Spectrum Awareness Under Co-Channel Usage Via Deep Temporal Convolutional Networks

Amir Ghasemi[*], Chaitanya Parekh, Paul Guinand
Communications Research Centre, Ottawa, Ontario, Canada
[*]Corresponding Email: firstname.lastname@canada.ca

Abstract—**Modulation recognition is a key component of spectrum environment awareness enabling radios to share the spectrum more effectively and allowing the regulators to monitor compliance and identify rogue users.**

In this paper we consider the problem of modulation recognition in circumstances where there is a co-channel signal. Both the problem where the type of the co-channel signal is known and the problem of modulation recognition for both of two unknown interfering signals are addressed. We use a temporal convolutional neural network which, for the case of a known co-channel signal type achieves a classification accuracy in excess of 80% at a signal-to-interference ratio (SIR) of 0dB. The modulation recognition of co-channel interfering signals in noise is of particular interest for spectrum regulation and compliance. We also show the efficacy of this architecture for the classical case of modulation recognition of a single signal. In this case, we can achieve 100% classification accuracy for signal-to-noise ratio (SNR) levels greater than 0dB.

I. INTRODUCTION

With the increasing demand for wireless data access and limited supply of radio spectrum, it is expected that in the future multiple systems will have to coexist in the same wireless channel. This in turn requires robust coexistence mechanisms to detect other users and avoid interfering with them. Identifying the modulation scheme of received signal(s) is a technique that might prove useful in such scenario. Historically, modulation recognition has received considerable attention due to its various applications in civilian and military communications [1]. Most of the techniques proposed over the years and deployed in the field today rely on specific features of transmitted signals which are derived analytically by domain experts. The analytical formulations and decision criteria in such cases are typically dependent on the specific assumptions made about the radio configuration, propagation characteristics, and receiver impairments, and deviations in any of these may either degrade the algorithm performance or require the analysis to be repeated. Furthermore, there are situations in practice where analytical solutions may be cumbersome (if not impossible) to derive due to the complexity of channel or transceiver impairments.

Over the past few years, deep neural networks have proven successful in learning effective representations of data in domains such as computer vision, thereby removing the need for feature engineering by experts. Motivated by this observation, researchers have recently shown that deep learning can achieve state-of-the-art performance in modulation classification, without any expert feature engineering, by directly processing the raw received I/Q samples [2], [3].

In this work we build upon the results of [2], [3] by leveraging recent advances in deep learning. More specifically, we use a deep temporal convolutional network architecture to classify the modulation of digital radio signals which shows an improved performance over the convolutional networks used in prior work. We further show that the same network can be trained to identify the modulation schemes of co-channel interfering signals in noise with applications in spectrum regulation and compliance (interference/jammer identification) and spectrum sharing (coexistence with other users).

The remainder of this paper is structured as follows. Section II describes the dataset used in this study. Section III details the deep learning architecture used for modulation classification while the results are presented in Section IV. Finally, concluding remarks are provided in Section V.

II. DATA PREPARATION

For our analysis, we build upon the publicly available RadioML dataset described in [4] which contains raw baseband I/Q (time-domain) vectors for several digital and analog modulations at various SNR levels, sampled at a rate of 8 samples/symbol. The synthesized data includes common radio communication impairments such as multi-path fading, additive white Gaussian noise (AWGN), frequency offset, and sample timing offset (see [4] for further details). To train our deep neural network, we re-synthesize this dataset using GNU Radio [5] to produce I/Q vectors with an increased length of 1024 samples (vs. the original 128) which, as we will show in Section IV, provides a higher classification accuracy.

For the single-signal case, for each modulation we generate baseband I/Q vectors at SNR levels ranging from -20 to 18dB in steps of 2dB. We synthesize 5000 vectors per SNR (100k total per modulation). For the scenario with co-channel signals, for each I/Q vector

we add a second signal with random phase (uniformly distributed between 0 and 2π). We synthesize the data at four SNR levels (-18,-6,6,18 dB) and five SIR levels (-10,-5,0,5,10 dB) again resulting in 100k I/Q vectors per class. Both SNR and SIR values are calculated with the first signal's power used as reference.

