Letters*, 9(11):1840–1844, 2020.

[18] Toshiki Matsumine, Toshiaki Koike-Akino, and Ye Wang. Deep learning-based constellation optimization for physical network coding in two-way relay networks. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–6, 2019.

[19] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *ArXiv*, abs/1811.03378, 2018. https://arxiv.org/abs/1811.03378 Last accessed May 22, 2022.

[20] Timothy O'Shea and Jakob Hoydis. An introduction to deep learning for the physical layer. *IEEE Transactions on Cognitive Communications and Networking*, 3(4):563–575, 2017.

[21] Yasir Rahmatallah and Seshadri Mohan. Peak-to-average power ratio reduction in ofdm systems: A survey and taxonomy. *IEEE Communications Surveys Tutorials*, 15(4):1567–1592, 2013.

[22] Sebastian Ruder. An overview of gradient descent optimization algorithms. *ArXiv*, abs/1609.04747, 2016.

[23] Laxman Singh. Geek culture: Forward and backward propagation, understanding it to master the model training process. Web article, 2015. Posted at https://medium.com/geekculture/forward-and-backward-propagation-understanding-it-to-master-the-model-training-process-3819727dc5c1 Last accessed May 22, 2022.

[24] Melika Vahdat, Koosha Pourtahmasi Roshandeh, Masoud Ardakani, and Hai Jiang. Papr reduction scheme for deep learning-based communication systems using autoencoders. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5, 2020.

[25] Benwei Wang, Qintuya Si, and Minglu Jin. A novel tone reservation scheme based on deep learning for papr reduction in ofdm systems. *IEEE Communications Letters*, 24(6):1271–1274, 2020.

[26] Chin-Liang Wang, Sheng-Ju Ku, and Chun-Ju Yang. A low-complexity papr estimation scheme for ofdm signals and its application to slm-based papr reduction. *IEEE Journal of Selected Topics in Signal Processing*, 4(3):637–645, 2010.

[27] Hao Ye, Geoffrey Ye Li, and Biing-Hwang Juang. Power of deep learning for channel estimation and signal detection in ofdm systems. *IEEE Wireless Communications Letters*, 7(1):114–117, 2018.

[28] Feng Zou, Zhijun Liu, Xin Hu, and Gang Wang. A Novel PAPR Reduction Scheme for OFDM Systems Based on Neural Networks. *Wireless Communications and Mobile Computing*, 2021:5574807, April 2021. Publisher: Hindawi.