A Deep Learning Approach to Radio Signal Denoising

Ebtesam Almazrouei[1], Gabriele Gianini[1,2], Nawaf Almoosa[1], Ernesto Damiani[1,2]
[1] Emirates ICT Innovation Center (EBTIC), Khalifa University, Abu Dhabi, UAE
[2] Computer Science Department, Università degli Studi di Milano, Italy
E-mail: {ebtesam.almazrouei,gabriele.gianini,nawaf.almoosa,ernesto.damiani}@ku.ac.ae

Abstract

This paper proposes a Deep Learning approach to radio signal de-noising. Being data-driven, it allows de-noising signals (corresponding to distinct protocols) without requiring explicit use of expert knowledge: in this way it grants higher flexibility. The core component of the Artificial Neural Network architecture used in this work is a Convolutional De-noising AutoEncoder. We report about the performance of the system in spectrogram-based denoising of the preamble of protocols from the IEEE 802.11 family, studied using simulation data. This approach can be used within a machine learning pipeline: the denoised data can be fed to a protocol classifier. A further perspective advantage of using the AutoEncoders in such a pipeline is that they can be co-trained with the downstream classifier (the protocol detector), to optimize its accuracy.

I. Introduction

Driven by spectrum scarcity, emerging radio standards are envisioned to involve spectrum sharing and coexistence of heterogeneous radio-access technologies. Shared-spectrum access can significantly improve capacity and spectrum utilization. However, it gives rise to new operational challenges, given the increased complexity of the radio-access environment and susceptibility to interference-induced disruption. Intelligent access monitoring and orchestration is therefore essential to ensure meeting the objectives of fair quality of service and regulatory compliance.

However, the adoption of intelligent techniques has to face some key challenges. Among the most important are the following. Firstly, the performance of the algorithms is required to generalize across radio-access scenarios, which vary greatly and may involve severe signal degradation (due to physical factors and interference). Secondly, the adopted solutions must be easily applicable across multiple radio standards: this implies factoring out standard-specific expert-engineered features.

Machine learning (ML) methods, and, more specifically, a set of recently developed techniques, known as Deep Learning (DL) [1], bear the potential of advancing the intelligence of radio devices, providing data-driven flexible solutions, without relying heavily on expert knowledge. Among the problems that the ML can target are signal denoising, protocol detection, and classification; further applications might include device or user profiling and classification, source counting.

In this paper, we address the problem of signal denoising: a deep learning model is applied to radio signals to improve the quality of received signals. By noise, we mean not only classic standard perturbations such as white noise etc., but the ensemble of effects of the channel, which distort the transmitted signal before it is received at the destination antenna.

Specifically, we propose a method to unfold the *spectrogram* of the transmitted signal from such noise by means of a type of Artificial Neural Network known as Convolutional Denoising AutoEncoders (CDAE, defined in the next Section). Being such a spectrogram equivalent to a 2D image, convolutional techniques turn out to be very effective. We demonstrate the method using two WLAN protocols: IEEE802.11n and IEEE 802.11a, and focus on the problem of denoising the protocol preamble. In this preliminary work, we use simulated data.

This approach to denoising based on CDAE allows subsequent use of a protocol classifier algorithm based on the denoised data. The classifier used downstream can be, for example, a standard or a convolutional multi-layer perceptron. A perspective advantage of using CDAE is that they can be co-trained with such a downstream classifier, to optimize its accuracy. Naturally, the two components can be trained separately if, as it happens with our case studies, the denoising is very effective. Indeed, we achieve a reconstruction accuracy (defined as $1-$loss metrics value) of the order of 85%. This loss is referred to the whole 2D spectrogram image: using only selected informative regions, the accuracy would be even higher.

It is important to highlight that, this work does not aim at comparing to the traditional well-established techniques for signal denoising. Most of those techniques are driven by domain expert knowledge, whereas the method we discuss is completely data-driven, thus it allows for high flexibility. Once the data from the transmitter and the corresponding data from the receiver are available, the proposed system can be trained to denoise the signal effectively.

The present paper is structured as follows: Section II recalls the definitions of ML, DL, and the related work of ML applications to radio signal processing. In section III, Denoising AutoEncoders are described. Data sets and analysis tools are discussed in Section IV. The architecture of the proposed Denoising AutoEncoders, loss function, and optimization algorithms are discussed in Section V. The experimental configurations and results are discussed in Section VI; conclusions are drawn and in Section VII.

II. Definitions and Related work

A. Machine Learning

When describing ML tasks and models/algorithms – where a task can be realized by using more models/algorithms – it is customary to organize them into few main classes, the most widely represented being those of supervised, unsupervised and reinforcement learning algorithms. For instance, in the supervised machine learning task known as classification, a labeled dataset is used to learn a predictor/classifier, later to be used to assign labels to unlabeled data. The performance of such model can be quantified using a test dataset endowed with labels, not used during the training: this dataset allows to estimate to which extent the learned model is able to generalize to unseen data [2]. An analogous task is regression: there the model learns to predict a numerical label for each input. Among the most popular supervised models are K-Nearest Neighbor (KNNs) classifiers [3], Support Vector Machines (SVMs) [4], and Naive Bayesian classifiers [5]. A special role in the recent developments has been taken by Artificial Neural Network algorithms (ANNs), whose most commonly used representative is the Multi-Layer Perceptron (MLP). MLPs learn from examples a certain set of input-output mappings by optimizing the weights that link successive layers of artificial neurons: MLPs are used both for classification and regression, and trained by the back-propagation (BP) algorithm [6].