III. DEEP TEMPORAL CONVOLUTIONAL NETWORK

Deep recurrent neural networks (RNN) have been traditionally preferred over CNNs for machine learning tasks involving a sequence (e.g., machine translation, language modeling, and audio synthesis) [6]. Recently though, certain deep convolutional architectures such as WaveNet [7] and ByteNet [8] have shown state-of-the-art results in some sequence problems beating RNNs. Motivated by this observation, in recent work we successfully applied a deep *temporal* convolutional network (TCN), similar to the one defined in [9], for the problem of detecting the presence of radio signals in noise (spectrum sensing) using raw time-domain sequences of I/Q samples [10]. In this work, we leverage the same architecture for the modulation classification of co-channel radio signals. This is typically much more challenging than the binary classification problem tackled in [10] as we are interested in detecting not only the presence of signal(s) but also the exact modulation scheme being used.

A prominent feature of the TCN architecture is its use of dilated causal convolutions with increasing dilation factors to achieve a larger receptive field without requiring many layers. This is shown in Fig. 1 with three layers of dilated causal convolutions with dilation factors of $d = 1$, 2, and 4 (for $d = 1$, the dilated convolution reduces to a regular convolution).

Fig. 2 shows the overall architecture of the TCN which is formed by using K stacks, each containing L residual blocks with increasing dilation factors for the 1D causal convolutions. Within each residual block (Fig. 2a), the dilated convolution layer is followed by rectified linear unit (ReLU) activation, weight normalization for faster convergence [11], and dropout to help avoid overfitting [12]. Similar to the original WaveNet architecture, we use skip connections as shown in Fig. 2.

TABLE I
TCN PARAMETERS

Description	Value
No. Stacks	3
No. Filters	64
Kernel Size	8
Dilations	2, 4, 16, 256
Dropout	0.5
Optimizer	Adam
Batch Size	300

Table I summarizes the configuration used for training of the deep temporal convolutional network. We use
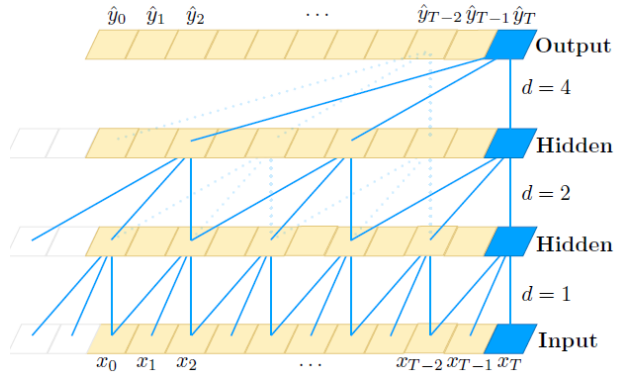

Fig. 1. Dilated causal convolutions with increasing dilation factors (denoted by d) effectively increase the receptive field of a convolutional network with less complexity [9]

three stacks as shown in Fig. 2 (K = 3) each containing four residual blocks (with dilation factors of 2, 4, 16, and 256, respectively). We also use 50% dropout within each residual block. Training is done using Adam optimizer with mini-batches of size 300.

IV. PERFORMANCE EVALUATION

As described in Section II, the dataset consists of 100k vectors of baseband I/Q vectors per class. We use 80% of the data for training, 10% for validation, and 10% for final testing of the trained model. We train the network on an NVIDIA Tesla V100 using the Adam optimizer, binary cross-entropy as the evaluation metric, and an early stopping patience of 10 epochs. The code uses Keras [13] with a Tensorflow [14] backend, with portions of the code derived from [15].

A. Single-Signal Classifier

First, we compare the performance of the proposed TCN architecture against the convolutional LSTM network (CLDNN) proposed in [3] for the task of identifying the modulation of a single signal in noise. We focus on the subset of 8 digitally modulated signals for which the CLDNN performs better. These include BPSK, QPSK, 8PSK, PAM4, QAM16, QAM64, CPFSK, and GFSK.

