Ensemble Wrapper Subsampling for Deep Modulation Classification

Sharan Ramjee, *Student Member, IEEE*, Shengtai Ju, *Student Member, IEEE*,
Diyu Yang, *Student Member, IEEE*, Xiaoyu Liu, *Student Member, IEEE*,
Aly El Gamal, *Senior Member, IEEE*, and Yonina C. Eldar, *Fellow, IEEE*

***Abstract*—Subsampling of received wireless signals is important for relaxing hardware requirements as well as the computational cost of signal processing algorithms that rely on the output samples. We propose a subsampling technique to facilitate the use of deep learning for automatic modulation classification in wireless communication systems. Unlike traditional approaches that rely on pre-designed strategies that are solely based on expert knowledge, the proposed data-driven subsampling strategy employs deep neural network architectures to simulate the effect of removing candidate combinations of samples from each training input vector, in a manner inspired by how wrapper feature selection models work. The subsampled data is then processed by another deep learning classifier that recognizes each of the considered 10 modulation types. We show that the proposed subsampling strategy not only introduces drastic reduction in the classifier training time, but can also improve the classification accuracy for the considered dataset. An important feature herein is exploiting the transferability property of deep neural networks to avoid retraining the wrapper models and obtain superior performance through an ensemble of wrappers over that possible through solely relying on any one of them.**

***Index Terms*—Deep learning, wireless modulation classification, data-driven subsampling.**

I. Introduction

Automatic modulation classification plays an important role in modern wireless communications. It finds applications in various commercial and military areas. For example, Software Defined Radios (SDR) use blind recognition of the modulation type to quickly adapt to various communication systems, without requiring control overhead. In military settings, friendly signals should be securely received, while hostile signals need to be efficiently recognized typically without prior information. Under such conditions, advanced real time signal processing and blind modulation recognition techniques are required. Modulation recognition is also important for identifying the source(s) of received wireless signals, which can enable various intelligent decisions for a context-aware autonomous wireless communication system.

A typical modulation classifier consists of two steps: signal preprocessing and classification algorithms. Preprocessing tasks may include noise reduction and estimation of signal parameters such as carrier frequency and signal power. In the second step, three popular categories of modulation recognition algorithms are conventionally selected: Likelihood-Based (LB) [2]–[7], Feature-Based (FB) [8]–[13] or using an Artificial Neural Network (ANN) [14]–[18]. The first compares the likelihood ratio of each possible hypothesis against a threshold, which is derived from the probability density function of the observed wave. Multiple likelihood ratio test (LRT) algorithms have been proposed: Average LRT [19], Generalized LRT [20], Hybrid LRT [7] and quasi-hybrid LRT [2]. For the FB approach, the classification decision is based solely on a subset of selected features. Both LB and FB methods require precise estimates in the first step and have only been derived to distinguish between few modulation types [4], [19], [21], [22]. ANN structures such as Multi-Layer Perceptrons (MLP) have been widely used as modulation type classifiers [14]. Traditional MLP performs well on modulation types such as AM, FM, ASK, and FSK. Recent work has shown that deep neural networks with cutting-edge structures could greatly improve the classification process (see, e.g., [23] and [24]), and deliver superior performance to state-of-the-art methods by enabling modulation recognition in the presence of a wide variety of modulation types, and with little or no requirements on the preprocessing step.

Deep neural networks have played a significant role in the research domain of video, speech and image processing over the past few years. The recent success of deep learning algorithms is associated with applications that suffer from inaccuracies in existing mathematical models and enjoy the availability of large data sets. Recently, the idea of deep learning has been introduced for modulation classification using a Pure

Manuscript received May 1, 2020; revised March 17, 2021; accepted August 19, 2021. Date of publication August 30, 2021; date of current version December 9, 2021. This work was supported in part by DARPA and AFRL under grant no. 108818. This article was presented in part at the Asilomar Conference on Signals, Systems and Computers [1]. The associate editor coordinating the review of this article and approving it for publication was T. O'Shea. *(Corresponding author: Aly El Gamal.)*

Sharan Ramjee was with the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA. He is now with Computer Science Department, Stanford University, Stanford, CA 94305 USA (e-mail: sramjee@stanford.edu).

Shengtai Ju, Diyu Yang, and Aly El Gamal are with the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: ju10@purdue.edu; yang1467@purdue.edu; elgamala@purdue.edu).

Xiaoyu Liu was with the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA. He is now with Oracle Inc., Austin, TX 78741 USA (e-mail: liuxiaoyu4321@gmail.com).

Yonina C. Eldar is with the Department of Math and Computer Science, Weizmann Institute of Science, Rechovot 7610001, Israel (e-mail: yonina@weizmann.ac.il).

Digital Object Identifier 10.1109/TCCN.2021.3108809

Convolutional Neural Network (PCNN) for distinguishing between 10 different modulation types [23]. Simulation results show that a PCNN not only demonstrates improved accuracy over current expert-based approaches, but also provides more flexibility in detecting various modulation types. Other deep neural network architectures like the Residual Network (ResNet) [25] were also recently introduced to strengthen feature propagation in deep neural networks by creating shortcut paths between different layers in the network. By adding the bypass connections, an identity mapping is created, allowing the deep network to learn simple functions. A ResNet architecture was shown to be successful for distinguishing between 24 different modulation types in [26]. A Convolutional Long Short-term Deep Neural Network (CLDNN) was also recently introduced in [27], by combining the architectures of the PCNN and the Long Short-Term Memory (LSTM) into a deep neural network and taking advantage of the complementarity of PCNNs, LSTMs, and conventional deep neural network architectures. The LSTM unit is a memory unit of a Recurrent Neural Network (RNN). RNNs are neural networks with memory that are suitable for learning sequence tasks such as speech recognition and handwritten recognition. LSTM optimizes the gradient vanishing problem in RNNs by using a forget gate in its memory cell, which enables the learning of long-term dependencies. The authors in [24] demonstrated the potential of LSTM units for accurately recognizing a wide range of modulation types.

In this work, we first present three different architectures that deliver higher classification accuracy than the PCNN introduced in [23] as well as the CLDNN of [24]. We design our own PCNN and CLDNN architectures for the modulation recognition task, as well as derive an optimized version of the ResNet architecture of [26] by tuning the number of residual stacks. In contrast to the 75% high SNR classification accuracy acheived by the PCNN of [23] using the RadioML2016.10b dataset that was first considered in the same work, our PCNN, CLDNN, and ResNet architectures deliver high SNR accuracy values of 83.8%, 88.5%, and 92%, respectively. However, we find that the performance of all these architectures, as well as the ones in [23] and [24], suffers degradation, even at high SNR, due to confusions between similar modulation types, in particular those of QAM16 and QAM64 and those of AM-DSB and WBFM. Another major challenge facing machine learning algorithms based on deep neural network architectures is the long training time. For example, for the problem at hand, even the simple PCNN architecture in [23] takes approximately 40 minutes to train using three Nvidia Tesla P100 GPU chips. This creates a serious obstacle towards the feasibility of applying such algorithms in real time, where online training is needed to adapt the network architecture to changing environmental conditions. In particular, applying deep learning to autonomous wireless communication systems anticipated in next-generation networks requires significant reduction in training time compared to state-of-the-art methods. In such systems, it is likely that training will be frequently needed to accommodate new environmental conditions. Hence, reducing training time becomes essential for the success of these algorithms. The third major challenge is hardware requirements due to sampling the received signal at high rates, which can be cumbersome in real time, particularly in wideband settings.

In summary, the motivating factors for this work can be described as follows.

1) Existing results in the literature shed light on the promise of sub-Nyquist sampling for reducing hardware cost and computational efficiency gains with minimal loss in performance [28], [29]. This potential has not been well investigated in the context of employing deep learning for modulation classification.
2) Existing subsampling schemes rely on fixed strategies rather than data-driven adaptive strategies that alter the subsampling mechanism based on intrinsic input features. On the other hand, the proven success of deep learning for wireless communications in general, and modulation classification in particular, suggests potential significant improvements due to data-driven techniques.
3) Connecting between recent advances in deep learning models for modulation classification - and in general for wireless communication systems - and the literature on Wrapper methods that employ machine learning models for feature selection.
4) State of the art machine learning models for modulation classification encounter difficulty in distinguishing between close pairs such as QAM16/QAM64 and AM-DSB/WBFM, which draws a relatively low ceiling on achievable performance. For example, typical PCNNs that are custom-tailored for classifying the ten modulation types of the RadioML2016.10b dataset, achieve a maximum classification accuracy of around 89% at high SNR.

We tackle these challenges by introducing a data-driven subsampling stategy that relies on an ensemble of the three deep neural network classifiers presented in this work, as well as the ResNet as a final deep neural network classifier that recognizes the modulation type. Our strategy relies on the learning transferability property of deep neural networks, as we determine the optimal set of samples based on simulations that employ a diverse set of architectures, all of which are suitable for the considered classification task. These simulations are inspired by how wrapper feature selection methods work through model-based evaluations of feature sets. The obtained results demonstrate that not only does the proposed data-driven subsampling strategy lead to significant reduction in the required training time, but it also leads to achieving unprecedented classification accuracy values. Further, it almost fully resolves the confusions - suffered by traditional methods as well as previous deep learning-based methods - between similar pairs of modulation types like QAM16 and QAM64 as well as AM-DSB and WBFM at higher SNR values (above 2 dB). Using the RadioML2016.10b dataset of [23], the ResNet high SNR classification accuracy increases with subsampling rates as low as $\frac{1}{16}$, and goes above 99% when subsampling with a rate of $\frac{1}{4}$ or higher. As further illustrated in Section VI, subsampling led to increase in classification accuracy in our experiments, due to its effect in reducing overfitting by projecting samples onto a lower dimensional subspace that admits a distinction between different classes through simple decision boundaries.

The contributions of this work can be summarized as follows.

1) We identify three deep neural architectures, namely a Pure Convolutional Neural Network (PCNN), a new Convolutional Long Short-term Deep Neural Network (CLDNN), and a new Residual Network (ResNet), each delivering state of the art performance for the task of modulation classification based on the extensively studied RadioML2016.10b dataset.
2) We then use the identified classifiers as *Ranker Models* for subsampling, via ranking the importance of the different samples through simulations of sample subset removals without re-training the model. This is inspired by the literature of Wrapper Feature Selection methods.
3) We next introduce the *Holistic Subsampler*, which relies on an effective strategy for combining the rankings provided by the three ranker models, for robust subsampling that generalizes to unseen testing data.
4) When analyzing the performance of the Holistic Subsampler, we observe drops in accuracy at certain SNR values, which indicates a potential for improving the set of selected sample indices in these cases. We introduce the *ϵ-Greedy Search* algorithm that identifies best sample indices whenever the Holistic Subsampler solely fails to do so.
5) Based on the above contributions, we propose the main method of this work: The *Ensemble Wrapper Subsampler*, which delivers performance superior to the state of the art and unveils the potential of deep-learning-based sub-Nyquist sampling, particularly via resolving confusions between difficult modulation pairs such as the QAM16/QAM64 and the AM-DSB/WBFM pairs.
6) We perform an ablation study to justify the addition of each component of the proposed method, and compare the final performance with that obtained through state of the art methods.

