Received 26 July 2021; revised 27 August 2021; accepted 10 September 2021. Date of publication 20 September 2021; date of current version 4 October 2021.

Digital Object Identifier 10.1109/OJCOMS.2021.3112939

An RFML Ecosystem: Considerations for the Application of Deep Learning to Spectrum Situational Awareness

LAUREN J. WONG [1,2], WILLIAM H. CLARK, IV [2] (Graduate Student Member, IEEE), BRYSE FLOWERS [3] (Graduate Student Member, IEEE), R. MICHAEL BUEHRER [1] (Fellow, IEEE), WILLIAM C. HEADLEY [2] (Senior Member, IEEE), AND ALAN J. MICHAELS [1,2] (Senior Member, IEEE)

[1]Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24060, USA

[2]Hume Center for National Security and Technology, Virginia Tech, Blacksburg, VA 24060, USA

[3]Department of Electrical and Computer Engineering, University of California at San Diego, San Diego, CA 92093, USA

CORRESPONDING AUTHOR: L. J. WONG (e-mail: ljwong@vt.edu)

ABSTRACT While deep learning (DL) technologies are now pervasive in state-of-the-art Computer Vision (CV) and Natural Language Processing (NLP) applications, only in recent years have these technologies started to sufficiently mature in applications related to wireless communications, a field loosely termed Radio Frequency Machine Learning (RFML). In particular, recent research has shown DL to be an enabling technology for Cognitive Radio (CR) applications as well as a useful tool for supplementing expertly defined algorithms for spectrum awareness applications such as signal detection, estimation, and classification. A major driver for the usage of RFML is that little, to no, *a priori* knowledge of the intended spectral environment is required, given that there is an abundance of representative raw Radio Frequency (RF) data to facilitate training and evaluation. However, in addition to this fundamental need for sufficient data, there are other key considerations, such as trust, security, and hardware requirements, that must be taken into account before deploying RFML systems in real-world wireless communication applications that largely go unaddressed in the current literature. This paper examines the prior works related to these major research considerations, with focus on the dependencies between them and factors unique to the RFML space.

INDEX TERMS Survey, deep learning, neural networks, radio frequency machine learning, spectrum awareness, dynamic spectrum access, cognitive radio, automatic modulation classification, specific emitter identification, signal detection.

I. INTRODUCTION

IN RECENT years, deep learning (DL) algorithms have been utilized in the wireless communications domain for facilitating spectrum situational awareness applications such as signal detection, signal parameter estimation, Automatic Modulation tagification (AMC), and Specific Emitter Identification (SEI). Given the initial successes in these areas, among others in the wireless communications domain, DL is considered a transformative technology in the upcoming 5G standard and is expected to be a core component of 6G technologies and beyond [1].

While the term RFML has been used in the literature to loosely describe any application of machine learning (ML) to the RF domain, RFML systems were first defined as systems [2].

- That utilize autonomous feature learning from raw data that can "learn the characteristics used to identify and characterize signals"
- Used to detect, identify, and recognize signals-of-interest
- Able to autonomously configure the RF sensor or communications platform to be most effective in changing communications environments

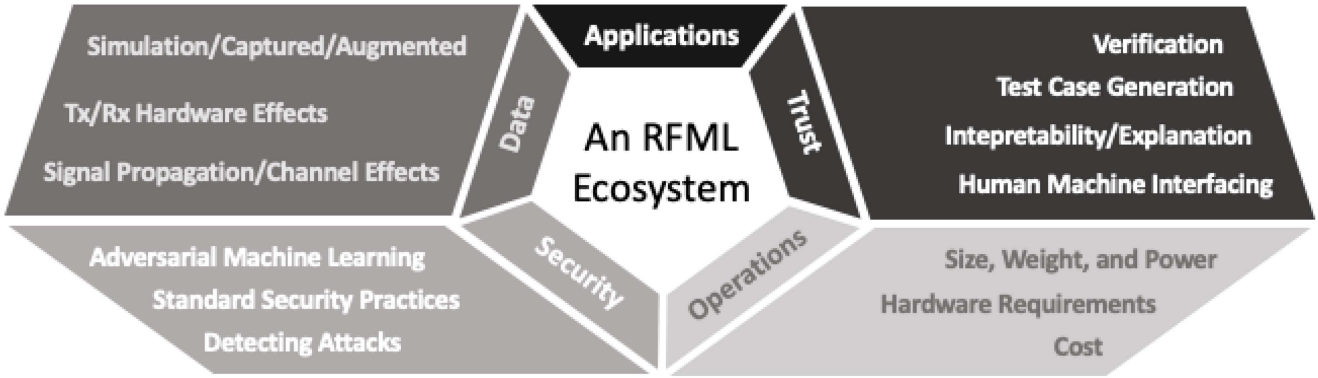

FIGURE 1. An RFML "Ecosystem" is made up of the major research thrust areas that must be considered holistically in order to utilize RFML systems in real-world applications.

