JS 20

FUTURE NETWORKS: 5G AND BEYOND

A light neural network for modulation detection under impairments

Thomas Courtat*†[1] ***and Hélion du Mas des Bourboux***‡[1]

[1] *Thales SIX Theresis, 1 av. Augustin Fresnel, 91120 Palaiseau*

Keywords: Machine learning, deep learning, modulation recognition, propagation channel, data augmentation

Abstract:
We present a neural network architecture able to efficiently detect modulation techniques in a portion of I/Q signals. This network is lighter by up to two orders of magnitude than other architectures working on the same or similar tasks. Moreover, the number of parameters does not depend on the signal duration, which allows processing stream of data, and results in a signal-length invariant network. In addition, we develop a custom simulator able to model the different impairments the propagation channel and the demodulator can bring to the recorded I/Q signal: random phase shifts, delays, roll-off, sampling rates, and frequency offsets. We benefit from this data set to train our neural network to be invariant to impairments and quantify its accuracy at disentangling between modulations under realistic real-life conditions.

1 Introduction

During the last few years, a lot of effort has been put into applying the performances of machine learning to the physical layer of radio transmission. Toward this goal, multiple directions are investigated; of interest in this study is modulation classification through supervised learning [1].
A large step in this direction was accomplished by [1] through the publication of a public data set for radio modulation classification, along with an artificial neural network architecture. Following this release multiple publications presented neural networks and analyses with respect to this data set, e.g. [2, 3, 4, 5, 6, 7, 8, 9]. [10] showed that machine learning (ML) based modulation classifiers already outperform traditional techniques based on higher order statistics. On the other hand [11] showed that even though ML based classifiers give better results, they can be less robust to data with impairments not present in the training set. This outlines the need to feed realistic and complete data sets to machine learning algorithms.
This study presents a novel neural network architecture that outperforms existing ones in the modulation recognition task. It is lighter than previously published networks [1, 10] and is built to be invariant under signal duration. We also develop a synthetic data set generator that allows to better control the sets of impairments and better understand their effects on the accuracy of our classifier.
This article is organized as follows. Section 2 presents all the data sets, either publicly available or developed here. Then, section 3 details the three state of the art architectures and the two developed for this study. They are compared in section 4 with respect to their accuracy at classifying modulations and we show that our network outperforms the others, while being two to ten times lighter. We study the performances of our architecture under variation of the signal length, and under frequency shifts in section 5. We present a conclusion in section 6.

2 Data sets

The industry standard data set, and its following updates, for modulation classification in radio is given by [12, 1, 13]. The first release, `RadioML2016.04C` (ML: Machine Learning), is composed of 11 modulations: 8PSK, AM-DSB, AM-SSB, BPSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK, and WBFM, with 20 evenly spaced bins in signal-to-noise ratio (SNR), ranging from -20 to 18 dB. The set is composed of 162,060 examples, consisting in 128 samples of I/Q (in-phase/quadrature) signals. The simulated synthetic data were produced using software defined radio programmed with GNU radio [14].
Three releases expanded and completed the set. `RadioML2016.10A` expanded to 220,000 the number of examples. `RadioML2016.10B` provides 1,200,000 examples on the same grid of SNR, but removes the AM-SSB modulation, leading to a total of 10 classes. `RadioML2018.01A` [10] provides a total of 2,555,904 examples, 1024 samples long, with a signal-to-noise ratio ranging from -20 up to 30 dB. Along with synthetic data, this set provides

*These authors contributed equally to this work.
†thomas.courtat'at'thalesgroup.com
‡helion.dumasdesbourboux'at'thalesgroup.com

Impairment	Range
$T_{\text{sample}}/T_{\text{symbol}}$	$[0.3, 0.5]$
Phase	$[0, 2\pi]$
Delay	$[0, 1]$
Roll off	$[0.1, 0.5]$
SNR	$\{0, 10, 20, 30, 40\}$
Relative frequency offset, Δf	$\pm[10^{-6}, 5 \times 10^{-1}]$

Table 1 – Set of impairments simulated in the `AugMod` *synthetic data set, developed in this study.*

