SNR-Boosted Automatic Modulation Classification

Clayton A. Harper
Darwin Deason Institute for Cybersecurity
Dallas, TX

Avi Sinha
Darwin Deason Institute for Cybersecurity
Dallas, TX

Mitchell A. Thornton
Darwin Deason Institute for Cybersecurity
Dallas, TX

Eric C. Larson
Darwin Deason Institute for Cybersecurity
Dallas, TX

***Abstract*—Automatic modulation classification is a desired feature in many modern software-defined radios; however, classification performance degrades with decreasing signal to noise ratios. We propose employing a deep convolutional signal to noise ratio estimation model to exploit relationships within signals of similar signal to noise ratio ranges through signal to noise ratio specific modulation classifiers. We utilize a two-stage process where the signal to noise ratio is first estimated and then demultiplexed into a modulation classifier that has been tuned on signals with similar signal to noise ratios. Using the proposed method, we build upon the current state-of-the-art and increase classification performance at decreasing signal to noise ratios.**

I. Introduction

A considerable amount of work has gone into Automatic Modulation Classification (AMC) for a variety of applications including cognitive radios, interference monitoring, and defense applications. AMC plays a critical role in these systems as it is a necessary step in demodulating an unknown signal. In systems that utilize adaptive modulation schemes, AMC can be used to determine the current channel conditions. With knowledge of the channel conditions, the transmitter can adjust the modulation scheme to maximize usage of the transmission medium.

Although application specific, AMC in many situations cannot assume any prior knowledge on the incoming signals. When this is the case, AMC must be able to classify a large variety of modulation schemes. Typical benchmarks are constructed on the premise that the AMC must classify not only the mode of modulation (e.g., QAM), but the exact variant of that mode of modulation (e.g., 32QAM). These architectures have proven to be effective at high signal to noise ratios (SNRs) but degrade significantly at low SNRs which occur often in real-world applications. Therefore, our aim in this work is to improve AMC as SNR decreases.

To perform classification, a large variety of input features have been investigated. Historically, AMC has been performed through statistical moments and higher order cumulants [1]–[3] derived from the received signal. Recent approaches [4]–[7] use raw in-phase (I) and quadrature (Q) components as features to predict the modulation variant of a signal. Further works have investigated additional features including IQ constellation plots [8]–[10].

After deriving input features, machine learning models are used to determine statistical patterns in the data for the classification task. Support vector machines, decision trees, and neural networks are commonly used classifiers [4]–[7],

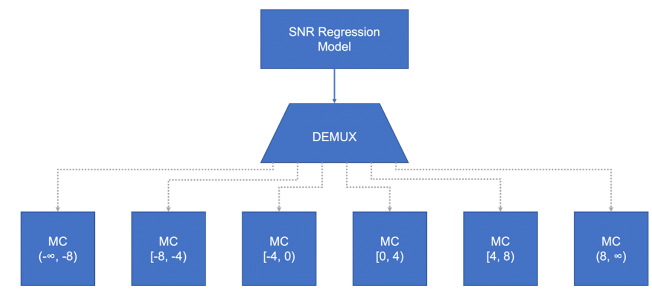

Fig. 1. Complete architecture using SNR regression and SNR-specific classifiers.

[11], [12]. Residual neural networks (ResNet) along with convolutional neural networks (CNN) have been shown to achieve high classification performance for AMC [4]–[8]. Deep learning in AMC has become more prevalent in recent years with promising performance and the ability to generalize for large, complex datasets.

A task that is often overlooked in AMC is SNR estimation. In previous work, models are trained to be as resilient as possible under different SNRs; however, classification is increasingly difficult as SNR degrades as a single model may not be able to represent features specific to different noise levels. At lower SNR values, the noise component of the signal becomes more dominant and the features employed by the classifier may not be reliable enough to make an informed classification.