Fig. 3 shows the classification accuracy as a function of SNR for different architectures with the original sample length of 128 that was used in [3] (short-duration dataset). We compare the performance of CLDNN with two TCN architectures using different sets of dilation factors. We observe that all networks perform closely with the TCN-based classifiers having a bit of advantage at higher SNRs. It is also worth noting that increasing the receptive field of the TCN (via higher dilation factors) is not helping much in this case due to the relatively short duration of each I/Q vector which limits the effective kernel size of the convolution for high dilation factors.
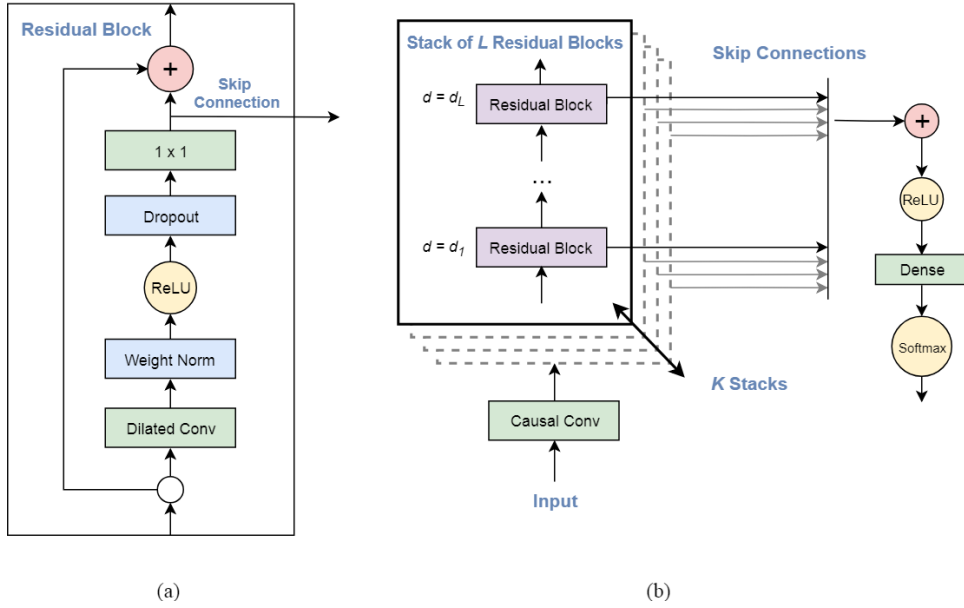
Fig. 2. (a) Configuration of a single residual block, (b) Overall TCN architecture with K stacks and L residual blocks per stack with d_i denoting the dilation factor of ith block
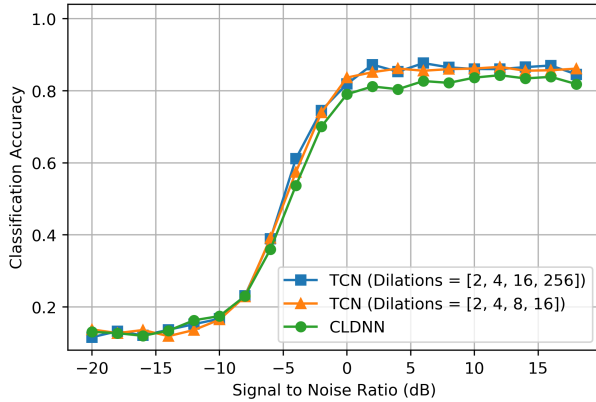

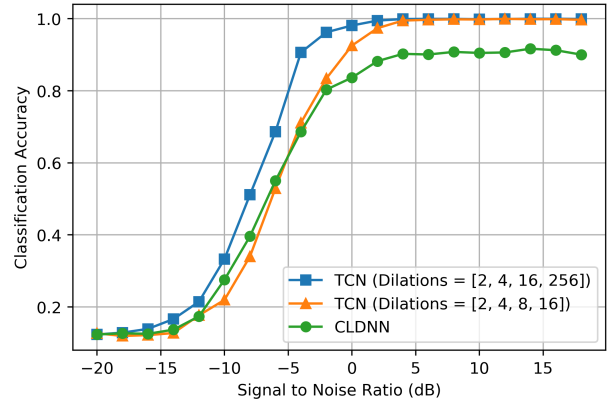Fig. 3. Single-signal classification accuracy vs. SNR for short-duration dataset

Fig. 4. Single-signal classification accuracy vs. SNR for long-duration dataset

For the longer-duration dataset with 1024 samples per vector, all architectures perform better while at the same time the performance gap between them starts to widen. Specifically, as shown in Fig. 4, the TCN architecture with dilation factors of (2,4,16,256) outperforms CLDNN at all SNR levels above -18dB, achieving 100% classification accuracy for SNRs greater than 0dB. Using a smaller receptive field for the TCN has no discernible impact at high SNRs but degrades the performance up to 4dB at SNRs below 2dB.