- "Able to digitally synthesize virtually any possible waveform"

We use these guiding principles to narrow the scope of the subject matter examined herein, and focus discussions and the literature review undertaken on techniques aiming to reduce the amount of expert-defined features and prior knowledge needed for the intended application. More specifically, *we focus discussion on works which utilize raw RF data as input to ML techniques*, while works utilizing or deriving pre-defined expert features as input to classical ML methods are drawn upon only for context. Further, the works discussed and cited herein almost exclusively use DL techniques (Deep Neural Networks (DNNs) in particular), as DL models are better equipped to handle high dimensional inputs than traditional ML models.

To date, the primary area of research in RFML has focused on providing novel solutions to spectrum awareness and cognitive radio tasks. Meanwhile, only limited attention has been paid to the impacts of the data on learned behavior, vulnerabilities of RFML in adversarial contexts, and the requirements for deploying these algorithms in real-world applications including testing, verification, and assurance. Further, even less work has directed attention towards the relationships between these less recognized areas of research and the application space. Ultimately, these limitations have hampered the widespread adoption of RFML algorithms thus far.

This paper aims to address this shortcoming through a holistic overview and survey of prior works related to five major research thrusts, namely RFML applications, dataset creation, security, trust and assurance, and operational considerations, herein called an *RFML Ecosystem* and illustrated in Figure 1. Additionally, particular attention is paid to

i. RF domain specific considerations, not present in fields such as image, audio, or natural language processing such as channel effects and hardware imperfections, and
ii. Relationships between components of an RFML ecosystem, as they are inextricably dependent and therefore must be considered in tandem
iii. How practical environment and hardware limitations affect the feasibility of using emerging RFML techniques in real-world systems

Thus, this work provides a holistic guide for RFML researchers and developers looking to develop realizable and deployable solutions for real-world applications and to promote the advancement of DL architectures and algorithms purpose-built for the RF domain.

This paper is organized as follows: Section II compares and contrasts this work to existing papers surveying the use of ML techniques in the RF domain and highlights the contributions of this work. In Section III, we survey the relevant RFML applications found in the literature to provide context for the sections that follow. Next, Section IV discusses the types of RFML datasets used in existing work and how to create them, including discussion of real-world and hardware effects on RF data and guidelines for using publicly available datasets versus custom datasets. Section V discusses general RFML security with a focus on adversarial RFML techniques and methods of defending RFML systems from attack. Given limited work in the area, Section VI highlights the need for work in verification, testing, and interpretation/explanation methods for RFML techniques, and surveys existing works in other ML and DL modalities that show promise for use in the RF domain. Section VII discusses operational considerations such as size, weight, power, and cost (SWaP-C). In Section VIII, we conclude the work by highlighting three of the key challenges and areas of future research needed to mature RFML for deployment spanning the five elements of the "ecosystem" presented.

II. PRIOR WORK AND CONTRIBUTIONS

Though there have been a multitude of papers in the past few years surveying the use of ML in wireless communications systems, as shown in Table 2, few have focused on RFML, as defined in the previous section. The vast majority of these existing surveys are generally algorithm and application focused, and overview, compare, and contrast the ML approaches used and highlight the variety of applications, operating conditions, and assumptions under which

TABLE 1. Nomenclature.

AMC	Automatic Modulation Classification
AWGN	Additive White Gaussian
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CNN	Convolutional Neural Network
CR	Cognitive Radio
CV	Computer Vision
DL	Deep Learning
DNN	Deep Neural Network
DSP	Digital Signal Processing
EA	Evolutionary Algorithm
FGSM	Fast Gradient Sign Method
FSK	Frequency Shift Keying
GAN	Generative Adversarial Network
GPU	Graphical Processing Unit
HMI	Human-Machine Interaction
IoT	Internet-of-Things
IQ	In-Phase and Quadrature
ML	Machine Learning
NLP	Natural Language Processing
NN	Neural Network
OTA	Over-the-Air
PAPR	Peak-to-Average Power Ratio
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency
RNN	Recurrent Neural Network
RFML	Radio Frequency Machine Learning
SEI	Specific Emitter Identification
SNR	Signal-to-Noise Ratio
SVM	Support Vector Machine
SWaP	Size, Weight, and Power
SWaP-C	Size, Weight, Power, and Cost
TPU	Tensor Processing Unit
UAV	Unmanned Aerial Vehicle

ML approaches are beneficial over traditional techniques. However, these existing works more broadly survey the use of ML and/or DL in the context of wireless communications systems, and as a result, they primarily review works utilizing hand-crafted features as input and not raw RF data, as addressed in this paper. For example, [3] focuses on applications of DL in wireless communications systems broken down by layer (PHY, data link, network, etc), and in [4], focus is placed on the contexts/applications under which Neural Networks (NNs) are useful in wireless communications networks such as for multiple radio access, edge computing, and in the Internet-of-Things (IoT).

Several existing surveys have focused their attention specifically on applications of ML in the context of wireless networking for problems such as routing, data aggregation, and query processing [5]–[8]. These surveys also focus on the feasibility of using ML to perform such tasks in constrained and decentralized environments, but because these surveys examine problem spaces farther up the network stack, they generally require synchronization and demodulation, unlike in this work.