In this work, we propose an architecture that leverages SNR estimation of modulated signals to enhance classification performance. By first predicting the SNR of the received signal, we can apply an SNR-specific modulation classifier (MC) that has been trained on signals with the predicted SNR. Utilizing this approach, different classifiers can tune their feature processing for differing SNR ranges (see Figure 1). In our implementation, we train an SNR regression model that is used to select the desired MC based on the estimated SNR of the signal.

In our previous work [4], we found that modulation classification performance plateaued in peak performance beyond +8dB SNR and approached chance classification performance below -8dB SNR. Therefore, in this work we aim to primarily increase classification performance in the range of -8dB to +8dB SNR. We build upon our previous work [4] exploiting latent space statistics for AMC. More precisely, this previous work made use of X-Vectors that are traditionally used in speech embeddings [13]. X-Vectors employ statistical moments, specifically mean and variance, across convolutional filter outputs. It can be theorized that taking the mean and variance of the embedding layer helps to eliminate signal-specific information, leaving modulation-specific characteristics. This X-Vector inspired architecture achieved state-of-the-art AMC performance; therefore, we aim to improve upon this architecture in this work.

II. Dataset

To evaluate different machine learning architectures, we chose the RadioML 2018.01A dataset that is comprised of 24 different modulation types [6], [14]. There are a total of 2.56 million labeled signals, $S(T)$, each consisting of 1024 time domain digitized baseband samples of in-phase (I) and quadrature (Q) signal components where $S(T) = I(T) + jQ(T)$.

The 24 modulation types are listed as follows: OOK, 4ASK, 8ASK, BPSK, QPSK, 8PSK, 16PSK, 32PSK, 16APSK, 32APSK, 64APSK, 128APSK, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM, AM-SSB-WC, AM-SSB-SC, AM-DSB-WC, AM-DSB-SC, FM, GMSK, and OQPSK. Each modulation type includes a total of 106,496 observations ranging from -20dB to +30dB SNR in 2dB steps for a total of 26 different SNR values.

For evaluation, we divided the dataset into 1 million different training observations and 1.5 million testing observations under a random shuffle split, stratified across modulation type and SNR. Because of this balance, the expected performance for a random chance classifier is 1/24 or 4.2%.

III. SNR Regression

A. SNR Regression Training

First, a regression model to predict the SNR level of an input signal is trained. Although the SNR values in the dataset are discrete, SNR is measured on a continuous scale in a deployment scenario and can vary over time. As a result, we utilize regression over classification to model SNR. Through this task, we hope to attain more information about the modulated input signal and improve AMC performance, especially at decreasing SNR levels, by directing the signal to a specialized classifier.

We investigate two different architectures for SNR regression—Random Forest Regression [15] and a deep convolutional neural network architecture similar to our previous work [4] (Table I). Each model requires the input features to be in a slightly different format. For the random forest, we aggregate statistics of the I and Q data as well as the magnitude of the signal, M, defined as $M = \sqrt{I^2 + Q^2}$. We include magnitude as an additional feature to the random forest regression model as SNR is based on the amplitude variations a signal. We calculate the signal statistics for range, variance, and kurtosis for the M, I, and Q signals similarly to [16].

In this work, we use the random forest regression algorithm from the Python library scikit-learn [17]. Default parameters were used—an ensemble of 100 trees, using mean squared error as the splitting criterion. The variable importance of each of these statistics can be calculated from the random forest using variable permutations as suggested by [15]. The importance values are graphed in Figure 2, where it can be observed that most features are deemed to have relatively equal importance, with magnitude-based statistics having slightly larger importance than I or Q-based signals alone.