Number of signal samples	RML-ConvNet	RML-CNN/VGG	RML-ResNet	Mod-LCNN (ours)	Mod-LRCNN (ours)
128	2,829,399	199,111	179,303	37,487	97,663
1024	21,179,479	256,455	236,647	37,487	97,663

Table 2 – Number of parameters of the five different networks for signals with 128 *or* 1024 *samples. For this table we choose* 7 *output classes, slightly different number of classes do not yield significant changes in the order of magnitude of the number of parameters.*

radio signals propagated through real indoor environment, transmitted and received via two universal software radio peripherals (USRP). This former data set expands to 24 the number of different modulations classes: 32PSK, 16APSK, 32QAM, FM, GMSK, 32APSK, OQPSK, 8ASK, BPSK, 8PSK, AM-SSB-SC, 4ASK, 16PSK, 64APSK, 128QAM, 128APSK, AM-DSB-SC, AM-SSB-WC, 64QAM, QPSK, 256QAM, AM-DSB-WC, OOK, and 16QAM. For simplicity, in this study we limit the data sets to positive SNR, we verify nonetheless that similar results are obtained on the whole range. All of these data sets are publicly available[1].

In order to independently study the performances of machine learning in modulation classification, we develop a synthetic custom data set. In addition, this module allows us to tune different parameters which are fixed or unknown in the previously defined data sets. As a consequence it allows us to study its robustness against parameters, while improving upon it. Seven linear modulations are simulated: BPSK, QPSK, PSK8, QAM8, QAM16, QAM32, and QAM64, with 5 evenly spaced bins of SNR from 0 to 40 dB. We generate 175,000 examples, i.e. 5000 per (SNR, modulation) pairs. The I/Q signal is produced for 1024 samples. A vast range of impairments brought by the propagation channel and the demodulator are added to the baseline data set: random phase shifts, delays, roll-off, sampling rates, and additive Gaussian noise. We also produce an additional data set enhanced with an extra impairment: relative frequency offsets. This allows us to better study its individual impact on modulation classification. Hereafter, we refer to this data set as `AugMod`, for "augmented modulation" data set. The range of the parameters are given in table 1.

As a result we benefit from five different data sets, with positive SNR, with both synthetic and indoor-propagated signals, to perform modulation classification under impairments. The first four data sets are public: `RadioML2016.04C` has 81,030 examples, `RadioML2016.10A` has 110,000 examples, `RadioML2016.10B` has 600,000 examples, and `RadioML2018.01A` has 1,572,864 examples. The fifth data set is private: `AugMod` with 175,000 examples. Each data set is split into two halves, one for training and the other for testing. Each individual signal is normalized by its root mean square, to have a power of 1.

3 Neural network architectures

Along with the available data set, [1] presents a convolutional neural network (ConvNet [15]) performing modulation classification, hereafter referred as "RML-ConvNet" (RML: Radio Machine Learning). This network treats the complex I/Q signal as a two-dimensional image, with a single "color" channel. As it is presented, this network has 2,829,399 parameters, when the I/Q signal has 128 samples and the data set has 7 different classes. The architecture is not invariant with the number of samples; this imposes to train a different network for every possible length of the input signal. Furthermore, a signal given with 1024 samples would multiply the number of parameters by approximately one order of magnitude, compared to the one for 128. This aspect produces a hardly scalable architecture for longer signals. Table 2 gives for two different length of signals, 128 and 1024, the number of parameters, or weights, of the neural networks studied here.

In a more recent release of their work, [10] presented an updated data set, `RadioML2018.01A`, with 1024 samples long signals. They also developed two extra neural networks: "RML-CNN/VGG" and "RML-ResNet". The first network builds upon the already developed RML-ConvNet network, but limits the explosion of the

[1] `https://www.deepsig.io/datasets`

number of parameters at 1024 samples through a VGG network (Visual Geometry Group [16]). It is modified to fit a 1-dimensional convolutional neural network (CNN). The second network has a residual architecture (ResNet [17]). ResNet has historically been invented to be easier to train for deep neural networks. Although both of these networks have less parameters than RML-ConvNet, as shown on table 2, they still suffer from the augmentation of the number of parameters with the signal length. For example, going from 128 to 1024 samples adds 30% more parameters. Because of this aspect, they lack the ability to adapt to signals of different sizes and, as for RML-ConvNet, must be re-trained for each signal length.

