Classification of Radio Modulation with Machine Learning

Han Zhao
Electrical and Computer Engineering
h1zhao@ucsd.edu

Jiawen Xu
Electrical and Computer Engineering
jix034@ucsd.edu

Abstract

Wireless communication has brought a great impact on modern society. As a fundamental aspect of wireless communication, understanding and analyzing radio modulation is important. Our work explores multiple machine learning models employed for solving radio modulation classification problems and comparatively evaluates their performance and efficiency. (**Link:** `https://drive.google.com/drive/folders/1NL0vIY3SBU0uoH9M7Ut_1w0nenuV6WxW?usp=sharing`)

1 Introduction

Wireless communication techniques play an important role in modern life, serving as the backbone for seamless connectivity, efficient information exchange, and transformative technological advancements, thereby shaping our interconnected world. In wireless communication, modulation is the process of converting the wireless signal before its transmission on a wireless channel. Modulation encodes information from the transmitted signal while demodulation extracts the information from the modulated signal [11]. Understanding the modulation scheme is important because it is the foundation of building wireless communication. Therefore, automatic modulation recognition (AMR) has gained increasing importance due to its ability to efficiently analyze and enhance wireless communication. The wide usage of AMR in spectrum interference monitoring and dynamic spectrum allocation, [8], and other military usages [3]. make AMR play a crucial role in optimizing wireless communication systems.

Given the increasing significance and demand for radio modulation classification using machine learning, this paper focuses on training multiple machine learning models. By analyzing the training outcomes and performance of each model, this paper thoroughly examines their respective strengths and weaknesses.

2 Related Works

While image recognition problems have received considerable attention and extensive research in recent times, automatic modulation recognition (AMR) has been comparatively less explored and discussed. Nonetheless, AMR has been the subject of investigation in several research papers, indicating a growing interest in this domain.

In Pijackova's work, they discuss the performance of the normal CNN model, and the CLDNN model, which is a hybrid model between CNN and a Long-Short-Term Memory Neural Network (LSTM) on this classification of radio problems. In addition, Pijackova also studies the impact of data normalization on the accuracy of the models. [7]

Additionally, Pi's paper explores a broader range of machine learning models, such as support vector machine (SVM), decision tree, and k-nearest neighbor (KNN), in an effort to address the classification problem of radio modulation. Moreover, Pi incorporates data augmentation techniques to generate

Preprint. Under review.

additional data from the original dataset, thereby enhancing the generalization capability of the trained models.[6]

The lack of discussion on the efficiency of the training process, specifically training time, in the referenced papers is a notable gap. For radio modulation classification models, swift and efficient training is crucial to adapt to the dynamic radio environment. Moreover, the papers did not explore newer and popular models or strategies like GRU and attention mechanisms for classification. Therefore, this paper aims to address these gaps by examining the performance and efficiency of unexplored models, incorporating additional relevant features, and analyzing their impact on model performance

3 Methodology

3.1 Dataset

RADIOML2016.10A is the data set used for the model training which was generated with GNU radio and contains 11 modulations (8 digital and 3 analog) at different SNRs. There are 11 modulation schemes in the data set to be classified: QPSK, 8PSK, QAM16M, QAM64, AMDSB, AMSSB, BPSK, CPFSK, GFSK, PAM4, WBFM. [2] The data set leverages the 2-wide I/Q to hold the in-phase and quadrature components as a 2x128 vector. For our data processing, we reshape it in a form such that the characteristic of the radio frequency data is sustained. A VT-CNN2 Mod-Rec is developed for this data set which presents a possible solution for RF modulation classification. [4] [5]

In the dataset, each label has an Signal-to-noise ratio (SNR) level within the range of {-20dB, -18dB, -16dB, $\cdots$, 16dB, 18dB}. Models are trained without separating the training dataset into different SNR. Only separte the test data into different SNR groups to test each SNR group individually for each model. SNR is a measure to compare the levels of signal to the background noise with expression as follow:

$$SNR_{dB} = 10\log_{10}(\frac{p_{signal}}{p_{noise}})$$

where p_{signal} represents the power of the signal and p_{noise} represents the power of noise. [5]