Similarly, there are copious works surveying aspects of CR [9]–[14]. However, as the primary goal CR systems is to adapting to changing channel conditions without the need for a human in the loop or time intensive re-configurations, differing from the goal of RFML as discussed in this work. RFML is typically discussed in these surveys as a tool used to improve CR capabilities, rather than being the focus of the work as it is here. Meanwhile, a number of surveys have taken a more introductory or tutorial style approach to applying ML techniques in the RF domain [15]–[17]. However, these works focus more on the algorithmic details of specific RFML techniques when compared to this work.

In all the surveys discussed above, little-to-no emphasis is placed on how the components of an RFML ecosystem might impact the works cited, with the primary focus being the algorithms and applications of interest. A few more unique surveys have also examined individual components of an RFML ecosystem including dataset generation considerations using tools such as GNU Radio [18] and security and privacy challenges faced in cognitive wireless sensor networks [19], [20]. Additionally, though discussed in the context of CR, [21] and [22] also discuss operational considerations for using RFML in a military setting and solutions for combating practical imperfections encountered in CR system (i.e., noise uncertainty, channel/interference uncertainty, hardware imperfections, signal uncertainty, synchronization issues), with discussion relevant to Sections IV and VII. However, these surveys fail to acknowledge the dependencies between the components of an RFML Ecosystem, one of the primary focuses of this work.

In contrast, this paper surveys RFML-related applications and solutions for context in Section III, but focuses primarily on holistically bringing together the works in RFML dataset creation, security, trust and assurance, and deployment, which bring to light a broader RFML ecosystem that underpins them. Through this discussion, a better understanding of the components of RFML systems, and their interplay, is developed, providing a framework for future research and development.

III. APPLICATIONS

An RFML Ecosystem, as the name implies, is composed of the supporting considerations in the development and deployment of RFML applications. Therefore, before we discuss the different facets of an RFML Ecosystem, it is important to provide context through a discussion of the relevant RFML applications found in the literature including AMC, signal detection, SEI, channel modeling or emulation, positioning or localization, and spectrum anomaly detection. Through this discussion, it is clear that the utility of DL techniques for various spectrum sensing applications has driven a sharp rise of RFML work in recent years, thereby increasing the need for work to support the deployment of these applications in real-world systems.

An overview of the algorithms described herein, including training data types and model types, is given in Table 3. It should be noted that the works cited provided herein are not exhaustive, and rather serve as quality examples of work in the area.

TABLE 2. Comparison of prior works surveying the use of ML in communications systems.

Reference	Publication	Year	Focus Area(s)	Approaches
[3]	IEEE Communications Surveys & Tutorials	2018	Intelligent Wireless Networks Applications (Anti-Jamming, Error-Correction, Interference Management, Modulation Classification, Signal Detection, Channel Resource Allocation/Management, Traffic Prediction, Link Evaluation, Routing, Scheduling, Intrusion Detection, Flow Identification)	DL
[4]	IEEE Communications Surveys & Tutorials	2019	Wireless Communications Applications (UAV networks and communications, Wireless Virtual Reality, IoT, Multi-Radio Access Technologies, Caching and Computing)	ANN
[5]	IEEE Communications Surveys & Tutorials	2014	Wireless Sensor Network Applications (Routing, Clustering and Data Aggregation, Event Detection and Query Processing, Localization and Object Targeting, Medium Access Control, Security and Anomaly Intrusion Detection, Quality of Service, Data Integrity, Fault Detection)	ML
[6]	IEEE International Conference on Information, Communications, and Signal Processing	2007	Wireless Sensor Network Applications (Energy Aware Communications, Optimal Node Deployment and Localization, Resource Allocation, Scheduling, Information Processing, Target Tracking, Event Classification and Identification)	ML
[7]	Springer Journal of Internet Services and Applications	2018	Networking (Traffic Prediction, Traffic Classification, Traffic Routing, Congestion Control, Resource Management, Fault Management, Quality of Service/Experience Management, Network Security)	ML
[8]	IEEE Communications Surveys & Tutorials	2019	Software Defined Networks (Virtualized, Edge computing, Optical, IoT, Vehicular, Wireless Sensors) and Applications (Traffic Classification, Routing, Quality of Service/Experience Prediction, Resource Management, Security)	ML
[9]	Springer Progress in Advanced Computing and Intelligent Engineering	2018	Cognitive Radio	ML
[10]	Springer International Conference on Cognitive Radio Oriented Wireless Networks	2015	Cognitive Radio (Spectrum Sensing, Modulation Classification, Power Allocation)	ML
[11]	IEEE Wireless Communications	2007	Cognitive Radio Applications (Capacity Maximization, Dynamic Spectrum Access)	ML
[12]	MDPI Sensors	2013	Cognitive Radio Wireless Sensor Networks (Spectrum Sensing, Spectrum Sharing, Prediction, Fairness, Routing, Reconfiguration, Environment Sensing, Trust and Security, Power Control)	Rule-based, ML
[13]	IEEE Communications Surveys & Tutorials	2013	Cognitive Radio (Spectrum Sensing, Medium Access Control, Signal Classification, Feature Detection, Power Allocation, Rate Adaptation, System Reconfiguration)	ML
[14]	IEEE Access	2020	Cognitive Radio-based Vehicular Ad Hoc Networks (Spectrum Sensing, Spectrum Mobility Management, Security, Road Accident Reduction, Traffic Congestion Reduction, Resource Allocation, Spectrum Aware Routing, Infotainment)	ML
[15]	Springer Development and Analysis of Deep Learning Architectures	2019	Wireless Communications (End-to-End Communications, Channel Modeling and Estimation, Signal Detection, Modulation Classification, Spectrum Situational Awareness, Adversarial Deep Learning	DL
[16]	IEEE Transactions on Cognitive Communications and Networking	2017	PHY Layer Applications (End-to-End Communications, Augmented Signal Processing, Modulation Classification)	DL
[17]	IEEE Transactions on Cognitive Communications and Networking	2018	Why, When, and How to use ML in Communications Systems	ML
[18]	IEEE Transactions on Cognitive Communications and Networking	2016	Simulated Dataset Generation with GNU Radio (Source Alphabet, Signal Modulation, Channel Simulation, Normalization, Formatting)	-
[19]	Cognitive Radio Technology Applications for Wireless and Mobile Ad Hoc Networks	2013	Attacks on, Security Mechanisms for, Security Vulnerabilities of, and Threats to Cognitive Wireless Sensor Networks	-
[20]	IEEE International Conference on Computing, Communication, and Networking Technologies	2020	Wireless Network Security (Routing Attacks, Capability Attacks, PHY Layer Attacks, Link Layer Attacks, Network Layer Attacks, Transport Layer Attacks)	ML
[21]	IEEE International Conference on Military Communications and Information Systems	2015	Framework to assess the military operational capability of cognitive radio	-
[22]	IEEE Communications Surveys & Tutorials	2015	Practical Imperfections in Cognitive Radio (Noise Variance Uncertainty, Noise/Channel Correlation, Signal Uncertainty, Channel/Interference Uncertainty, Cognitive Radio Transceiver Imperfections)	Rule-based, ML
This Paper		2021	RFML Ecosystem: Dataset Creation, Security, Trust, and Deployment	DL with raw IQ input