We propose a lighter convolutional neural network to perform modulation classification, invariant of the input signal length: Light Modulation Convolutional Neural Network, "Mod-LCNN". The complex I/Q signal is treated as a one-dimensional signal with two channels. These channels are expanded to higher dimension space through consecutive 1-dimensional convolutional layers. Then through an average pooling layer, the time dimension is collapsed to produce a one-dimensional layer of dimension that of the last convolutional layer, which is fed into a fully connected layer, and a softmax [18] layer to perform classification. Each convolutional layer of kernel size 7, along with the first fully connected layer, are followed by the rectified linear unit (ReLU [19]) activation function. During training, we apply dropout [20] to the output weights of the first fully connected layer, thus preventing overfitting.

We develop two different networks: "Mod-LCNN" and "Mod-LRCNN". Both are presented on figure 1. These two networks have the structure presented above, they differ in how each convolutional layer is applied. In the case of Mod-LCNN (top panel), we use a regular CNN, Mod-LRCNN (bottom panel) is a ResNet [17]. As a consequence each convolution step is split into three simple convolutions. The first one has a kernel size 1, allowing to expand the filter dimension [21], the two following convolutions have a kernel size of 7. The output of these last two consecutive convolutions is added to the output of the first one, through a skip connection (see figure 1).

For these two networks, the number of parameters does not depend on the signal duration. The consequence of this design is that the same trained network can be used for signals of different lengths. The resulting networks have 37,487 parameters for Mod-LCNN and 97,663 for Mod-LRCNN (table 2). As shown on figure 1, these two networks can be modeled as two blocks. The first one is a "latent space embedding", i.e. it extracts latent features of the signal, invariant of its length. The second block is a fully connected network that performs the "classification". The average pooling layer serves thus as a bottleneck between these two blocks.

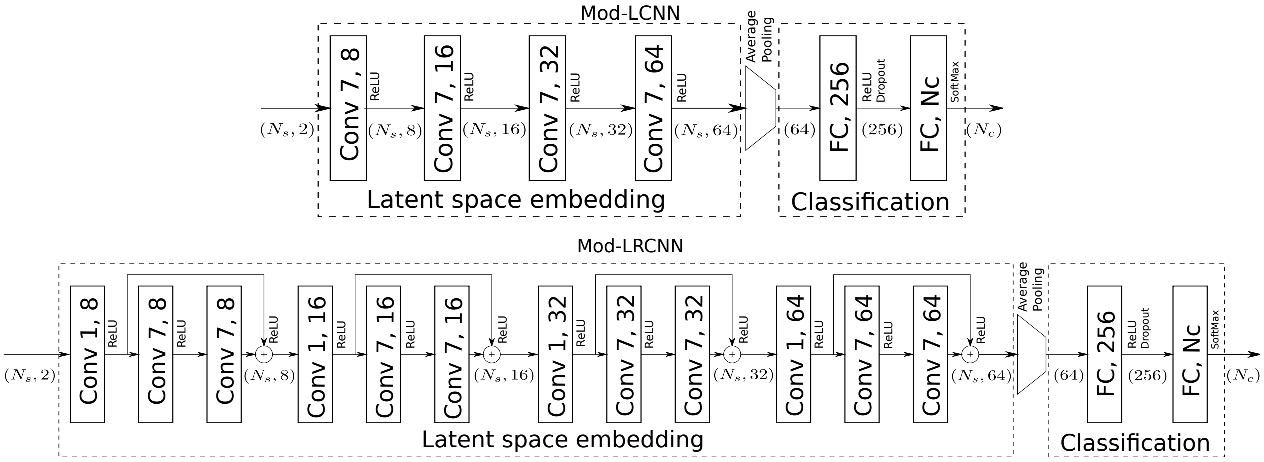

Figure 1 – Architecture of Mod-LCNN (top) and Mod-LRCNN (bottom), the neural networks developed in this study. N_s is the number of samples: 128 *or* 1024 *in this study, N_c is the number of output classes:* 7*,* 10*,* 11 *or* 24 *in this study. The 1-dimensional convolutions have a kernel size of* 7*. During training the dropout rate is* 0.5.

