CLOUD-D RF: Cloud-based Distributed Radio Frequency Heterogeneous Spectrum Sensing

Dylan Green*, Caleb McIrvin*, River Thaboun*, Cora Wemlinger*, Joseph Risi†, Alyse M. Jones*, Maymoonah Toubeh*, William C. Headley*

* Virginia Tech National Security Institute

† Amazon Web Services

{dylang,alysemjones,mtoubeh,cheadley}@vt.edu

Abstract

In wireless communications, collaborative spectrum sensing is a process that leverages radio frequency (RF) data from multiple RF sensors to make more informed decisions and lower the overall risk of failure in distributed settings. However, most research in collaborative sensing focuses on homogeneous systems using identical sensors, which would not be the case in a real world wireless setting. Instead, due to differences in physical location, each RF sensor would see different versions of signals propagating in the environment, establishing the need for heterogeneous collaborative spectrum sensing. Hence, this paper explores the implementation of collaborative spectrum sensing across heterogeneous sensors, with sensor fusion occurring in the cloud for optimal decision making. We investigate three different machine learning-based fusion methods and test the fused model's ability to perform modulation classification, with a primary goal of optimizing for network bandwidth in regard to next-generation network applications. Our analysis demonstrates that our fusion process is able to optimize the number of features extracted from the heterogeneous sensors according to their varying performance limitations, simulating adverse conditions in a real-world wireless setting.

CCS Concepts

• Computing methodologies → Distributed artificial intelligence; **• Networks → Cloud computing**.

Keywords

spectrum sensing, heterogeneous, distributed, cloud, wireless communications, artificial intelligence

ACM Reference Format:

Dylan Green*, Caleb McIrvin*, River Thaboun*, Cora Wemlinger*, Joseph Risi†, Alyse M. Jones*, Maymoonah Toubeh*, William C. Headley*. 2024. CLOUD-D RF: Cloud-based Distributed Radio Frequency Heterogeneous Spectrum Sensing . In *Proceedings of The 30th Annual International Conference on Mobile Computing and Networking (ACM MLNextG '24).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3636534.3698249

ACM MLNextG '24, November 18, 2024, Washington, DC

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM

https://doi.org/10.1145/3636534.3698249

1 Introduction

Collaborative sensing has become increasingly common across domains within the last decade due to its extensive applicability, such as capturing data from sensors over a wide area or obtaining multiple "views" of an environment [6]. It leverages data from multiple sensors, a concept more formally known as sensor fusion, so that a system is able to make more informed decisions and lower the overall risk of failure with more sources of information [3].

One method for implementing collaborative sensing that is of particular interest is offloading sensor fusion via cloud-based processing rather than on-device processing. A cloud-based architecture such as this reduces the processing and network burden of edge devices, allowing these devices to maintain low size, weight, and power (SWaP) requirements, and reduce the complexity of the network topology. With this, edge devices can be deployed at lower cost and for longer periods of time [9]. Furthermore, the elasticity of the cloud enables signal processing compute to scale up and down based on network traffic, while also bolstering the network security with end-to-end Transport Layer Security (TLS) encryption and access control policies.

In practice, collaborative sensing in the cloud is limited by real world impairments, such as limited network bandwidth available to edge devices. Thus, fusion in the cloud should aim to understand the resource constraints of edge devices as it learns. Additionally, almost all research in collaborative sensing focuses on homogeneous systems with identical sensors or systems where the data from all sensors are weighted equally [6]. However, in wireless communications, especially in mobile settings, devices have varying processing and computing capabilities and are spread throughout different geographical locations to measure signal activity in the RF environment, leading to major differences in sensor results. Thus, heterogeneous collaborative spectrum sensing is required to capture how each sensor measures a signal across spatial and computing differences. Therefore, the goal of this paper is to study the effectiveness of three machine learning-based RF sensor fusion methods, using heterogeneous deep learning-based RF sensors at the edge. We validate our methods by leveraging the cloud to perform sensor fusion to improve the performance of modulation classification, a common use case for spectrum sensing, where there exists resource constraints at edge devices and heterogeneity among RF sensors. The approaches detailed in this work may then be applied to more challenging spectrum sensing tasks in the future, including specific emitter identification. Our contributions are summarized as follows.