A. AUTOMATIC MODULATION CLASSIFICATION (AMC)

One of the earliest, and perhaps the most researched, applications of RFML for spectrum situational awareness is that of modulation classification, likely due to the historical success of ML techniques on classification tasks across modalities. Traditional modulation classification techniques typically consist of two signal processing stages: feature extraction and pattern recognition. The feature extraction stage has typically relied on the use of so-called "expert features" in which a human domain-expert pre-defines a set of signal features that allow for statistical separation of the modulation classes of interest, examples of which can be found in [51]. These expert-defined signal features are extracted from the raw received signal during a potentially time intensive and computationally expensive pre-processing stage, then used as input to a pattern recognition algorithm, which may consist of decision trees, support vector machines, NNs, among many others.

RFML-based approaches aim to replace the human intelligence and domain expertise required to identify and characterize these features using deep neural networks and advanced architectures, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to both blindly and automatically identify separating features and classify signals of interest, with minimal pre-processing and less *a priori* knowledge [18], [23]–[30]. Given the significant research in RFML-based modulation classification, it can be argued that AMC is one of the most mature fields in RFML, and has been deployed in real-world products [52].

TABLE 3. An overview of the dataset and model types used in popular RFML works.

Application	Reference	Dataset Type			Model Type				
		Real	Synthetic	Augmented	MLP	CNN	RNN	GAN	Other
AMC	[23]	X				X			
	[24]–[26]		X			X			
	[27]		X				X		
	[28]		X			X	X		
	[29]	X							X
	[30]		X	X		X	X		
Signal Detection	[31]	X				X			
	[32]		X			X			
	[33]		X			X	X		
	[34]		X						X
	[35]		X			X			
	[36]		X			X	X		
	[37]		X						X
	[38]		X			X			
SEI	[39]	X				X			
	[40], [41]		X			X			
Channel Modeling/Emulation	[42], [43]	X						X	
	[44]	X	X					X	
Positioning/Localization	[45]		X			X			
	[46]						X		
	[47]	X			X				
	[48]	X				X			
Anomaly Detection	[49]		X						X
	[50]		X				X		

B. SIGNAL DETECTION

Another area of spectrum situational awareness seeing a particular increase in the RFML literature is signal detection [31], [32], [34]. Most often, signal detection is discussed in the context of spectrum sensing as a step in identifying a specific or primary user of the spectrum [33], [35]–[38], and is traditionally performed using various energy detection methods and/or matched filtering.

