Design and Analysis of Convolutional Neural Network for RF Signal Modulation Classification for In-Orbit Deployment

Chris Yakopcic
Prixarc, LLC
Beavercreek, Ohio, USA
c.yakopcic@prixarc.com

Tarek M. Taha
Brisk Computing, LLC
Centerville, Ohio, USA
ttaha@ieee.org

Sanjeevi Sirisha Karri
Prixarc, LLC
Beavercreek, Ohio, USA
sanjeevi@prixarc.com

Guru Subramanyam
Prixarc, LLC
Beavercreek, Ohio, USA
guru@prixarc.com

Aaron D. Smith
NASA Glenn Research Center
Cleveland, Ohio, USA
aaron.smith@nasa.gov

Janette C. Briones
NASA Glenn Research Center
Cleveland, Ohio, USA
janette.c.briones@nasa.gov

***Abstract*— To effectively transmit data to and from satellites requires a complex and robust RF communication system. Using adaptive link techniques, several different types of signal modulations may be used to maximize satellite efficiency depending on a variety of unexpected channel impairments. We propose a neural network algorithm capable of learning these RF signal modulations using a supervised learning technique designed for low power, high-efficiency in-orbit deployment. Thus, as signal modulation changes, a receiving satellite would be able to recognize this modulation update without adjustment from an operator, or any delay due to control signal transmission.**

The work presented demonstrates a convolutional neural network (CNN) capable of learning and recognizing a set of modulation schemes commonly used to transmit RF information. We are capable of recognizing the modulation scheme from the I and Q data channels directly, with no preprocessing or data conversion required other than breaking the incoming signal into a set of uniform normalized samples. We perform a network design and size analysis, showing that we can obtain reasonably high accuracy using networks with a relatively low number of trainable parameters. Given that a user of a system such as this may wish to receive a signal using a modulation scheme that the network has not previously learned, we demonstrate that transfer learning can learn new modulation schemes by retraining only the fully connected layers in the CNN. Thus, this type of network would excel in outer space deployment using high-efficiency transfer learning hardware. Modulation recognition can be performed through rapid feedforward computation, and the CNN training process is significantly simplified when learning new modulations is required.

Keywords—deep learning, automatic modulation classification, convolutional neural network, transfer learning

I. Introduction

In space environments, particularly for low-powered satellites such as CubeSats, onboard neural processing has many applications. These include signal processing to reduce the amount of data transmitted, cognitive radio applications, and system self-diagnosis.

Deep neural networks have been pushing recent performance boundaries for various machine learning tasks in fields such as computer vision, natural language processing, and speech recognition. Since its inception, DL has created even more significant disruptions, showing outstanding success in almost every application domain. *Learning* is a procedure consisting of estimating model parameters so that the learned model (algorithm) can perform a specific task. Deep neural networks consist of several layers between the input and output layers, which allows for many stages of non-linear information processing units with hierarchical architectures to be exploited for feature learning and pattern classification [1-3]. Some articles describe DL as a universal learning approach that can solve many types of problems in different application domains. In other words, DL is not task-specific [4]. Thus, implementing a DL engine in space applications would enable the tremendous potential for edge computing outside of Earth's atmosphere. A key challenge with this is that neural processing algorithms tend to be highly power-consuming. To reduce power consumption, several commercial neural processors have been developed, particularly for deep learning applications.

To study the efficacy of deep learning hardware deployed in Earth's orbit, we implemented a CNN capable of successfully classifying different modulation schemes for data transmission. Automatic signal modulation classification (AMC) is a

978-0-7381-2565-7/21/$31.00 ©2021 IEEE

significant research direction of signal recognition. AMC is the automatic identification of the modulation format of the transmitted signals by observing the received and often corrupted data samples. It is an intermediate operation between signal detection and data demodulation. Modulation recognition is a challenging and critical task for intelligent radios [15,20], and it is our goal to develop algorithms that satisfy the following criteria:

1. The recognition algorithm needs to sense different signal types under different signal conditions (different transmission rates, modulation methods, and sensitivities).
2. It should minimize the requirements for signal preprocessing.
3. It should detect and provide high performance even for low signal-to-noise ratio (SNR) signals.