In unsupervised ML models, the classes of training dataset are not labeled (there are no correct answers to guide the training process). Among unsupervised algorithms are the clustering algorithms, that group the trained data into clusters (and possibly sub-clusters): elements within the same clusters are assumed to be similar in some sense, more than elements of different clusters. A common clustering algorithm is the k-means algorithm [7]. Another example of unsupervised ML is represented by techniques for dimensionality reduction, where a new representation is learned for the dataset [8] (e.g. in this representation redundant or irrelevant information is removed): the new representation can be used to feed a classifier, improving its performance: common models of this type are Independent Component Analysis (ICA) [9], and Principal Component Analysis (PCA) [10].

In Reinforcement Learning (RL) [11] the model learns not based on labels but based on rewards, e.g. positive rewards are provided when the algorithm is taking the right direction. Examples of Reinforcement Learning Algorithms are the Genetic Algorithms (GA), Ant Colony (ACO) algorithms and Q-learning.

B. Deep Learning: Convolutional AutoEncoders

The DL methods represent a part of ML developed in the latest years, that allows the training of larger and more complex models, thus facing more challenging tasks, using reasonable computational time and resources.

The DL techniques related to the present work are Convolutional Neural Networks (CNNs) architectures and AutoEncoders (AEs): those two techniques are used together to create Convolutional AutoEncoders (CNN-AE).

Differently from standard fully connected Multi-layer Perceptrons, where in principle every neuron (of non-output layers) feeds all the neurons of the next layer, in CNNs there is a limited number of connections from one layer to another, and the same pattern of weights is sought for a layer to the next; this, together with the use of special ”pooling” layers (down-sampling layers) reduces considerably the number of parameters of the network. CNNs are particularly effective in image processing because they can exploit the local coherence/correlation of the image areas.

AutoEncoders are ANNs characterized by a special architecture and a peculiar training procedure: they are used to find a new representation, an *encoding*, of the data that is endowed by desired characteristics such as extra compactness, or extra sparseness, depending on the cases. Their structure is a multilayered architecture, but they are trained by means of a self-supervised algorithm: they do not use labels associated with input data, but try to replicate every input example on the output, i.e. use the input record as its own (complex) label. The architecture consists of at least three layers: an input layer, a hidden layer, and an output layer. The encoding layer is smaller than the input if one looks for a more compact representation, larger if one looks for a sparser representation. AEs are trained typically using the BP algorithm: after convergence, the weights are frozen and the layer(s) beyond the central (encoding) one are discarded. What is left is a function that maps the input into a different representation: when presenting the data in this new representation to a classifier learning algorithm (e.g. an ordinary MLP), one typically obtains improvements in the classification performance.

By stacking successively trained AEs (the encoding found by one becomes the input of the next), one can build very deep models, optimizing a moderate number of parameters at a time (a moderate set of parameters for each layer involved).

AEs can be used even without the need of feeding a downstream classifier: finding a better (in some sense) representation has a value by itself: for instance Denoising AEs (described in the next section) are trained so as to find a representation where the signal is unfolded from noise.

CNN-AEs merge the advantages of the AE representation learning process with the reduced complexity of the CNN paradigm. In CNN-AEs the down-sampling layers in the initial encoding part of the network are paralleled by up-sampling layers in the decoding part of the network.

When trained for yielding a denoised representation the CNN-AE is called Convolutional De-noising AutoEncoder (CDAE).

C. Machine Learning in Radio Signal Denoising

ML models have been applied in the radio signals area for signal classification, device classification, spectrum detection, noise estimation, radio optimization, and anomaly detection and other problems [12]. Hierarchical-SVMs (H-SVMs) have been applied for channel noise estimation and location estimation to classify mobile nodes [13]. H-SVMs were implemented for Channel selection in cognitive radio networks for users classification [14]. Also, SVMs have been designed for anomaly detection in sensor networks [15].

Q-learning has been applied for channel selection in Cognitive Radio to optimize the spectrum [16]. Bayesian learning has been used in spectrum sensing for spectrum optimization in Cognitive Radio [17]. Moreover, it has been employed for users detection in Cellular network [18] and other Cognitive Radio applications [19]. CNNs have been proposed for spectrum sensing and modulation classification in Cognitive Radio for spectrum classification [20].

There are a few works who considered DL for modulation classification in cognitive radio for users classification [21], [22] or for applying DL for indoor localization for Wi-Fi signals to identify antenna's locations [23]. AE has been applied in digital receiver design including Multiple Input Multiple Output (MIMO) detection to replace end-to-end receivers by using a deep learning architecture [24]. Also, AE has been proposed within as new Physical Layer design for communication systems based on ANNs [25].