Spectrogram-based signal detection is prime example of a setting in which an image processing techniques have directly been applied to solve an RFML problem. More specifically, in [31], [32], the raw In-Phase and Quadrature (IQ) samples were converted into spectrum waterfall plots to allow the spectrum information to be viewed as an image on a time-frequency plane. This has allowed a rich class of existing image processing techniques to be applied directly to perform near real-time signal detection in positive Signal-to-Noise Ratio (SNR) environments. Additional work in [38] explores the use of Generative Adversarial Network (GAN) networks to improve the signal detection performance of systems using compressive sensing.

C. SPECIFIC EMITTER IDENTIFICATION (SEI)

The goal of Specific Emitter Identification (SEI), also known as RF Fingerprinting, is to identify the transmitter responsible for sending a signal of interest. Slight but consistent differences between emitters, such as IQ imbalances, amplifier non-idealities, and other imperfections caused during the manufacturing process [40] make SEI possible. These differences not only exist between transmitter brands and models, but amongst transmitters of the *same* brand and model, which may even have been manufactured side-by-side. Further, work presented in [53] showed geographical differences including propagation channels and angle of arrival to have a dramatic effect on SEI performance as well.

Given the vast number of existing devices, each exhibiting nearly imperceptible differences from another, SEI in particular has benefited greatly from the advent of RFML [39]–[41]. While traditional SEI techniques have focused on the difficult and laborious task of defining expert features to distinguish between emitters [54], recent RFML-based solutions have used CNNs to learn the discriminating features for identifying transmitters more reliably than the hand-crafted features, and have shown the ability to identify unknown emitters [39], [40].

D. CHANNEL MODELING/EMULATION

The channel plays a defining role in the performance of RFML systems. As a consequence, including realistic channel effects, captured or simulated, into the training of RFML systems is critical to achieving top performance. In the case that sufficient data can not be captured, channel modeling is a critical component of creating realistic simulations of RF systems.

Traditionally, channel modeling requires understanding the multi-path propagation effects of a wireless channel and stochastically recreating those characteristics using mathematical approximations during simulation. However, such approaches are often computationally expensive [55]. The

area of RFML-based channel modeling and/or emulation is currently limited, but continues to grow as the need for data grows. For example, in [42], [43], a ML-based channel "stand-in" is used, which allows for channel emulation within an end-to-end RFML training routine. Alternatively, in [44], the goal is channel translation, where signal captures collected in one channel environment are augmented to resemble a different channel environment.

E. POSITIONING/LOCALIZATION

Positioning and localization play a crucial role in both military and commercial communications. For example, as the quantity of consumer-focused wireless devices continue to grow, positioning and localization become increasingly useful in emergency and safety applications, such as search and rescue operations [45], [46].

Traditionally, localization techniques have relied on expert-defined features such as received signal strength [47], [56]. However, in recent years a more rich set of RF measurements including channel transfer functions, frequency coherence functions, and channel state information have been used [46], [48]. While channel state information has been used to reach state-of-the-art and cm-level accuracy on indoor positioning tasks [46], little-to-no work has made progress towards performing localization using raw RF data.

F. SPECTRUM ANOMALY DETECTION

An emerging RFML application area is that of anomalous event detection where DL models are used to learn a baseline environment and subsequently detect/classify deviations from this baseline (so-called anomalies). An example of this budding area of research can be found in [49], where RF spectrum activities are monitored and analyzed using deep predictive coding NNs to identify anomalous wireless emissions within spectrograms. Similarly, in [50], the authors utilized recurrent neural predictive models to identify anomalies in raw IQ data. Such approaches also show promise as methods for detecting adversarial attacks or identifying out-of-distribution examples, as discussed further in Sections V and VI.

IV. DATASET CREATION

In any application of ML, representative and well-labeled datasets are of critical importance for training and/or evaluation. For RFML, observations in the dataset take the form of time samples of an RF signal, most commonly in complex baseband format with IQ notation, referred to as *raw IQ* data.

In this section, we begin by examining the three types of RFML datasets that can be created (simulated, captured, and augmented), and how these types of datasets affect the resulting RFML model. Then, we identify two categories of real-world effects, namely hardware variations and channel effects, that must be considered when developing RFML datasets. From this discussion, we derive guidelines for creating and labeling general and application-specific RFML datasets. Finally, we discuss the datasets used in existing works, considerations for using publicly available RFML datasets, and best practices for publishing work when using custom datasets.

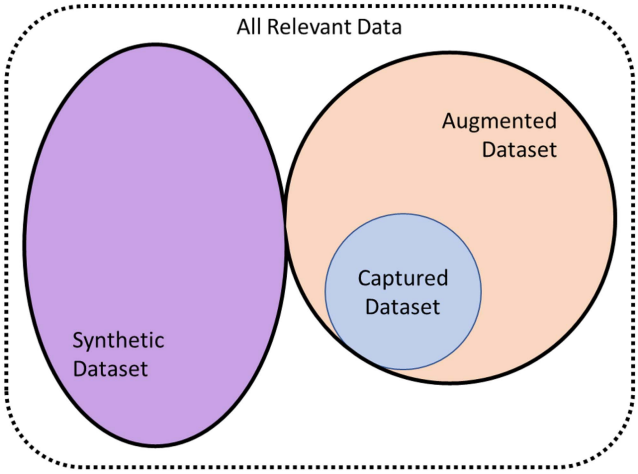

FIGURE 2. A conformal map of all relevant data to a RFML application bounded by the dotted line. While the solid black line constrains the data actually available to be used in the training of a particular system while overlaying the relationship between the three dataset types present in RFML systems.