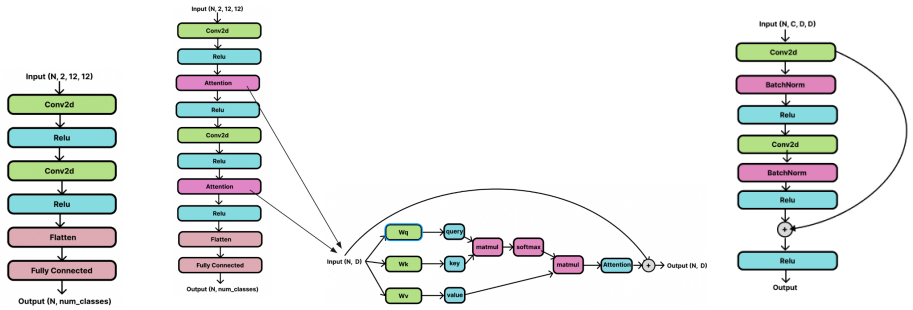
3.2 Feature Extraction

The types of modulations can be categorized into three main types: amplitude modulation (AM), frequency modulation (FM), and phase modulation (PM). Amplitude modulations, such as AM-DSB and AM-SSB, vary the carrier wave's amplitude. Frequency modulations, including CPFSK and GFSK, alter the carrier wave's frequency. Phase modulations, such as 8PSK and BPSK, modify the carrier wave's phase. To enhance the accuracy of the models, this paper tries to extract the amplitude, frequency, and phase features from the raw I/Q data and incorporating them into the training data, which is described in Algorithm 1. By including these additional features, the model may differentiate different types of modulations more effectively.

Algorithm 1: Extracting Amplitude, Frequency, and Phase Features from I/Q Samples

Input: X: I/Q samples
Output: data_combined: Extracted features (amplitude, frequency, phase)

$I \leftarrow X[:, 0, :]$; // get real part data
$Q \leftarrow X[:, 1, :]$; // get imaginary data
$IQ \leftarrow I + iQ$; // Complex representation of I/Q samples
$amplitude \leftarrow np.abs(IQ)$; // Amplitude of IQ samples
$phase \leftarrow np.angle(IQ)$; // Phase of IQ samples
$instantaneous\_phase \leftarrow np.unwrap(phase)$; // Unwrapped phase
$frequency\_deviation \leftarrow np.diff(instantaneous\_phase)$; // Frequency deviation
$features \leftarrow np.array([amplitude[:, :-1], phase[:, :-1], frequency\_deviation])$; // Combine features
$data\_original \leftarrow np.array([I[:, :-1], Q[:, :-1]])$; // Original I/Q data
$data\_combined \leftarrow np.concatenate([data\_original, features], axis = 2)$; // Combined data with features
return $data\_combined$



(a) Vanilla CNN (b) CNN with attention (c) ResNet Block

Figure 1: CNN Models

3.3 Models

3.3.1 Vanilla CNN

A Convolutional Neural Network (CNN) is used in processing and analyzing structured grid-like data such as images. Our dataset has two channels - IQ channels, which makes CNN a good option for classification tasks. The structure of Vanilla CNN as shown in figure 1a, and its performance is compared with adding an attention layer and residual block in the following section.

3.3.2 CNN with attention

To explore more CNN implementation, two attention layers are added into the previous CNN network as shown in figure 1b. The attention layer will calculate the attention score and add it to the input as the output. The attention mechanism is more widely used in computer vision and language processing tasks, which can better focus on specific parts of the input data and suppress irrelevant noise features. In general, it will improve the performance of models by enabling them to capture dependencies and match input to outputs. The attention score will be calculated as follow: [9]:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

The query, key, and value are linearly transformed from our input data with three separate sets of weights.

3.3.3 ResNet

Other than the attention mechanism, the residual network is applied to the CNN. ResNet, or Residual Network can address the problem of vanishing gradients and enable the training of much deeper neural networks. For each residual block, it sets up a connection that can "skip" over some layers and directly propagate the input to the deeper layers, also called a "shortcut" connection. The vanishing gradient problem occurs when the gradients become really small as they propagate through multiple layers. The residual structure can effectively improve this situation. Figure 1c presents one of our ResNet blocks, and some of the blocks may not necessarily have the Conv2d layer in the shortcut. [1]

3.3.4 GRU

GRU is a modern RNN that has the memory to learn the long-term dependency of the data and prevent the vanishing gradient of the simple. A GRU layer is composed of two gates, a reset gate r, and an update gate z, instead of three gates. The reset gate controls how to incorporate the input data with the previous memory, and the update gate controls how much previous data is stored in the memory unit.[10] The equations of GRU can be found below and the structure diagram is illustrated in Figure2 .