- We present a novel framework that utilizes ML-based RF sensor fusion in the cloud, utilizing the features for modulation

classification from the layers of the edge devices' ML model as input.
- We optimize the machine learning-based RF sensor fusion by performing intelligent feature selection before fusion through Recursive Feature Elimination (RFE) and Reinforcement Learning (RL) methods.

2 Background

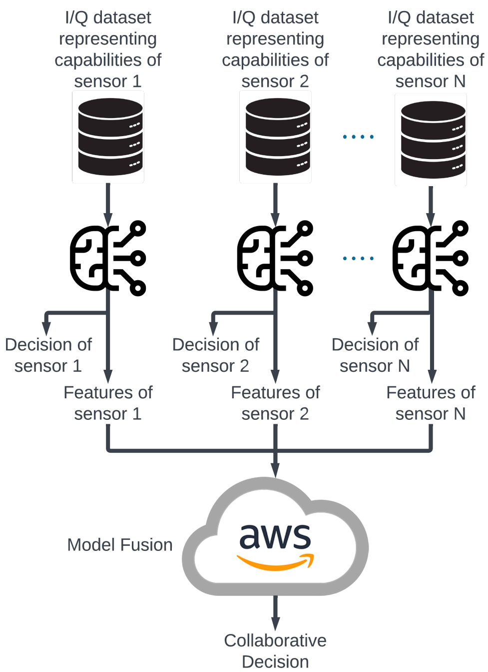


Figure 1: Diagram of the CLOUDD-RF System Model.

We propose a sensor fusion approach, shown in Figure 1, applied and analysed for the radio frequency (RF) modulation classification problem to address the need for improved fusion and SWaP among resource-constrained heterogeneous RF sensors in a distributed wireless communication setting. We use deep learning-based edge sensor models to perform modulation classification based on their heterogeneous views and device capabilities. Prior work shows that deep learning-based modulation classification, specifically using convolutional neural networks (CNN), are effective at identifying raw RF signal data, often referred to as I/Q data [5]. Thus, our system utilizes raw I/Q inputs to train edge devices for modulation classification using a CNN. However, a single CNN is unlikely to be effective at sensing all possible scenarios as it assumes homogeneity of the sensors. In a real-world RF environment, the physical location of sensors causes each to observe a different view of the signal, hence the proposed need for intelligent fusion. Therefore, our work investigates the effectiveness of combining the heterogeneous sensing results using a machine learning-based fusion approach.

A relevant prior work employs a collaborative spectrum sensing system which dynamically assigns weight factors to clusters based on their contributions after each sensing process to determine the influence each cluster will have on the final collaborative decision [8]. However, this system focuses on how implementing different sensing durations and different fading channels will improve collaborative sensing performances, whereas our paper looks at the effects of a wider set of nuisance parameters. To provide more robust performance to sensor differences, our models include variations in signal-to-noise ratio (SNR), bandwidth, carrier frequency, in addition to duration. We also investigate a more intelligent feature ranking using Recursive Feature Elimination (RFE) [4] and directly use raw I/Q data for training.

Another prior work investigates how different machine learning algorithms evaluate data collected by a spectrum environment awareness system built on cloud-based services [7]. Their results verified the power of the XGBoost algorithm with large datasets for efficiency and accuracy. Our work also leverages the XGBoost algorithm in fusing the latent space features of our individual models together. However, prior work in RF modulation classification and specific emitter identification (SEI) [10, 12] shows that training on the extracted features of a model, rather than raw data, can help classification performance. Thus, the inputs to all of our tested fusion approaches come from deep learning-based feature extraction. We show that fusion on the features, rather than raw inputs, provides an intelligent first step at pruning fusion inputs and also reduces bandwidth requirements of the edge devices. These features provide a condensed, processed form of the raw inputs, of which an arbitrary amount may be selected for fusion based on their contribution to the final classification. This reduction in total data required to transmit to the cloud makes fusion on features advantageous compared to fusion on raw data in the resource-constrained environments edge devices operate in.

Feature selection for effective sensor fusion is an important component of our study. A prior work proposes using a feature selection algorithm to address reduced accuracy in the cloud due to large-scale datasets [11]. We employ a similar logic to our fusion approach, recognizing that not every RF device can reliably send all their data to the cloud. Thus, we implement ML-based algorithms, specifically RFE, to select the top-N features to reduce the bandwidth burden on RF edge sensors.