The presence of noise affects the quality of the received signal. Several researchers designed ML models to improve the radio signal quality. In [13], H-SVMs have been proposed to estimate the noise level for an Additive White Gaussian Noise (AWGN) channel in MIMO wireless network. In [26], the CNN classifier is proposed for modulation classification in presence of different Signal-to-Noise Ratios (SNRs). In [27], CNNs are used for spectral identification over different SNRs.

To the best of our knowledge, we are not aware of attempts to improve the signal quality using ML model *before* signal identification, under fading channel conditions. In our approach, DL is explored for improving the received signal quality using CDAEs.

In this paper, we apply the CNN-AEs to denoise the radio signal from the fading channel effects. This process of denoising is data-driven, thus carried on without resorting to domain experts knowledge.

III. De-noising AutoEncoders

Here we briefly describe the Denoising AutoEncoders [28], which are used to reconstruct data from a corrupted input. In contrast to standard AEs – which receive in input an example and is trained to reconstruct the input as faithfully as possible –, the Denoising AE are given in input a noisy example and forced to reconstruct a denoised version. To do so one needs both the non-noisy and the noisy version of each example: those can be obtained for instance from a simulation environment, or by corrupting artificially non-noisy data.

AutoEncoders [28] consists of an encoder which represents the mapping F_Θ of the input vector X into hidden representation y.

$$y = f_\Theta(x) = s(Wx + b) \tag{1}$$

where s is nonlinear function such as Rectified Linear Unit (ReLU) or a sigmoid, while Θ represents the parameters $\Theta = (W, b)$. W is the weight matrix and b is the offset vector.

The y representation is then decoded to yield the reconstruction z, which is the same size as the input x.

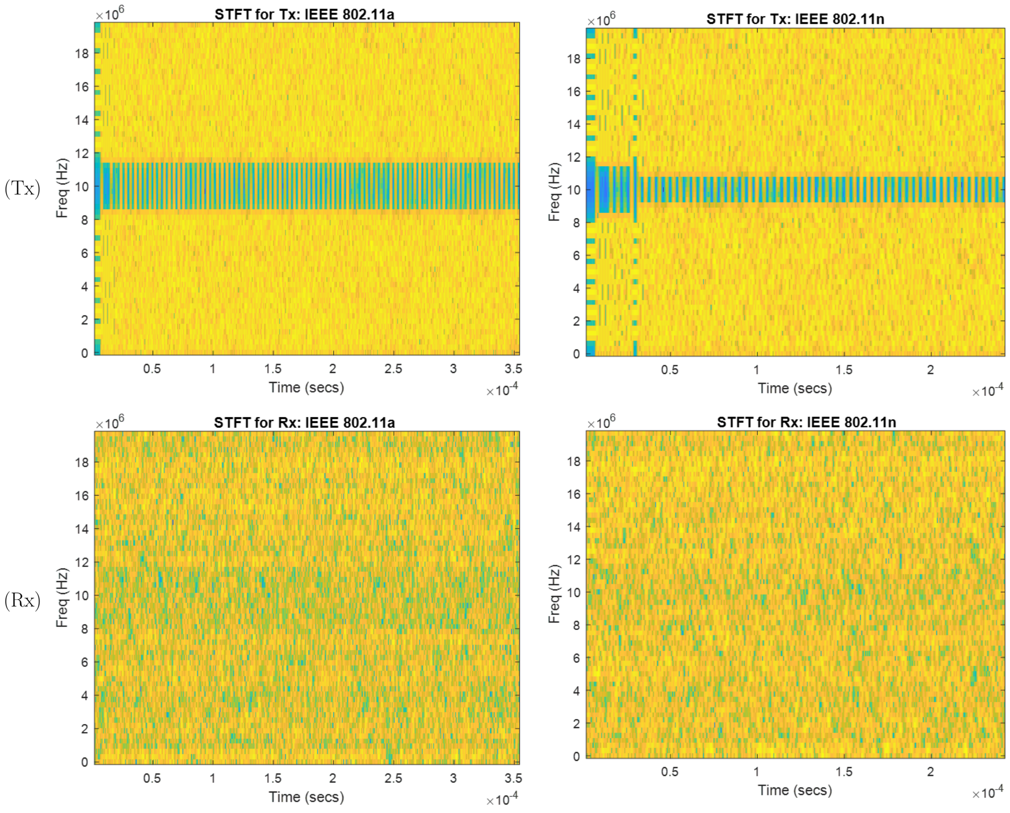
$$z = g_\Theta(y) = s(W'y + b') \tag{2}$$

The parameters W, W', b, and b' are optimized by minimizing a suitable loss function. For binary data where $x \in [0, 1]^d$, the loss function typically used is the following mutual information:

$$-\sum_j [x_j log(z_j) + (1 - x_j) log(1 - z_j)] \tag{3}$$

De-noising AutoEncoders (DAEs) follow the general concept of AutoEncoder but with the addition of noise n to the input x, to obtain a noisy input $\hat{x} = x + n$. The reconstruction of z is however compared to the clear signal x in the loss function. The loss is minimized by means of the stochastic gradient descent algorithm [29].

