WIRELESS SIGNAL CLASSIFICATION VIA MACHINE LEARNING

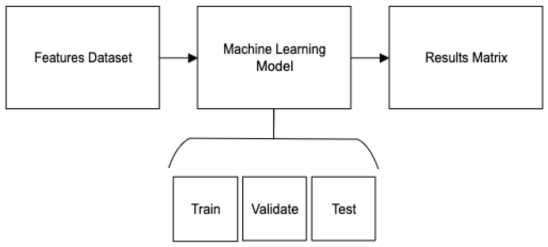
Jacob Ramey, Paras Goda, Curtis Zgoda, Anas Qutah

ECE 5424 Advanced Machine Learning - Proposal

October 11th, 2024

1. ABSTRACT

This study aims to classify radio signals based on their time series values or features. We will compare the performance of different models to determine the most effective approach. We will use publicly available datasets from DeepSig and MATLAB to extract key features, such as bandwidth, center frequency, and signal power (including peak, average, and standard deviation). Labeled data with these features will be used to train, validate, and test various machine-learning models as needed. The goal is to classify the modulation type of signals and compare the performance of several models to existing classifiers, such as those proposed by Flowers & Headly (2024) and Roy (2020). The performance of Support Vector Machines (SVM), Decision Trees (DTrees), Recurrent Neural Networks (RNNs), and Gradient-Boosted-Trees (XGBoost) will be evaluated. By comparing these models, we aim to find the most effective approach for modulation classification in wireless signals. The baseline for performance will be the RFML Deep Neural Network (DNN), which uses the I/Q data directly with labels such as FM, AM, QAM, PSK, etc (Flowers & Headly, 2024).



2. SPECIFIC AIMS

Digital wireless signals can be uniquely identified by their modulation scheme. Labeled I/Q Data will be used to classify a signal based on a set of features (i.e., frequency, bandwidth, power, etc.) using supervised machine-learning techniques. Different models such as Support Vector Machine (SVMs), Decision Trees, or various Neural Networks (NN) will be used and evaluated on a zero-one loss error rate. Data stored in I/Q format can be split, trained with, and then tested to validate machine learning models. The raw I/Q data may be directly input into a Neural Network. However, some other models, like SVM and decision trees, may require us to create engineered features as part of our work. For example, if a signal's bandwidth is 250kHz, we will classify it as 'FM.' LTE will be more suitable if its bandwidth is closer to 20MHz, 40MHz+. The RadioML data from DeepSig is provided with eleven modulation classifications, including FM, AM x 2, and 8 digitally modulated signals (CPFSK, GFSK, BPSK, QPSK, 8PSK, PAM4, QAM16, QAM64).

In the future, we can expand this classification using unsupervised models to identify new or unknown signals, as the industry is rapidly growing, and to create a future-proof algorithm. Clustering or using deep learning could show some advantages here. Another future improvement is to collect live over-the-air (OTA) data and feed it through the trained algorithm. Lastly, an expansion of this research could be to use computer vision to collect the data from a spectrum analyzer waveform vs digital I/Q samples. These future thoughts are outside this proposal's scope but are presented here for reader consideration.

3. BACKGROUND

In RF applications, a two-dimensional sinusoidal signal is sampled with a complex component, adding a third dimension. With time on one axis, the in-phase and quadrature-phase components are known as the I/Q data points. The formulas below give them, where A is the amplitude, f is the frequency, and ϕ is the phase as a function of time, *t*.

$$I(t) = A * \cos(2\pi f t + \phi) \qquad Q(t) = A * \sin(2\pi f t + \phi)$$

A time series vector of I/Q data points is provided along with their label. The dataset will be separated into training, validation, and testing sets (70/15/15 split). We will use the datasets provided by DeepSig to train our classifier. Features will also be extracted and labeled for training models requiring reduced dimensional inputs. By using existing data, we can focus on our model's performance.

Using multiple supervised machine learning algorithms described in section 4, we can compare the performance of various models. The permutations of each model, such as data set sizes or hyperparameters, will be adjusted to achieve the best accuracy.