For the scenarios with co-channel signals that follow next, we only show results for the TCN with higher dilation factors as it provided the best performance on the longer-duration dataset.

B. Interference Classifier

We extend the analysis of single-signal modulation classification to the case where the modulation of the desired signal is known and fixed while there might be a potential interferer whose modulation needs to be identified. This scenario is of particular interest in the spectrum regulation and compliance context where there is a need to identify unauthorized use of a channel which is licensed to a specific user. We use the class label notation (a,b) to indicate I/Q vectors consisting of co-channel modulations a and b. It is worth emphasizing that the training is done with respect to a fixed type of incumbent user, the type of the fixed signal is not a direct input to the inference. Thus this represents an example of a classifier tuned to a specific impairment
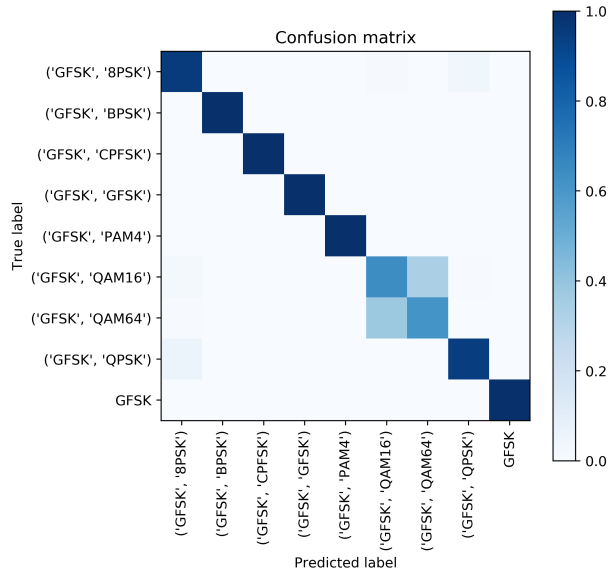
Fig. 5. Confusion matrix for interference classification across the full SIR range when the desired signal is GFSK modulated
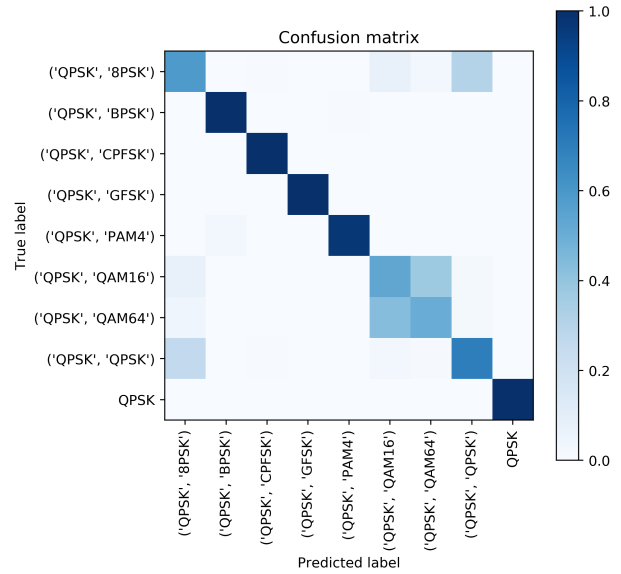

Fig. 6. Confusion matrix for interference classification across the full SIR range when the desired signal is QPSK modulated

that would be difficult to deal with analytically.

Fig. 5 and 6 show the confusion matrices of the TCN-based classifier for two different cases where the desired (known) signal uses GFSK and QPSK modulations, respectively. In each case, the performance is averaged over all five SIR levels with a fixed SNR of 10dB. To account for the situation with no interference, the final class in each dataset constitutes the signal of interest only (with channel impairments and receiver noise added). We use a total of 180k vectors (20k per class) with a 80/10/10 split for training, validation, and testing, respectively.