In this work, we are proposing a convolutional neural network that learns signal modulation types. Then, we show that transfer learning can teach the CNN new modulation types while only training the fully connected part of the network. This method alleviates the computation that would be required to train the entire CNN in-situ.

Given the commercial success of several different methods for implementing low-power neural networks, the proposed AMC method would allow for the deployment of real-time, high rate, low-power, and functional neural networks for RF communications. Many products have already been released in this space, including the Intel Movidius processor, NVIDIA's TX2 GPU, and the IBM TrueNorth processor. The Movidius and TX2 systems use a global memory to store weights and quickly implement different networks without any issues. IBM's TrueNorth chip is an asynchronous digital design with memory within the cores, developed to implement spiking neurons. This system has lower power consumption than traditional computing systems, but requires significant compiler-level work to map convolution networks to its 4096 cores. We have implemented many networks on this processor [5-9], and our results show that this system has meager energy consumption. We have also evaluated the recent Intel Loihi spiking neural network processor and implemented mission planning AI algorithms and Boolean satisfiability in spiking from [10-12]. This work involved converting non-spiking algorithms into spiking form. Although the Loihi does have online learning, the binary neuron outputs (due to their spiking nature) make the online training of deep networks extremely challenging. Thus, our proposed work could also be advantageous if deployed on specialized spiking processors that would benefit if some aspects of CNN training were to be simplified.

No method currently exists for deploying a CNN with transfer learning (TL) on satellites, which can be used for many applications, including cognitive radio applications, as shown in [13,14]. The development of new energy-efficient and high-performance methods for these applications have the potential to impact many commercial systems in next-generation 5G networks, smart infrastructure, autonomous systems, electronic warfare, and military systems. Work in [15] provides a strong foundation showing that deep neural networks can accurately classify signal modulation types. Our work expands on these ideas by showing that (1) transfer learning can teach the CNN additional modulation types, and (2) network size can be reduced to alleviate the computation load without a dramatic loss of accuracy. Parameter reduction will be essential when developing the minimum required network for outer space deployment.

This paper is organized as follows: Section II provides a brief background on the algorithms used in this work, and Section III describes the application to which we are applying these deep neural networks. Section IV discusses the results obtained, and Section V provides a brief conclusion.

II. Background

A. Convolutional Neural Networks

Fukushima first proposed the CNN network structure in 1980 [16]. However, it was not widely used because the training algorithm was challenging to implement. In the 1990s, LeCun et al. applied a gradient-based learning algorithm to a CNN and obtained successful results [17]. CNNs have been designed to imitate human visual processing [18], and they contain highly optimized structures to process 2D images [18]. In addition to the standard advantages of deep neural networks, CNNs have additional desirable properties. Composed of sparse connections with tied weights, the CNN has significantly fewer parameters than a fully connected network of a similar size. Furthermore, a CNN is trainable with a gradient-based learning algorithm and suffers less from the diminishing gradient problem [17].

The overall architecture of the CNN consists of two main parts, the feature extractor, and classifier. In the feature extraction layers, each layer of the network receives an input from the preceding layer. The CNN architecture consists of three types of layers: convolution, subsampling, and classification. The convolution and subsampling layers alternate within the feature extraction section of the network. The outputs of the convolution and subsampling layers are organized into multiple 2D planes known as feature maps. Convolution layers extract the features from the input images using convolution operations, and the subsampling layers abstract the feature maps through an averaging filter. The size of the features is reduced (in terms of pixels) as they propagate through the network, depending on the convolutional and subsampling kernel size. However, the number of feature maps increases as data is propagated through the network, which allows for more efficient processing through weight sharing. The input of the last convolution layer of the CNN is output to a fully connected classification layer. The output node with the highest value designates the class assigned to a given input image in the classification layer.

B. Transfer Learning

Transfer learning is a concept proposed to alleviate some of the resource constraints of neural network training. As neural networks become larger and more complex, training time can become prohibitively long. Thus, it may be advantageous to pre-train some layers in a network and then update the last few layers when new data streams through the system. Thus, the system can classify new data without the time-consuming and computationally expensive process of retraining the convolutional layers. Furthermore, the final classification layer

can be modified to include more classes than those trained initially. For example, if a user trained and deployed an AMC system to recognize a set of four different modulation types, the proposed system could employ transfer learning to learn a fifth modulation type while only training (at a minimum) the final fully connected layer in the CNN.