MORE MODEL BACKGROUND

SVM

The support vector machine model looks to minimize the function:

$$\min_{\omega,b} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{n}\zeta_i \qquad \text{s.t. \\ } (\langle \chi_i, \omega\rangle + b)y_i \geq 1 - \zeta_i \text{ and } \zeta_i \geq 0$$

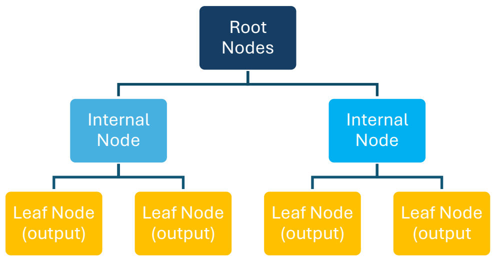
where we assume we can find a linear hyperplane as our decision boundary. ω is the vector of engineered features from our raw I/Q data set. Included is a slack cost/loss function,

$$C\sum_{i=1}^{n}\zeta_i,$$

where C is a constant parameter to adjust for best results; this cost function equals 0 if we find a linear decision boundary. It is expected based on the data, there will be a need to find a non-linear boundary by using kernel functions to map our data to a higher dimension. (GeeksForGeeks, 2024). The options to select from are polynomial, Gaussian RBF, or sigmoid. Multiple kernel functions will be evaluated to get the best accuracy which will have a different equation to minimize then above.

DECISION TREES (IBM, 2024)

A decision tree is a nonparametric supervised learning algorithm which is utilized for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. It has a hierarchical tree structure consisting of decision and leaf nodes, as shown in the figure below. Decision nodes contain conditions (from data features) to split the data, and the latter helps to decide the class of the data point.



At each Node, our model needs to learn which features to take and the corresponding threshold values to optimally slit the data. To achieve that, we will use Information Gain (IG) for splitting criterion. The model will choose the split (A) from the dataset (S) that maximizes the information gain (how much information a particular feature gives us about the output class)

$$IG\ (A,S) = H(S) - \sum_{j=1}^{v} \frac{|S_j|}{|S|} H(S_j)$$

H : Entropy, how uncertainty or impurity of the data points at each node for m classes. It is defined as:

$$H = \sum_{i=1}^{m} -P_i * log_2(P_i)$$

The Decision Tree model compares each possible split and takes the one that maximizes IG.

GRADIENT BOOSTED TREES

Gradient-boosted trees utilize a weak machine learning model (decision tree) and a strong machine learning model composed of multiple weak machine learning models. Each iteration trains a decision tree to predict the error of the current strong model (Google Developers, 2022). This is called the pseudo response.

$$F_{i+1} = F_i - f_i$$

F_i is strong model at step i

f_i is the weak model at step i

Shrinkage is a parameter that we can apply to the weak model to slow the learning rate of the strong model to reduce the effects of overfitting. Our new formula is

$$F_{i+1} = F_i - vf_i$$

The weak model can be trained to predict the gradient of the loss according to the strong model through

$$z_i = \frac{\partial L(y, F_i)}{\partial F_i}$$

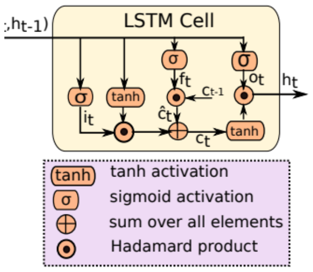
Where L(y,p) is the loss function with y being the label and p the prediction. Our strong model at step i now becomes

$$F_{i+1} = F_i - vz_i$$

RNN

For the RNN, we will implement an LSTM (long short-term memory) like the one used by Roy, as shown in **Figure 1** below. ReLU is a function that passes through the input if it is greater than 0; otherwise, it passes through 0 (Kızılırmak, 2023). **Figure 2** shows the plot of the ReLU function.