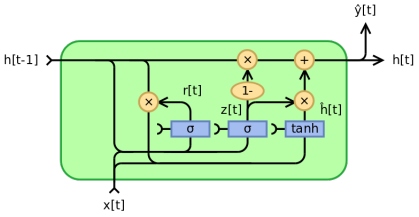
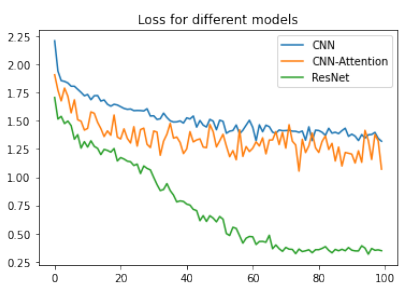
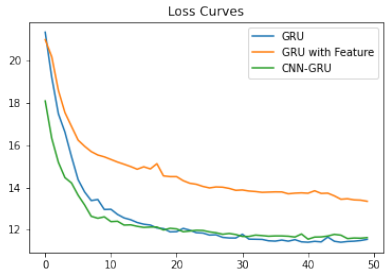


Figure 2: GRU



(a) Loss of Vanilla CNN, Atention CNN, ResNet



(b) Loss of GRU, GRU-feature, CNN-GRU

Figure 3: loss curves

3.3.5 CNN-GRU

In recent studies, there has been growing popularity in employing hybrid models that combine both CNN layers and RNN layers for radio modulation classification. While these models typically utilize the CLDNN architecture, which combines CNN and LSTM layers, none of the existing research has explored the combination of CNN and GRU. The GRU model, known for its fewer gates and parameters compared to LSTM, holds the potential for improved efficiency over LSTM-based models. Therefore, it presents an intriguing alternative for incorporating the strengths of CNNs with a more compact RNN variant. This paper builds a new architecture of the CNN-RNN hybrid model by using the CNN layers and GRU layers to solve the classification problem of the radio modulations. The details of this model can be found in Table 1

3.4 Training and Evaluating Process

	CNN	Attention CNN	ResNet	GRU	Featured GRU	CNN-GRU
Batch Size	1000	100	1000	1000	1000	1000
Learning Rate	1e-3	1e-3	1e-4	1e-3	1e-3	1e-3
Number of Epochs	100	100	100	50	50	50
Filter size	5x5->3x3	5x5->3x3	7x7->(3x3->3x3)*	N/A	N/A	N/A
Hidden Head	N/A	N/A	N/A	200	200	200

Table 2: Hyperparameters (* means multiple block concatenated)

All the models discussed in this paper are trained in UCSD datahub with 8 CPU, 16G RAM, and 1 GPU. The dataset is divided into two equal parts, with one part allocated for training the model, while the other part is utilized for testing and evaluating the accuracies. During the training phase, 80% of the dataset is utilized as the training dataset, while the remaining 20% is allocated for validation purposes. To improve the training process, an Adam optimizer is introduced, which can adjust the learnig rate based on the epoch and the momentum of gradient. The loss curve of the training process and the hyperparameters used for training each model can be found at Figure 3 and Table 2

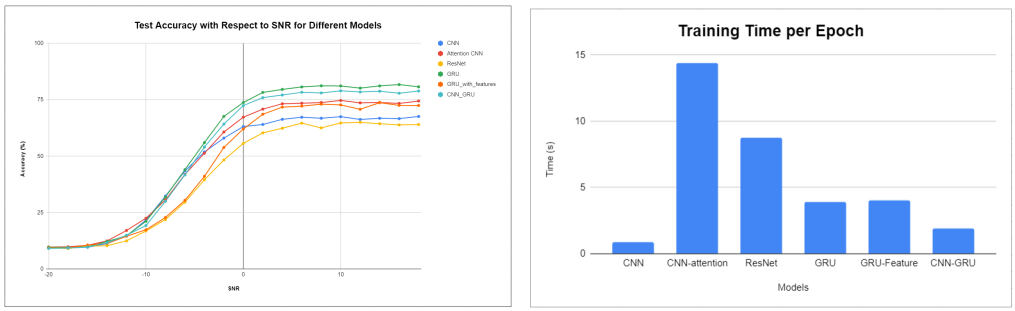
Layers	Output dimensions
input	128x2
Convolutional layer	121x64
Maximum pooling layer	60x64
GRU layer	64x200
Dropout layer	64x200
GRU layer	200x200
feed-forward layer	200x11

Table 1: ARCHITECTURE OF THE CNN_GRU MODEL

4 Result and Discussion

4.1 Signal-To-Noise Ratio (SNR) Characteristics

Figure 4a presents the test result with test set separated into different Signal-to-Noise Ratio (SNR) level. The result shows that the test accuracy is impacted by the level of SNR. As the SNR increases, the test accuracy increases.When SNR is -20dB. the test accuracy is around 9% for all the trained models, while the test accuracy ranges from 60% to 75% with SNR greater than 0dB. This result



(a) Test accuracy with respect to SNR

(b) Training Time for Different Models

Figure 4: Accuarcy and Efficiency

is as expected as it is intuitive that higher noise level will negatively impact the accuracy of the RF modulation classification.