III. Application Description

In this work, we show that a CNN is capable of performing modulation classification. The dataset used for this analysis is the RadioML 2018.01A dataset found in [19]. This dataset consists of 2,555,904 data samples that include 24 different modulation types at 26 different attenuation levels, and thus 4096 samples from each unique class (assuming class differentiation in both the modulation and attenuation direction). There are 98,304 samples available at each attenuation level and 106,496 samples available for each modulation type. Each of the 2,555,904 examples in the dataset is 1024 samples long. Table 1 lists the 24 different modulation types present in the 2018 RadioML dataset according to the numerical classification label. The order of the modulation types in Table I is matched to the order listed in [15] as 'Difficult Classes.' A number of these modulations use phase-shift keying, in addition to a few AM and FM signals as well. Fig. 1 displays several waveform samples present in the dataset to display some examples of the type of data available.

TABLE I. Different Modulation Types Present in the RadioML 2018 Dataset

Class Number	Mod. Class
1	OOK
2	4ASK
3	8ASK
4	BPSK
5	QPSK
6	8PSK
7	16PSK
8	32PSK
9	16APSK
10	32APSK
11	64APSK
12	128APSK
13	16QAM
14	32QAM
15	64QAM
16	128QAM
17	256QAM
18	AM-SSB-WC
19	AM-SSB-SC
20	AM-DSB-WC
21	AM-DSB-SC
22	FM
23	GMSK
24	OQPSK

Fig. 1 shows data samples from two different modulation types: BPSK and 32PSK at two different SNRs: 30dB and 8dB. Many of these modulation types look similar when they are within the same subset (for example, 8PSK and 16PSK), which sometimes makes ACM difficult in these cases. Thus, we investigate complex machine learning algorithms to carry out this classification task. This will allow the possibility of classifying incoming data based in I and Q channels directly, without the need of extracting secondary features such as cyclo-stationary moments [15].

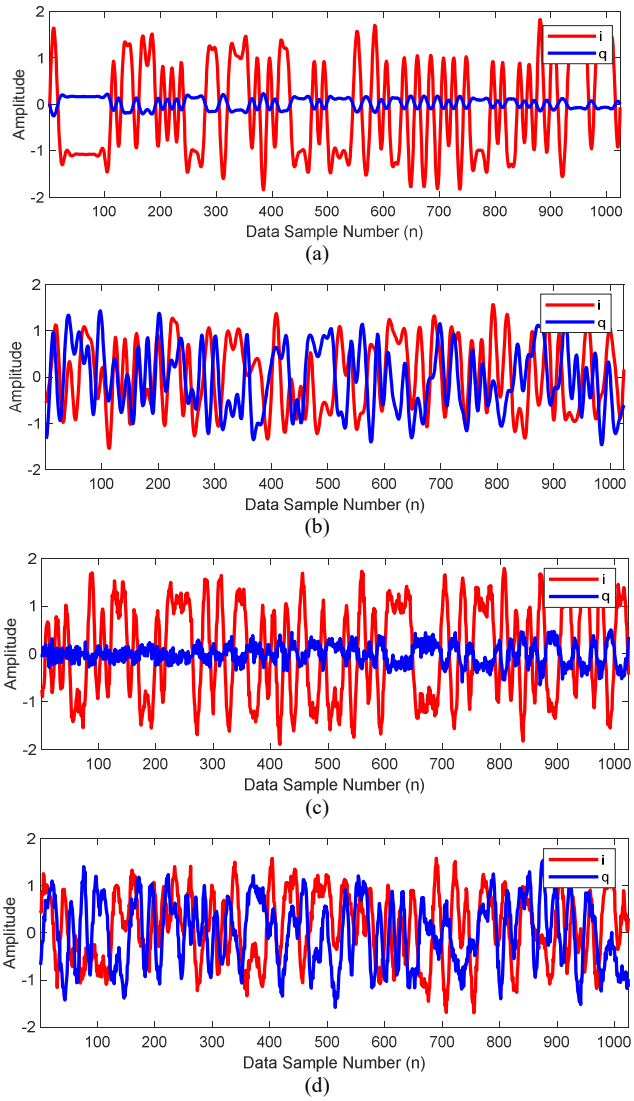


Fig. 1. Example waveforms from the RadioML dataset for the (a) BPSK and (b) 32PSK waveforms at 30dB SNR and (c) BPSK and (d) 32PSK at 8dB SNR.