-$h = \max(0, a)$ where a($a = Wx + b$) is any real number



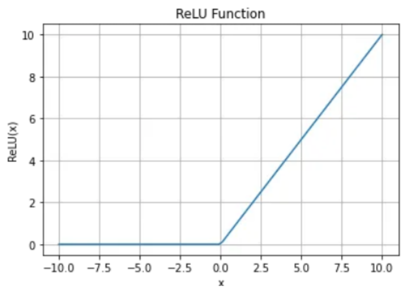


Figure 1 The architecture of an LSTM Cell **Figure 2 Plot of ReLU function**

The model will use two LSTM layers with 1024 and 256 units. Next, there will be two fully connected layers with 512 and 256 nodes, respectively. Batch normalization will be used on the output, then passed through a dense layer of 8 nodes. ReLU will be used as an activation function for the LSTM layers, and 'tanh' will be used for the dense layers. Finally, stochastic gradient descent (SGD) improves the model with categorical cross-entropy training. (Roy, 2020). A representation of RNN using LSTM is shown below in **Figure 3.**

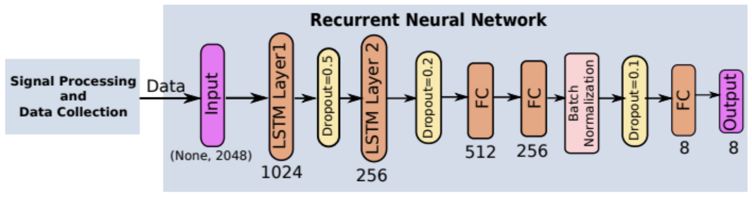


Figure 3: Layout of RNN-LSTM

4. RESEARCH DESIGN & METHODS

FEATURES

Using DSP techniques, we will engineer most, if not all, of the features below to allow for classification. This process will use existing tools that are readily available online. We hope to get good classification accuracy with just the primary features. However, more features are noted if improvement is necessary.

The main signal features for classification are shown below

FEATURE	DEFINITION	USAGE
Bandwidth	The range of frequencies occupied by a signal. Bandwidth helps distinguish between different modulation types, as modulated signals have characteristic bandwidths.	Some modulation schemes occupy a more comprehensive range of frequencies (e.g., OFDM), while others are narrower (e.g., AM, FM). Bandwidth can also help filter out unwanted noise or adjacent signals.
Center Frequency	The frequency at the center of the signal's bandwidth is critical in wireless communications and signal classification.	Different signals are transmitted at specific frequency bands (e.g., Wi-Fi, Bluetooth). Knowing the center frequency helps identify and classify modulation types based on expected frequency ranges.
Modulated	Indicates whether the signal is modulated and, if so, specifies the modulation type (e.g., AM, FM, PSK, QAM).	Modulation type provides essential information about how the signal is transformed, allowing models to differentiate between amplitude-modulated, phase-modulated, and frequency-modulated signals.
Signal Power (peak, average, std. Dev.)	Measures the strength of the signal. Peak power captures the highest value; average power gives the mean and standard deviation measures variability.	Power characteristics help distinguish between modulation schemes, which may show varying power levels across time or frequency. For instance, PSK has consistent power, while QAM shows significant variation in power.

Some other optional features are shown below

FEATURE	DEFINITION	USAGE
Phase Information	The phase of the I/Q samples provides insights into modulation schemes relying on phase shifts (e.g., PSK, QAM).	Extracting phase information helps the model learn phase relationships, aiding the classification of modulations like PSK and QAM.

FEATURE	DEFINITION	USAGE
Instantaneous Frequency	The rate of change of the signal's phase over time reflects how fast frequency content changes.	It helps distinguish frequency-modulated signals (e.g., FM) from others, as their frequency varies over time.
In-Phase (I) and Quadrature (Q) Statistics	Direct features derived from I/Q data, including mean, variance, skewness, and kurtosis.	These statistics can highlight differences between modulation schemes, revealing patterns like phase modulation signals showing unique I/Q distributions.
Signal-to-noise ratio (SNR)	The ratio of signal power to noise power indicates signal quality.	Incorporating SNR improves model performance in noisy conditions by enabling better decision-making.
Higher-Order Statistics (HOS)	Skewness, kurtosis, and other moments of the signal's statistical distribution.	It captures non-linearities and subtle variations in the signal that first-order statistics might miss, helping to classify modulation types accurately.
Time-Frequency Representations	Is frequency content over time using STFT, Wavelet Transform, or Wigner-Ville distribution.	It offers a detailed view of how the signal frequency changes over time, aiding in classifying non-stationary signals, such as modulated ones.
Hilbert Transform Features	Extracts the analytic signal, providing instantaneous amplitude, phase, and frequency.	It is helpful for signals where instantaneous characteristics, such as amplitude or frequency, are critical for correct classification.