The rest of this paper is organized as follows. In Section II, we describe the problem. We then provide a detailed description of the proposed approach in Section III, and highlight the obtained results in Section IV. We provide a detailed justification for every step of the proposed approach through a benchmarking and ablation study in Section V. Finally, we discuss our results in Section VI and provide concluding remarks in Section VII.

II. Problem Description

We study the classification of the modulation type of received wireless signals, using deep neural network classifiers and subsampling techniques. We consider ten widely used modulation schemes: eight digital and two analog modulations. These consist of BPSK, QPSK, 8PSK, QAM16, QAM64, BFSK, CPFSK, and PAM4 for digital modulations, and WBFM, and AM-DSB for analog modulations.

A general expression for the received baseband complex envelope is

$$r(t) = s(t;\boldsymbol{u}_{\mathbf{i}}) + n(t), \tag{1}$$

where for $0 \leq t \leq KT$,

$$s(t;\boldsymbol{u}_{\mathbf{i}}) = a_i e^{j2\pi\triangle ft} e^{j\theta} \sum_{k=1}^{K} e^{j\phi_k} s_k^{(i)} \times g(t-(k-1)T-\varepsilon T), \tag{2}$$

is the baseband complex envelope of the received signal, and *n*(*t*) is the instantaneous channel noise at time *t*. In (2), a_i is the received signal amplitude, Δf is the carrier frequency offset, θ is the time-invariant carrier phase introduced by the propagation delay, ϕ_k is the phase jitter, $\{s_k^{(i)}, 1 \leq k \leq K\}$ denotes K complex symbols taken from the i^{th} modulation format, T represents the symbol period, ε is the normalized epoch for time offset between the transmitter and signal receiver, $g(t) = P_{pulse}(t) \otimes h(t)$ is the composite effect of the residual channel with *h*(*t*) denoting the channel impulse response and $\otimes$ denoting mathematical convolution, and $P_{pulse}(t)$ is the transmit pulse shape. We denote $\{a_i,\ \Delta f,\ \theta,\ \varepsilon,\ g(t),\ \{\phi_k\}_{k=1}^{K},\ \{s_k^{(i)}\}_{k=1}^{K}\}$ by $\boldsymbol{u}_{\mathbf{i}}$; the multidimensional vector that includes the deterministic unknown signal or channel parameters for the i^{th} modulation type.

Our goal is to recognize the modulation type *i* from a sampled version of the received signal *r*(*t*). This is achieved through a supervised machine learning algorithm that has access to labeled sample vectors. We assume that the data available for training and testing are equi-sized and equally split across the ten modulation types. We further study this problem under constraints on the allowed sampling rate. Such constraints could reflect a training time limitation, which is analyzed in this work, as well as hardware requirements (e.g., of RF sensors).

Using the RadioML2016.10b dataset that consists of samples taken at around 6 times the Nyquist rate and 8 samples per symbol, a PCNN architecture was shown to achieve 75% classification accuracy at 18 dB SNR [23]. As detailed below, we first present three deep neural network architectures that deliver state-of-the-art performance, with classification accuracy values reaching 92% at high SNR. Then, we present a data-driven subsampling strategy that employs the ensemble of the presented architectures and relies on wrapper-based recursive simulations, to deliver accuracy values that exceed 99% at high SNR with sampling rates around the Nyquist rate, and remain above the no subsampling accuracy with sampling rates at or above 37.5% of the Nyquist rate. To the best of our knowledge, the accuracy values obtained by applying our method with subsampling rates at or above $\frac{1}{16}$ are higher at high SNR than those obtained by applying existing methods in the literature on the same dataset, even with no subsampling. This superior performance is uniform across the studied SNR range from -20 dB to 18 dB when applying our method with subsampling rates at or above $\frac{1}{4}$.

III. Designing the Ensemble Wrapper Subsampler

The proposed strategy utilizes training data, originally sampled at a high rate, to search for the optimal set of sample indices using an ensemble of deep neural network architectures that were found empirically to be well fit for the considered task. Once the sample indices are determined, we only sample at the corresponding times for training and testing the modulation type classifier. We will show in the sequel that samples chosen by this strategy lead to classification accuracy values that are drastically higher than the state-of-the-art. It is important to note that even if the process of determining the best sample indices is computationally intensive, the overall cost can be negligible as this process needs to be carried out only occasionally with significant environmental changes, and the subsampler can be used in real-time without frequent re-training. It is also worth mentioning that we assume a perfect SNR estimate available to the subsampler both during its training and for selecting possibly different sample indices for different SNR values. Relaxing this SNR availability assumption is left for future investigations.

We begin by introducing three deep classifiers, each achieving high classification results on fully sampled data. Then, we build wrapper models - that we call Subsampler Nets - using each of the three architectures. We then use the ensemble of these models to build a Holistic Subsampler that exploits the diversity in performance delivered by the three models. Finally, we introduce a deterministic variant of ϵ-Greedy search that finesses the obtained classification performance, by exploiting the available wrapper-based sample ranking. We present results obtained through the proposed approach and justify the need for each of its components in Sections IV and V, respectively.

A. Deep Neural Network Architectures

Our strategy employs a PCNN, CLDNN, and ResNet, whose details we provide below. We chose these architectures through experimental trials as well as an extensive literature survey. In particular, we found these three architectural types besides the pure Long Short-Term Memory (LSTM) to be the most commonly successful across recent studies on deep learning for modulation classification. However, as we illustrate below, since our criterion was not only to find a good classifier, but also to find a model that would work well for ranking samples as part of a wrapper subsampling approach, pure LSTM architectures were found experimentally to not be good candidates for our purpose.

For all architectures, we use the Adam optimizer and the categorical cross entropy loss function. We also use ReLu activation functions for all layers, except the last dense layer, where we use Softmax activation functions. **Robustness** and **diversity** were the **key design factors** that guided our choice of architectures. The former indicates that each architecture is well fit for the task, even at low sampling rates, which we verified through experimental results; the latter indicates that the three architectures are independently trained and rely on different mechanisms for capturing task-relevant features. While a PCNN relies on a fixed hierarchical representation that first extracts lower-level features through a large number of convolutional kernels, and then captures higher-level semantics through less kernels whose outputs have a wide input receptive field, the ResNet relies on shortcut connections between convolutional layers that are far apart, which provides stable training for deeper layers and allows for dynamically choosing an *effective architecture* while training (see [29, Chs. 8–9] for more illustration). Also, the ResNet is significantly deeper than the PCNN, which makes it likely to reach very different solutions. Unlike these two architectures, the CLDNN includes a gated LSTM layer that captures long-term temporal correlations in the convolutional output feature maps.

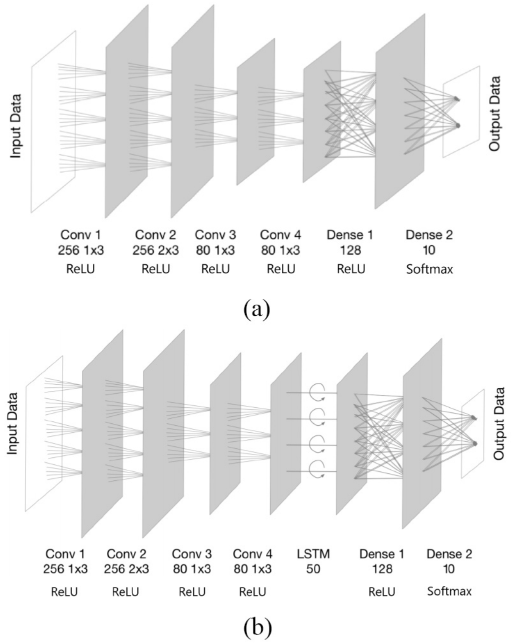


Fig. 1. Architecture diagrams of (a) PCNN, and (b) CLDNN.

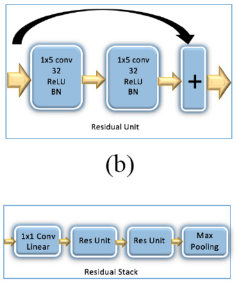
1) PCNN: We modify the CNN2 architecture, that was proposed in [23] by having four, in lieu of two, convolutional layers, and two dense layers, as depicted in Figure 1a, as we found this modification to lead to better classification accuracy than the original CNN2 architecture. The first parameter below each convolutional layer in the figure represents the number of filters in that layer, while the second and third numbers show the size of each filter. For the two dense layers, we use 128 and 10 neurons in order of their depth in the network.

2) CLDNN: Inspired by [27], we proposed a CLDNN in [1] by adding an LSTM layer into the PCNN architecture. The detailed architecture considered for the CLDNN is shown in Figure 1b. The extra LSTM layer is placed between the convolutional layers and the dense layers. In our experiments, an LSTM layer with 50 cells provided the best accuracy.

3) ResNet: As neural networks grow deeper, their learning performance is challenged by problems like vanishing or exploding gradient and overfitting, and therefore both training and testing accuracy start to degrade after the network reaches a certain depth. The Deep Residual Network (ResNet) architecture that was introduced in the ImageNet and COCO 2015 competitions [25], tackles accuracy degradation issues

Layer	Output dimensions
Input	2x128
Residual Stack	32x64
Residual Stack	32x32
Residual Stack	32x16
FC/ReLU	128
FC/ReLU	128
FC/Softmax	10

(a)



(b)

(c)

Fig. 2. (a): The ResNet architecture, (b) A residual unit, and (c) A residual stack.

in deeper neural networks and has been shown to be a robust choice for a wide range of machine learning tasks. Inspired by the ResNet architecture in [26], we designed a similar ResNet but with three residual stacks instead of six, as we found that choice to lead to increased classification accuracy. In our network, three residual stacks are followed by three fully connected layers, where each residual stack consists of one convolutional layer, two residual units, and one max-pooling layer. As seen in [26], for each residual unit, a shortcut connection is created by adding the input of the residual unit with the output of the second convolutional layer of the residual unit. Finally, each convolutional layer in the residual unit uses a filter size of 1x5 and is followed by a batch normalization layer for optimization stability. The overall architecture is observed in Figure 2. As we illustrate below, this architecture delivered the best - or very close to the best - performance among all considered architectures for a wide range of SNR values that only excludes extremely low values.