We observe that in both cases the interference modulation is classified with high accuracy. As expected, there is some confusion between QAM16 and QAM64 which might require more symbols to be distinguished from each other. When both signals are QPSK-modulated, there is also some confusion with 8PSK being detected as the interferer's modulation.

Fig. 7 shows the classification accuracy as a function of SIR. As intuitively expected, the classifier becomes more accurate at lower SIRs as it would be easier to identify a stronger interference source.

It is sometimes of interest to visualize what the deep neural networks are learning. One approach is to extract the (typically high-dimensional) activations of a given layer within the architecture and project it into two or three dimensions for visualization. There are several methods for achieving this with perhaps the most well-known being the t-SNE [16]. In this paper, we apply a recently-proposed technique known as Uniform Manifold Approximation and Projection (UMAP) [17] which has been shown to be faster than t-SNE while
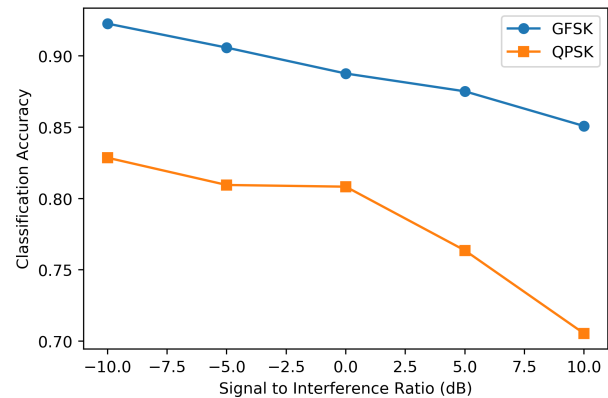

Fig. 7. Classification accuracy vs. SIR for different desired-signal modulations

producing competitive projections. In particular, UMAP projection tries to preserve structure of the data in the original higher-dimensional space as far as possible. This in turn allows one to interpret proximity of samples in the resulting 2D plane as similarity in the original higher-dimensional space.

We re-train the TCN with a dense (embedding) layer of size 32 added before its final output layer. This (slightly more complex) network reaches similar accuracy as the original while allowing us to peek inside the representations learned by the network. The 32-dimensional embedding vector is calculated for each of the samples in the test dataset and passed onto UMAP for projection into a 2D plane. Fig. 8 shows the 2D UMAP projection (with color-coded classes) for the case of desired signal having GFSK modulation. Inspecting
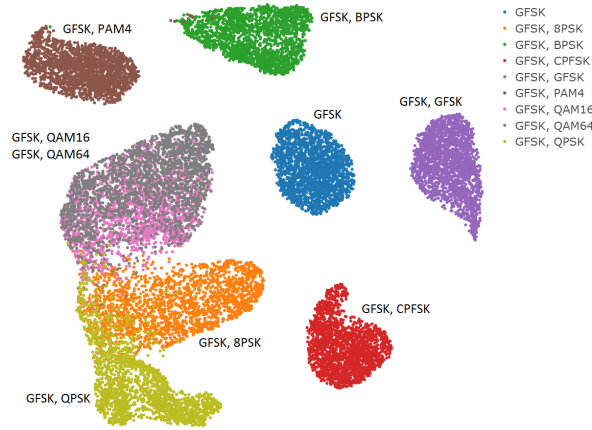
Fig. 8. 2D UMAP projection of TCN embeddings with GFSK-modulated desired signal
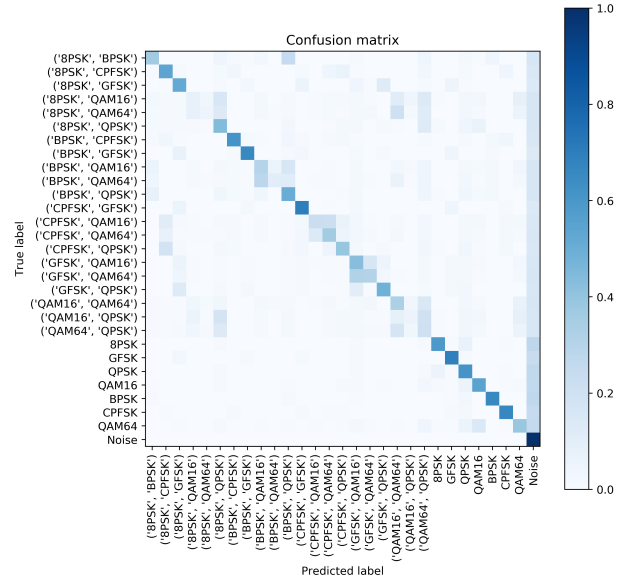