A. SIMULATED VS. CAPTURED VS AUGMENTED DATASETS

As shown in Table 3, the data used in existing RFML works can be categorized into one of three types: simulated, captured/collected, and augmented. Simulated datasets refer to synthetically generated data, in which the transmitter, channel, and receiver are all modeled in interconnected software and/or hardware systems. In contrast, captured datasets contain signals that have been transmitted over a wireless channel. Finally, augmented datasets combine simulated and captured data by adding synthetic perturbations to captured data and/or placing synthetic signals within channel captures. For clarity, Fig. 2 depicts the relationship between the three dataset types discussed herein, and acknowledges that no dataset will ever consist of all relevant data. A more descriptive comparison of quality and quantity for these three dataset types is discussed in [57].

Simulated datasets are the most commonly used in current RFML literature, as they are the most straightforward to compile and label using publicly available toolsets such as GNU Radio [58], liquid-dsp [59], and MATLAB [60], among others. Therefore, simulated datasets are particularly well-suited to initial development. The same equations and processes used to transmit waveforms in real RF systems can be used directly in simulation [18], unlike in image processing [61]. Additionally, for *simplistic* environments, mathematical models can be used to reasonably describe common degradations such as additive interference, channel effects, and transceiver imperfections. As a result, synthetically generated RFML datasets can be good analogs for captured RFML datasets,

if carefully crafted and known models exist for the *simplistic* environment. However, a recent AMC analysis [57] showed that, without considering channel effects, models trained on simulated datasets are insufficient when applied to real captured data (i.e., during real-world deployment).

Because a capture from the real environment will include all of the different degradations that are of concern in practical RF situations, some of which may be missing from a simulated dataset due to inaccurate modeling, captured data is critical for test and evaluation prior to real-world deployment. This improved realism also reduces end-user resistance and doubt surrounding the system. However, captured data requires significant labor and resources to both gather sufficiently diverse captures for producing a training and/or evaluation datasets and to label it correctly [24]. This is the primary reason that augmented datasets are used, which combine simulated and captured data to increase the quantity of data available for training, or to incorporate more realism into a dataset over using additional synthetic data alone.

Augmented datasets aim to provide a "best of both worlds" approach by combining simulated and captured data to increase the quantity of data available for training or to incorporate more realism into a dataset. A simple augmented dataset may shuffle a small subset of real-world data captures into a larger synthetic dataset. Using this approach, the intent is to use the synthetic data to teach the DL model the features and characteristics of signals that can be well modeled in software, such as modulation schemes and simple channel models, and to use the captured data to teach the DL model the features and characteristics of signals that cannot be modeled well, such as transmitter/receiver imperfections. A more complex augmented dataset might include injecting synthetic waveforms into captured spectrum, or overlaying multiple captured observations to create a more congested observation [62]. Such datasets are useful in testing detection and classification performance of signals in a congested or interference-heavy environment with real-world transmitted signals. An additional augmentation technique often used includes adding synthetic noise to real world captures, which decreases the SNR without performing additional signal captures, thereby increasing the range of test SNRs [30], [63].

Though augmented datasets minimize the limitations of synthetic datasets (i.e., real-world model accuracy) and reducing the amount of captured data needed, augmented datasets do not yield the highest performance per observation, when compared to captured data [57]. Further, there are a multitude of open research questions related to the development and use of augmented datasets. Perhaps the most important of these open questions is how to balance the amount of real, synthetic, and augmented data used in training datasets to avoid network bias. Work in [30] and [57] both examined data augmentation in the context of AMC, with results in [57] showing that for a fixed quantity of captured data, augmentations which consider the distribution of receiver degradations (SNR, Frequency Offset, Sample Rate Mismatch) can improve the performance of a model over captured data alone. While both works showed that augmentation of the original data result in increased performance, particularly when the degradation distribution is considered, no conclusions were drawn as to how performance changes in response to varying levels of augmentation. Further, no known work has examined how to balance increases in performance with the added computational burden of augmentation, or whether such trends will be consistent across other applications such as those discussed in Section III.

B. REAL-WORLD CONSIDERATIONS

The primary difference between laboratory-measured or synthetic data and observed data is typically that the laboratory or synthetic environment is pristine in comparison to an observed environment. This is largely due to the multiple overlapping phenomena, not typically encountered in simulation or a laboratory, that degrade signals which have propagated in the physical world [64]. These real-world effects can generally be categorized as hardware variations and channel effects, and can significantly impact RFML performance, if not considered when developing the training, validation, and test datasets, as discussed in the following subsection.