IV. Results

The results in this work describe two different studies rooted in CNNs for AMC: (1) transfer learning to learn new modulation types previously unknown to the network and (2) reduction in network parameters to determine if it is possible to uphold network accuracy with smaller networks.

A. Transfer Learning

We started with a relatively large CNN that contained two convolution layers and two fully connected layers (see Fig. 2). To show the impact of a fairly aggressive transfer learning problem, we broke the 24 modulation classes within the RadioML 2018 dataset into two groups. Classes 1 through 12 were considered Group A, and classes 13 through 24 were considered Group B (see Table I for class numbers). Table II

shows the results for four different training combinations, where the latter two are the transfer learning cases. The results show that accuracy remains relatively high even when learned classes are replaced when transfer learning 12 new classes. For Group A, about 7% accuracy is lost by transferring instead of training the entire system. However, only about 2% accuracy is lost when transferring learning Group B instead of training the entire system. Figs. 3 and 4 show the confusion matrices where Group A is entirely trained and where Group A is transfer learned from a CNN trained using Group B. The results are very similar, with only a bit more confusion in the center of the plot in Fig. 4.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 32, 1024, 1]           1,632
              ReLU-2          [-1, 32, 1024, 1]               0
         MaxPool2d-3           [-1, 32, 512, 1]               0
            Conv2d-4           [-1, 64, 512, 1]          51,264
              ReLU-5           [-1, 64, 512, 1]               0
         MaxPool2d-6           [-1, 64, 256, 1]               0
           Dropout-7                [-1, 16384]               0
            Linear-8                 [-1, 1000]      16,385,000
            Linear-9                   [-1, 24]          24,024
================================================================
Total params: 16,461,920
Trainable params: 16,461,920
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.01
Forward/backward pass size (MB): 1.38
Params size (MB): 62.80
Estimated Total Size (MB): 64.19
----------------------------------------------------------------
```

Fig. 2. Network Structure of the CNN used for this transfer learning experiment.

TABLE II. TRANSFER LEARNING RESULTS WHEN TRAINING THE ENTIRE CNN FOR 12 MODULATION CLASSES AND TRANSFER LEARNING THE OTHER 12

Metric	Train A (40 Epochs)	Train B (40 Epochs)	Train A Transfer B (40 + 40 Epochs)	Train B Transfer A (40 + 40 Epochs)
Accuracy	91.15%	80.40%	78.32%	83.34%

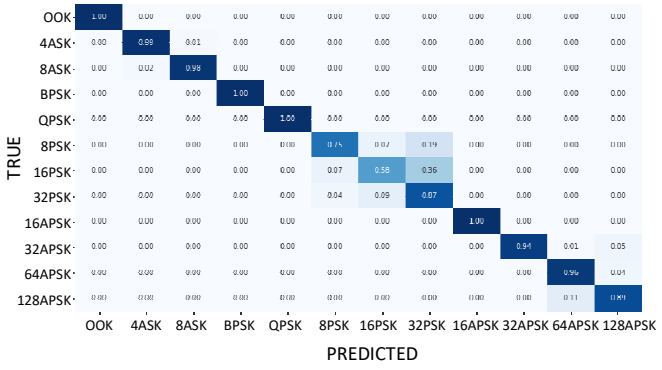


Fig. 3. Normalized confusion matrix when learning modulation Group A by training the entire network.

B. *Reduced Network Transfer Learning*

The transfer learning experiment performed in the previous subsection was performed using a reasonably extensive network. Outer space deployment may require minimization of computation throughput. Thus, the network in Fig. 5 contains less than 1000 times the parameters of the network in Fig 2. Here, we show a simplified transfer learning experiment in the confusion matrices in Figs. 6 through 8.

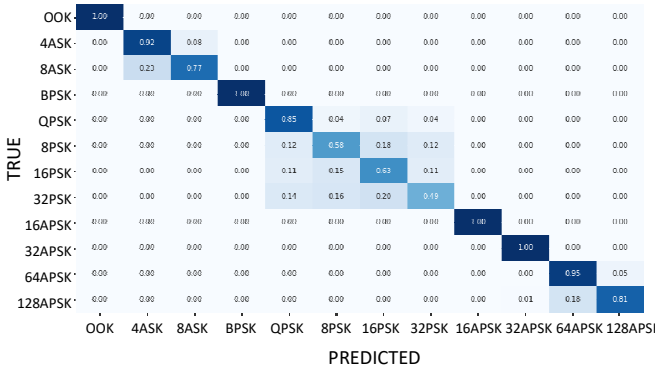


Fig. 4. Normalized confusion matrix when transfer learning modulation Group A by training only the fully connected layers (after training the entire network to learn Group B).

Fig. 6 shows the confusion matrix for the network after training to learn four modulation classes within the RadioML 2018 dataset with high accuracy. Then, in Fig. 7, the inference was performed using a dataset that contained samples from the four learned classes in addition to a fifth unlearned class. This unlearned modulation was classified incorrectly 100% of the time. However, transfer learning was then applied, and the fully connected layers in the network were retrained based on samples from the four original classes and the fifth class (QPSK). Fig. 8 shows that not only are the QPSK samples now recognized correctly, but accuracy has improved for the 8ASK case as well.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 12, 1024, 1]              72
              ReLU-2          [-1, 12, 1024, 1]               0
         MaxPool2d-3           [-1, 12, 512, 1]               0
            Conv2d-4           [-1, 12, 512, 1]             432
              ReLU-5           [-1, 12, 512, 1]               0
         MaxPool2d-6           [-1, 12, 256, 1]               0
            Conv2d-7           [-1, 12, 256, 1]             432
              ReLU-8           [-1, 12, 256, 1]               0
         MaxPool2d-9           [-1, 12, 128, 1]               0
           Conv2d-10           [-1, 12, 128, 1]             432
             ReLU-11           [-1, 12, 128, 1]               0
        MaxPool2d-12            [-1, 12, 64, 1]               0
           Conv2d-13            [-1, 12, 64, 1]             432
             ReLU-14            [-1, 12, 64, 1]               0
        MaxPool2d-15            [-1, 12, 32, 1]               0
           Conv2d-16            [-1, 12, 32, 1]             432
             ReLU-17            [-1, 12, 32, 1]               0
        MaxPool2d-18            [-1, 12, 16, 1]               0
          Dropout-19                  [-1, 192]               0
           Linear-20                   [-1, 64]          12,288
           Linear-21                   [-1, 12]             768
================================================================
Total params: 15,288
Trainable params: 15,288
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.01
Forward/backward pass size (MB): 0.46
Params size (MB): 0.06
Estimated Total Size (MB): 0.53
----------------------------------------------------------------
```