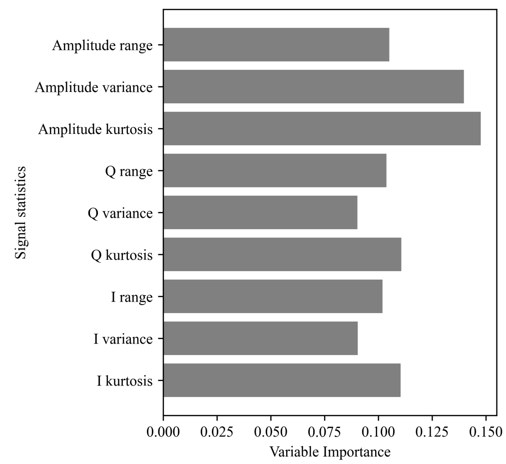

Fig. 2. Variables used in random forest regression and importance to accuracy.

The convolutional network does not use any derived features, but instead learns directly from the I and Q signals in the time domain. The filters learned in the network extract features that can be directly used for SNR regression. We found incorporating the magnitude in the deep convolutional architecture did not increase performance. That is, there was no statistically significant advantage to adding magnitude as a feature. Consequently, we only use I and Q as input modalities to the convolutional model. Both the random forest regressor and convolutional network are trained to convergence with the best models being saved according to the lowest mean squared error achieved while training.

TABLE I
Deep Convolutional SNR Regression Model

Layer	Output Dimensions
Input	2 x 1024
Conv 1D (ReLU)	64 x 1024
Conv 1D (ReLU)	64 x 1024
Conv 1D (ReLU)	64 x 1024
Conv 1D (ReLU)	64 x 1024
Conv 1D (ReLU)	64 x 1024
Conv 1D (ReLU)	64 x 1024
Conv 1D	64 x 1024
Average Pooling 1D	64
Variance Pooling 1D	64
Concatenate	128
FC/SeLU	128
FC/SeLU	128
FC/Linear	1

B. SNR Regression Results

Figure 3 summarizes the performance of each regression model. The deep convolutional regression model outperforms

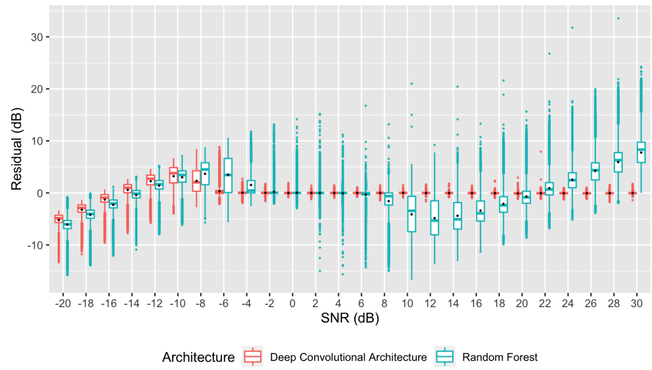

Fig. 3. Deep convolutional architecture and random forest residual values for each SNR value with the mean residual dB shown in black.

the random forest in terms of residual dB particularly for true SNRs above -8dB. The residual is defined as:

$$residual = true - predicted$$

A residual closer to zero is desirable as the predicted value would be closer to the true value. The variance in the predictions becomes unstable below -8dB SNR. This behavior was also observed in Figure 4 where classification performance becomes no better than chance below -8dB SNR. This behavior may indicate that signals below -8dB SNR are too noisy to obtain consistent predictions. When comparing the regression results with an F-Test of residual variance, it was found that the difference is statistically significant across all SNRs. We therefore choose to use the convolutional architecture as our SNR prediction model for the remainder of this work.

SNR values above -8db exhibit small variances along with mean residuals of approximately zero. As stated previously, our main goal in this work is to improve classification performance for SNR levels between -8dB and $+8$dB. With this goal in mind and our regression model performing well in this range, we can predict SNR and pass the signal in question to the appropriate SNR-specific modulation classifier with confidence for signals between -8dB and $+8$dB SNR.

IV. Modulation Classification

A. Modulation Classification Training

From our previous work, we know classification performance plateaus outside the range of approximately -8dB SNR to $+8$dB SNR (see Figure 4 from [4]). Due to this behavior, we decided to create multiple modulation classifiers (MCs) to exploit nuances for varying SNR groups specifically in the -8dB to $+8$dB SNR range.

