STRATEGIES FOR ENHANCED SIGNAL MODULATION CLASSIFICATIONS UNDER UNKNOWN SYMBOL RATES AND NOISE CONDITIONS

Ruixuan Wang, Yue Qi, Mojtaba Vaezi, Xun Jiao, and Moeness Amin
Center for Advanced Communications, Villanova University, USA

ABSTRACT

Radio frequency signal modulation classifications find broad applications in cognitive sensing and RF spectrum coexistence. Recently, deep neural networks have been shown to be a powerful tool for automatic modulation classification (AMC). Accounting for different signal variations is paramount towards reliable classifications. In this paper, we examine the performance of AMC under varying sampling rates and signal-to-noise ratio (SNR). We consider a dynamic environment where the signal modulation and channel conditions can be assumed constant over a number of consecutive observations. We also show that a single ResNet can be used for both modulation classification and estimating SNR which allows network training and testing at the same noise levels. It is shown that significant signal modulation classification accuracy improvement can be achieved using multiple observations and known SNR.

***Index Terms*—** Deep learning, modulation classification.

1. INTRODUCTION

Automatic modulation classification (AMC) is of great significance in a variety of applications that involve radio frequency sensing and co-existence. These include radar, communications, radio telescopes, satellite navigation, and telemetry. In essence, AMC has become an integral part for cognitive radio networks, electronic warfare, dynamic spectrum access, and interference monitoring [1, 2].

Signal modulation classification has been extensively studied over the past decades [1, 3, 4]. Feature-based methods [1] have constituted the traditional approach for modulation classification and predicated on the extraction of hand-crafted features to inform classification decisions. These features can be defined in the time domain, frequency domain, time-frequency domain, and constellation domain, and may be deterministic or/and statistical in nature [5]. Feature engineering, however, is time consuming and inaccurate, and may overlook important discriminatory features.

Recently, deep learning (DL) have successfully been applied to the AMC problem [3, 6–9]. The in-phase and quadrature-phase (I/Q) time-domain observation samples were used in [3] as input to deep neural network (DNN). Convoluntional neural networks (CNNs), ResNet [4], VGGNet [6], AlexNet [10], and GoogLeNet [11] were subsequently employed to deliver accurate classification decisions. More recently, long short-term memory (LSTM) was proposed to learn long term temporal representations from time-domain amplitude and phase samples [7, 12]. CNN-based modulation classification is considered in [8], where channel impairment effects on the classification results were discussed. The effect of the input data dimension on modulation classification was examined in [9]. The work in [13] evaluated the influence of up-sampling on classification accuracy. Both references [14] and [13] considered signal-to-noise ratio (SNR) mismatch of the training and testing data and its adverse impact on the confusion matrix.

One of the fundamental questions in AMC using deep learning is how the network generalizes to different channel conditions, changes in symbol rates vs sampling rates, and variations in SNR. Most existing works assume ideal sampling in the sense that the number of samples per symbol (SPS) is an integer. Further, in an attempt to improve accuracy, it was assumed that the SNR in model training and testing is fixed and precisely estimated. These assumptions are neither realistic nor practical. The imperfections in either or both cases may degrade performance below acceptable accuracy, causing intolerable misclassifications [9]. The effects of varying SPSs and mismatched SNRs for a wide class of signal structures have not been sufficiently examined in the literature.

In this paper we propose utilizing two important means to improve AMC performance under discrepancies in SNRs and SPSs. First, we employ the same deep learning network, devised for AMC, to provide SNR estimates. This is achieved using ResNet which can predict SNR nearly perfectly, even for SNR ≤ -10dB. DNN-based SNR estimation enables improved modulation classification performance by matching the test data with the network trained with the corresponding SNR. Second, we exploit the fact that neither modulation constellations nor channel conditions change instantaneously. Rather, they remain constant over a segment of time. This permits the classification decision to be rendered incorporating multiple consecutive inputs, in lieu of only one network input observation for one network decision. It is shown that the above attributes improve classification accuracy from 74% to 95%, even at SNR as low as -2dB.

The paper is structured as follows. Section 2 introduces the ResNet structure we employed in the underlying problem. Section 3 discusses the AMC under varying SNRs and SPSs. The SNR estimation and post-processing strategies are discussed in Section 4. Furthermore, we present the experimental result in Section 5 and conclude the paper in Section 6.