4 Comparison of the different architectures

We benefit from the five different data sets presented in section 2 to train and compare the five artificial neural networks of section 3. RML-ConvNet implementation is publicly provided by the author[2], in Keras [22], with TensorFlow backend [23], we thus use the same framework for all the other network architectures. Following the publicly available implementation of RML-ConvNet, we initialize all weights using the "Glorot" uniform initializer [24] for convolutional layers and through "He" normal initializer [25] for fully connected layers. The training is ran on a Nvidia 1080 Ti.

[2] `https://github.com/radioML/examples`

Data set	RML-ConvNet	RML-CNN/VGG	RML-ResNet	Mod-LCNN (ours)	Mod-LRCNN (ours)
128 samples					
`RadioML2016.04C`	93	93	95	93	**95**
`RadioML2016.10A`	84	83	90	90	**91**
`RadioML2016.10B`	89	91	93	93	**93**
`RadioML2018.01A`	50	70	76	68	**78**
`AugMod` (ours)	64	60	71	75	**75**
1024 samples					
`RadioML2018.01A`	61	87	88	85	**89**
`AugMod` (ours)	74	76	83	89	**89**

Table 3 – Accuracy of the five different neural network architectures on the different data sets. The performances are given for a signal of size 128 *for all data sets, and for* 1024 *samples when available. Boldface texts highlight the best results for each data sets.*

The neural network weights are learned using the training set through the Adam optimizer [26], to minimize the categorical cross-entropy loss function. Among the five data sets, two have 1024 samples long signals: `RadioML2018.01A` and `AugMod`. We train the networks on these two data sets twice: once on the full signal duration, and another time keeping only the first 128 samples. The training is performed for 200 iterations, or epochs, through each data set with a batch size of 512 examples. Because of computation time, all networks are trained for only 50 iterations for `RadioML2018.01A`, when using the full 1024 samples long signals.

Table 3 presents the accuracy on the test sets for the five data sets, over the five different neural networks. The accuracy, in percent, is given by the proportion of correctly assigned modulations on the test set, after the end of training. Boldface texts highlight the best results for each data set. All networks perform relatively equally well on `RadioML2016.04C` and on `RadioML2016.10B`. RML-ConvNet and RML-CNN/VGG do not manage to reach as good performances as other networks on other data sets. This is explained by the too large number of parameters for the first network, resulting in overfitting the training set. For the second network this is explained by the depth of the network, preventing the gradient updates to efficiently propagate through the network.

We confirm the results noted by [10], RML-ResNet gives indeed the best performances over all the data sets, when compared to RML-ConvNet and RML-CNN/VGG. Mod-LCNN, developed in this study, outperforms RML-ResNet when testing on the `AugMod` data set, however it fails at giving good results on `RadioML2018.01A`. This can be interpreted by the too small number of parameters. Adding more layers would reduce the performances by producing a too deep architecture, harder to train. Mod-LRCNN manages to outperforms all the other networks, building on Mod-LCNN performances, but adding a residual network architecture. It increases the accuracy by up to 2% on `RadioML2018.01A`, with 128 samples, and by up to 6% on `AugMod`, with 1024 samples.

Figure 2 presents the learning curves, i.e. the error rate as a function of the number of epochs, for all the networks, on the `AugMod` data set, with 1024 samples. Unbroken curves give the results on the test sets, and dotted curves on the training sets. This figure outlines the advantages of the Mod-LRCNN architecture: it outperforms other architectures with the lowest error rate, converges faster and continuously, and is less prone to overfitting.