Fig. 9. Confusion matrix for mixed-signal classification across the full SNR and SIR range

Fig. 10. Confusion matrix for mixed-signal classification at SIR = -10dB and SNR = 18dB

overlapping classes in Fig. 8 shows the same patterns as the confusion matrix of Fig. 5. Moreover, we are able to visualize how the proposed deep neural network perceives different classes in terms of their similarity. For instance, in Fig. 8 a BPSK-modulated interference is most similar to one with PAM4 modulation. While the network is distinguishing these two classes well, one might expect them to be at a higher risk of confusion as the SIR is increased further.

C. Mixed-Signal Classifier

In Section IV-B we considered two co-channel signals where one of the signals' modulation was known and fixed. In what follows we analyze the performance of TCN for a co-channel scenario where neither of the two signals' modulation is *a priori* known. We consider all possible pairwise combinations of BPSK, QPSK, 8PSK, QAM16, QAM64, CPFSK, and GFSK (21 classes) as well as possibility of having a single signal (7 classes) or pure AWGN noise, resulting in a total of 29 classes.

The TCN network is trained, as before, using an Adam optimizer which converges after approximately 90 hours using a single NVIDIA Tesla V100. The confusion matrix averaged over all SNR and SIR values is depicted in Fig. 9 showing good overall performance with QAM16 and QAM64 being the main sources of confusion as before. Fig. 10 shows the confusion matrix for SIR and SNR of -10 and 18 dB, respectively. Comparing with the average confusion matrix in Fig. 9, we see much better overall classification performance. Specifically, single modulated signals and pure noise are almost perfectly classified. Classification accuracy of co-channel signals is also significantly improved, although there is still some residual confusion due to the QAMs. It is interesting to note that with a strong QPSK interference (SIR of -10dB), both QAM16 and QAM64 signals are often being classified as 8PSK.
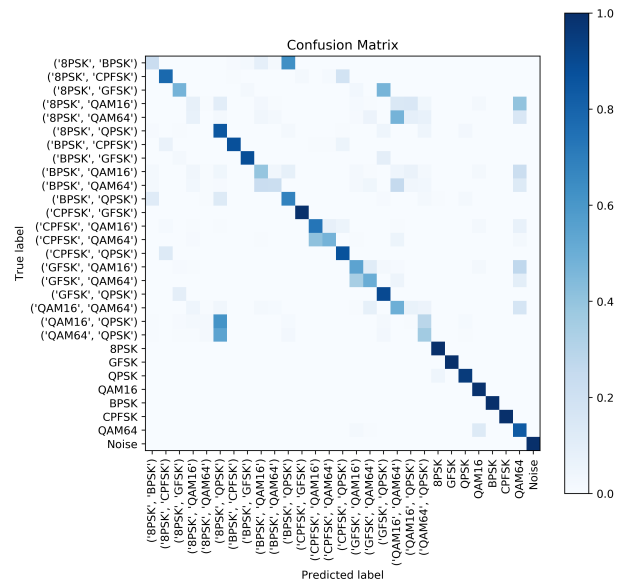
In the performance evaluations so far, we have been choosing the Softmax output index with the highest probability as the detected class. Rather than reporting the single class with the highest Softmax activation as the predicted class, sometimes one might be interested in *soft* prediction where all Softmax outputs are reported and treated as probabilities of individual classes. In this case, it is more appropriate to look at the distribution of true class label's rank among all soft predictions. This distribution is plotted in Fig. 11 showing that the probability of the true modulation pair being in the top-
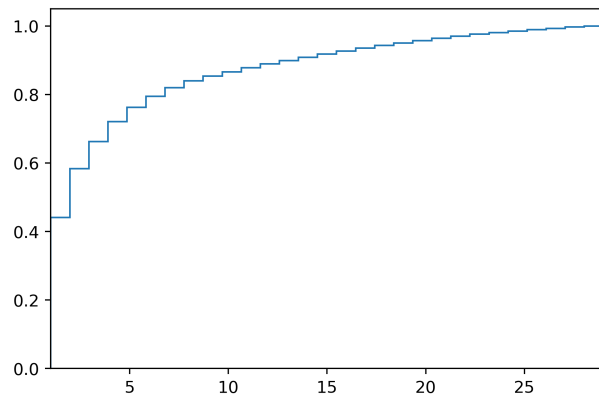
Fig. 11. Probability distribution of true label's rank among the predicted labels