2. RESNET-BASED AMC

In this paper, we apply the ResNet [6] for AMC, a representative state-of-the-art deep learning model that has achieved tremendous success. The architecture is illustrated in Fig.1.

The input size of the ResNet is (2, 1024) which corresponds to the I/Q data in the dataset. In our experiment, we employ the basic ResNet structure mentioned, called residual stack in [6], which contains 5 convolutional layers, 1 max-pooling layer, and also 2 skip connections. Our architecture employs 6 residual stacks. The size of the output vector from these 6 residual stacks is (40, 16). This vector is then inputted to 3 dense layers with 640, 128, and 128 nodes respectively. Finally, we use 1 softmax layer to generate a prediction vector containing the probability of each modulation.

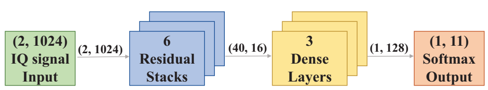


Fig. 1: The ResNet architecture and data flow

The employed ResNet includes 224588 trainable parameters in total and the size of our model is 3MB. We build the ResNet model using Tensorflow [15] running on a Tesla P100 GPU and trained the network for 150 epochs.

3. AMC UNDER VARYING SPSS AND SNRS

3.1. Modulation classification Dataset

We generate an RF signal dataset that consists of nine digital modulations (BPSK, QPSK, 8PSK, 4QAM, 16QAM, 64QAM, 16APSK, 32APSK, CPFSK) and two analog modulations (WB-FM, AM-DSB) similar to [6, 16]. In this dataset, each input sample is a (2, 1024) size floating point vector, which represents the real part and the imaginary part of a 1024-length complex time-domain signal. The power of each signal is normalized and Gaussian white noise is added with SNR in the range of $[-18, 18]$dB with 4dB step, e.g., $\{-18\text{dB}, -14\text{dB}, \ldots, 14\text{dB}, 18\text{dB}\}$.

3.2. Training and Testing under Different Symbol Rates

The relation between symbol rate and SPS is defined as $R_s = \frac{f_s}{SPS}$ where R_s and f_s denote symbol rate and sampling rate. If f_s is fixed, then by changing the SPS, the symbol rate would change correspondingly. The mismatch between the SPS of training and testing data may lead to a dramatic performance drop during the inference phase [13]. It is shown in [13] that the highest accuracy is achieved only when training and testing over the same up-sampling factor. It suggests the prior knowledge of the up-sampling factor is critical.

In this paper, we examine the classification performance under varying symbol rates with SPS granularity, using fractional sampling to simulate the asynchronization property. We generate 5 datasets with non-integer SPS in {6.7, 7.8, 7.9, 8.1, 8.2} obtained by interpolation methods. Each set contains 9 digital modulation signals. We first train a ResNet model based on signals with SPS = 6.7, then test the pre-trained ResNet Model based on each of the 5 aforementioned SPS signals separately. The results are shown in Fig. 2. One important observation is that the highest accuracy is achieved when the training and testing signals have the same SPS = 6.7. In addition, there is no significant performance drop if training and testing over two different SPSs with small deviation, which illustrates the robustness of the ResNet for mild asychronization.