To determine the SNR groupings, we had to ensure there was sufficient training data in each group to not overfit the classifiers. We also had to ensure enough granularity such that each grouping provided more value than a single classifier trained to be as resilient as possible under varying SNR levels. We created 6 different groupings as seen in Table II to fit

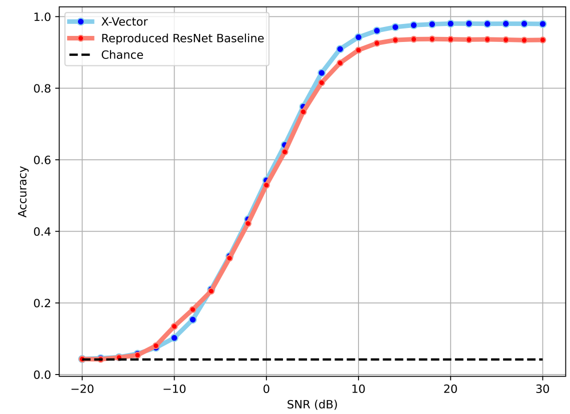

Fig. 4. Accuracy values for each SNR value in the dataset as seen in [4].

TABLE II
SNR Groupings for Training SNR-Specific Classifiers and Demultiplexed Classification Ranges for Each Predicted SNR.

Training Range (dB)	Demultiplexed Classification Range (dB)
[-20, -8]	$(-\infty$, -8)
[-8, -4]	[-8, -4)
[-4, 0]	[-4, 0)
[0, 4]	[0, 4)
[4, 8]	[4, 8)
[8, 30]	[8, ∞)

these constraints. Values below -8dB SNR are grouped and values above $+8$dB SNR are grouped as diminishing returns in classification performance were observed in Figure 4. Values below -8dB were also grouped as Figure 3 illustrates high variability in SNR predictions below this level.

Each MC is based upon the same architecture described in Table I where the input modalities are the I and Q signal components; however, each output is a 24 unit softmax output for the 24 different modulation types. The complete architecture can be seen in Figure 1 where the SNR regression output is demultiplexed into a specialized classifier for the final modulation type prediction.

B. Classification Results

Using our X-Vector inspired architecture used in [4], we were able to achieve a maximum accuracy of 98% at high SNR values improving over [6] as seen in Figure 4. Highlighting improvements across SNR, Figure 5 shows the overall performance improvement (in percentage accuracy) compared to [4].

While we see a slight decrease in performance for -8dB and a larger decrease for -2dB, we improve upon [4] under most SNR conditions particularly in our target range of -8dB to $+8$dB. A possible explanation for these decreases in performance is that the optimization for a particular MC helped overall performance for a grouping at the expense of a single value in the group. That is, the MC for $[-4, 0)$ boosted the overall performance by performing well at -4 and 0dB at the expense of -2dB. Due to the large size of the testing set, the small percentage gains are impactful because thousands more classifications are correct. All results are statistically significant based on a McNemar's test [18], therefore achieving a new state-of-the-art performance.

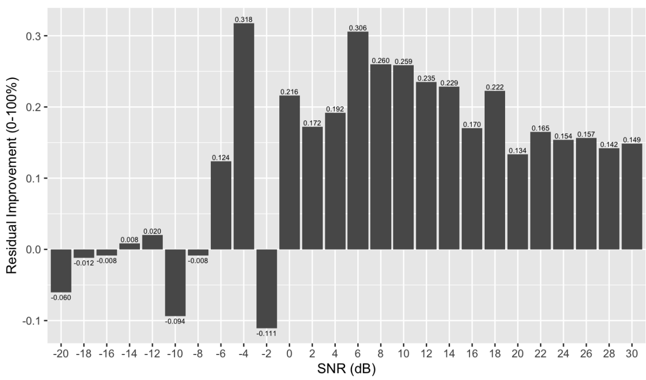

Fig. 5. Proposed approach residual improvement in accuracy over [4].