In our approach for de-noising the preamble, the Convolutional De-noising AutoEncoder (CDAE) is applied, i.e. the layers are convolutional layers like in a CNN: the parameters W and b of a patch of the image are shared among all the locations and enforce spatial locality. In general, CDAEs perform better than classical DAEs in image processing [29].



(802.11a) (802.11n)

Fig. 1. Spectrograms for radio signals for IEEE 802.11a (left) and IEEE 802.11n (right). Top: transmitted signals (Tx). Bottom: received signals (Rx) under the hypothesis of the NLOS fading channel described in Table II (standard model E from reference [33]).

IV. Simulation Data

Several experiments have been performed in various scenarios, with the two protocols IEEE 802.11a, and IEEE 802.11n, to study the following performances:

- the performance of different CDAE models to de-noise various radio signals coming from one protocol;
- the performance of different CDAE models to de-noise the received noisy signals for multiple protocols.

We investigated the denoising performance of the CDAEs by means of simulated data, by focusing on the reconstruction of the preamble of each protocol.

A. Simulation and analysis tools

We used mainly MATLAB for generating simulated data and the TensorFlow and Keras libraries (in Python) for building the Deep Learning models.

Specifically used the MATLAB/SIMULINK suite to simulate the whole model of the physical layer of the IEEE wireless local area network (WLAN). Specifically, the WLAN System Toolbox can be used to design, simulate, analyze, and test various IEEE WLAN protocols such as IEEE 802.11b/a/g/n/j/p and 802.11ac/ad/ah standards: it allows the configuration of the physical layer waveform for each IEEE standard, it makes possible to design the transmitter, the channel model, and the receiver [30]. For example, we used the *wlanTGnChannel* component to pass input signals Tx through an 802.11n (TGn) multipath fading channel. Various parameters could be set up for a specific WLAN scenario: the sample rate fs, channel bandwidth cbw, large-scale fading effect, path loss and shadowing, delay profile, and the channel model.

Several tools have been used for building artificial neural networks. TensorFlow (TF) is a set of Python libraries to develop deep neural networks [31]. We used the Keras libraries to control the TF back-end [32].

The simulation was carried on both under Windows 10 and Linux exploiting graphical processing units (GPUs).

B. Noise model used in the data

The data from the WLAN environment has been the primary source of the radio signals for this study: we focussed on the IEEE 802.11 physical layer standards, for Wi-Fi signals operated in 2.4GHz and 5GHz range. Specifically, we generated the end-to-end radio signal environment for the following standards: IEEE 802.11a and IEEE 802.11n. The design parameters for each protocol (channel bandwidth BW, carrier frequency f_c, modulation scheme, channel model, number of transmitters and receivers, etc.) are summarized in Table I [34]–[36].

TABLE I
WLAN STANDARDS

Standard	**Year**	**Maximum Data rate**	**Channel Bandwidth**	**Frequency Band (GHz)**	**Modulation**
IEEE 802.11n	2009	600 Mbps with four spatial streams	20MHz, 40MHz	2.4 GHz, 5 GHz	OFDM
IEE 802.11a	1999	54Mbps	20MHz	5GHz	OFDM

In the practical radio spectrograms, the radio signals suffer from noise due to the No-Line of Sight (NLOS), multi-path, intersymbol interference, Doppler effect, and fading effects. In our experiments, the radio signals are modeled by a Rayleigh channel model and NLOS effects, where the distance between the transmitter T_x and the receiver R_x is not less than 5 meters. Several standard indoor channel models have been developed in [33]. In our research, we chose to study model E, corresponding to a typical large open space, indoor or outdoor, with large delay spread (we set 250 ns delay spread), a highly dispersive channel. Table II details path delays and average path gains for Model E.

TABLE II
RAYLEIGH CHANNEL MODEL E

Path Delays (nanoseconds)	[0 10 20 40 70 100 140 190 240 320 430 560 710 880 1070 1280 1510 1760]
Average path gains (dB)	[-4.9 -5.1 -5.2 -0.8 -1.3 -1.9 -0.3 -1.2 -2.1 0.0 -1.9 -2.8 -5.4 -7.3 -10.6 -13.4 -17.4 -20.9]

C. The datasets

The dataset consisted in 1000 radio spectrogram images corresponding to as many preambles for each studied IEEE 802.11 protocol.

1) Spectrograms: Each spectrogram represented the Short Time Fourier Transform (STFT) [37] of the raw time series $x(t)$ corresponding to the signal of a preamble. More specifically, a spectrogram pictures the STFT $S_x(\tau, \omega)$ as a function of the (discretized) time τ and frequency ω. Each preamble was partitioned into 3782 time intervals, for each of which 64 frequencies were computed. Thus, each image consisted in 64×3782 gray-level pixels, where gray-level expressed the modulus $|S_x(\tau, \omega)|$.

An example of such spectrograms is provided in Figure 1. One can notice the difference between the two transmitted spectrograms and the received signal spectrograms, i.e. the severe degradation undergone by both protocol preambles, due to the channel.