4.2 Training Time Analysis

Figure 4b shows the average training time for each epoch for all the models. Vanilla-CNN takes the least amount of time while CNN-attention takes the longest time for each epoch. The reason is due to memory restraint from the UCSD datahub, we have to reduce the batchsize of CNN-attention to 100, while other models run with a batch size of 1000.

4.3 Classification of Overall Test Results Visualization with Confusion Matrix

We represent our result using a confusion matrix where X-axis represents the predicted labels and Y-axis represents the true labels (See Figure 5). The boxes on the diagonal line represent correct predictions and other boxes represent incorrect predictions. We can see our models have most of the results locating on the diagonal line, which means most of our prediction results are correct. The confusion matrix is plotted with the test result without regarding SNR, which means the test set is not divided into different SNR levels for this result visualization. [5] [4]

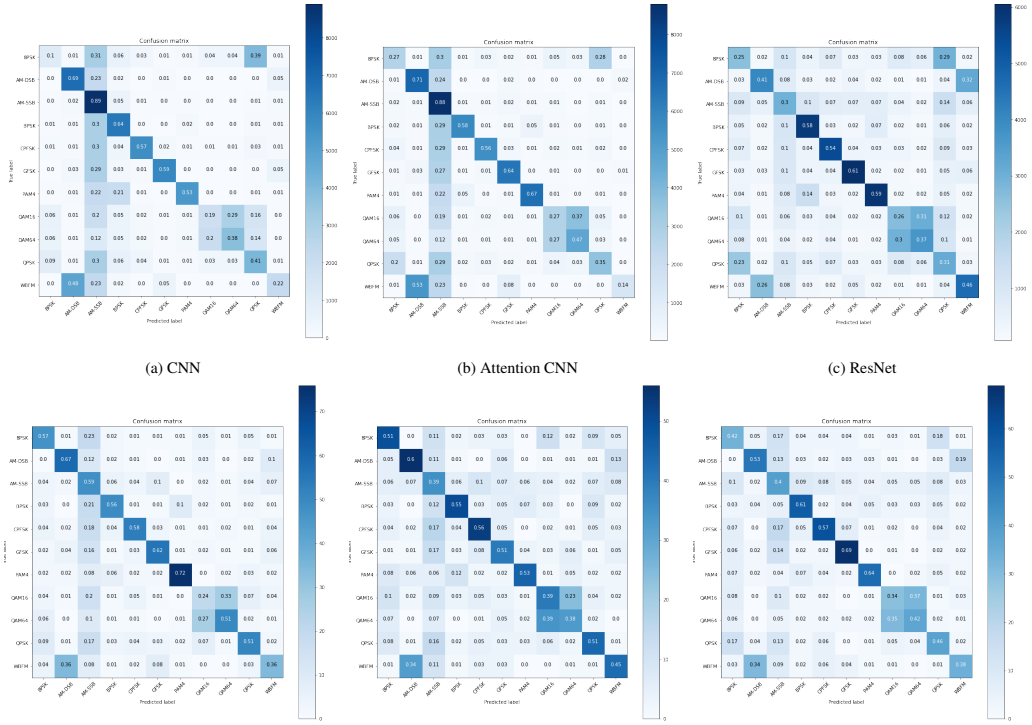
4.4 Model Analysis

4.4.1 CNN v.s. CNN with Attention v.s. ResNet

In this paper, three CNN models are used: Vanilla CNN, Attention CNN, ResNet. Compared to Vanilla CNN, Attention CNN improves the accuracy of classification because attention mechanism can bring focus to specific relevant data. ResNet suffers from over-fitting even with high regularization configuration. The training accuracy of ResNet is up to 90% while test accuracy and validation accuracy is lower than Vanilla CNN. From the Figure 3a, it shows that ResNet achieves lowest loss comparing to other model, but the accuracy on test data set does not become higher. Although Attention CNN has higher accuracy than CNN, the training time is much higher than Vanilla CNN, which is because of the change of batch size due to the memory restraint. The attention layer in CNN increases the memory consumption which may not be practical in reality usage.