We compare the performances of each neural network at classifying modulations, on the `AugMod` data set, as a function of signal-to-noise ratio. The results are presented on the left panel of figure 3. The panel gives the error rate as a function of SNR. Mod-LRCNN, developed for this study, performs more than 40% better than the best architecture of previous studies, RML-ResNet, at $SNR = 0$. In the $SNR \in [0, 30]$ range, Mod-LRCNN effectively improves the performances by ~ 5 dB.

We assess the training time by looking at the time per epoch when running on the `AugMod` data set, with 1024 samples. Other data sets give similar results. Mod-LRCNN runs in 3.1 ms per example, resulting in 27 seconds per epoch, with 512 examples per batch, for a total training time of 1.5 hours with 200 epochs. Mod-LCNN and RML-CNN/VGG are twice as fast, however, RML-ResNet is 1.25 times longer. Finally RML-ConvNet runs in twice as long due to the large number of parameters (table 2). The fact that Mod-LRCNN runs each epoch in twice the time compared to Mod-LCNN is balanced by both its higher accuracy (table 3) and the fact that less epochs are needed to converge (figure 2).

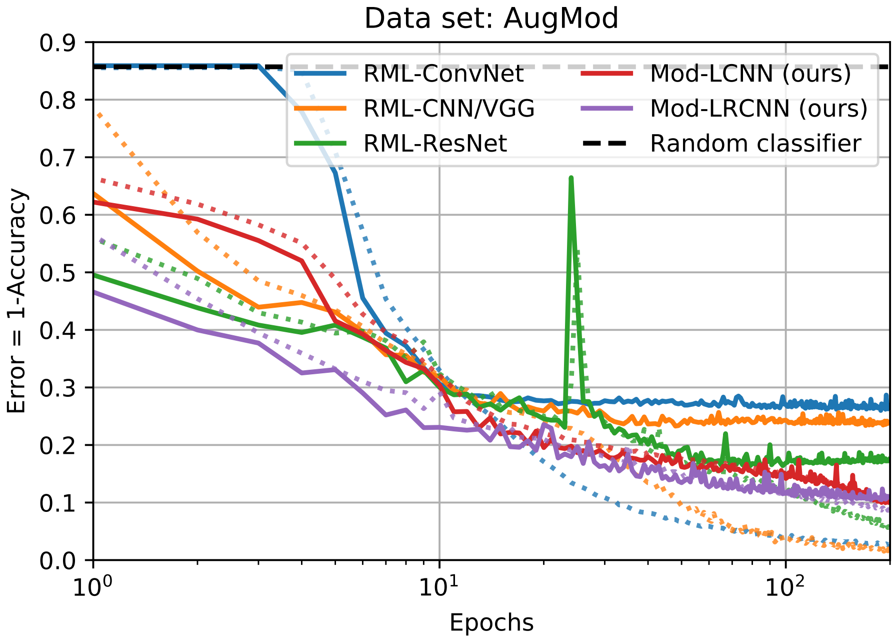

Figure 2 – Error rate as a function of the number of epochs for the five different neural network architectures, compared with the performances of a random classifier. Solid curves are for the test set and dotted curves for the training set. The comparison is performed with the `AugMod` data set on 1024 *samples long signals.*

5 Specific performances of Mod-LRCNN

As discussed previously in section 4, the Mod-LRCNN architecture outperforms all other architectures in accuracy. We investigate in this section its performances on different signal lengths, and under different sets of impairments.

5.1 Signal duration

Mod-LCNN and Mod-LRCNN's strength are their invariance under the signal duration. This means that once the network has been trained, it can be used to infer the signal modulation, whatever its length. We test this property on three different training strategies for Mod-LRCNN. The following results are given through the implementation of Mod-LRCNN in PyTorch [2]. This choice gives us more flexibility during training.

The right panel of figure 3 presents the classification error rate as a function of the signal length. These results are given on the `AugMod` data set. The first strategy is to train Mod-LRCNN on 128 samples long signals. The second strategy is to train on 1024 samples long signals. On the test set, we limit each example to the first $\{16, 32, 64, 128, 256, 512, 1024\}$ samples, infer the modulation class, and give the resulting error rate.