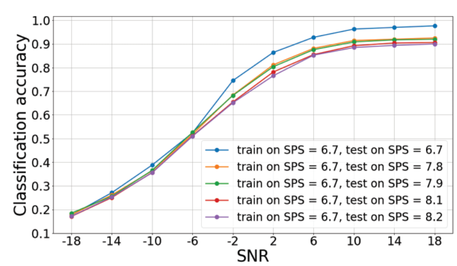


Fig. 2: Classification accuracy vs SNR under different testing SPS

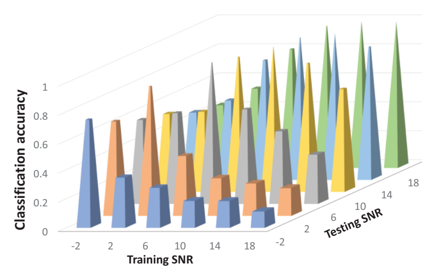


Fig. 3: Training and testing under different SNRs

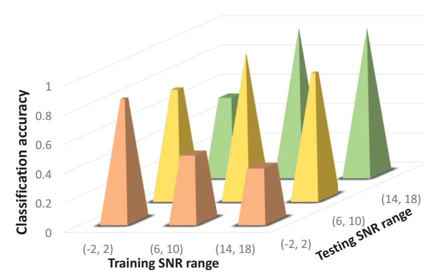


Fig. 4: Training and testing under different SNR ranges

3.3. Training and Testing under Different SNRs

Similar to the performance deterioration under SPS mismatch, the SNR mismatch between the training and testing data also leads to reduced classification accuracy.

In this experiment, we select different signals with SNR in $\{-2\text{dB}, 2\text{dB}, 6\text{dB}, 10\text{dB}, 14\text{dB}, 18\text{dB}\}$ and build 6 independent datasets. Then, we pre-train these 6 different ResNet models respectively based on these specific SNR signals. During the inference process, we test each pre-trained model using only signals with the same SNR as well as each of the other 5 SNR values. The results are captured in the matrix shown in Fig. 3. The highest accuracy is achieved when the training and testing SNR = 18dB, which is 99.81% whereas the lowest accuracy occurs when the SNR deviation between training and testing signals is the greatest. That is, when training on SNR = 18dB and testing on SNR = -2dB, the accuracy is only 11.05%.

3.4. Training and Testing under Different SNR Ranges

In lieu of exact match between SNR of the training and testing data, and different from total SNR mismatch between these datasets, we examine the case where the training and testing data assume the same SNR range. In our experiment, we use 3 different datasets whose SNRs are in the SNR ranges $[-2, 2]$dB, $[6, 10]$dB and $[14, 18]$dB. We pre-train three different ResNet models on each specific SNR range and test on signals from each of the above 3 ranges. As evident from Fig. 4, the best case occurs when both training and testing on the SNR range in the range of $[14, 18]$dB, achieving 99.81% modulation classification accuracy. The worst case is associated with the case when training on signals with SNR in the range of $[14, 18]$dB and testing in the range of $[-2, 2]$dB, where the accuracy drops to 37.67%. As generally expected, our experiment shows that the best performance comes from the highly matched SNRs between the training and testing data.

4. POST-PROCESSING STRATEGIES TO IMPROVE THE CLASSIFICATION

In this section, we first establish a ResNet model and train the ResNet using the time-domain I/Q data. During the inference process, for each (2, 1024) size query input we predict the corresponding modulation, which is presented by a softmax output vector. The length of this vector equals to the number of classes (11 in our modulation classification experiment). It is worth mentioning that since the ResNet structure is fixed, the size of the input layer implicitly determines the time segment over which modulation is held constantly. However, in reality, the modulation does not change rapidly and can last several folds of the input size of the ResNet. Hence, the prediction can benefit from several consecutive input data segments instead of one single segment. To this end, we propose a post-processing scheme to refine the classification decision.

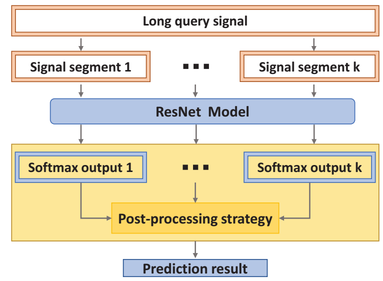


Fig. 5: Post-processing strategy to enhance the modulation classification accuracy

4.1. Consecutive Segments

We consider a long query input signal which is fed into the ResNet model, illustrated in Fig.5. Since it may not fit the input size of ResNet model, we divide one long signal into several segments fitting the input length of the ResNet and perform inference on the corresponding segments. For instance, we select (2, 1024*k) samples of input signal and proceed to divide them into k consecutive segments, each of size (2, 1024). Using ResNet, we perform the inference on each segment and the k prediction results are post-processed for a final decision. The prediction result for each segment is a softmax output vector representing the probability of each modulation. To achieve our post-processing strategy, we propose two selection methods, average voting and majority voting.