B. Subsampler Nets: A Wrapper Feature Selection Approach

The first building block in our ensemble method that employs the architectures provided above, is a supervised wrapper feature selection algorithm that we call the **Subsampler Net**, which uses a deep neural network to rank the importance of each sample in accordance to the relative drop in classification accuracy that occurs when that sample is removed (set to 0). Similar to other wrapper feature selection methods, a Subsampler Net relies on a classifier to rank sample importance. We will refer to this classifier as the **Ranker Model**. We use pre-trained models for each of the above three architectures as Ranker Models to construct three separate Subsampler Nets.

Suppose we want to obtain k samples from a pool of d samples. As shown in Figure 3, the Subsampler Net first starts by setting a sample to 0 (setting both the real and imaginary parts of that sample to 0) in a batch of training validation examples and evaluating it using the Ranker Model, which will then provide us with a classification accuracy for that batch. Setting a sample to 0 means that the input neurons to the model for the two features corresponding to that complex sample are dead, which allows us to simulate the removal of that sample from the signal as all the weights from these neurons in the input layer will not contribute to the outcome of the model. This is done d times by setting each of the d samples to 0. After evaluating each of the d samples, we are left with d classification accuracies that correspond to the ability of the model to classify the signal if each of the samples were to be removed. The most important sample, which is the sample whose removal results in the lowest classification accuracy, is then permanently set to 0 for this batch of training examples and added to the final set of samples. Now, we are left with $d - 1$ samples and this process is repeated until we are finally left with k samples. The Subsampler Net construction is detailed in Algorithm 1.

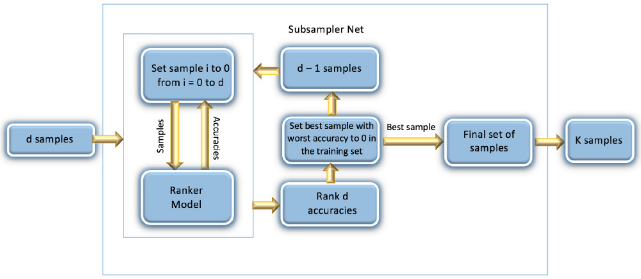


Fig. 3. Subsampler Net that chooses k representative samples out of a pool of d samples.

Algorithm 1: Subsampler Net

```
Inputs: k: Final No. of samples; trainSet: Training
Dataset; rankerModel: Trained Model that ranks samples
Outputs: sampleList: List of k selected sample indices
  function SUBSAMPLERNET(k, trainSet, rankerModel)
    Initialize sampleList to an empty list
    for $i = 0$ to $k$ do
      Initialize accList to an empty list
      Set candidateList as set of sample indices not in
      sampleList
      for j in candidateList do
        Set sample with index j to 0 in trainSet
        accuracy = rankerModel(trainSet)
        Append (j, accuracy) to accList
        Set sample with index j back to original value
      end
      Sort accList by order of increasing accuracy
      Append sample index with lowest accuracy in
      accList to sampleList
      Permanently set this sample to 0 for all examples
      in trainSet
      return (sampleList)
  end function
```

While attempting this method, we found that normalizing the data by setting the mean of each sample to 0 and the variance to 1 improves performance because when we set an input sample to 0, we are effectively setting it to the mean, and lower variances now manifest as lower weights in the input layer [30]. We also observed, from experiments that rely on a discrete set of SNR values, that the sample indices chosen for batches of validation examples belonging to the same SNR

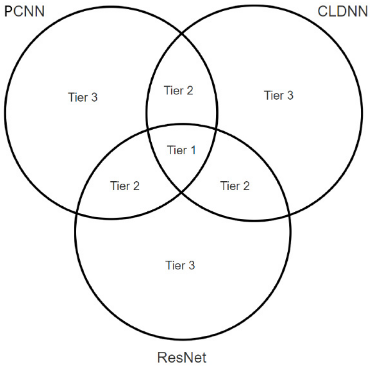


Fig. 4. Tier Division.

value were the same in most cases, while they were likely to be distinct from those chosen for batches belonging to different SNR values. Therefore, we divide the available data according to SNR value and obtain a set of k sample indices for each SNR. While presenting our experimental results in Secion IV, we further highlight this SNR sensitivity by showing the percentage of overlap between selected sample sets among different SNR values.

C. The Holistic Subsampler: The Best of All Worlds

We next introduce the notion of the Holistic Subsampler, which combines the ability of all three Ranker Models in order to capitalize on their diversity of performance. After the sets of samples, that match the required sampling rate, are collected for each of the three Ranker Models, we divide these samples into three tiers, as illustrated in the Venn diagram in Figure 4. Tier 1 consists of the intersection of all three sets of samples, Tier 2 consists of the samples that belong to two of the three sets of samples, and Tier 3 consists of samples that are selected by only one Subsampler Net. The samples within each of the tiers are sorted according to the sum of the classification accuracy values that occur when that sample is removed using the corresponding Ranker Models. For example, for Tier 2 samples, we sort the samples using the sum of the two obtained classification accuracy values when the sample is removed (set to 0). Akin to the Subsampler Net, the Holistic Subsampler is a recursive algorithm that first selects the best sample, then sets the value of the corresponding sample index to 0 for the whole training set, and calls itself again to find the next best sample. This is done until the desired number of samples k is reached. To find the best sample, the top sample - corresponding to the lowest sum of classification accuracy values - is selected from Tier 1. If Tier 1 is empty, then the top sample from Tier 2 is selected. If Tiers 1 and 2 are both empty, then the top sample from Tier 3 is selected.

D. ϵ-Greedy Search: The Final Piece of The Puzzle

We note that the Subsampler Net merely selects the best sample at each iteration, without regard to how the selection of a subsequent sample will affect the importance of the currently selected sample to the classification task. We chose to do this in the interest of saving training time of Subsampler Nets in order to render the implementation of the proposed method feasible using low-power communication devices. We next propose a variant of the ϵ-greedy algorithm [31] in order to explore candidate combinations for subsequent best samples while taking into account dependence relationships between the selected samples.

Algorithm 2: ϵ-Greedy Search

```
Inputs: d: Total No. of samples; k: Final No. of samples;
$\epsilon$: exploration factor that is the fraction of total samples
to be explored; prevSnrAcc: Classification accuracy at
the preceding SNR value; trainSet: Training Dataset
Outputs: finalIdx: Set of sample indices whose removal
leads to the lowest accuracy among combinations
explored; finalSNRAcc: Accuracy obtained using trainSet
when the sample indices in finalIdx are selected
  function $\epsilon$-GREEDY($k$, $\epsilon$, prevSnrAcc, trainSet)
      Call SubsamplerNet using PCNN, CLDNN, and
  ResNet as the Ranker Models
      Set sampleIdx as the ordered set of the $k$ sample
  indices selected by the Holistic Subsampler
      Set currSnrAcc as accuracy of ResNet architecture
  when trained with selected samples
      if $k = 0$ then
        return (sampleIdx, currSnrAcc) if currSnrAcc $>$
    prevSnrAcc
        return (NULL, NULL) otherwise
      else
        for $i = 0$ to $\min(k, \epsilon d)$ do
            Set trainSet[sampleIdx[$i$]] to 0
            Set (finalIdx, finalSnrAcc) =
        $\epsilon$-GREEDY($k - 1$, $\epsilon$, prevSnrAcc, trainSet)
            Set trainSet[sampleIdx[$i$]] back to original
        value
            Add sampleIdx[$i$] to finalIdx
            return (finalIdx, finalSnrAcc) if returned
        values are not NULL
        done
        return (NULL, NULL)
      end
  end function
```

According to Algorithm 2, we introduce ϵ, the exploration factor that determines the number of candidate samples considered for selection at each step. If $\epsilon = 0.1$, then in every step, we explore the 10% best samples according to the ranking provided by the Holistic Subsampler. This is unlike the conventional ϵ-greedy algorithm, where ϵ represents the probability that the decision taken deviates from the top greedy choice. Our variant of the algorithm explores all of the top routes and ϵ is the parameter that determines the number of top routes explored. The rationale behind our deviation from the conventional ϵ-Greedy Algorithm is as follows. The Multi-Armed Bandit problem [32], which is one of the most popular applications of the traditional ϵ-Greedy Algorithm, is

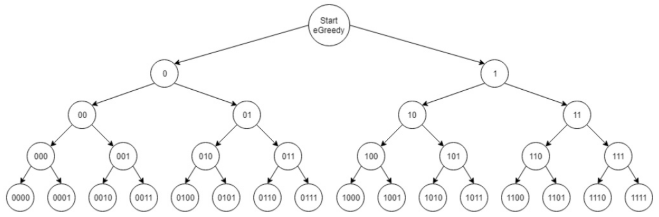


Fig. 5. Illustration of ϵ-Greedy Search with $\epsilon = \frac{2}{d}$, where d is the total number of available samples, and a subsampling rate of $\frac{4}{d}$. A 0 corresponds to removing the best sample, while a 1 corresponds to removing the second best sample, according to the ranking of the Holistic Subsampler. The samples are re-ranked after each removal.

based on a scenario where there is a single top choice and all the other choices are of the same importance before exploration. However, in our case, the choices apart from the top choice have a ranking that reflects their relative importance. Therefore, unlike the Multi-Armed Bandit problem, we do not need to waste time exploring unknown paths.

As observed in Figure 5, the ϵ-Greedy Search can be represented as a traversal algorithm over an ϵd-ary tree whose depth is equal to the desired number of samples k, where the subsampling rate is $\frac{k}{d}$. Each node in the tree corresponds to the selection of a combination of sample indices. The nodes at the same depth are arranged by increasing order of accuracy when removed, which implies that the combination of samples corresponding to the left child of a node has higher priority than that corresponding to the right child of the same node, for the case when $\epsilon d = 2$. The root node does not represent any sample and is added just for the sake of illustration of tree formation. The leaf nodes are searched from left to right until a classification accuracy that is satisfactory is reached.

We note that setting $\epsilon = \frac{1}{d}$ is equivalent to having an unaltered Holistic Subsampler that greedily chooses the best sample at each iteration, which leads to the leftmost leaf of the tree in Figure 5. This corresponds to node 0000 because only a tree with a single branch is formed with the nodes 0, 00, 000, and 0000. Increasing the value of ϵ expands the scope of exploration and exponentially increases the size of the tree.