In the right panel of figure 3 the blue dashed curve presents the results for the first strategy, and the orange for the second. One could have expected Mod-LRCNN trained on 1024 to outperform the first strategy on the full range. It is the case for signals more than 256 samples long, however it is not the case bellow. This indicates a tendency of Mod-LRCNN, trained on 1024, to overfit long signals.

We develop a third strategy where we modify dynamically the length of the signal during training. At each batch iteration we randomly pick an integer $N_s \in [16, 1024]$, and limit the signal duration to the first N_s samples. The resulting accuracy on the test set is given in the green unbroken curve. We observe that indeed this new training scheme allows to get good performances for short and long signals.

5.2 Frequency shift

The `AugMod` synthetic data set is reproduced adding a relative frequency offset (table 1) on top of the other baseline impairments. We span a wide range of values, from positive and negative 50% of the carrier frequency, with a logarithmic scale: $\Delta f \in \pm[10^{-6}, 5{\times}10^{-1}]$. The effect of this latter impairment is to drift the constellations into circular patterns with a typical time scale $\Delta f \cdot T_{sample}$. The results are presented on the left panel of figure 4, for Mod-LRCNN trained on the `AugMod` data set, with 1024 samples long signals.

In this figure, the unbroken blue curve gives the result when Mod-LRCNN is trained on the `AugMod` data set without the frequency shift impairment, with variable length of signals (sec. 5.1). This curve thus displays the

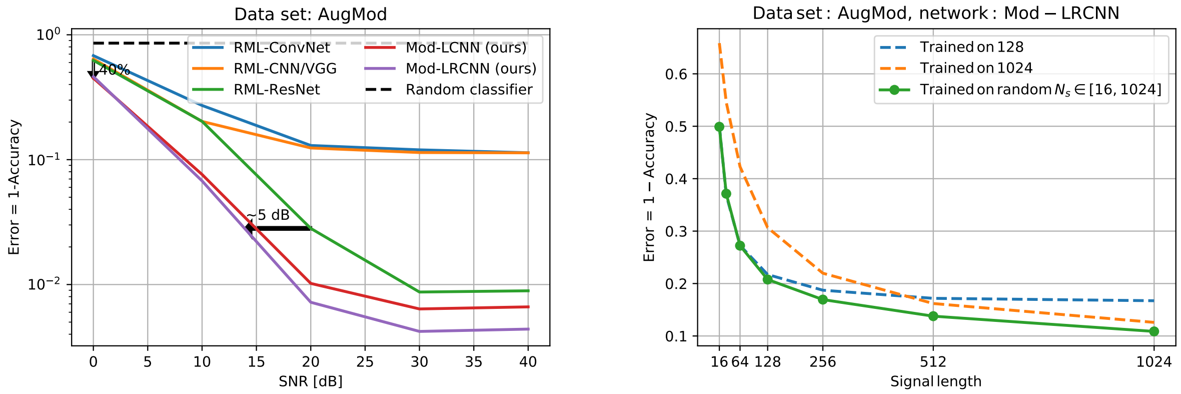

Figure 3 – ***Left:*** *Error rate as a function of the signal-to-noise ratio for the five different networks. The performances are given for the* `AugMod` *data set, with* 1024 *samples long signals.* ***Right:*** *Error rate of the Mod-LRCNN architecture, developed for this study, on the* `AugMod` *data set as a function of the signal length. The blue dashed curve gives the performances for a model trained on* 128 *samples long signals, the orange dashed curve for a model trained on* 1024*, and the green unbroken curve for a training with signals of dynamically random sizes* $N_s \in [16, 1024]$.

ability of the network to generalize to out of distribution example signals. The dotted blue curve presents the same results, but for a training with fixed 1024 samples long signals. We observe that the accuracy starts to drop at $|\Delta f| = 10^{-4}$ and falls out at 10^{-2}. This behavior is even more drastic when the network is trained on fix 1024 samples long signals. This later behavior confirms the tendency of networks trained on fixed size signals to overfit long signals and thus be less robust to time varying impairments.
The orange curves show the accuracy on the test set when Mod-LRCNN is trained on half of the `AugMod` data set, impaired with frequency shifts. We recover good performances at large frequency shifts. Following the methods of curriculum learning [27], only few epochs are needed to perform this re-training, if the weights are initialized to the best values found when trained on the simpler `Augmod` data set.