V. Conclusion

In this work we have examined several different architectures for improving modulation classification performance. While many architectures focus on being as resilient as possible under varying SNR conditions, a single model may not be able to fully characterize discrepancies between signals at different SNR levels. Our proposed architecture makes use of SNR estimates to refine the classification task. We found a deep convolutional architecture to be superior in SNR regression compared to a random forest regression in terms of residuals. After estimating SNR, we are able to direct the signal to a specialized classifier for the given SNR estimate. Through leveraging the deep convolutional regression model, we were able to create SNR-specific classifiers that improved classification performance compared to the current state-of-the-art.

References

[1] S.S. Soliman and S.-Z. Hsue, "Signal Classification using Statistical Moments," *IEEE Transactions on Communications*, vol. 40, no. 5, pp. 908–916, 1992.

[2] A. Swami and B.M. Sadler, "Hierarchical Digital Modulation Classification using Cumulants," *IEEE Transactions on Communications*, vol. 48, no. 3, pp. 416–429, 2000.

[3] Mahi Abdelbar, William H. Tranter, and Tamal Bose, "Cooperative Cumulants-Based Modulation Classification in Distributed Networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 446–461, 2018.

[4] C. A. Harper, L. Lyons, M. A. Thornton, and E. C. Larson, "Enhanced Automatic Modulation Classification Using Deep Convolutional Latent Space Pooling," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, 2020.

[5] T.J. O'Shea, J. Corgan, and T.C. Clancy, "Convolutional Radio Modulation Recognition Networks," in *International Conference on Engineering Applications of Neural Networks.*, 2016, p. 213–226.

[6] T.J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-Air Deep Learning Based Radio Signal Classification," in *IEEE Journal of Selected Topics in Signal Processing*. IEEE, 2018, vol. 12, pp. 168–179.

[7] Thien Huynh-The, Cam-Hao Hua, Jae-Woo Kim, Seung-Hwan Kim, and Dong-Seong Kim, "Exploiting a Low-Cost CNN with Skip Connection for Robust Automatic Modulation Classification," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6.

[8] S. Peng, H. Jiang, H. Wang, H. Alwageed, and Y. Yao, "Modulation Classification using Convolutional Neural Network Based Deep Learning Model," in *2017 26th Wireless and Optical Communication Conference (WOCC)*, 2017, pp. 1–5.

[9] S. Peng, H. Jiang, H. Wang, H. Alwageed, Y. Zhou, M. M. Sebdani, and Y. 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, 2019.

[10] F. Wang, Y. Wang, and X. Chen, "Graphic Constellations and DBN Based Automatic Modulation Classification," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, 2017, pp. 1–5.

[11] Kais Hassan, Iyad Dayoub, Walaa Hamouda, Crepin Nsiala Nzeza, and Marion Berbineau, "Blind Digital Modulation Identification for Spatially-Correlated MIMO Systems," *IEEE Transactions on Wireless Communications*, vol. 11, no. 2, pp. 683–693, 2012.

[12] Mahi Abdelbar, Bill Tranter, and Tamal Bose, "Cooperative Modulation Classification of Multiple Signals in Cognitive Radio Networks," in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 1483–1488.

[13] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN Embeddings for Speaker Recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[14] DeepSig Incorporated, "RF Datasets for Machine Learning," `https://www.deepsig.ai/datasets`, 2018, Accessed: 2021-04-29.

[15] Leo Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[16] Tara Salman, Ahmed Badawy, Tarek M. Elfouly, Tamer Khattab, and Amr Mohamed, "Non-Data-Aided SNR Estimation for QPSK Modulation in AWGN Channel," in *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. Oct. 2014, IEEE.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[18] Quinn McNemar, "Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.