2) Quantization: The spectrogram has very high dimensionality. To reduce it, the values $|S_x(\tau, \omega)|$ have been quantized and encoded with binary values $b_x(\tau, \omega)$ as follows:

$$b_x(\tau, \omega) = \begin{cases} 1 & if\, |S_x| > \langle S_x(\tau, \omega) \rangle \\ 0 & if\, |S_x| < \langle S_x(\tau, \omega) \rangle \end{cases} \tag{4}$$

where the average $\langle \cdot \rangle$ is taken over the pixels of the spectrogram.

Examples of quantized spectrograms are shown in Fig. 2.

The experimental configurations for which the quantized spectrograms have been generated are shown in Section VI.

V. DENOISING WITH CDAE

The problem we address hereafter is the denoising the preamble spectrograms of a single radio protocol at a time. In other words, we adopt the simplifying assumptions that, in the sensed area, only one protocol is used and only one transmitter is operational. Furthermore, we posit that we are able to identify the time interval in which lies the preamble and distinguish it from the subsequent part of the frame. The latter simplifying assumption can be considered plausible if one postulates the presence of a suitable segmentation algorithm (such an algorithm is out of the scope of the present paper).

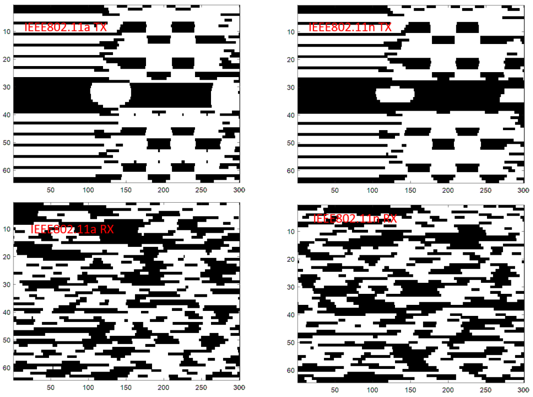


Fig. 2. Spectrograms for transmitted (top) and received (bottom) preambles for IEEE 802.11a (left) and IEEE802.11n (right).

Adopting a configuration in which we denoise spectrograms implicitly assumes that the radio signal has been previously received and decoded into a time-domain signal representing the actual preamble and subsequently transformed into a spectrogram by means of Short Time Fourier Transform.

A. *Loss function and optimization algorithm*

To de-noise the radio spectrograms, we used the CDAEs introduced in Section III.

From the example of a spectrogram is shown in Figure 1, one can see that such a 2D signal presents regions of homogeneous behavior: e.g. one can notice a horizontal central strip, consisting in vertical strips of low values, alternated to higher values. The spectrogram can be considered a sort of picture, with consistent regions.

The reasons why we chose to experiment with convolutional versions of the AE are several, however, we counted on the fact that the convolutional mechanisms are known to be effective in processing/learning images.

It is true that the special kind of image under study had rather unbalanced dimensions (64×3782 pixels per spectrogram instance), but the internal coherence of different regions of the image suggested that the spectrogram could be dealt with almost like an ordinary gray-scale image.

Among the main parameters to choose for setting up a CDAE are the loss function and the optimization algorithm for the neurons' link weights. As an evaluation metric for CDAE several loss functions can be used, such as cross-entropy and mean square error. In our model, we used binary cross entropy defined as follows [38]:

$$CE = \sum P(x)\log(Q(x)) + (1 - P(x))\log(1 - Q(x)) \tag{5}$$

where $P(x)$ represents the target distribution and $Q(x)$ the distribution obtained at the CDAE output.

The AutoEncoder is trained over the stochastic channel model at learning rate 0.001 [25]. The optimization we chose for our model is based on a variant of Stochastic Gradient Descent (SGD), the Adaptive Gradient Descent (AGD) [39]. The main difference between SGD and AGD is that in the latter, the learning rate of the optimization weight gradient descent algorithm is not fixed, but depends on the training loss model. Fig. 3 shows, for comparison, the accuracy of training using SGD and AGD: the latter converges faster, and has higher accuracy. Therefore, our CDAE is trained and optimized using AGD algorithms.

Among the reasons why we have preferred AGD over SGD is the fact that in general AGD is know to work better for images.

B. *Operation and architecture of the CDAE*

The CDAE follows the standard approach of de-noising AutoEncoders with encoded and decoded convolution neural layers but uses also some convolutional layers. Convolutional layers are known to preserve the spatial image distribution so that the corresponding networks perform better than non-convolutional neural networks in image processing [29].

Fig. 4 shows the operation of a CDAE network. The input consists in both noisy and non-noisy data: in our case, the noisy data are the spectrograms of the received signal, which was generated by propagating the clean transmitted signal in the non-ideal channel. Those input spectrograms were processed by the encoder, which consists of multiple convolutional layers, and by the decoder: the output is compared to the spectrogram of the clean signal. By optimizing the loss function, the network finds a compressed representation which is to some extent denoised.