6 Conclusion

This study presented an artificial neural network architecture allowing to classify modulations: the light residual convolutional neural network for modulation classification, “Mod-LRCNN”. This architecture is lighter than previously published networks. Its architecture is invariant to the signal length, allowing it to adapt perfectly to signals recorded on more or less samples, without a need for re-training. The network is designed to search for the natural symmetries of the signals, extract latent features and use them to classify modulations. It simply builds statistical significance with the signal duration, and thus can process data stream.
It performs better than three public networks [1, 10] on all four publicly available data sets, e.g. `RadioML2018.01A`, and on a custom made data set, `AugMod`. It is defined by up to two orders of magnitude less parameters. In the SNR $\in [0, 30]$ range, Mod-LRCNN effectively improves the threshold by ~ 5 dB (up to 10 dB) compared to previously published networks.
We characterize some of the performances of the network. When trained on dynamically changing examples lengths, between 16 to 1024 samples, the network is able to give very good accuracy whatever the inferred signal lengths. This training technique prevents overfitting long signals, and thus gives good performances on evolving impairments, e.g. frequency shift. We show the ability of the network to efficiently classify signals under frequency shift impairment, even when they are out of the distribution given in the training set. Even better performances can be obtained through curriculum learning, by training the network in few epochs, if the weights are initialized at their values for the simpler data set.
The data set introduced in this study has allowed us to train our network to create signal representation invariant to real life impairments. We aim at adding more complexity to this set, e.g. non-linear modulations, multi-path propagation, and test the network under more real indoor and outdoor propagated signals.

7 References

[1] T. J. O’Shea, J. Corgan, and T. C. Clancy, “Convolutional radio modulation recognition networks,” in *Engineering Applications of Neural Networks* (C. Jayne and L. Iliadis, eds.), (Cham), pp. 213–226, Springer International Publishing, 2016.

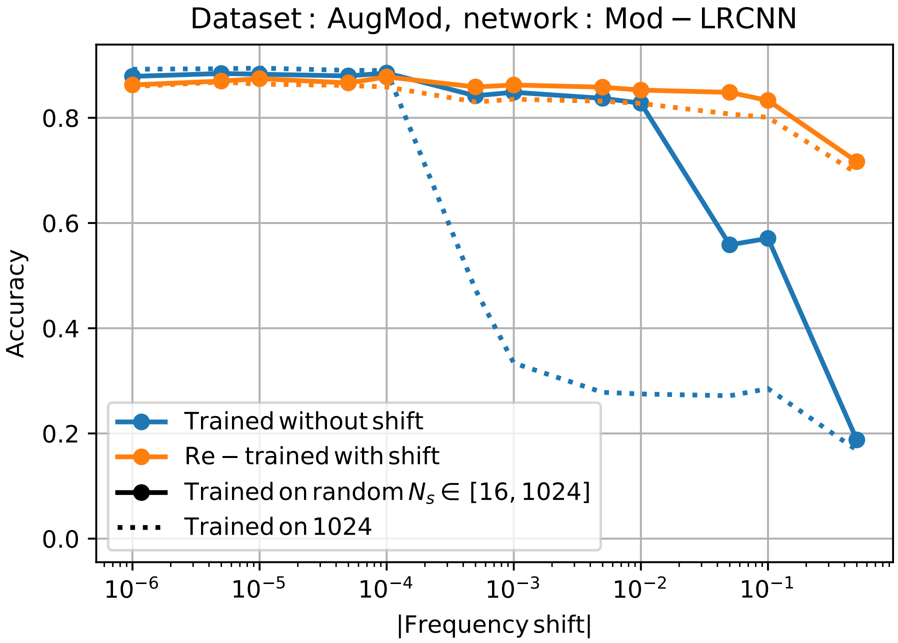

Figure 4 – Accuracy of the Mod-LRCNN architecture, developed for this study, on the `AugMod` *data set, with* 1024 *samples long signals, enhanced with frequency shift impairments: in blue the results for the network trained on a data set without carrier shift, and in orange for a re-training on the data set including it. Unbroken curves are for a training with variable random signal lengths,* $N_s \in [16, 1024]$*, and dotted curves for training on fixed* 1024 *samples long signals.*

[2] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.