4.2. Average Voting and Majority Voting

In post-processing we deploy an element-wise averaging over the corresponding k softmax output vectors. For instance, if softmax output $S1 = [0.6, 0.2, 0.1, 0.1]$ and softmax output $S2 = [0.4, 0.3, 0.1, 0.2]$, then the average voting over $S1$ and $S2$ $S_{average} = [0.5, 0.25, 0.1, 0.15]$. Unlike the average voting post-processing described above, in the majority voting method, we first compute the class for each segment based on the corresponding softmax output, which is the index of the maximum value in the softmax output vector. Then, we construct a voting vector of size k, corresponding to the k segments. Finally, we apply the majority voting to determine the most likely label of the long query input. For example, if the $k = 10$ and the prediction results include one "64QAM", two "16APSK", two "QPSK" and five "BPSK", then a majority voting strategy would determine that "BPSK" is the predicted signal modulation since "BPSK" has the max voting score.

5. EXPERIMENTAL RESULT

5.1. Post-processing Configuration

To further improve the modulation classification performance, we apply the post-processing scheme. Fig. 6 shows the classification performance of our proposed post-processing methods. We fix the SPS = 8 and train the model over all SNRs. As illustrated in Fig. 6, without post-processing, when SNR is 6dB or higher, we achieve over 90% accuracy over 11 signal modulations. We can observe that after post-processing, the classification accuracy is significantly improved.

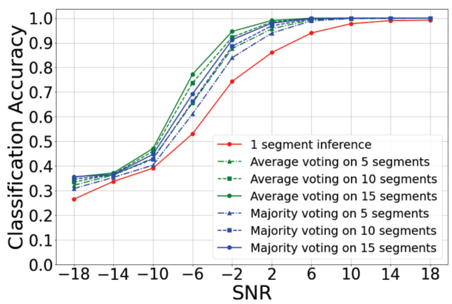


Fig. 6: Classification accuracy vs SNR with post-processing

For example, at SNR = -2dB, we can still achieve over 95% accuracy, which is 20% higher than the accuracy of 1 segment inference at 74%. Overall, the simulations show that the average voting method performs better than the majority voting method. As we change the number of segments k, the more segments used, the more accurate the classification. For k = 15 consecutive segments, the post-processing method gives the highest performance improvement. On average across all the 10 SNRs, this improvement is 7.3% compared to the 1 segment inference.

Following section 3.2 on the effect of varying SPS, we train and test ResNet on different SPSs. The average accuracy drop is over 5% in the worst scenario, which occurs when training the ResNet on SPS = 6.7 and testing on SPS = 8.2, as illustrated by the blue curve in Fig. 7. To mitigate the influence of SPS deviation between the training and testing signals, we apply the averaging voting scheme over k = 10 segments. As the orange curve indicates in Fig. 7, the performance of ResNet benefits from post-processing. According to the average voting result, we can achieve the same level of average accuracy at the worst scenario compared to the baseline case, where we train and test on the same SPS. Meanwhile, compared to the 1 segment inference, the average voting on 10 segments achieves 6.8% accuracy improvement.

5.2. SNR Estimation

As shown in Fig. 3 and Fig. 4, training and testing at the same/similar SNR would provide the best classification accuracy. Thus, it would be beneficial to know the SNR at the receiver so the model trained with the same SNR can be used for inference. We hence propose a possible solution to this issue — SNR estimation. We also employ the ResNet model and train the ResNet model using the SNR labels, which will be used to predict the SNR of a query input signal. The confusion matrix of SNR estimation is illustrated in Fig. 8(a) and Fig. 8(b). During the training process, we use the SNR of each signal as the label and the same dataset configuration as the modulation classification. According to our experimental result, the average accuracy of SNR estimation over SNR in the range of $[-18, 18]$dB is 94.1%. It is worth mentioning that the SNR estimation can achieve 100% accuracy when SNR higher than -6dB. Based on our experiment, we use an averaging voting and select k = 10. The result shows 99.49% accuracy over SNR in the range of $[-18, 18]$dB. We can estimate the SNR of a query signal even under SNR less than -10dB with a relatively high accuracy.