4.4.2 GRU v.s. GRU with Features

Section 3.2 discusses the potential improvement of the models by adding the amplitude, phase shift, and frequency to the training data. However, the result doesn't support this idea. According to Figure4a, the accuracy from the GRU models trained with more features is lower than the accuracy from the GRU trained with the original data. This deterioration might be because the modulation in the data set is more complex. Some of the modulations combine both AM, FM, and PM. Adding more features may confuse the models. However, according to Figure 5e and Figure 5d, the accuracy of GRU with more features on some modulation like WBFM and QAM16 are higher than the accuracy of normal GRU. This difference may imply extracting more features may help to recognize some modulations. Moreover, as demonstrated in Figure 4b, the integration of three additional features into the training dataset only marginally escalates the training time per epoch. This suggests



(a) CNN (b) Attention CNN (c) ResNet

(d) GRU (e) GRU with features (f) CNN-GRU

Figure 5: Test Result Visualization with Confusion Matrix

that augmenting the dataset with more features could potentially enhance model accuracy without substantially compromising training efficiency.

4.4.3 GRU v.s. CNN-GRU

Section 3.3.5 introduces a new architecture that combines CNN layers and GRU layers. Figure 4a demonstrates that the CNN-GRU model achieves a similar level of accuracy as the conventional GRU model. However, CNN-GRU model performs a little better with high noise data, which might suggest that CNN-GRU models can learn from high-noise data better. In addition, the Figure 4b shows that the training time of CNN-GRU layers is much shorter than the training time of normal GRU models. This might be because the CNN layers combine the data and pass a smaller-dimension input to the GRU layers which makes CNN-GRU models more efficient.

5 Conclusion

The classification of radio modulation is essential for advancement of wireless communication. Utilizing machine learning method can effectively increases the possibility of having a RF modulation classifier with higher accuracy and better performance. This paper explores two families of machine learning models(CNN and GRU) and compare their predictive accuracy and performance such as training time. In summary, GRU outperforms the CNN models while the hybrid model CNN-GRU model is the best one in both accuracy and efficiency. For future work, more attention models like the Transformer and more hyprid models can be discussed and applied to solve this automatic modulation recognition problem.

6 Individual Contribution

Han implements, trains, and analyzes with GRU, GRU with features, and CNN-GRU (50%)
Jiawen implements, trains and analyzes with CNN, CNN with attention, and ResNet (50%)

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[2] DeepSig Inc. Rf datasets for machine learning.

[3] Asoke Kumar Nandi and Zhechen Zhu. *Automatic Modulation Classification: Principles, algorithms and applications*. Wiley-Blackwell, 2015.

[4] Timothy J O'Shea, Johnathan Corgan, and T. Charles Clancy. Convolutional radio modulation recognition networks. *arXiv preprint arXiv:1602.04105*, 2016.

[5] Timothy J O'Shea and Nathan West. Radio machine learning dataset generation with gnu radio. *Proceedings of the 6th GNU Radio Conference*, 2016.

[6] Shuang Pi, Shuanggen Zhang, Shumin Wang, Bochi Guo, and Wei Yan. Improving modulation recognition using time series data augmentation via a spatiotemporal multi-channel framework. *School of Integrated Circuit Science and Engineering, Tianjin University of Technology*, 2023. Correspondence: shgzhang@tjut.edu.cn; Tel.: +86-022-1362-207-2392.

[7] Kristyna Pijackova and Tomas Gotthans. Radio modulation classification using deep learning architectures. In *2021 31st International Conference Radioelektronika (RADIOELEKTRONIKA)*, pages 1–5, 2021.

[8] Dragoslav Stojadinovic, Prasad Netalkar, Carlos E. Caicedo Bastidas, Igor Kadota, Gil Zussman, Ivan Seskar, and Dipankar Raychaudhuri. A spectrum consumption model-based framework for dsa experimentation on the cosmos testbed. In *Proceedings of the 15th ACM Workshop on Wireless Network Testbeds, Experimental Evaluation CHaracterization*, WiNTECH'21, page 77–84, New York, NY, USA, 2021. Association for Computing Machinery.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[10] Denny Wild. Recurrent neural network tutorial, part 4 – implementing a gru/lstm rnn with python and theano. *WildML*, October 2015. Archived from the original on 2021-11-10.

[11] Fuqin Xiong. *Digital Modulation Techniques, (Artech House Telecommunications Library)*. Artech House, Inc., 2006.