3 System Model

Figure 1 provides an overview of our system model. The input to our system are multiple augmented datasets, representing heterogeneous sensor inputs, each representing a different view of the RF spectrum environment. The edge computing involves multiple CNNs, each trained on one of the heterogeneous datasets and having a unique architecture. Features are extracted from each CNN by taking the outputs of the intermediate hidden layers. Specifically, instead of taking the final modulation class decision, the "features" are taken from the outputs of the second to last fully-connected layer.

Our system model aims to investigate how features extracted from each CNN at the edge devices can be fused together in the cloud to perform collaborative decision making. To ensure each network was unique, four teams created their own CNN to classify signal modulation type, without access to the other teams' models to maintain heterogeneity. Table 1 shows the varying model architectures of the four distinct CNNs trained on the heterogeneous

Table 1: Sensor models

Sensor	# Conv	# Linear	Max Pool	Batch Norm	Activation(s)
1	3	4	Yes	No	ReLU
2	4	2	Yes	Yes	ReLU
3	3	4	No	No	ReLU, Tanh
4	4	3	Yes	No	ReLU

Table 2: Sensor dataset parameters

Sensor	Seed	Duration	Bandwidth	Carrier Frequency Offset	SNR
1	1234	2048	[0.1,0.5]	[-0.01, 0.01]	[5, 15]
2	4321	1024	[0.25,0.5]	[-0.01, 0.01]	[0, 20]
3	1337	512	[0.25,0.5]	[-0.05, 0.05]	[5, 20]
4	7331	256	[0.25,0.5]	[-0.01, 0.01]	[5, 20]

datasets. Using a fusion strategy allows the system to overcome the limitations of each individual edge device, allowing for greater accuracy and confidence in the final classification.

The datasets for each team's model contained signals from a range of modulation types and nuisance parameters, like signal bandwidth and SNR, as shown in Table 2. For example, the dataset for team 1's model had an observation length of 2,048 samples, while team 3's dataset only had an observation length of 512 samples. In effect, this simulates that the device team 1's model is placed on can store more samples before running out of space in its buffer. With this approach, the system can easily emulate sensors and edge devices with differing performance capabilities, i.e. heterogeneity. Other examples of this include emulating sensors that can listen for larger bandwidths or are more tolerant to changes in the target signal's carrier frequency.

Another notable aspect of our system is that sensor fusion occurs in the cloud. This allows the system to support a large, dynamic number of edge devices limited by computational and network resources. In our system, features used in the classification of a given signal can be extracted from the CNNs on each edge device and transmitted to a fusion algorithm running in the cloud. The number of features transmitted from each edge device must be set based on a balance of information provided to the fusion algorithm and network bandwidth available to the device. The fusion strategies used to address this are explained in detail in Section 4.

4 Fusion Approach

Fusion is implemented in the cloud by combining the features extracted from each model to train a new centralized machine learning model. This approach is particularly advantageous for collaborative sensing with heterogeneous sensors, as it supports an arbitrary number of features extracted from each model and allows for dynamic weighting of features. These benefits are possible because the model used to perform the fusion will learn which features contribute most to the final classification, instead of weighting each feature equally. More practically, it allows for sensors that provide different "views" of the original signal and situations where some sensors perform better at signal classification than others. For example, this could include scenarios where a sensor is physically closer to the signal source or better at classifying signals at lower SNR.

Table 3: Number of extracted features per model.

Sensor	Number of Extracted Features
1	65
2	512
3	64
4	256

There are various methods to optimize the fusion process, with the primary goal in this paper being to reduce the number of features transmitted by each edge device to the cloud. This is important for NextG networks as they are expected to have a massive number of connected devices; efficient use of network bandwidth is critical. We utilize two forms of feature reduction to find the top contributing features and create a bandwidth-optimized fusion process: one setup using recursive feature elimination (RFE) and one using reinforcement learning.

4.1 Baseline Fusion Method

First, a baseline method is needed to better evaluate the performance of these feature reduction techniques. Our baseline method performs fusion using all features extracted from the individual models. This ensures that the fused model gets the most information possible, but is wasteful in terms of network bandwidth due to the likelihood of redundant or unnecessary features being transmitted. We implemented this method with an XGBoost classifier [2], using the histogram method, fitting it to the full set of extracted features. After initial analysis for our problem, the second to last fully-connected layer in all CNN models was used to extract all features as this is where the highest-level features will have been learned. There are a total of 897 extracted features used to train the classifier using this method, with a breakdown of features per model shown in Table 3.