1 and top-5 outputs is approximately 43% and 75%, respectively.

Finally, as in Section IV-B, we re-train the network with an embedding layer of size 32 added before the output layer. Fig. 12 shows the 2-dimensional UMAP projection of the 32-dimensional embedding vectors for test data at SNR and SIR of 18 and -10 dB, respectively. It is interesting to note that I/Q vectors corresponding to pure noise are clustered together and are surrounded by clusters of data points representing the single-signal classes. The data points corresponding to pairs of co-channel signals are placed further away from the noise and seem to be grouped together according to the modulation scheme of the interfering signal. Of note is also the GFSK modulation which, based on the TCN embeddings, shows the least proximity/similarity to other modulations.

V. CONCLUDING REMARKS

In this paper we have identified and described problems of modulation recognition in situations where there is co-channel interference as well as noise. These include the problem of identifying the modulation type of a signal interfering with a signal of a known type and the identification of modulation types for two unknown interfering signals. We also consider the classical problem of modulation recognition for a single signal in noise. For all of these problems we have shown the efficacy of a temporal convolutional network architecture as a classifier. For the classical case, we achieve essentially 100% accuracy at an SNR of 0dB. For the problem with two signals, one of which is of known type, we correctly identify the unknown second signal type more than 80% of the time at an SIR of 0dB. For the case of classifying two unknown signals, both signals were correctly classified 43% of the time averaging over a range of signal power ratios from -10 to 10 dB. Visualization techniques based on UMAP were used to

show the clustering of the signal types achieved by the temporal convolutional network.

A major motivating factor in the consideration of co-channel interference analysis described in this paper is its potential role in future spectrum management and regulation. If regulatory agencies are to manage the limited (and much in demand) spectrum resource more effectively, it is anticipated that they will increasingly need to understand what is actually happening in the spectrum both in frequency and geography. To do this will require the capability to make observations/measurements and to analyze the results and implications of those measurements. In particular, efficient geographical packing of channel assignments requires understanding of co-channel interference under specific circumstances. This research is, in part, intended to support this understanding. Also, a future data-driven regulatory regime is likely to require more active enforcement mechanisms that would be enhanced with better understanding of the co-channel activity situation. This would include issues around both interferers and jammers.

REFERENCES

[1] O. A. Dobre, A. Abdi, Y. Bar-Ness, W. Su, "Survey of Automatic Modulation Classification Techniques: Classical Approaches and New Trends," *IET Communications*, vol. 1, no. 2, pp. 137-156, 2007.

[2] T. J. O'Shea, J. Corgan, T. C. Clancy, "Convolutional Radio Modulation Recognition Networks," *in Proc. 17th International Conference on Engineering Applications of Neural Networks (EANN)*, Aberdeen, UK, Sept. 2016, available online: https://arxiv.org/pdf/1602.04105

[3] N. E. West, T. J. O'Shea, "Deep Architectures for Modulation Recognition," *in Proc. IEEE DySPAN 2017*.

[4] T. J. O'Shea, N. E. West, "Radio Machine Learning Dataset Generation With GNU Radio," *in Proc. of the GNU Radio Conference*, vol. 1, no. 1, 2016.

[5] GNU Radio, http://www.gnuradio.org.

[6] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2016.

[7] A. van den Oord *et al.*, "WaveNet: A Generative Model for Raw Audio," *9th ISCA Speech Synthesis Workshop*, Sunnyvale, CA, Sept. 2016, available online: https://arxiv.org/pdf/1609.03499

[8] N. Kalchbrenner *et al.*, "Neural Machine Translation in Linear Time," available online: https://arxiv.org/pdf/1610.10099

[9] S. Bai, J. Z. Kolter, V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," available online: https://arxiv.org/abs/1803.01271, April 2018.