E. Ensemble Wrapper Subsampling

We here finalize the specification of the proposed approach. Given input data with d samples per example, we first initialize ϵ to $\frac{1}{d}$ and invoke the ϵ-Greedy Search. As mentioned earlier, this is the same as invoking the Holistic Subsampler on its own. Next, we proceed to the next SNR value available in the training set. The ϵ-Greedy Search is invoked with an ϵ value of $\frac{1}{d}$ for this next training set belonging to the next SNR value. If the accuracy is lower than the accuracy for the previous SNR value, then ϵ-Greedy Search is invoked again after doubling the ϵ value to $\frac{2}{d}$. The ϵ-Greedy Search stops exploring once the accuracy is greater than the accuracy obtained for the previous SNR value. We repeatedly double ϵ and invoke ϵ-Greedy Search until this stopping criterion is met. The pseudocode for this strategy is given in Algorithm 3.

The function described in Algorithm 3 returns the selected set of k sample indices for each SNR value. Note that doubling the value of ϵ for the ϵ-Greedy Search corresponds to searching for combinations of sample indices in a tree that has twice the arity.

Algorithm 3: Ensemble Wrapper Subsampler

Inputs: d: Total No. of samples; k: Final No. of samples; trainSet: Training Dataset
Outputs: idxDict: Dictionary with SNR as key and sets of k sample indices each as values

```
function ENSEMBLEWRAPPERSUBSAMPLER($k$, trainSet)
    Divide trainSet based on SNR
    Initialize idxDict as an empty dictionary
    Initialize prevSnrAcc to 0
    for snrValue in set of SNR values do
        Initialize $\epsilon = \frac{1}{d}$
        Initialize snrIdx as NULL
        while snrIdx is NULL do
            Set (snrIdx, snrAcc) = $\epsilon$-GREEDY(k, $\epsilon$, prevSnrAcc, trainSet)
            Set $\epsilon = 2\epsilon$
        done
        Set snrIdx as the value to snrValue key in idxDict
        Set prevSnrAcc as snrAcc
    done
    return idxDict
end function
```

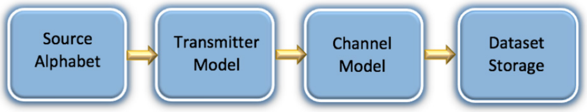


Fig. 6. High-level framework for data generation.

IV. Experimental Results

In this section, we present an experimental evaluation of the proposed method. First, we specify the dataset used, the programming environment, and the hyperparameter settings. Then, we present the obtained classification accuracy results while highlighting important insights, and finally quantify the reduction in training time for the final classifier with different subsampling rates.

A. Dataset

We use the RadioML2016.10b synthetic dataset generated in [23] as the input data. Details about the generation of this dataset can be found in [33]. Figure 6 shows a high-level framework for the data generation process. For digital modulations, the entire Gutenberg works of Shakespeare in ASCII is used, with whitening randomizers applied to ensure equiprobable symbols and bits. For analog modulations, a continuous voice signal consisting of acoustic voice speech with some interludes and off times is used as input. The modulation rate is 8 samples per symbol.

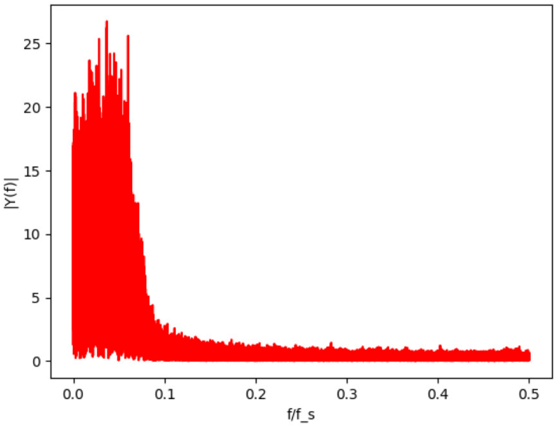


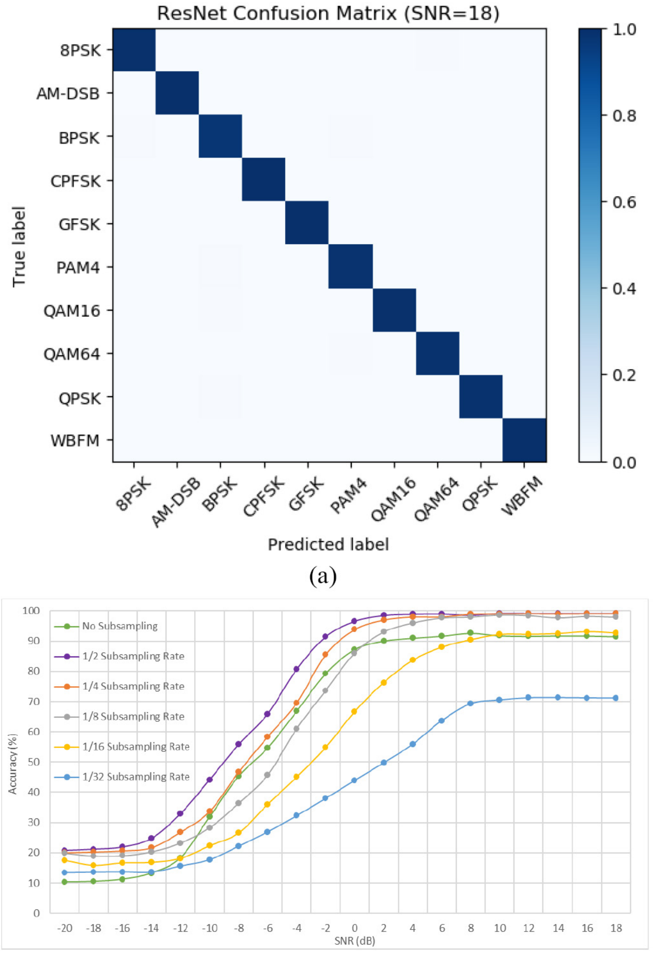
Fig. 7. One-sided normalized FFT for a BPSK signal. A value of 0.5 on the horizontal axis corresponds to the Nyquist rate (Bandwidth is half the sampling rate). Most of the signal energy is within a band of around $\frac{1}{12}$ of the sampling rate.

The dataset is split equally among all ten considered modulation types. For the channel model, physical environmental noises like thermal noise and multipath fading were simulated. The models for generating random channel and device imperfections, that determine the parameters in (2), are detailed in [33].[1] When packaging data, the output stream of each simulation is randomly segmented into vectors as the original dataset with a sample rate of 1M samples per second. Similar to the way that an acoustic signal is windowed in voice recognition tasks, a sliding window extracts 128 samples with a shift of 64 samples, which forms a sample vector in the dataset. 160,000 sample vectors generated using the GNU-radio library developed in [33] are segmented into training and testing datasets. Each example consists of 128 samples, that are represented as a 2 × 128 vector with real and imaginary parts separated. The SNR of the samples is uniformly distributed from -20 dB to 18 dB, with a step size of 2 dB, i.e., the dataset is equally split among all SNR dB values in $\{-20, -18, -16, \ldots, 18\}$.

We note from the frequency domain representation of the input waveform depicted in Fig. 7 that the sampling rate of the input waveform is around 6 times the Nyquist rate.

B. Implementation Details

We used Keras with TensorFlow as a backend, and a GPU server equipped with three Tesla P100 GPUs with 16 GB memory. For all architectures, we used a batch size of 1024, and a learning rate of 0.001. **Only the training set is used by the subsampling algorithm** described in Section III with a validation split of 0.25. After selecting the set of sample indices for each of the 20 considered SNR values, we train the ResNet classifier with the corresponding samples, as we found it to deliver the best performance among the three considered architectures.[2]



(a)

(b)

Fig. 8. (a) ResNet Confusion Matrix after Ensemble Wrapper Subsampling using a subsampling rate of $\frac{1}{2}$ at 18 dB SNR. (b) Accuracy vs SNR for ResNet with Ensemble Wrapper Subsampling.

C. Classification Accuracy

We present the obtained results at different sampling rates in Figure 8. From our results, we note the following.

- *Subsampling can Lead to Higher Accuracy:* Applying the proposed ensemble wrapper subsampling strategy can result in dramatic improvements in classification accuracy, particularly at low SNR values. The direct cause for this phenomenon at high SNR is resolving confusions between the QAM16/QAM64 and AM-DSB/WBFM pairs, as we illustrate in Section V. To the best of our knowledge, state-of-the-art methods fail at resolving these confusions. We believe that this is because our subsampling strategy reduces overfitting, as we elaborate in Section VI-B.
- *Sub-Nyquist Sampling:* As noted above, the considered data is originally sampled at around 6 times the Nyquist rate. A subsampling rate of $\frac{1}{16}$ hence corresponds to around 37.5% of the Nyquist rate, and leads to slightly higher classification accuracy at very high SNR and significantly higher accuracy at very low SNR, than the case with no subsampling.

[1] Dataset generation parameters are also available at https://github.com/radioML/dataset.

[2] Code is available at: https://github.com/dl4amc/dds.

TABLE I
COMPARISON OF TRAINING TIME AND HIGH SNR ACCURACY FOR THE RESNET AFTER ENSEMBLE WRAPPER SUBSAMPLING

Samples	Time per Epoch	Epochs	Total Training Time	Accuracy (18 dB SNR)
All	32.0642s	37	1186.3754s	91.49%
1/2	26.9615s	33	889.7295s	99.27%
1/4	24.2615s	29	703.5835s	99.13%
1/8	21.8361s	27	589.5747s	97.94%
1/16	17.1167s	26	445.0342s	92.67%
1/32	11.6828s	24	280.3827s	71.14%

This observation could carry important implications in practice, as the sampling hardware requirements can be dramatically simplified (see [28], [29] for further illustration).

- *Minimal Sample Set:* Based on the loss in classification accuracy, we can choose the smallest set of samples (smallest value of k) that gives us a classification accuracy higher than a given classification accuracy requirement. For instance, 20 is the smallest number of samples (around Nyquist rate) that can be selected such that the classification accuracy has to be higher than 99% at 18 dB SNR. Similarly, 8 is the smallest number of samples (around 37.5% of Nyquist rate) that can be selected given that the classification accuracy has to be higher than 90% at 18 dB SNR.

D. Training Time

As a result of subsampling, the training time of the classifier is reduced due to the reduced input dimensions. We show the reduction in training times and high SNR classification accuracy of the ResNet classifier for different subsampling rates in Table I. Note that a subsampling rate as low as $\frac{1}{16}$, which corresponds to the sub-Nyquist regime with around 37.5% of the Nyquist rate, and results in approximately $\frac{1}{3}$ of the original training time, still results in a classification accuracy higher than that without subsampling.