4.2 Recursive Feature Elimination Fusion Method

The first machine learning-based fusion method we use is XGBoost with Recursive Feature Elimination (RFE). This follows the same process as the baseline, but adds an RFE feature reduction step before training the XGBoost classifier. The RFE process starts with the full feature set, then decreases by powers of 2 from 512 features, selecting the top features for that round and keeping that subset to select from in the next round. This process stops once the desired number of top contributing features has been found. In this paper, our RFE fusion method stops once the top 16 features have been found. We choose to stop at 16 after initial analysis of the full feature set resulted in no substantial improvements in accuracy beyond that number, proving diminishing returns.

4.3 RL-based Fusion Method

The second method we use for fusion uses Reinforcement Learning (RL). This is similar to RFE, except it uses RL to iteratively weight features by their contribution to the final classification. An RL-based approach is expected to be advantageous for real-time wireless sensing in dynamic environments, as it can intelligently select features

Table 4: Individual Model Validation Accuracy

Sensor	2-ASK	4-ASK	8-ASK	BPSK	QPSK	16-QAM	Tone	P-FMCW	Overall
1	99.92%	95.42%	74.76%	85.36%	99.78%	98.89%	99.97%	100%	94.30%
2	99.00%	85.37%	84.59%	99.89%	98.90%	97.30%	99.71%	99.71%	95.53%
3	77.44%	65.88%	47.38%	19.98%	63.76%	37.57%	94.00%	96.32%	62.95%
4	96.87%	50.76%	82.78%	96.98%	93.07%	87.68%	78.82%	87.08%	84.12%

under varying channel conditions. Our implementation uses the OpenAI Gym library and Proximal Policy Optimization (PPO) algorithm. The environment is set up such that, over 4,096 timesteps, the RL agent selects 16 features from the full feature set to perform fusion with, keeping a fair comparison with the RFE approach. For a given timestep, the agent's action is to add a single feature to its list of top 16 features, resetting the list once it becomes full. The agent's observation consists of 16 features it has chosen, while the reward is the accuracy obtained by training a new XGBoost classifier using the histogram tree method and five estimators on those features. As the agent selects greater-contributing features, its reward will increase in turn.

By discovering features that increase its reward the most, the agent is able to learn the top contributing features and converge to a selection similar to the one produced by RFE. However, because an RL agent is able to explore its environment, there is much greater flexibility in what kinds of data it can be given. In the context of spectrum sensing, an RL agent would be able to more intelligently select which features to transmit when given additional information about the channel conditions detected by each sensor. For example, the agent could place greater importance on features from a sensor receiving signals with a higher SNR than others, due to physical proximity to the signal source or placement of obstacles in the environment. This allows an RL feature reduction process to have much greater flexibility in its application compared to RFE.

5 Results

5.1 Simulation Settings

To simulate data collected by heterogeneous sensors, we specify four sets of generation parameters, as shown in Table 2. By carefully modifying the individual characteristics of the receivers, we create a scenario where a fused model may be able to take advantage of the differing information from the individual models to improve overall performance. For example, Model 2 is the only model trained on very low SNR data ($0 - 5dB$), so it is expected to perform more effectively when given noisy I/Q data. Whereas, Model 1 was trained on low-bandwidth data ($0.1 - 0.25$), so it is likely to be proficient when given poorly isolated I/Q data. This is expected to encourage the fusion approaches to incorporate features from multiple models, resulting in a more robust model with increased performance.

To ensure consistency between sensors, we generate 105,000 "ideal" training examples evenly distributed across eight distinct modulation schemes: 16-QAM, 2-ASK, 4-ASK, 8-ASK, BPSK, QPSK, P-FMCW, and Constant Tone. The I/Q samples are then artificially degraded from the "ideal" high SNR, full duration, high bandwidth, and nominal carrier frequency to match the individual sensor dataset parameters. By providing individual sensors degraded I/Q samples artificially generated from the same initial "ideal" data, we ensure data consistency across models, necessary when combining extracted features for sensor fusion, as described in Section 4.