Fig. 5. Network Structure of the CNN used for the reduced network transfer learning experiment.

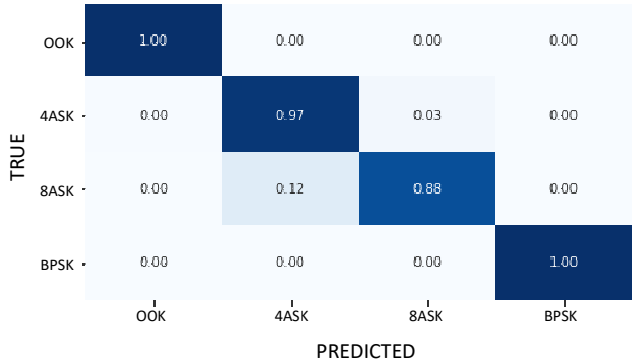


Fig. 6. Normalized confusion matrix when learning four modulations in the RadioML 2018 dataset.



Fig. 7. Normalized confusion matrix when performing inference with the four learned modulation classes in addition to a fifth.

C. *Neural Network Reduction Study*

Given that this work aims to prepare a CNN algorithm capable of performing AMC for outer space deployment, it may be advantageous to reduce the computational load of the CNN as much as possible. Table III presents the results of a study that shows how accuracy is impacted when reducing the number of trainable parameters when the entire CNN is trained.