V. BENCHMARKING AND ABLATION STUDY

Supported by experimental results, we first provide an analysis of our proposed approach with regard to traditional approaches, and then provide a justification for each of its components. We begin by motivating the need for deep learning via analyzing the performance of a Bayes classifier. Then, we present the results obtained with the considered deep neural network architectures with no subsampling, and highlight the modulation type pairs that are difficult to distinguish even at high SNR. We then compare the results obtained through our proposed approach with conventional subsampling and feature selection schemes. Finally, we present an ablation study to demonstrate the performance degradation caused when removing any of the components of the proposed method.

TABLE II
GAUSSIAN NAIVE BAYES CLASSIFIER RESULTS FOR DIFFERENT MODULATION PAIRS AT ALL SNRS AND AT 18 DB SNR

Modulation Pairs	Classification Accuracy (All SNR)	Classification Accuracy (18 dB)
PAM4 vs QAM64	68.5%	82%
QAM16 vs QAM64	50%	51%
AM-DSB vs BPSK	70%	81%
AM-DSB vs WBFM	53%	53%

A. Gaussian Naive Bayes Classifier

The Gaussian naive Bayes classifier can be described through the conditional probabilities:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}}, \tag{3}$$

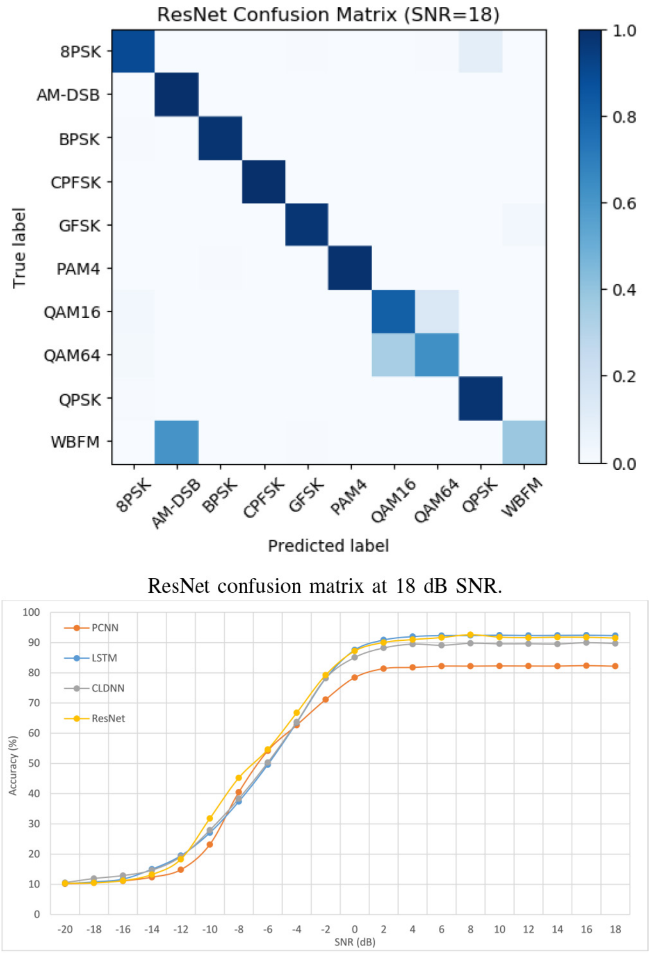
where $P(x_i|y)$ is the likelihood of an observed instance x_i belonging to a certain class y, σ_y^2 is the observed variance of class y, and μ_y is the observed mean of class y. The predicted output of x_i is the class that maximizes the likelihood function. Instead of trying to classify all ten modulation types, we only used certain pairs to further demonstrate the performance of the Gaussian naive Bayes classifier for simpler tasks.

From the results in Table II, we note that even when the Bayes classifier is trained to distinguish pairs that are not challenging at the maximum SNR value, its maximum accuracy is 82%. Further, for challenging pairs, the performance is similar to random guessing even at high SNR.

B. Deep Learning With No Subsampling

We present in Figure 9 the classification accuracy of the considered architectures using the outlined dataset with no subsampling. We note that - similar to previous work on deep learning for modulation classification - most of the misclassifications at high SNR are due to confusions between the AM-DSB/WBFM and the QAM16/QAM64 pairs, which is evident from the ResNet 18 dB confusion matrix depicted in Figure 9a. We observe by comparing to Figure 8 how the proposed data-driven subsampling method leads deep neural network classifiers to clear this confusion. We believe that this is due to overfitting reduction, as further illustrated in Section VI. We further note that we also considered a pure LSTM architecture by fine tuning that of [34] for the task. Even though this architecture delivered good performance with no subsamlping as shown in Figure 9b, we chose not to use it in our proposed method as it suffered drastic performance degradation with subsampling. We believe that this is due to extreme sensitivity of the captured temporal correlations to absence of few samples.

In Figure 10, we provide the results obtained when using recent architectures that were introduced at the time of writing - or slightly after - the first draft of this work. These architectures are the MCNet [35], the Accu-polar CNN [36], and the Depthwise and Depthwise Separable CNNs [37]. We note that the performance obtained is very similar to our considered architectures, as the aforementioned confusions present a challenge even at high SNR. We plan to investigate in future



ResNet confusion matrix at 18 dB SNR.

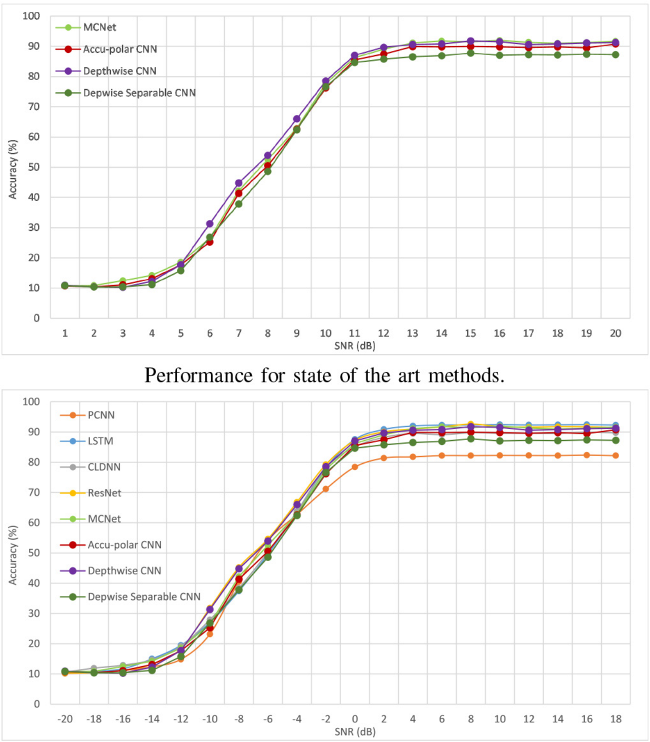
Accuracy vs SNR for each of the architectures.

Fig. 9. Baseline results with no subsampling for modulation classification.

work the use of the CNN variants introduced in these works in our holistic subsampler.

C. Conventional Subsampling

We provide in Figure 11a a comparison between the proposed approach and four different subsampling techniques; namely: 1) Uniform Subsampling: where a sample is taken every fixed amount of time, 2) Random Subsampling: where the indices of selected samples are determined randomly with equal probabilities, 3) Magnitude Rank Subsampling: where the indices corresponding to samples with top magnitude values in each example are selected, and 4) Principal Component Subsampling (PCS), where first Principal Component Analysis (PCA) is done over the training set, and the indices corresponding to samples with the top PCA coefficient total magnitude values are selected. A more thorough explanation of these methods is available in [38]. We note that the proposed method leads to - uniformly across all SNR values - superior performance than all these methods. The figure demonstrates results slightly below the Nyquist rate, but this observation extends to all considered subsampling rates. In particular, random subsampling delivers better performance that the other three methods at lower rates, which agrees with the intuition in [28], but this performance remains lower than that of the proposed approach.



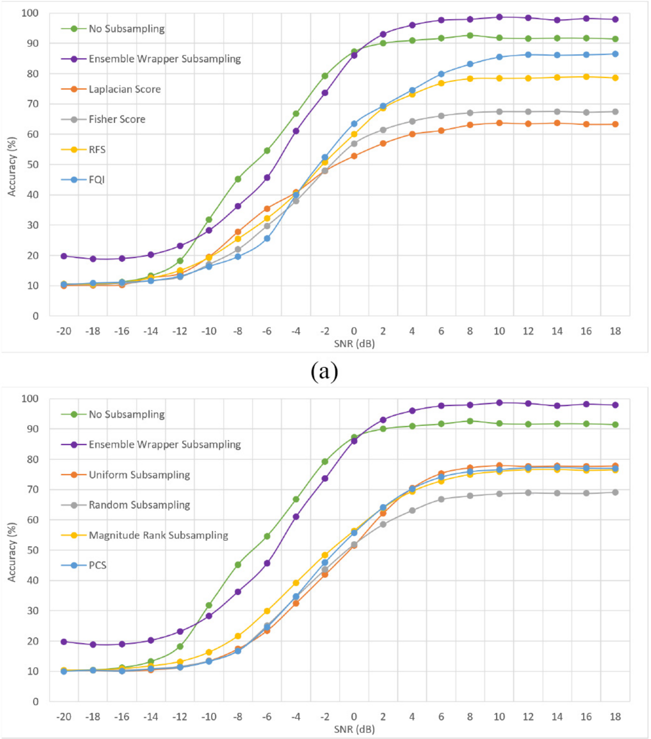
Performance for state of the art methods.

Comparison with considered architectures.

Fig. 10. Deep learning with no subsampling - state of the art results.