Each model was trained over ten epochs as a CNN classifier over modulation schemes, using the Adam optimizer, with a learning rate of 0.001. Note that due to the heterogeneous nature of the sensors being modeled and the resulting lack of control over model types in the problem being modeled, any fusion approach must make no assumptions on the characteristics of the individual models, including their being trained identically or even being trained to convergence. It is also important to note that although we use CNNs, our approach is meant to be model agnostic, and can be applied to other deep learning or machine learning-based architectures. We base our approach on the current state-of-the-art.

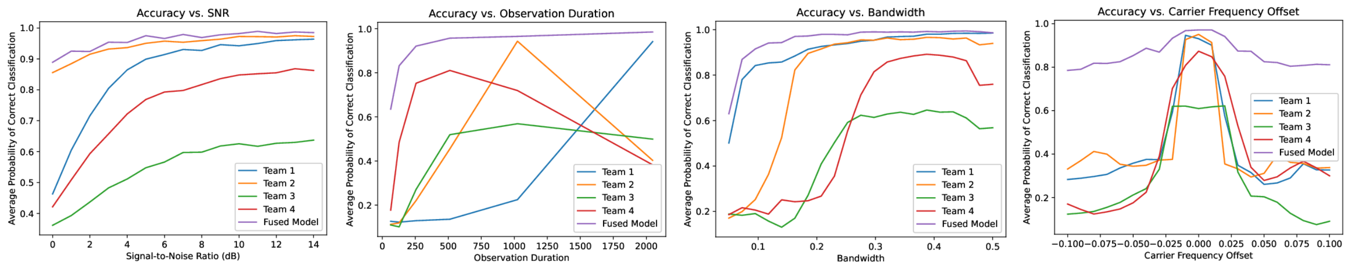
5.2 Individual Model Results

To verify individual models' performance, we generate accuracy values over the modulation schemes being classified with a validation dataset. Results for individual models are given in Table 4.

As we see from the table, all of the models perform well on certain modulation schemes, such as P-FMCW and Constant Tone. We expect this phenomenon is primarily due to the inherently distinctive nature of the samples generated using these individual modulation schemes, making them easier to separate at lower SNR values. Similarly, many of the models struggle to distinguish between 2-ASK, 4-ASK, and 8-ASK, likely due to the inherent similarity of these transmission schemes.

There are also clear differences in performance across the models. Team 1, 3, and 4's models have some trouble classifying BPSK and QPSK signals, whereas Team 2's model did so with near perfect accuracy. This is likely due to these signals being more difficult to distinguish at lower SNRs, along with Team 2's model being trained on signals reaching an SNR of 0 instead of a minimum of 5 like the other models. Team 4's model was the only one to have difficulty classifying Constant Tone and P-FMCW signals. We believe this may be due to Team 4's model having the shortest observation.

In addition, we contrast model performance across ranges of nuisance parameters, as shown in Figures 2a through 2d. For each nuisance parameter, we calculate model accuracy over 15,000 testing samples at a series of predefined levels to visualize model performance across values of the parameter. Classification performance increases with signal bandwidth and SNR, and worsens as carrier frequency deviation increases for all models. Team 2's model performs with above 85% classification accuracy for all SNRs, as it was trained on the widest SNR range. Similarly, Team 1's model was trained on the widest range of signal bandwidths and performs well for most bandwidths tested. Even though Team 2's model was trained on signals with a minimum bandwidth of 0.25, its accuracy



(a) SNR **(b) Duration** **(c) Bandwidth** **(d) Carrier Frequency Offset**

Figure 2: Accuracy for varying nuisance parameters

only begins to drop off at a bandwidth of approximately 0.18. Model behavior is not as consistent in Figure 2b; Team 2 and 4's models experience a drop in performance after reaching their training data's observation duration whereas Team 3's model plateaus. This may be due to the training implementation used, as the training data used for models 2 through 4 are zero-padded to reach an observation duration of 2048 samples. Extending the I/Q data into this region which was filled with zeros during training does not align with what the models had learned, thus reducing classification accuracy. Despite the aforementioned differences, all models have reduced performance on data with an observation duration lower than that of their training data.

5.3 Fused Model Results

Table 5 shows classification accuracy results across various modulation types for each of the fused models. These results were produced using the same validation dataset as the confusion matrices for the non-fused models, taking features from the second-to-last fully-connected layer of each edge model and feeding those into the fused models as input. For each method, the fused models attained greater classification accuracy than that of all individual models, as can be seen in Fig. 2.