The architecture of CDAE we used is summarized in Table III. It encompasses a sequence of sixteen layers: an input layer, seven 2D convolutional layers for the *encoder*, and eight 2D convolutional layers for *decoder*. The 2D *encoder* includes

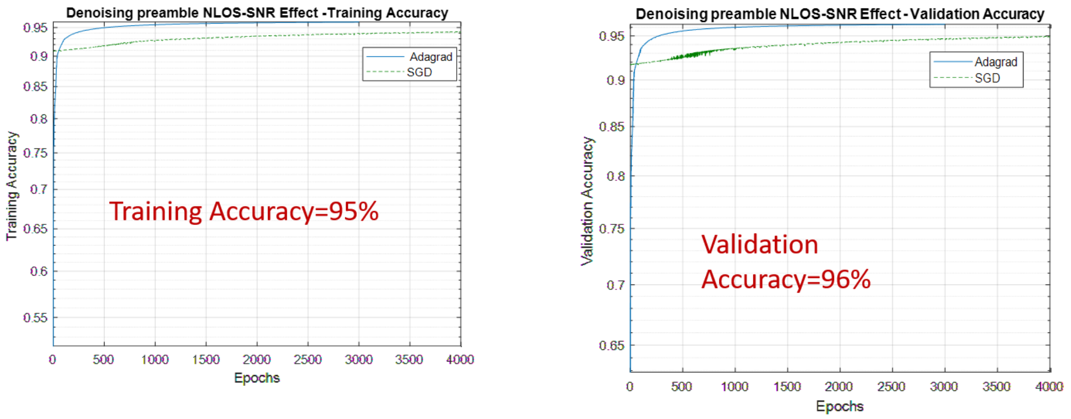


Fig. 3. Training and validation accuracies as a function of the number of epochs: the use of the standard Stochastic Gradiend Descent (SGD) and the Adaptive Gradient Descent (Adagrad) algorithms are compared. One can observe that Adagrad performs remarkably better than SGD.

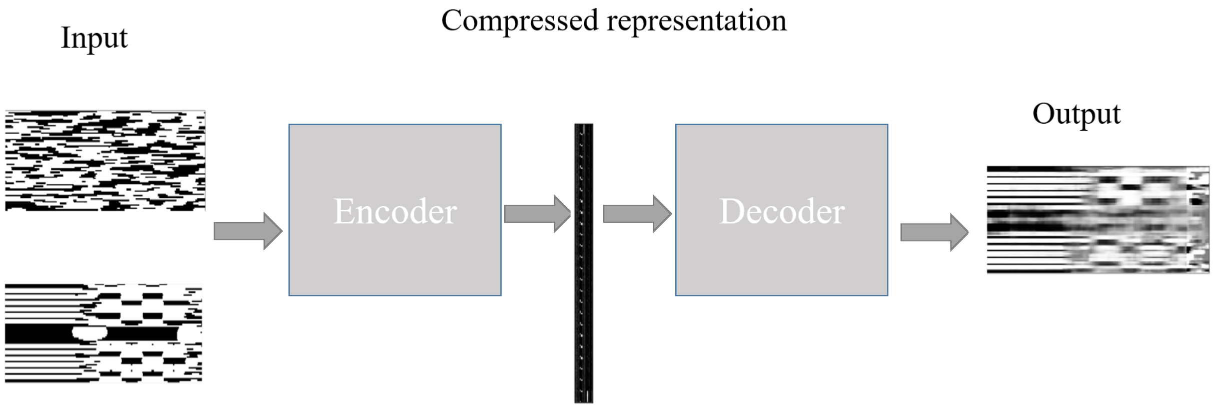


Fig. 4. Schematic view of the operation of a de-noising auto-encoder.

convolutional layers, use *rectified linear unit (ReLU)* layers, a *dropout* layer, and a 2D *max-pooling* layer. The *ReLU* layer computes element-wise the maximum between a value and a threshold. The *dropout* layer is used for better generalization and to avoid overfitting. The *max-pooling* performs a deterministic down-sampling operation to reduce the spatial dimensions (width, height) of the convolutional layer.

The first hidden layer is a convolutional layer with 16 feature maps. Each feature map is connected to a different kernel. Each kernel has size 3 by 3 in each feature map. The *decoder* convolutional layers includes a *ReLU* layer, an *up-sampling* layer, and a *dropout* layer. The *up-sampling* layer is adapted to change the output of convolutional layers in the decoder to a higher resolution that matches the original input size image.

VI. Experimental Configurations and Results

A. Studied Configurations

The WLAN environment for both IEEE 802.11a and IEEE 802.11n has been modeled as Single Input Single Output (SISO) consists of one transmitter Tx, a channel model, and one receiver Rx. The specification parameters for IEEE standards as mentioned in section IV-B were followed. The channel has been modeled as a Rayleigh channel. Different levels of Signal-to-Noise-Ratio (SNR) were also introduced to study the robustness of the neural network performance. The simulation model included also the Doppler Effect.

For all transmitted and received packets for both IEEE standards, the signal was transformed into the corresponding spectrogram.

B. Results and Discussion

Overall, we generated $40,000$ spectrogram images, a half for IEEE 802.11a and a half for IEEE 802.11n): 90% was used for the training and 10% for the test. Half of the training spectrograms were clean, half were noisy. Table IV summarizes these data.