D. Conventional Feature Selection

Feature selection algorithms aim at identifying important input vector elements. In the considered setup, a direct application of a feature selection algorithm for 2 × 128 input vectors, would treat each of the 256 elements separately. We handle this with a slight modification to tie the real and imaginary parts of each sample, as illustrated below. In Figure 11b, we show a comparison between the proposed method and four popular feature selection algorithms; namely: 1) Laplacian Score [39]: which is an unsupervised filter feature selection algorithm that selects features with the objective of preserving the data manifold structure through a graph representation [40], 2) Fisher Score [41]: which is a supervised filter feature selection algorithm that selects features such that the features of samples within the same modulation type are similar while the features of samples belonging to other modulation types are as distinct as possible [40], 3) Efficient and Robust Feature Selection (RFS) [42]: which is a computationally efficient embedded feature selection method that exploits the noise robustness - through rotational invariance - property of the joint $\ell_{2,1}$-norm loss function [40], [43], by applying the $\ell_{2,1}$-norm minimization on both the loss function and its associated regularization function, and 4) Feature Quality Index (FQI) [44]: which is a wrapper feature selection algorithm that utilizes the output sensitivity of the considered model to changes in the input to rank features. FQI can be considered as a simplified version of our Subsampler Net that relies on the Mean Squared



(a)

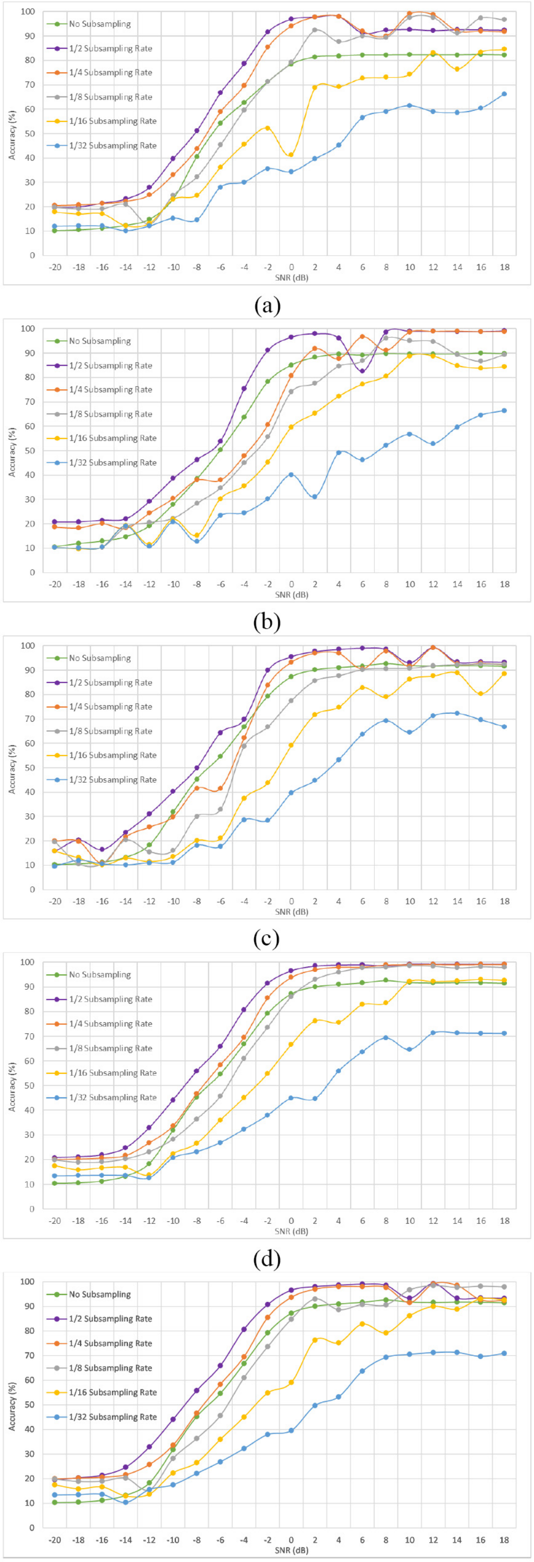
(b)

Fig. 11. Accuracy vs SNR comparisons of the proposed Ensemble Wrapper Data-Driven Subsampler with (a) Conventional Subsampling Techniques and (b) Feature Selection Techniques for the ResNet classifier at $\frac{1}{8}$ subsampling rate.

Error (MSE) loss instead of the model's loss and uses only the initial sample ranking that we use to select the first sample. For each of the techniques examined, except FQI where we have sample scores, we add the two scores obtained for the two features belonging to a sample to obtain a sample score before proceeding to rank the samples. We note how the proposed method delivers significantly better performance than all considered methods, and that this holds for all other considered subsampling rates not shown in the figure. Particularly, our method delivers better performance than FQI, which demonstrates the need for re-ranking samples after each iteration in the Subsampler Net described in Section III-B. Also, we note that this re-ranking does not require model retraining, unlike most other wrapper feature selection algorithms, which makes our method computationally feasible in a wide range of settings.

E. Ablation Study

We observe from Figure 12 how the relative performance of the three considered Subsampler Nets changes with different sampling rates and SNR values. This is the main motivation behind the Holistic Subsampler that benefits from the performance diversity among the three architectures. However, even the Holistic Subsampler, suffers from significant drops in classification accuracy for a wide range of SNR values at sampling rates well below the Nyquist rate. We believe that this phenomenon takes place due to *ranker overfitting* while selecting the sample indices via sample removal simulations that do not take into account potential statistical discrepancies between training and testing data. This motivated our ϵ-Greedy step of the proposed approach. In particular, the slope of



(a)

(b)

(c)

(d)

(e)

Fig. 12. Accuracy vs SNR for ResNet classifier with (a) PCNN Subsampler Net, (b) CLDNN Subsampler Net, (c) ResNet Subsampler Net, and (d) Holistic Subsampler, (e) Holistic Subsampler with ϵ-Greedy Search using only PCNN and ResNet.

the classification accuracy curve for the Holistic Subsampler becomes negative towards -12, 4, and 8 dB with a $\frac{1}{16}$ subsampling rate and towards -12, 2, and 10 dB with a $\frac{1}{32}$ subsampling rate. To obtain the results shown in Figure 8, our ensemble wrapper data-driven subsampling algorithm, illustrated in Algorithm 3, applied ϵ-Greedy Search with $\epsilon = \frac{1}{64} = \frac{2}{d}$ for the 4 and 8 dB SNR values with $\frac{1}{16}$ subsampling rate, as well as at 10 dB SNR with $\frac{1}{32}$ subsampling rate. For the -12 dB SNR value with both $\frac{1}{16}$ and $\frac{1}{32}$ subsampling rates, an $\epsilon = \frac{1}{32}$ had to be used because having $\epsilon = \frac{1}{64}$ was insufficient, and the same held for 2 dB SNR with $\frac{1}{32}$ rate. It is important to note that our ϵ-Greedy step has a time complexity of the Order $O((\epsilon d)^k)$, if the number of explored combinations has the same order as the number of the constructed tree leaves. Fortunately, this step is typically needed only at very low subsampling rates, where the value of k is small, and with small values of ϵ.

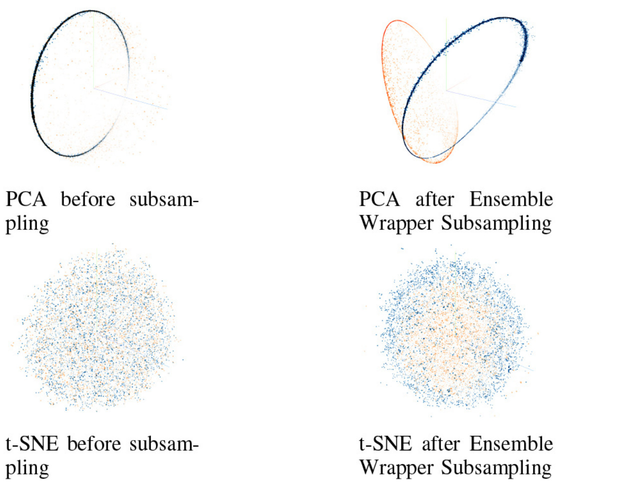
We also show in Fig. 12 (e) the results obtained by our complete ensemble wrapper subsampling strategy when using only the PCNN and ResNet ranker models. This ablation study was performed to investigate the utility of having the CLDNN architecture in the ensemble, especially as the pure LSTM ranker had detrimental impact. The results show drops in accuracy at certain SNR values, even with using ϵ-Greedy Search with the same ϵ values as detailed above, which justifies the inclusion of the CLDNN ranker model.

VI. Discussion

A. Exploiting Transfer Learning

The Holistic Subsampler achieves better results than any individual Subsampler Net, even though the final classifier relies on a single ResNet architecture. Furthermore, even though each of these architectures is trained to classify the data when all the samples are present at the input, when used as Subsampler Nets, one or more of these samples are set to 0. Hence, we use the trained deep neural network classifiers in two ways other than their intended application that they are trained on: 1- They are used to select samples for another classifier, 2- They are used with only a subset of samples present. This is only possible because of the transferability property of these deep neural network architectures. In general, we believe that exploiting transferability has great potential for various wireless communication tasks that rely on processing received signal samples. In future work, we plan to investigate the compatibility of different combinations of deep neural network architectures in light of that transferability property. For example, we know from this work that an LSTM Subsampler Net is not compatible with a ResNet classifier, but a combination of a PCNN, CLDNN, and ResNet Subsampler Nets are. This future study can further shed light on architectural properties and hyperparameter settings that enable these compatibility relationships. Further, we plan to study transferability across different SNR values. More specifically, our goal would be to identify combinations of SNR values that are ideal for training Subsampler Nets for each test SNR range.

The ability to use the ranker model in presence of only a subset of samples, without requiring re-training, significantly reduces the computational cost of the sample selection procedure. Otherwise, we would need a differently trained model for every simulation of sample set removal. Although difficult to assess due to excessive computational time, for the few experiments we ran to investigate the performance when re-training for every sample combination instead of exploiting transferability, we observed only negligible performance improvements that would not justify the computational cost in practice. For example, the training time increased by approximately ten-fold for subsampling rates of $\frac{1}{2}$ and $\frac{1}{4}$ while the average accuracy improvement across the considered SNR range was consistently less than 0.25%.



PCA before subsampling

PCA after Ensemble Wrapper Subsampling

t-SNE before subsampling

t-SNE after Ensemble Wrapper Subsampling

Fig. 13. (Upper) PCA visualization of the training dataset for the AM-DSB (blue) and WBFM (orange) classes before and after Ensemble Wrapper Subsampling with a subsampling rate of 1/2 at 18 dB SNR. (Lower) t-SNE visualization for QAM16 (blue) and QAM64 (orange) at same rate and SNR.

B. Subsampling Leads to Higher Accuracy

In Section V-B, we saw that all the deep learning architectures suffered from the same drawback of the AM-DSB/WBFM and QAM16/QAM64 misclassification when no subsampling is used. To further analyze why the proposed subsampling method leads to higher classification accuracies with fewer samples, we use Principal Component Anaysis (PCA) [45] and t-Distributed Stochastic Neighbor Embedding (t-SNE) [46] to visualize how subsampling allows us to reduce overfitting, particularly for the aforementioned class pairs. We first subsample the training dataset at 18 dB SNR with a rate of 1/2. After subsampling, we have 64 samples, corresponding to 128 features. Finally, we implement PCA and t-SNE to obtain a 3-Dimensional projection of the training dataset for better visualization. We chose to implement both PCA and t-SNE because PCA clarifies the distinction between AM-DSB and WBFM while t-SNE clarifies the distinction between QAM16 and QAM64 after subsampling, as shown in Figure 13.[3] Observing the figure, we believe that the higher accuracy values stem from the subsampling strategy enabling simpler decision boundaries to distinguish, with high

[3] The t-SNE plots were generated with a perplexity value of 20, a learning rate of 10, and were run for 250 iterations.