The results from the RFE and RL fused models are striking, given how drastic a reduction there was in feature space: going from 897 to 16 features. Overall, classification accuracy dropped by only 1.4% to a minimum of 96.29% with the model produced by the RL fusion method. This means a large majority of the features extracted from the individual models are unnecessary or redundant when making the final classification. Reducing the number of features transmitted to 16 constitutes a 98% decrease in network bandwidth required to transmit feature data from edge devices to the cloud, a highly beneficial trade-off for devices in bandwidth-constrained environments.

5.4 Fusion Analysis

To better understand these results, we also performed an analysis of how each team's model contributed to the fused models. We accomplished this by following the same training process as in the baseline fusion method, but replacing the XGBoost classifier with a traditional decision tree classifier. We make use of the sci-kit learn implementation [1]. A list of importance values corresponding to each feature can be read directly from the decision tree, with each value representing the proportion of the final classification that feature contributes. Figure 3 shows the features that contribute the most to the fused model, color coded by which team's model they originate from.

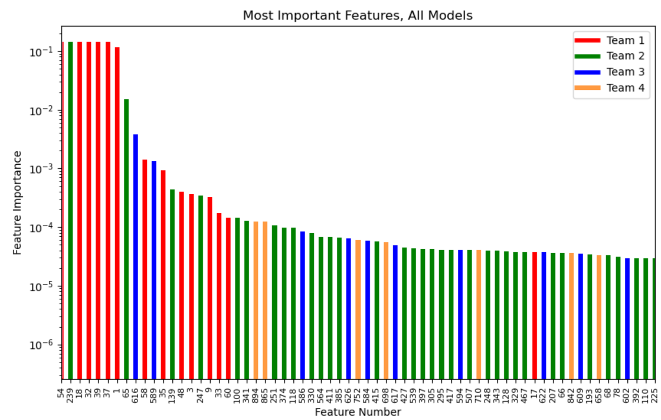


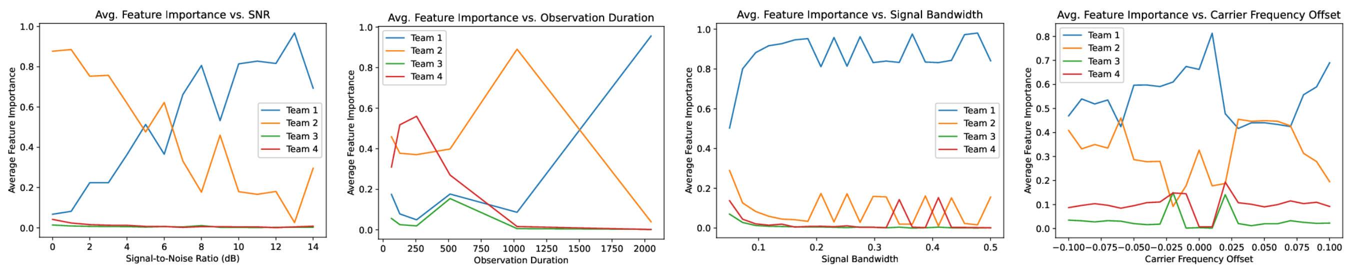
Figure 3: Top Features, All Models

As can be seen in Figure 3, it is clear that Team 1's model contributes the vast majority of the top features. Approximately 98% of the total contribution is made by only seven features.

Last, we perform an analysis of the proportion each team's model contributed to the fused model with respect to the various nuisance parameters. Our approach to this follows the same training process as in the baseline fusion method, but uses a testing dataset where one nuisance parameter is kept constant at a time. The objective in this test is to confirm that fusion is able to dynamically weight features in scenarios where some models perform better than others. For example, if one model performs better at low SNRs than others, there should be a corresponding rise in average feature importance compared to the other models. This is indeed what we see in Figure 4a, as Team 2's model takes up the majority of the feature contribution at low SNRs, while Team 1's model takes the majority at SNRs between 7 and 14. This holds true across the board, as Table 2 shows that Team 1's model was trained on signals with SNRs between 5 and 15, whereas Team 2's model was trained on SNRs between 0 and 20.