Hardware variations refer to the variances between transmitter and receiver hardware platforms and the resulting impact on the received waveform. More specifically, different transmitter and receiver pairs distort waveforms from the ideal to varying degrees as a result of manufacturing variations, environmental operating conditions (i.e., temperature), and access to supporting devices like reference oscillators. These distortions take the form of non-linearities, additive noise, timing offsets, frequency offsets, phase offsets, sample rate mismatches, and/or amplitude offsets, all of which may be time varying. Depending on the application, distortions to the waveform caused by the transmitter may be a parameter of interest, or may be considered a *nuisance parameter*. In the latter case, an ensemble of transmitters is required to model an *average* transmitter, and as a result, adding varying transmitter imperfections to the training data is critical for model generalization. For example, applications such as SEI depend upon transmitter imperfections to distinguish between transmitters. Meanwhile, for applications such as AMC, transmitter imperfections are considered nuisance parameters, as the goal of AMC is to identify the modulation class, regardless of the emitter. Similarly, in the case of receiver distortions [24], [56], natural reception variations such as sampling rate differentials, frequency offsets, and varying SNR, must also be varied in the training data to encourage generalized learning [65].

Lack of synchronization between devices will also exacerbate the distortion caused by the transmitter and receiver, as well as the channel itself. To improve synchronization, detection and isolation routines are used to select spectrum

of interest. However, these algorithms introduce measurement errors in the form of time, frequency, and phase offsets between the devices which must also be modeled in order create a realistic simulated dataset. It should also be noted that higher quality hardware such as military transmitters tend to cause less severe distortions than lower quality hardware such as IoT transmitters, and the non-linearities that contribute to these variations are often dependent upon technology and hardware configurations.

The second category of real-world consideration for RFML system performance, signal propagation and/or channel effects, add noise and further degrade the signal of interest. While the baseline simulated or laboratory training environment used in mose RFML works is an Additive White Gaussian Noise (AWGN) channel, real-world channels have time-varying, often colored spectra, and uncontrolled RF interference sources such as other signals, impulsive noise (i.e., lightning), and non-linear effects associated with bursty packet transmissions. While many of these effects may be approximately modeled [66], [67], preliminary work in [62] has shown superposition of a live captured effect onto synthetic datasets through augmentation to yield better performance. However, additional work is needed to confirm this hypothesis.

The physical medium (channel) through which the signal propagates can also change over time, if the transmitters/receivers or environment is mobile, causing delayed imperfect reflections of the signal to overlap with the direct path resulting in time and frequency varying interference. Therefore, relative motion between platforms, co-channel/adjacent channel interference, and multi-path must also be considered in the development of RFML datasets. Many of these channel variations can be modeled stochastically. However, it is important that the training dataset not be biased so heavily towards learning the channel that it fails to learn the desired behavior [2].

C. GENERAL GUIDELINES FOR CREATING A DATASET

Given the discussions above, what follows are general guidelines that should be observed when creating a new RFML dataset. The first step, no matter the type of dataset being created or intended application, is identifying the expected degradations in the deployed environment (i.e., channel types, transmitter imperfections, SNR, etc.) and categorizing whether each potential degradation is fundamental to the application or a limiting nuisance parameter. For example, a waveform's modulation is fundamental to AMC, while the transmitter imperfections are fundamental to SEI, yet both applications are significantly affected by the channel over which they are observed and is therefore regarded a nuisance parameter [57], [64].

Once the expected degradations have been identified and categorized, the next steps in the dataset creation process are dependent on the type of dataset being created. More specifically, in the case of simulated datasets, the next steps are to:

- Define (mathematically) how the degradation is applied, and model the signals, channels, and imperfections, of interest (use toolkits GNU Radio, liquid-dsp, or MATLAB, if desired).
- Run "collects," sweeping over both fundamental and nuisance degradations and recording all generation parameters as metadata.

For captured datasets, the next steps are to:

- Identify the conditions under which the fundamental degradations can be collected (i.e., hardware used, collection environment, etc). For each identified nuisance degradation, attempts should be made to generalize over observations of the degradation, such as sweeping over the impairment range in simulations or changing the transmission devices or environment in some way while capturing the dataset.
- Set up transmit and receive hardware, with supporting infrastructure for power and environment. A shared timing basis is particularly important for time-varying waveforms.
- Run data collects, recording time-synchronized metadata as available. In the case of unknown metadata information, efforts should be made to characterize the distribution of the observed parameters.

For augmented datasets, the dataset creation process builds upon each of the synthetic and captured dataset methods, adding the following post-processing steps to improve the generalization of the learned behaviors:

- Identify dominant parameter variations (e.g., SNR, frequency offset) over which learned behaviors must be generalized.
- Construct an appropriate statistical distribution (usually uniform) of the parameter variations. Note that these variational parameters typically compound each other, leading to a combinatorial explosion in actual data points.
- Implement parameter variations as combinations of operations applied to the synthetic and collected signal baselines. Follow similar guidelines for separating out training, validation, and testing datasets.
- Label metadata for each augmented input.