TABLE III
CDAE MODEL ARCHITECTURE

Layer Type	Output Shape
Input layer	(None, 300, 64, 1)
2D convolutional layer (Conv2D)	(None, 300, 64, 16)
Dropout layer	(None, 300, 64, 16)
MaxPooling	(None, 150, 32, 16)
Conv2D	(None, 150, 32, 16)
MaxPooling	(None, 75, 16, 16)
Conv2D	(None, 75, 16, 16)
MaxPooling (output:compressed representation)	(None, 25, 8, 16)
Conv2D	(None, 25, 8, 16)
Upsampling	(None, 75, 16, 16)
Conv2D	(None, 75, 16, 16)
Upsampling	(None, 150, 32, 16)
Conv2D	(None, 150, 32, 16)
Dropout	(None, 150, 32, 16)
UpSampling	(None, 300, 64, 16)
Conv2D	(None, 300, 64, 1)

TABLE IV
SIZE OF THE DATASETS IN TERMS OF NUMBER OF IMAGES

Data Type	**Noisy Datasets**	**Clean Datasets**	**Testing Datasets**	**Training Datasets**
IEEE 802.11a	10000	10000	2000	18000
IEEE 802.11n	10000	10000	2000	18000
IEEE 802.11a and IEEE 802.11n	20000	20000	4000	36000

Fig. 5 shows a sample results of the denoising process. It is obvious that the developed CDAE is capable of performing an effective denoise operation for the IEEE protocol preambles considered. This is due to the application of convolutional layers as it preserves the spatial locality of the input image. Fig. 5 provides a blow-up of the original, noisy and denoised spectrograms. One can see that visually meaningful features have been partially restored.

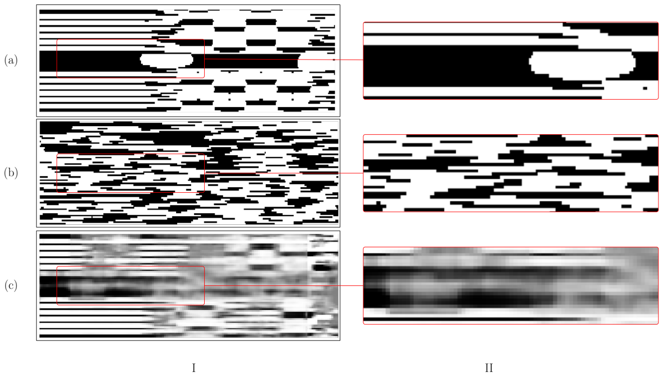


Fig. 5. An example of the de-noising of the IEEE 802.11n preamble. Left: a) spectrogram of the transmitted signal, b) spectrogram of the the received signal, c) the denoised spectrogram. Right: the blow-up of a central region of the three spectrograms.

The overall accuracy of signal de-noising is assessed with a different type of metrics. Fig. 6 shows the behavior of the training and testing accuracy for de-noising of radio spectrograms. The accuracy of de-nosing reached 86% after 2000 epochs. The training was stopped at 2000 epochs because the gains around that number of epochs resulted to be negligible. One can observe that the training accuracy and testing accuracy follow the same overall curve, which suggests that there was no overfitting by the developed model.

VII. CONCLUSION

We proposed a system for signal de-noising based on machine learning algorithms and demonstrate its effectiveness using various WLAN protocols. The advantage of such systems is that they can perform such a task without relying on expert knowledge: this potentially can provide high flexibility. The de-noising algorithm for radio protocols was developed using a

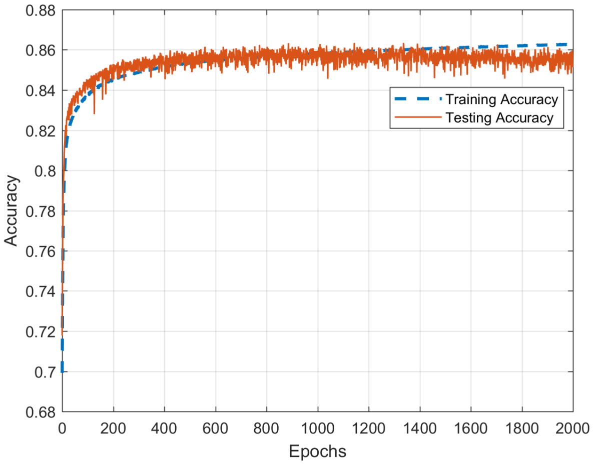


Fig. 6. The training and testing accuracy for the developed CDAE model.

Convolutional De-noising AutoEncoder (CDAE). The results show that the system is capable of performing de-noising for radio protocol and of reconstructing the transmitted radio signals even in the presence of severe noise in the radio spectrograms. The overall training accuracy is about 86%.

Although we demonstrated the approach for protocols of the IEEE 802.11 family, this approach could be generalized to all radio protocols: further studies are required to examine the challenges and the effectiveness of such generalization.

The current study was performed using simulation data; we plan to apply this approach to data sets of real radio signals.

Among the assumptions of the current study was that only one radio source was active. Future investigations will consider multiple concurrent sources and the problem of filtering out the reciprocal interference.

We plan, also, to optimize the Convolutional AutoEncoder structure within an actual classification pipeline in order to improve the classification performance, which would represent the main application scenario for this kind of denoising. The advantage of using CDAE in such a context is that they can be trained in isolation, but also can be co-trained with the classifier algorithm, so as to improve the overall performance.