METHODS

The team aims to test multiple machine learning algorithms, with each member responsible for one algorithm. All members will share the same dataset to compare methods. A zero-one loss error rate (in percentage) on the final test dataset, the generalization error, will be calculated. Another goal will be for the members to select and test various parameters for their model to minimize the loss function. We can use a training data set to train the models and a separate validation dataset to validate the error rate for each model. We will also use a third test dataset on each model for final comparison and research purposes.

Our data consists of 128 data points per sample or signal. This translates to 128 dimensions. A dimension reduction method could use DSP techniques to input I/Q data and find engineered features. For example, for center frequency and bandwidth, an effective way to handle small and width bandwidths involves an algorithm to find a peak and then locate the edges by finding the place when the power drops by 3dB from the peak. Modulation types, such as 'CW, pulsed, wideband, and hopping, can be left unadorned.'

THE TEAM WILL COMPARE THE FOLLOWING MODELS:

- **Support Vector Machines (SVM) - *Paras***: A Support Vector Machine (SVM) traditionally is a binary classifier. We can try to layer multiple SVMs in sequence to further classify the signal based on features, e.g., high frequency vs low frequency as a first layer, and then bin them into different bandwidths over multiple added layers. The SVM would learn to find patterns, such as differences in amplitude-frequency range or bandwidth, to differentiate between signals like a.m. FM or LTE by finding the best hyperplane boundary.

- **Decision Trees (DTrees) - *Anas***: They are simple and interpretable but prone to overfitting. A decision tree could identify signal types, like how a flow diagram works, using specific conditions at each split. A signal within a particular frequency range could be 'FM, AM, or LTE.' (IBM, 2024)

- **Gradient Boosted Trees (XGBoost) - *Curtis*:** Sequentially builds decision trees, where each tree corrects the errors of the previous ones. They are great at "capturing complex patterns and are resilient to outliers and noisy data" but "can sometimes struggle with overfitting and may not generalize well to unseen data." (Choudhary, 2023) Gradient-boosted trees can learn the intricate relationships within features, such as frequency ranges or amplitude variations, by combining weaker machine learning algorithms (decision trees) to reduce error. This makes them well-suited for distinguishing between AM, FM, or LTE signal types. Parameters such as the maximum depth of the tree and shrinkage rate may need to be adjusted to regulate training and reduce overfitting.

- **Recurrent Neural Networks (RNNs)—*Jacob*:** A recurrent neural network can simplify detection in time series data, a common scenario for live, real-time data sources like Software-Defined Radios (SDRs). According to Roy, RNNs are "useful for capturing and exploiting the temporal correlations of time series data." (Roy, 2020) This might be a viable choice since we can access the time series data. A neural network should allow us to use the raw I/Q data and see the effects without reducing the dimensions. There are three types of RNNs we can explore," long short-term memory (**LSTM**), (ii) gated recurrent unit (GRU), and (iii) convolutional long, short-term memory (ConvLSTM)." (Roy, 2020). I will implement the LSTM model.

Additional Models to Explore:

- **Random Forests (RF)**
- **K-Nearest Neighbors (KNN)**
- **Autoencoders**
- **Transformer Networks**
- **Ensemble Learning**
- **Gate Recurrent Unit, ConvLSTM**