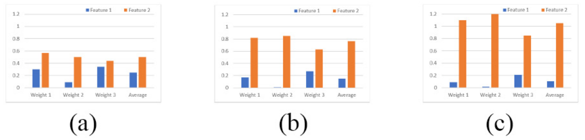


Fig. 14. Input layer weight magnitudes for toy example after training for (a) 10, (b) 100, and (c) 1000 epochs.

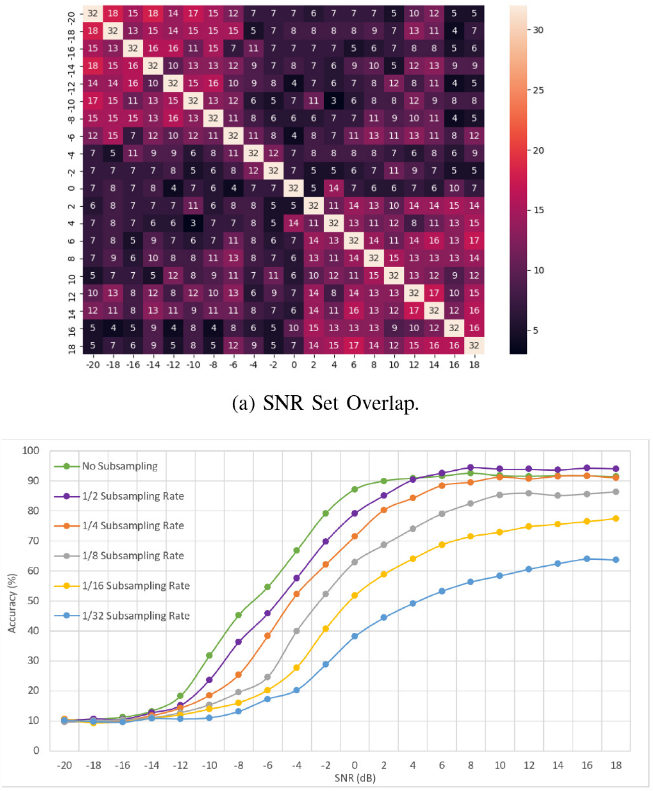
fidelity, between the considered class pairs, which improves generalization performance and reduces overfitting.

C. Designing the Ranker Models

The performance of a Subsampler Net heavily depends on the performance of the model used to rank the features. For some learning tasks, however, even the state-of-the-art models do not have high classification accuracy values. In such cases, we believe that better feature selection results can be obtained with a Subsampler Net by training the ranker model for more epochs beyond what is suggested by the Early Stopping algorithm (see, e.g., [29, Ch. 8]). This is as we found this strategy to be useful in multiple settings and further observed that it can significantly increase the discrepancy in weight magnitudes across the different features. For example, we considered a toy example constructed in TensorFlow Playground with a single-hidden-layer network that distinguishes between two classes based on two features, and the second is more salient as it enables forming a decision boundary that allows for better classification. Each of these features have three weights in the input layer and as expected, the weights connected to Feature 2 quickly manifest, after several epochs, into weights of higher average magnitude than those belonging to Feature 1 as shown in Fig. 14. As the number of training epochs increases - even from 100 to 1000 which is way more than needed for this small network - the difference in the average magnitudes of the weights increases. This implies that the ranker will be able to better rank the saliency of features because the accuracy difference will increase when suppressing each of these features.

D. Sensitivity to SNR Estimate

The selected set of sample indices is different for each SNR value, and hence, we expect a real time system employing this method to have an accurate estimate of the SNR value, in order to know the right set of sample indices. We made this choice, as we found it to deliver a significantly superior performance to the extreme alternative, where the same set of sample indices is selected for all SNR values. In Fig. 15 (a), we investigate the impact of errors in such an estimate, by comparing the different sets of selected sample indices for pairs of SNR values. Note that, as expected, the overlap is larger between close by SNR values (lower right and upper left corners). However, the size of the overlap is approximately half the set size for adjacent SNR values, which indicates performance vulnerability for small SNR estimate errors. We observed similar phenomena for other subsampling rates as well. In future work, we plan to benefit from analyzing these sets of sample indices, to better understand the roles of different ranker models at different SNR values. In Fig. 15 (b), we show the performance of the proposed method when imposing the constraint that the same sample indices are selected for all SNR values. Note the lower performance across the whole considered SNR range, in comparison to the SNR-aware subsampling considered in this work (see Fig. 8 (b)).



	-20	-18	-16	-14	-12	-10	-8	-6	-4	-2	0	2	4	6	8	10	12	14	16	18
-20	32	18	15	18	14	17	15	12	7	7	7	6	7	7	7	5	10	12	5	5
-18	18	32	13	15	14	15	15	15	5	7	8	8	8	8	9	7	13	11	4	7
-16	15	13	32	16	16	11	15	7	11	7	7	7	7	5	6	7	8	8	5	6
-14	18	15	16	32	10	13	13	12	9	7	8	7	6	9	10	5	12	13	9	9
-12	14	14	16	10	32	15	16	10	9	8	4	7	6	7	8	12	8	11	4	5
-10	17	15	11	13	15	32	13	12	6	5	7	11	3	6	8	8	12	9	8	8
-8	15	15	15	13	16	13	32	11	8	6	6	6	7	7	11	9	10	11	4	5
-6	12	15	7	12	10	12	11	32	11	8	4	8	7	11	13	11	13	11	8	12
-4	7	5	11	9	9	6	8	11	32	12	7	8	8	8	8	7	6	8	6	9
-2	7	7	7	7	8	5	6	8	12	32	7	5	5	6	7	11	9	7	5	5
0	7	8	7	8	4	7	6	4	7	7	32	5	14	7	6	6	7	6	10	7
2	6	8	7	7	7	11	6	8	8	5	5	32	11	14	13	10	14	14	15	14
4	7	8	7	6	6	3	7	7	8	5	14	11	32	13	11	12	8	11	13	15
6	7	8	5	9	7	6	7	11	8	6	7	14	13	32	14	11	14	16	13	17
8	7	9	6	10	8	8	11	13	8	7	6	13	11	14	32	15	13	13	13	14
10	5	7	7	5	12	8	9	11	7	11	6	10	12	11	15	32	13	12	9	12
12	10	13	8	12	8	12	10	13	6	9	7	14	8	14	13	13	32	17	10	15
14	12	11	8	13	11	9	11	11	8	7	6	14	11	16	13	12	17	32	12	16
16	5	4	5	9	4	8	4	8	6	5	10	15	13	13	13	9	10	12	32	16
18	5	7	6	9	5	8	5	12	9	5	7	14	15	17	14	12	15	16	16	32

(a) SNR Set Overlap.

(b) SNR-agnostic Subsampling.

Fig. 15. In (a), we show the number of overlapping sample indices among the 32 sample sets selected for different SNR values at the subsampling rate of $\frac{1}{4}$. In (b), we show the performance of the proposed method with SNR-agnostic subsampling.

VII. Concluding Remarks

In this work, we considered the problem of recognizing one out of ten modulation types with a constraint on the sampling rate in an erroneous wireless environment that is difficult to model. We first identified three deep neural network architectures that are well fit for the task and deliver state-of-the-art classification accuracy, namely a PCNN, CLDNN and ResNet. We then presented a wrapper data-driven subsampling approach that employs all three architectures - as an ensemble - for selecting a set of samples that maximizes the classification accuracy via recursive simulations aided by ϵ-Greedy deterministic explorations. Our experimental results, using the RadioML2016.10b dataset of [23], indicate that using the proposed method with a ResNet classifier leads to very high classification accuracy values, that to the best of our knowledge, have not been reached before even at sampling rates well above the Nyquist rate. Further, even in the sub-Nyquist regime, we achieve almost perfect classification (accuracy above 99%) at high SNR. We also noted the drastic reduction in the classifier's training time as a result of subsampling. We plan to further investigate in future work the potential of employing deep learning for subsampling in wireless communication systems, as we believe that the insights distilled from this work carry practical significance beyond the considered modulation classification task.

References

[1] X. Liu, D. Yang, and A. El Gamal, "Deep neural network architectures for modulation classification," in *Proc. IEEE Asilomar Conf. Signals Syst. Comput.*, Pacific Grove, CA, USA, 2017, pp. 915–919.

[2] J. A. Sills, "Maximum-likelihood modulation classification for PSK/QAM," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Atlantic City, NJ, USA, 1999.

[3] A. Polydoros and K. Kim, "On the detection and classification of quadrature digital modulations in broad-band noise," *IEEE Trans. Commun.*, vol. 38, no. 8, pp. 1199–1211, Aug. 1990.

[4] P. C. Sapiano and J. D. Martin, "Maximum likelihood PSK classifier," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, McLean, VA, USA, 1996, pp. 1010–1014.

[5] B. F. Beidas and C. L. Weber, "Asynchronous classification of MFSK signals using the higher order correlation domain," *IEEE Trans. Commun.*, vol. 46, no. 4, pp. 480–493, Apr. 1998.

[6] P. Panagiotou, A. Anastasopoulos, and A. Polydoros, "Likelihood ratio tests for modulation classification," in *Proc. 21st Century Mil. Commun. Archit. Technol. Inf. Superiority*, vol. 2. Los Angeles, CA, USA, 2000, pp. 670–674.

[7] L. Hong and K. C. Ho, "Antenna array likelihood modulation classifier for BPSK and QPSK signals," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Anaheim, CA, USA, 2002, pp. 647–651.

[8] S.-Z. Hsue and S. S. Soliman, "Automatic modulation recognition of digitally modulated signals," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Boston, MA, USA, 1989, pp. 645–649.

[9] L. Hong and K. C. Ho, "Identification of digital modulation types using the wavelet transform," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Atlantic City, NJ, USA, 1999, pp. 427–431.

[10] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Trans. Commun.*, vol. 48, no. 3, pp. 416–429, Mar. 2000.

[11] G. Hatzichristos and M. P. Fargues, "A hierarchical approach to the classification of digital modulation types in multipath environments," in *Proc. IEEE Asilomar Conf. Signals Syst. Comput.*, Pacific Grove, CA, USA, 2001, pp. 1494–1498.