[10] A. Ghasemi, C. Parekh, P. Guinand, "Spectrum Sensing for Modulated Radio Signals Using Deep Temporal Convolutional Networks," *IEEE WCNC Workshop on Smart Spectrum*, Morocco, April 2019.

[11] T. Salimans, D. P. Kingma, "Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks," *in Proc. NIPS 2016*.

[12] N. Srivastava *et al.*, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol 15, pp.1929-1958, 2014.

[13] F. Chollet *et al.*, "Keras: The Python Deep Learning Library," Software available online: https://keras.io

[14] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," Software available online: https://www.tensorflow.org/

[15] P. Rémy, Keras TCN, (2018), available at: https://github.com/philipperemy/keras-tcn
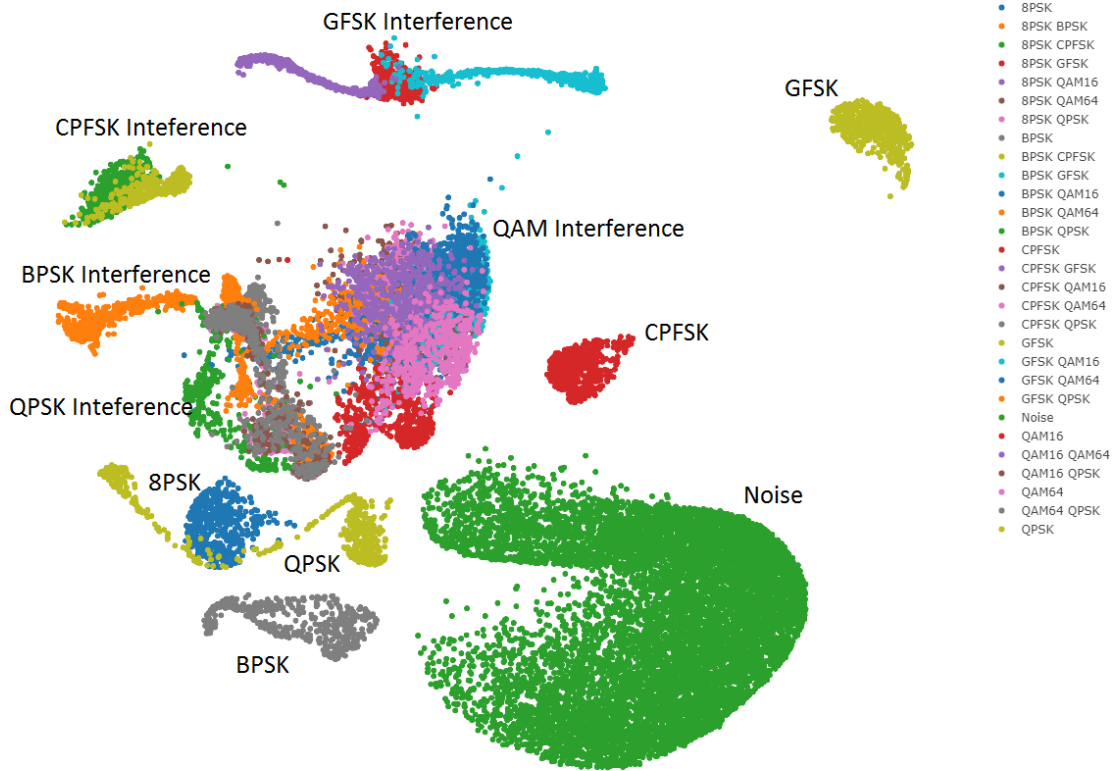
Fig. 12. 2-dimensional UMAP projection of the TCN embeddings for mixed-signal classification

[16] L. J. P. van der Maaten and G. E. Hinton, "Visualizing High-Dimensional Data Using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579-2605, November 2008.

[17] L. McInnes *et al.*, "UMAP: Uniform Manifold Approximation and Projection," *The Journal of Open Source Software*, vol. 3, no. 29, pp. 861, 2018.