[3] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "Oracle: Optimized radio classification through convolutional neural networks," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 370–378, April 2019.

[4] M. Sadeghi and E. G. Larsson, "Adversarial attacks on deep-learning based radio signal classification," *IEEE Wireless Communications Letters*, vol. 8, pp. 213–216, Feb 2019.

[5] S. Ramjee, S. Ju, D. Yang, X. Liu, A. El Gamal, and Y. C. Eldar, "Fast Deep Learning for Automatic Modulation Classification," *arXiv e-prints*, p. arXiv:1901.05850, Jan 2019.

[6] L. Huang, W. Pan, Y. Zhang, L. Qian, N. Gao, and Y. Wu, "Data augmentation for deep learning-based radio modulation classification," *IEEE Access*, vol. 8, pp. 1498–1506, 2020.

[7] Y. Shi, K. Davaslioglu, Y. E. Sagduyu, W. C. Headley, M. Fowler, and G. Green, "Deep learning for rf signal classification in unknown and dynamic spectrum environments," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–10, IEEE, 2019.

[8] K. Tekbıyık, A. Rıza Ekti, A. Görçin, G. Karabulut Kurt, and C. Keçeci, "Robust and Fast Automatic Modulation Classification with CNN under Multipath Fading Channels," *arXiv e-prints*, p. arXiv:1911.04970, Nov 2019.

[9] C.-F. Teng, C.-Y. Chou, C.-H. Chen, and A.-Y. Wu, "Accumulated Polar Feature based Deep Learning with Channel Compensation Mechanism for Efficient Automatic Modulation Classification under Time varying Channels," *arXiv e-prints*, p. arXiv:2001.01395, Jan 2020.

[10] 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, pp. 168–179, Feb 2018.

[11] B. Luo, Q. Peng, P. C. Cosman, and L. B. Milstein, "Robustness of deep modulation recognition under awgn and rician fading," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pp. 447–450, Oct 2018.

[12] T. O'Shea and N. West, "Radio machine learning dataset generation with gnu radio," *Proceedings of the GNU Radio Conference*, vol. 1, no. 1, 2016.

[13] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Unsupervised representation learning of structured radio communication signals," in *2016 First International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE)*, pp. 1–5, IEEE, 2016.

[14] E. Blossom, "Gnu radio: Tools for exploring the radio frequency spectrum," *Linux J.*, vol. 2004, pp. 4–, June 2004.

[15] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Proceedings of the 2Nd International Conference on Neural Information Processing Systems*, NIPS'89, (Cambridge, MA, USA), pp. 396–404, MIT Press, 1989.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, June 2016.

[18] J. S. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," in *Advances in Neural Information Processing Systems 2* (D. S. Touretzky, ed.), pp. 211–217, Morgan-Kaufmann, 1990.

[19] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (G. Gordon, D. Dunson, and M. Dudík, eds.), vol. 15 of *Proceedings of Machine Learning Research*, (Fort Lauderdale, FL, USA), pp. 315–323, PMLR, 11–13 Apr 2011.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[21] M. Lin, Q. Chen, and S. Yan, "Network In Network," *arXiv e-prints*, p. arXiv:1312.4400, Dec 2013.

[22] F. Chollet *et al.*, "Keras." `https://keras.io`, 2015.

[23] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[24] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (Y. W. Teh and M. Titterington, eds.), vol. 9 of *Proceedings of Machine Learning Research*, (Chia Laguna Resort, Sardinia, Italy), pp. 249–256, PMLR, 13–15 May 2010.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV 15, (USA), p. 10261034, IEEE Computer Society, 2015.

[26] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv e-prints*, p. arXiv:1412.6980, Dec 2014.

[27] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML 09, (New York, NY, USA), p. 4148, Association for Computing Machinery, 2009.