Acknowledgements

Authors GG and ED acknowledge the support by the EU H2020, research programme, through the projects Toreador (No. 688797) and Threat-Arrest (No. 786890).

References

[1] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[2] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, pp. 3–24, 2007.

[3] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[4] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[5] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[7] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[8] H. Kamyshanska and R. Memisevic, "On autoencoder scoring," in *International Conference on Machine Learning*, 2013, pp. 720–728.

[9] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[10] A. Mackiewicz and W. Ratajczak, "Principal components analysis (pca)," *Computers and Geosciences*, vol. 19, pp. 303–342, 1993.

[11] R. S. Sutton and A. G. Barto, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.

[12] K. Umebayashi, K. Moriwaki, R. Mizuchi, H. Iwata, S. Tiiro, J. J. Lehtomäki, M. Lopez-Benitez, and Y. Suzuki, "Simple primary user signal area estimation for spectrum measurement," *IEICE Transactions on Communications*, vol. 99, no. 2, pp. 523–532, 2016.

[13] V.-s. Feng and S. Y. Chang, "Determination of wireless networks parameters through parallel hierarchical support vector machines," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 3, pp. 505–512, 2012.

[14] K. G. M. Thilina, E. Hossain, and D. I. Kim, "Dccc-mac: A dynamic common-control-channel-based mac protocol for cellular cognitive radio networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3597–3613, 2016.

[15] Z. Li, Q. Ding, S. Haeri, and L. Trajković, "Application of machine learning techniques to detecting anomalies in communication networks: Classification algorithms," *Cyber Threat Intelligence*, pp. 71–92, 2018.

[16] V. Raj, I. Dias, T. Tholeti, and S. Kalyani, "Spectrum access in cognitive radio using a two-stage reinforcement learning approach," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 20–34, 2018.

[17] Y. Xu, P. Cheng, Z. Chen, Y. Li, and B. Vucetic, "Mobile collaborative spectrum sensing for heterogeneous networks: A bayesian machine learning approach," *IEEE Transactions on Signal Processing*, 2018.

[18] R. Amorim, J. Wigard, H. Nguyen, I. Z. Kovacs, and P. Mogensen, "Machine-learning identification of airborne uav-ues based on lte radio measurements," in *Globecom Workshops (GC Wkshps), 2017 IEEE*. IEEE, 2017, pp. 1–6.

[19] A. Assra, J. Yang, and B. Champagne, "An em approach for cooperative spectrum sensing in multiantenna cr networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1229–1243, 2016.

[20] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.

[21] B. Kim, J. Kim, H. Chae, D. Yoon, and J. W. Choi, "Deep neural network-based automatic modulation classification technique," in *Information and Communication Technology Convergence (ICTC), 2016 International Conference on*. IEEE, 2016, pp. 579–582.

[22] S. Peng, H. Jiang, H. Wang, H. Alwageed, Y. Zhou, M. M. Sebdani, and Y.-D. Yao, "Modulation classification based on signal constellation diagrams and deep learning," *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1–10, 2018.

[23] X. Wang, L. Gao, S. Mao, and S. Pandey, "Deepfi: Deep learning for indoor fingerprinting using channel state information," in *Wireless Communications and Networking Conference (WCNC), 2015 IEEE*. IEEE, 2015, pp. 1666–1671.

[24] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Deep learning based mimo communications," *arXiv preprint arXiv:1707.07980*, 2017.

[25] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep learning based communication over the air," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, 2018.

[26] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International conference on engineering applications of neural networks*. Springer, 2016, pp. 213–226.

[27] T. J. O'Shea, T. Roy, and T. Erpek, "Spectral detection and localization of radio events with learned convolutional neural features," in *Signal Processing Conference (EUSIPCO), 2017 25th European*. IEEE, 2017, pp. 331–335.

[28] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[29] D. Lee, S. Choi, and H.-J. Kim, "Performance evaluation of image denoising developed using convolutional denoising autoencoders in chest radiography," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 884, pp. 97–104, 2018.

[30] W. Mathworks, "System toolbox," 2017.

[31] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[32] F. Chollet *et al.*, "Keras," 2015.

[33] J. Medbo, "Channel models for hiperlan/2 in different indoor scenarios," *ETSI/BRAN*, 1998.

[34] S. Banerji and R. S. Chowdhury, "On ieee 802.11: Wireless lan technology," *arXiv preprint arXiv:1307.2661*, 2013.

[35] "Wi-fi: Overview of the 802.11 physical layer and transmitter measurements," http://download.tek.com/document/37W_29687_0_press.pdf, accessed: 2018-09-09.

[36] V. D. Khairnar and K. Kotecha, "Performance of vehicle-to-vehicle communication using ieee 802.11 p in vehicular ad-hoc network environment," *arXiv preprint arXiv:1304.3357*, 2013.

[37] A. Mertins and D. A. Mertins, *Signal analysis: wavelets, filter banks, time-frequency transforms and applications*. John Wiley & Sons, Inc., 1999.

[38] J. Shore and R. Johnson, "Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy," *IEEE Transactions on information theory*, vol. 26, no. 1, pp. 26–37, 1980.

[39] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.