5. DATASETS

Existing open-source datasets like RadioML will be used (DeepSig, 2024). Data, consisting of 11 modulation classes, is labeled and stored. Each modulation consists of samples at 20 different SNR levels (-20dB to 18dB, in 2dB steps). While there are 11 classes, based on model complexity, team members may fragment data into a smaller classification subset, such as 2-4 modulation classes. There are one thousand samples for each SNR/Modulation combination. This gives us 220,000 samples to train/validate/test on. Each sample consists of a 2 x 128-time series vector of I/Q data points. We assume our data is independent and identically distributed (i.i.d); all our data come from the same distribution. The dataset will be split into training, validation, and testing sets (e.g., 70/15/15 split). We also assume no cleanup or duplicates exist in our dataset since DeepSig provides it already clean. Typically, we must clean and pre-process the data to fit our model. Also, in the real world, data will come with some variations that must be considered. Adding noise and other physical parameters may affect signal quality and accuracy during classification. The datasets hold signals at various SNR levels, so it should also be able to classify them at a lower SNR. We hypothesize that the higher the SNR, the higher the confidence during classification. The lower SNR signals may be challenging for the classifier.

6. EXPERIMENTS

We will classify training datasets from RadioML and R2016A using the different ML models mentioned above and compare them.

6.1 EXPERIMENTAL SETUP

- **Data Splitting**: The dataset will be divided into training, validation, and testing sets (70/15/15 split). Each split will be uniformly distributed across modulation classes and SNR levels to ensure balanced evaluation.
- **Preprocessing**: The I/Q data may be normalized, then we will run feature extraction (e.g., computing signal power, bandwidth, or modulation properties) and calculate the FFT
- **Baseline Models**: The RadioML project provides an NN model for a performance baseline. We will compare its performance to that of the other model (Flowers & Headly, 2024).

6.2 TRAINING AND HYPERPARAMETERS

Training Procedure: Each team member will decide the right training procedure for their respective model.

- Number of epochs, batch size, and the optimizer used (e.g., Adam, SGD).
- Evaluation of loss functions: Each model will be evaluated on two loss functions, then the results will be compared.
  - Cross-Entropy Loss: This is a Log loss function inversely proportional to the probability of the predicted value.

$$L = -\frac{1}{m}\sum_{i=1}^{m} y_i \ . \ \log(y_i^*)$$

  - Hinge Loss: it measures error associated with the distance between the predicted output ($f(x)$) the desired output (y). Hinge loss penalizes the wrong predictions or the improper classification.

$$L = \max\big(0{,}1 - y * f(x)\big)$$

- Regularization methods (e.g., dropout for neural networks or pruning for decision trees) to prevent overfitting.
- **Hyperparameter Tuning**: Will we experiment with grid search, random search, or more advanced methods like Bayesian optimization to select the best hyperparameters (e.g., learning rate, tree depth, and kernel parameters)
- Kernel Functions and slack/cost variables for SVM

6.3 EVALUATION METRICS

- **Accuracy**: The percentage of correct classifications across the test set.
- **Precision, Recall, F1-Score**: Especially useful for imbalanced classes to better understand the model's ability to classify modulation types.
- **Confusion Matrix**: Provide a detailed look into misclassifications between modulation classes. This is what was used in the whitepaper for Over-the-Air Deep Learning Based Radio Signal Classification (O'Shea, Roy, & Clancy, 2018) and the RadioML repository so we can compare our results to theirs.

- **ROC/AUC**: Evaluate performance over varying thresholds; this is applicable if you are interested in probabilistic outputs.

6.4 EXPERIMENTS WITH SNR VARIATIONS

- **Performance Across SNR Levels**: An essential part of our experiments will focus on how models perform under different SNR levels. We will include a subsection that analyzes model performance across varying noise conditions and explains how models handle low-SNR samples versus high-SNR samples.

6.5 COMPARISON OF MODELS

- Accuracy comparison across all modulation classes.
- Memory and computational efficiency, especially for real-time implementation.
- Robustness to noise (particularly for models like RNNs or CNNs).
- **Statistical Significance**: If time allows, we will also consider performing statistical tests (e.g., t-test, Wilcoxon test) to determine whether model performance differences are significant.

7. TIMELINES

Date	Deliverable
10/11/2024	Decide on features, algorithms, and datasets.
10/18/2024	Each team member selects the model they will use
10/25/2024	Using existing datasets, extract features and labels to train our model.
11/01/2024	Train system on the data
11/08/2024	Evaluate and iterate on the model.
11/15/2024	Compare the model’s performance and start drafting the report.
11/22/2024	Work on drafting the final report.
11/29/2024	Finish drafting the final report.