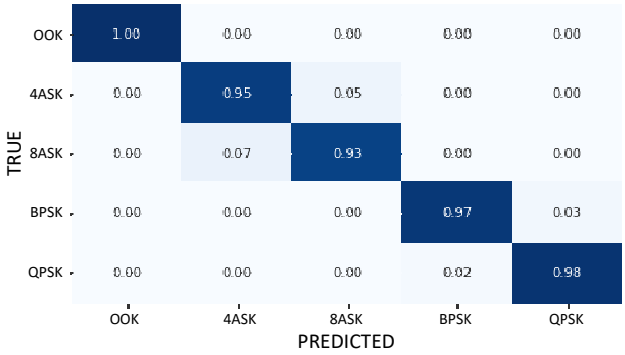


Fig. 8. Normalized confusion matrix when performing inference with the four learned modulation classes in addition to a fifth after transfer learning a dataset containing all five classes.

In this study, several aspects of the CNN were altered to determine the best way to increase accuracy while reducing the computational load. The number of trainable parameters within each network varies from about 500,000 to about 4,000. The accuracy vs. computation tradeoff will depend on the use case's requirements, but it would appear that the network containing 40,024 parameters with greater than 80% accuracy is favorable. For instances that require a higher classification accuracy, we plan to investigate expanded training experiments and post processing techniques to improve classification. When comparing this network to the 521,042 parameter network, the smaller one can achieve about the same training accuracy and better testing accuracy while using less than ten times the trainable parameters.

In the future, we plan to study these networks further to determine if there are ways to improve accuracy more. An argument could be made that all 24 modulations are not helpful for a given application and that some could be omitted to improve accuracy. Also, it is unlikely that a signal is being transmitted that would constantly be switching its modulation type. Therefore, perhaps multiple samples could be considered before a classification decision is made to improve overall

TABLE III. SET OF REDUCED PARAMETER CNNS TO DETERMINE OPTIMAL TRADEOFF BETWEEN COMPUTATIONAL LOAD AND ACCURACY

Convolution Layers	Filter Size	FC Layers	Epochs	Parameters	Training Accuracy (%)	Testing Accuracy (%)
2→16→16→16	1 by 5	2048→250→24	40	521,042	**82.0**	**74.8**
	1 by 3		40	519,954	**78.9**	**71.7**
2→16→16→16→16→16	1 by 3	512→250→24	40	137,522	**77.4**	**75.9**
		512→128→24	40	72,008	**71.4**	**70.28**
2→16→16→16→16→16→16	1 by 3	256→128→24	40	40,024	**76.7**	**75.6**
			80	40,024	**82.7**	**81.4**
2→12→12→12→12→12→12	1 by 3	192→128→24	40	30,104	**66.3**	**65.2**
			80	30,104	**75.7**	**74.8**
2→12→12→12→12→12→12	1 by 3	192→64→24	80	16,216	**77.8**	**76.5**
			160	16,216	**78.2**	**77.4**
2→8→8→8→8→8→8	1 by 3	128→64→24	80	10,872	**68.3**	**67.4**
			160	10,872	**72.5**	**71.7**
2→8→8→8→8→8→8	1 by 3	128→24	160	4,152	**54.6**	**54.06**

performance, even if the CNN does not correctly classify every sample.

V. Conclusion

In this paper, we demonstrate a CNN for automatic modulation classification. We build on previous work by showing that the network can learn additional classes after deployment, as long as the system can retrain the fully connected layers of the CNN in-situ. Furthermore, it may be possible to significantly reduce the number of trainable parameters within a CNN required to perform this task, thus significantly reducing the computational load for a device meant to operate in outer space. In the future, we plan to improve accuracy further when trainable parameter count is low, and we will examine the types of hardware that may be suitable for executing this network.

References

[1] Schmidhuber, J. Deep Learning in Neural Networks: An Overview. Neural Netw. 2015, 61, 85–117.

[2] Bengio, Y.; LeCun, Y.; Hinton, G. Deep Learning. Nature 2015, 521, 436–444. Electronics 2019, 8, 292 53 of 67

[3] Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. IEEE Trans. Pattern Anal. Mach. Intell. 2013, 35, 1798–1828.

[4] Bengio, Y. Learning deep architectures for AI. Found. Trends Mach. Learn. 2009, 2, 1–127.