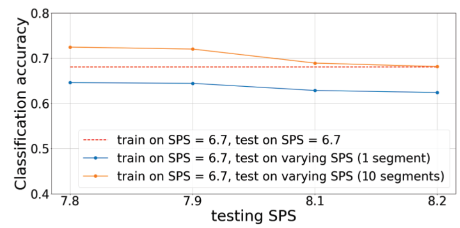
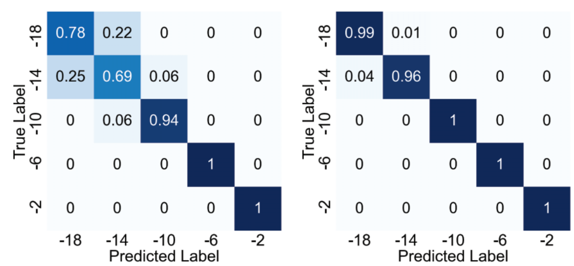


Fig. 7: Average classification accuracy vs testing SPS over SNR in the range of [-18, 18]dB



(a) 1 segment inference (b) 10 segments averaging

Fig. 8: Confusion matrix of SNR estimation

6. CONCLUSION

This paper examined DNN-based AMC under variations in signal and noise parameters. We accounted for a mismatch in both SPS and SNR values between the training and testing data. It was shown that the best classification accuracy is achieved when the training and testing data agree on SPS and SNR. The larger the disagreement and discrepancy, the lower the accuracy. In order to boost this agreement, we proposed an SNR estimation approach based on DNN to predict the SNR of query input signals. Since communication signal emitters do not typically change their signal modulations every few symbols, even in highly dynamic settings, consecutive input data segments were incorporated in the final classification decision. A post-processing scheme based on scoring and averaging of the SoftMax outputs was then applied, achieving more than 7% accuracy improvement.

7. REFERENCES

[1] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET Communications*, vol. 1, no. 2, pp. 137–156, 2007.

[2] S. Peng, S. Sun, and Y.-D. Yao, "A survey of modulation classification using deep learning: Signal representation and data preprocessing," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[3] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. International Conference on Engineering Applications of Neural Networks*, pp. 213–226, Springer, 2016.

[4] X. Liu, D. Yang, and A. El Gamal, "Deep neural network architectures for modulation classification," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, pp. 915–919, 2017.

[5] A. Hazza, M. Shoaib, S. A. Alshebeili, and A. Fahad, "An overview of feature-based methods for digital modulation classification," in *Proc. International Conference on Communications, Signal Processing, and their applications (ICCSPA)*, pp. 1–6, 2013.

[6] 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.

[7] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 433–445, 2018.

[8] K. Tekbıyık, A. R. Ekti, A. Görçin, G. K. Kurt, and C. Keçeci, "Robust and fast automatic modulation classification with CNN under multipath fading channels," in *Proc. Vehicular Technology Conference (VTC-Spring)*, pp. 1–6, 2020.

[9] F. Meng, P. Chen, L. Wu, and X. Wang, "Automatic modulation classification: A deep learning enabled approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10760–10772, 2018.

[10] S. Peng, H. Jiang, H. Wang, H. Alwageed, and Y.-D. Yao, "Modulation classification using convolutional neural network based deep learning model," in *Proc. IEEE Wireless and Optical Communication Conference (WOCC)*, pp. 1–5, 2017.

[11] 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*, vol. 30, no. 3, pp. 718–727, 2018.

[12] Z. Ke and H. Vikalo, "Real-time radio modulation classification with an lstm auto-encoder," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4935–4939, 2021.

[13] E. Perenda, S. Rajendran, G. Bovet, S. Pollin, and M. Zheleva, "Learning the unknown: Improving modulation classification performance in unseen scenarios," in *Proc. IEEE INFOCOM Conference on Computer Communications (INFOCOM)*, pp. 1–10, 2021.

[14] X. Xie, Y. Ni, S. Peng, and Y.-D. Yao, "Deep learning based automatic modulation classification for varying SNR environment," in *Proc. IEEE Wireless and Optical Communications Conference (WOCC)*, pp. 1–5, 2019.

[15] 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," 2016. Software available from tensorflow.org.

[16] T. J. O'shea and N. West, "Radio machine learning dataset generation with GNU radio," in *Proc. the GNU Radio Conference*, vol. 1, 2016.