8. REFERENCES

Bhuiya, S. (2020, October 31). *Disadvantages of CNN models*. Retrieved from Medium: https://sandeep-bhuiya01.medium.com/disadvantages-of-cnn-models-95395fe9ae40

Choudhary, J. (2023, September 20). *Mastering XGBoost: A Technical Guide for Machine Learning Practitioners*. Retrieved from Medium: https://medium.com/@jyotsna.a.choudhary/mastering-xgboost-a-technical-guide-for-intermediate-machine-learning-practitioners-f7ad167c6865

DeepSig. (2024, October 1). *Datasets*. Retrieved from DeepSig AI: https://www.deepsig.ai/datasets/

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification (2nd ed.).* New York: John Wiley & Sons, Inc.

Flowers, B., & Headly, W. C. (2024, September 24). *Radio Frequency Machine Learning (RFML) in PyTorch*. Retrieved from Github: https://github.com/brysef/rfml

GeeksForGeeks. (2024, October 10). *Support Vector Machine (SVM) Algorithm*. Retrieved from GeeksForGeeks: https://www.geeksforgeeks.org/support-vector-machine-algorithm/

Gish, H. (2006, August 06). A probabilistic approach to the understanding and training of neural network classifiers. *International Conference on Acoustics, Speech, and Signal Processing.* Albuquerque: IEEE Xplore. Retrieved from IEEE Explore: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=115636

Google Developers. (2022, September 28). *Gradient Boosted Decision Trees*. Retrieved from Google Developers.: https://developers.google.com/machine-learning/decision-forests/intro-to-gbdt

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning.* Springer.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation.* Prentice-Hall, Inc.

Hutomo, I. S. (2021, October 8). *https://github.com/alexivaner/Deep-Learning-Based-Radio-Signal-Classification*. Retrieved from Github: https://github.com/alexivaner/Deep-Learning-Based-Radio-Signal-Classification

IBM. (2024, October 9). *What is a decision tree?* Retrieved from IBM: https://www.ibm.com/topics/decision-trees

IBM. (2024, October 9). *What is random forest?* Retrieved from IBM: https://www.ibm.com/topics/random-forest

Keylabs AI. (2024, September 13). *K-Nearest Neighbors (KNN): Real-World Applications*. Retrieved from Keylabs AI: https://keylabs.ai/blog/k-nearest-neighbors-knn-real-world-applications/

Kızılırmak, S. (2023, February 11). *Rectified Linear Unit (ReLU) Function: Understanding the Basics*. Retrieved from Medium: https://medium.com/@serkankizilirmak/rectified-linear-unit-relu-function-in-machine-learning-understanding-the-basics-3770bb31c2a8

O'Shea, T. J., Roy, T., & Clancy, T. C. (2018). Over-the-Air Deep Learning Based Radio Signal Classification. *IEEE Journal of Selected Topics in Signal Processing*, 168-179.

O'Shea, T., & West, N. (2016, September 6). *Radio Machine Learning Dataset Generation with GNU Radio.* Retrieved from Gnuradio: https://pubs.gnuradio.org/index.php/grcon/article/view/11/10

Qiu, Y., Zhang, J., Chen, Y., & Zhang, J. (2023, April 20). *Radar2: Passive Spy Radar Detection and Localization Using COTS mmWave Radar*. Retrieved from IEEE Explore: https://ieeexplore.ieee.org/document/10105863

Roy, D. (2020). *MACHINE LEARNING BASED RF TRANSMITTER CHARATERIZATION IN THE*. Retrieved from Northeastern University College of Engineering: https://www1.coe.neu.edu/~droy/Doctoral_Dissertation_Debashri_Roy.pdf

Virginia Tech. (2024, October 9). ECE5424 LN12.pdf. Blackburg, VA, United Stated of America.

Viso.AI. (2024, October 9). *Ensemble Learning: A Combined Prediction Model (2024 Guide)*. Retrieved from Viso.AI: https://viso.ai/deep-learning/ensemble-learning/