[5] Md Zahangir Alom, Brian Van Essen, Adam T. Moody, David Peter Widemann, and Tarek M. Taha, "Convolutional Sparse Coding on Neurosynaptic Cognitive System," IEEE International Joint Conference on Neural Networks, May 2017.

[6] Md Zahangir Alom, and Tarek M. Taha, "Network Intrusion Detection for Cyber Security on Neuromorphic Computing System," IEEE International Joint Conference on Neural Networks, May 2017.

[7] Md Zahangir Alom, Brian Van Essen, Adam T. Moody, David Peter Widemann, and Tarek M. Taha, "Quadratic Unconstrained Binary Optimization (QUBO) on Neuromorphic Computing System," IEEE International Joint Conference on Neural Networks, May 2017.

[8] Stefan Westberg, Anthony Reiling, William Mitchell, Tarek Taha, Eric Balster, and Kerry Hill, "Coarse Grain Power Modeling for Low Power Convolutional Neural Network Inference Hardware," NAECON 2018.

[9] C. Yakopcic, N. Rahman, T. Atahary, M. Z. Alom, T. M. Taha, A. Beigh, and S. Douglass, "Spiking Neural Network for Asset Allocation Implemented Using The TrueNorth System," IEEE Cognitive Communications for Aerospace Applications Workshop, pp. 1-6, Cleveland, OH, June, 2019.

[10] Chris Yakopcic, Nayim Rahman, Tanvir Atahary, Tarek Taha, Alex Beigh and Scott Douglass, "High Speed Cognitive Domain Ontologies for Asset Allocation Using Loihi Spiking Neurons," IEEE International Joint Conference on Neural Networks, July 2019.

[11] C. Yakopcic, N. Rahman, T. Atahary, T. M. Taha, and S. Douglass, "Solving Constraint Satisfaction Problems Using the Loihi Spiking Neuromorphic Processor," IEEE Design, Automation and Test in Europe, 2020, pp.1079-1084, Grenoble, France, March, 2020.

[12] C. Yakopcic, N. Rahman, T. Atahary, T. M. Taha and S. Douglass, "Leveraging the Manycore Architecture of the Loihi Spiking Processor to Perform Quasi-Complete Constraint Satisfaction," International Joint Conference on Neural Networks, 2020 (Accepted)

[13] Timothy M. Hackett, Sven G. Bilén, Paulo Victor Rodrigues Ferreira, Alexander M. Wyglinski, Richard C. Reinhart, and Dale J. Mortensen, "Implementation and On-Orbit Testing Results of a Space Communications Cognitive Engine," IEEE Transactions On Cognitive Communications and Networking, Vol. 4, No. 4, December 2018.

[14] Paulo Victor Rodrigues Ferreira, Randy Paffenroth, Alexander M. Wyglinski, Timothy M. Hackett, Sven G. Bilén, Richard C. Reinhart, and Dale J. Mortensen, "Multiobjective Reinforcement Learning for Cognitive Satellite Communications Using Deep Neural Network Ensembles," IEEE Journal On Selected Areas In Communications, Vol. 36, No. 5, May 2018.

[15] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-Air Deep Learning Based Radio Signal Classification," IEEE Journal of Selected Topics in Signal Processing, vol. 12. no. 1 pp. 168-179, Feb. 2018.

[16] G. Indiveri and S.-C. Liu, "Memory and Information Processing in Neuromorphic Systems," Proceedings of the IEEE, 103(8), pp. 1379-1397, 2015.

[17] R. LiKamWa, Y. Hou, J. Gao, M. Polansky, L. Zhong, "RedEye: Analog ConvNet Image Sensor Architecture for Continuous Mobile Vision," ACM/IEEE 43rd Annual International Symposium on Computer Architecture, pp. 255-266, June, 2016.

[18] Y. Shim, A. Sengupta, K. Roy, "Low-Power Approximate Convolution Computing Unit with Domain-Wall Motion Based "Spin-Memristor" for Image Processing Applications," 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), August, 2016.

[19] RF Datasets for Machine Learning: https://www.deepsig.io/datasets

[20] O. A. Dobre, "Signal Identification for Emerging Intelligent Radios: Classical Problems and New Challenges," IEEE Instrumentation and Measurement Magazine, vol. 18 no. 2, pp. 11-18, April, 2015.