[12] S. S. Soliman and S.-Z. Hsue, "Signal classification using statistical moments," *IEEE Trans. Commun.*, vol. 40, no. 5, pp. 908–916, May 1992.

[13] L. Lichun, "Comments on signal classification using statistical moments," *IEEE Trans. Commun.*, vol. 50, no. 2, p. 195, Feb. 2002.

[14] L. Mingquan, X. Xianci, and L. Lemin, "AR modeling-based features extraction of multiple signals for modulation recognition," in *Proc. IEEE Int. Conf. Signal Process.*, Beijing, China, 1998, pp. 1385–1388.

[15] B. G. Mobasseri, "Digital modulation classification using constellation shape," in *Signal Process.*, vol. 80, no. 2, pp. 251–277, 2000.

[16] L. Mingquan, X. Xianci, and L. Leming, "Cyclic spectral features based modulation recognition," in *Proc. Int. Conf. Commun. Technol. (ICCT)*, Beijing, China, 1996, pp. 792–795.

[17] E. E. Azzouz and A. K. Nandi, "Modulation recognition using artificial neural networks," *Signal Process.*, vol. 56, no. 2, pp. 165–175, 1997.

[18] K. E. Nolan, L. Doyle, D. O'Mahony, and P. Mackenzie, "Modulation scheme recognition techniques for software radio on a general purpose processor platform," in *Proc. Joint IEI/IEE Symp. Telecommun. Syst.*, Dublin, Ireland, 2001, p. 1–5.

[19] K. Kim and A. Polydoros, "Digital modulation classification: The BPSK versus QPSK case," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, San Diego, CA, USA, 1988, pp. 431–436.

[20] N. E. Lay and A. Polydoros, "Per-survivor processing for channel acquisition, data detection and modulation classification," in *Proc. IEEE Asilomar Conf. Signals Syst. Comput.*, Pacific Grove, CA, USA, 1994, pp. 1169–1173.

[21] C.-S. Park, J.-H. Choi, S.-P. Nah, W. Jang, and D. Y. Kim, "Automatic modulation recognition of digital signals using wavelet features and SVM," in *Proc. Int. Conf. Adv. Commun. Technol.*, Gangwon, South Korea, 2008, pp. 387–390.

[22] L. De Vito, S. Rapuano, and M. Villanacci, "Prototype of an automatic digital modulation classifier embedded in a real-time spectrum analyzer," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 10, pp. 2639–2651, Oct. 2010.

[23] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. Int. Conf. Eng. Appl. Neural Netw.*, 2016, pp. 213–226.

[24] N. E. West and T. O'Shea, "Deep architectures for modulation recognition," in *Proc. Int. Symp. Dyn. Spectr. Access Netw. (DySPAN)*, Baltimore, MD, USA, 2017, pp. 1–6.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778.

[26] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.

[27] T. N. Sainath, O. Vinyals, A. W. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, South Brisbane, QLD, Australia, 2015, pp. 4580–4584.

[28] Y. C. Eldar, *Sampling Theory: Beyond Bandlimited Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[30] S. Ramjee and A. E. Gamal, "Efficient wrapper feature selection using autoencoder and model based elimination," 2019. [Online]. Available: arXiv:1905.11592.

[31] B. C. Stadie, S. Levine, and P. Abbeel, "Incentivizing exploration in reinforcement learning with deep predictive models," 2015. [Online]. Available: arXiv:1507.00814.

[32] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Found. Trends Mach. Learn.*, vol. 5, no. 1, pp. 1–122, 2012.

[33] T. O'Shea and N. West, "Radio machine learning dataset generation with GNU radio," in *Proc. GNU Radio Conf.*, 2016.

[34] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 3, pp. 433–445, Sep. 2018.

[35] T. Huynh-The, C.-H. Hua, Q.-V. Pham, and D.-S. Kim, "MCNet: An efficient CNN architecture for robust automatic modulation classification," *IEEE Commun. Lett.*, vol. 24, no. 4, pp. 811–815, Apr. 2020.

[36] C.-F. Teng, C.-Y. Chou, C.-H. Chen, and A.-Y. Wu, "Accumulated polar feature-based deep learning for efficient and lightweight automatic modulation classification with channel compensation mechanism," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15472–15485, Dec. 2020.

[37] J.-A. Lee and M. Usman, "AMC-IoT: Automatic modulation classification using efficient convolutional neural networks for low powered IoT devices," in *Proc. IEEE Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Jeju, South Korea, 2020, pp. 288–293.

[38] S. Ramjee, S. Ju, D. Yang, X. Liu, A. E. Gamal, and Y. C. Eldar, "Fast deep learning for automatic modulation classification," 2019. [Online]. Available: arXiv:1901.05850.

[39] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2006, pp. 507–514.

[40] J. Li *et al.*, "Feature selection: A data perspective," *ACM Comput. Surveys*, vol. 50, pp. 1–45, Jan. 2018.

[41] Q. Gu, Z. Li, and J. Han, "Generalized fisher score for feature selection," 2012. [Online]. Available: arXiv:1202.3725.

[42] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint $l2,1$-norms minimization," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2010, pp. 1813–1821.

[43] C. Ding, D. Zhou, X. He, and H. Zha, "R_1-PCA: Rotational invariant L_1-norm principal component analysis for robust subspace factorization," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 281–288.

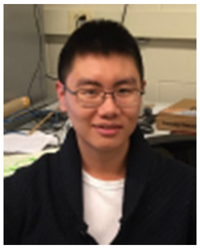
[44] K. De Rajat, N. R. Pal, and S. K. Pal, "Feature analysis: Neural network and fuzzy set theoretic approaches," *Pattern Recognit.*, vol. 30, no. 10, pp. 1579–1590, 1997.

[45] G. Ivosev, L. Burton, and R. Bonner, "Dimensionality reduction and visualization in principal component analysis," *Anal. Chem.*, vol. 80, no. 13, pp. 4933–4944, 2008.

[46] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

Sharan Ramjee (Student Member, IEEE) received the B.S. degree in computer engineering from Purdue University in 2020. He is currently pursuing the M.S. degree in computer science with a concentration in Artificial Intelligence with Stanford University. His research interests include deep learning and signal processing.

Shengtai Ju (Student Member, IEEE) received the B.S. degree (with Highest Distinction) in electrical and computer engineering from Purdue University in 2019. He is currently pursuing the Ph.D. degree in electrical and computer engineering. His research focuses on image processing, computer vision, and video analytics.



Diyu Yang (Student Member, IEEE) received the B.S. degree in electrical engineering and applied mathematics from the University of Illinois at Urbana Champaign in 2017. He is currently pursuing the Ph.D. degree in electrical and computer engineering from Purdue University. From 2020 to 2021, he was with Tesla, where he worked on the research and development of active noise cancelation. His research interests are focused on computational imaging and more specifically on high-performance computed tomography reconstruction.

Xiaoyu Liu (Student Member, IEEE) received the B.S. degree from Shanghai Jiao Tong University in 2016, and the M.S. degree from the Electrical and Computer Engineering Department, Purdue University in 2018. She is currently a Software Developer with Oracle Inc. Her research interests include signal processing and deep learning.

Aly El Gamal (Senior Member, IEEE) received the B.S. degree in computer engineering from Cairo University in 2009, the first M.S. degree in electrical engineering from Nile University 2007, and the second M.S. degree in mathematics and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 2013 and 2014, respectively.

He is an Assistant Professor with the Electrical and Computer Engineering Department, Purdue University. His research interests include information theory and machine learning. He has received a number of awards, including the Purdue Seed for Success Award, the Purdue Office of the Provost Award for Innovative Course Design and Use of Technology, the Purdue Engineering Outstanding Teaching Award, and the DARPA Spectrum Collaboration Challenge Contract Award and Preliminary Events 1 and 2 Team Awards. He is currently serving as an Associate Editor in the area of Machine Learning and AI for Wireless at the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.

Yonina C. Eldar (Fellow, IEEE) received the first B.Sc. degree in physics and the second B.Sc. degree in electrical engineering from Tel-Aviv University (TAU), Tel-Aviv, Israel, in 1995 and 1996, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 2002.

She is currently a Professor with the Department of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel. She was previously a Professor with the Department of Electrical Engineering, Technion. She is also a Visiting Professor with MIT, a Visiting Scientist with Broad Institute, and an Adjunct Professor with Duke University, and was a Visiting Professor at Stanford. She has authored the book *Sampling Theory: Beyond Bandlimited Systems* and coauthored of four other books published by Cambridge University Press. Her research interests are in the broad areas of statistical signal processing, sampling theory and compressed sensing, learning and optimization methods, and their applications to biology, medical imaging and optics.

Dr. Eldar has received many awards for excellence in research and teaching, including the IEEE Signal Processing Society Technical Achievement Award in 2013, the IEEE/AESS Fred Nathanson Memorial Radar Award in 2014, the IEEE Kiyo Tomiyasu Award in 2016, the Michael Bruno Memorial Award from the Rothschild Foundation, the Weizmann Prize for Exact Sciences, the Wolf Foundation Krill Prize for Excellence in Scientific Research, the Henry Taub Prize for Excellence in Research (twice), the Hershel Rich Innovation Award (three times), the Award for Women with Distinguished Contributions, the Andre and Bella Meyer Lectureship, the Career Development Chair at the Technion, the Muriel & David Jacknow Award for Excellence in Teaching, and the Technion's Award for Excellence in Teaching (two times). She received several best paper awards and best demo awards together with her research students and colleagues, including the SIAM outstanding Paper Prize, the UFFC Outstanding Paper Award, the Signal Processing Society Best Paper Award, and the IET Circuits, Devices and Systems Premium Award, was selected as one of the 50 most influential women in Israel and in Asia, and is a highly cited researcher. She was a Co-Chair and a Technical Co-Chair of several international conferences and workshops. She is the Editor-in-Chief of Foundations and Trends in Signal Processing, a member of the IEEE Sensor Array and Multichannel Technical Committee and serves on several other IEEE committees. In the past, she was a Signal Processing Society Distinguished Lecturer, a member of the IEEE Signal Processing Theory and Methods and Bio Imaging Signal Processing technical committees, and served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the *EURASIP Journal on Advances in Signal Processing*, the *SIAM Journal on Matrix Analysis and Applications*, and the *SIAM Journal on Imaging Sciences*. She was a Horev Fellow of the Leaders in Science and Technology Program at the Technion and an Alon Fellow. She is a member of the Israel Academy of Sciences and Humanities (elected 2017), and a EURASIP Fellow. She was a member of the Young Israel Academy of Science and Humanities and the Israel Committee for Higher Education.