Table 5: Fused Model Validation Accuracy

Method	2-ASK	4-ASK	8-ASK	BPSK	QPSK	16-QAM	Tone	P-FMCW	Overall
Baseline	99.97%	93.35%	88.73%	99.97%	99.76%	99.78%	100%	100%	97.69%
RFE	99.97%	93.11%	88.24%	99.84%	99.70%	99.84%	99.95%	100%	97.46%
RL	99.55%	91.38%	90.97%	99.46%	97.28%	99.73%	99.01%	99.47%	96.29%



(a) SNR **(b) Duration** **(c) Bandwidth** **(d) Carrier Frequency Offset**

Figure 4: Average feature importance for each model over the nuisance parameters, detailing the key success and failure areas of the models that impact fusion

6 Conclusions and Future Work

In this paper, we propose a method of implementing collaborative sensing in the cloud given heterogeneous sensors placed on edge devices, applying our system model to the RF signal modulation classification problem. Collaborative sensing is done using a machine learning-based approach, specifically XGBoost, RFE, and RL. Heterogeneous edge sensing is simulated using four separate CNN architectures each trained on its own synthetically augmented dataset. We propose the use of features extracted from the CNNs as input to the fusion models. When comparing intelligent feature selection with a baseline which uses all possible input features before fusion, we found negligible improvement after a certain number of features, proving diminishing returns. Our approach, as well as our analysis, provides insights into a state-of-the-art solution to heterogeneous collaborative sensing in a central cloud.

To further test the capabilities of our heterogeneous sensor fusion approach, it would be valuable to test fusion approaches in scenarios where no individual model performed well. Also, given the synthetic data used in our analysis, a natural extension of our work is testing on live RF data. While RFE was more effective than the reinforcement learning fusion approach in this initial approach, we suspect that the RL approach may be more effective in an online, dynamic setting. Developing an environment where properties of individual sensor's captures change over time would allow us to more accurately assess the ability of the RL model to respond to changes in the environment. Finally, we plan to extend our fusion approach from modulation classification to more challenging domain-specific classification tasks, including specific emitter identification.

References

[1] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 108–122.

[2] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. ACM, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785

[3] Wilfried Elmenreich. 2002. *Sensor Fusion in Time-Triggered Systems*. Ph. D. Dissertation. Vienna University of Technology.

[4] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002. Gene selection for cancer classification using support vector machines. *Machine learning* 46 (2002), 389–422.

[5] Steven C Hauser, William C Headley, and Alan J Michaels. 2017. Signal detection effects on deep neural networks utilizing raw IQ for modulation classification. In *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*. IEEE, 121–127.

[6] Shibo He, Kun Shi, Chen Liu, Bicheng Guo, Jiming Chen, and Zhiguo Shi. 2022. Collaborative Sensing in Internet of Things: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials* 24, 3 (2022), 1435–1474. https://doi.org/10.1109/COMST.2022.3187138

[7] Li Li and Amir Ghasemi. 2019. IoT-Enabled Machine Learning for an Algorithmic Spectrum Decision Process. *IEEE Internet of Things Journal* 6, 2 (2019), 1911–1919. https://doi.org/10.1109/JIOT.2018.2883490

[8] Xingyan Liang, Zhiquan Bai, Peng Gong, Bing Wang, and Kyungsup Kwak. 2012. Weighted collaborative spectrum sensing with different sensing durations. In *2012 International Symposium on Communications and Information Technologies (ISCIT)*. 268–273. https://doi.org/10.1109/ISCIT.2012.6380904

[9] Dave McCarthy. 2020. AWS at the Edge: A Cloud Without Boundaries. *International Data Corporation Accessed via https://d1. awsstatic. com/IoT/IDC-AWS-at-the-Edge-White-Paper. pdf* 1, 1 (2020), 1–13.

[10] Timothy J O'Shea, Nathan West, Matthew Vondal, and T Charles Clancy. 2017. Semi-supervised radio signal identification. In *2017 19th International Conference on Advanced Communication Technology (ICACT)*. IEEE, 33–38.

[11] Wenjuan Wang, Xuehui Du, and Na Wang. 2019. Building a Cloud IDS Using an Efficient Feature Selection Method and SVM. *IEEE Access* 7 (2019), 1345–1354. https://doi.org/10.1109/ACCESS.2018.2883142

[12] Lauren J Wong, William C Headley, Seth Andrews, Ryan M Gerdes, and Alan J Michaels. 2018. Clustering learned CNN features from raw I/Q data for emitter identification. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE, 26–33.