D. METADATA, LABELING, AND APPLICATION DEPENDENCIES

During the dataset creation process, whether through simulation or collection, correctly and completely labeling the data is of the utmost importance. Ideally, though not practically, every parameter should be recorded as metadata associated with the observations in the dataset, in order to increase the number of applications pertinent to the dataset, and qualitative descriptions should be used to provide as much description as is feasible [68]. Minimally, the parameter of interest to the application should be recorded; for example, the modulation class in the case of AMC. However, the value of generating and providing datasets with significant

TABLE 4. Categorization of the types of relevant metadata to the specified application. The proof of concepts will revolve around the Dominant Metadata, while more investigative research will explore the effects from the Supporting Metadata. When moving from investigative and exploratory toward deployment the relevance and inclusion of the Ancillary Metadata becomes critical.

Application	Dominant Metadata	Supporting Metadata	Ancillary Metadata
AMC	Waveform	Frequency Offset, SNR, Bandwidth	Channel Environment
Signal Detection	Time/Frequency Bounding	SNR, Bandwidth, Signal Count	Channel Environment
SEI	Transmitter Device	Receiver Device, Bandwidth, Frequency Offset, SNR	Channel Environment
Channel Modeling/ Emulation	Channel Environment	Waveform, SNR, Bandwidth, Transmitter/Receiver Device	Carrier Frequency, Transmitter/Receiver Location
Positioning/ Location	Transmitter/Receiver Location	Channel Environment	Transmitter/Receiver Device
Spectrum Anomaly Detection	Time and Date	Receiver Carrier Frequency, Receiver Sample Rate	Channel Environment

diversity, documentation, and open usage rights should not be overlooked, as the gains observed in the image processing domain were realized with the help of crowd sourcing efforts [69].

Given that each of the applications discussed previously require and benefit from the recording of different metadata parameters during the dataset creation process, Table 4 details the types of metadata that are most relevant to each application in Section III. It should be noted that Table 4 does not include all of the metadata available to be collected, but rather focuses on the metadata that has been shown to affect each application space. In other words, if ever a general dataset for all RFML applications is created, these are the fields that should be included at minimum. However, for datasets intended for one of these applications (or perhaps a small subset), Table 4 details the metadata which should be included for each application.

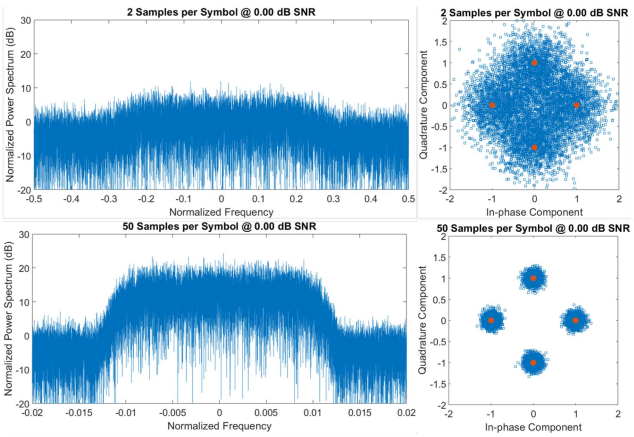

FIGURE 3. An example of the difficulties of direct comparison when the dataset's parameters are not explicitly defined. In this case, both signals can claim an SNR value of 0 dB, but the second is significantly oversampled and allows for either preprocessing or learning a filter-like behavior raising the apparent SNR observed during processing.

E. DATA USED IN EXISTING WORKS

A non-exhaustive search for publicly available RFML datasets identifies those released by Geotec [70] for Emitter Localization, DeepSig [71] for AMC, and by Genesys at Northeastern University [72] for RF fingerprinting, with additional datasets continually being registered by the IEEE Communications Society [73]. These published datasets were generated for and used in original published works [18], [23], [64], [74], and create a valuable common point of comparison for different RFML approaches within the literature. However, whenever using publicly available RF datasets, knowledge of how the signals in the dataset were generated and how to extend/modify said dataset is critical. Otherwise every signal (and the associated metadata, if applicable) should go through some form of validation by the user to ensure correctness, but validation processes are often both computationally prohibitive and time intensive, so are often overlooked.

Given the limited availability of publicly available RFML datasets which provide the requisite documentation to allow for replication and/or validation, the majority of existing works utilize custom datasets. When publishing work which uses a custom dataset, it is critical to describe the parameter space from which the data was generated, for reproducibility. To highlight the importance of describing the data generation parameters, consider the signal shown in Figure 3, where two signals have been generated with the same SNR but vastly different sampling rates. Traditional Digital Signal Processing (DSP) dictates that the bottom signal can achieve a higher maximum SNR using a matched filter, as is evident in the constellation plots. Given that most RFML applications describe performance as a function SNR, not including parameters such as sampling rates can not only impede the ability to reproduce results, but lead to false comparisons in subsequent publications.

F. DISCUSSION

Like in CV, NLP, and all other applications of ML, dataset quality and relevance are is of the utmost importance when designing an algorithm that might be deployed in the real world. More specifically, the data used to train, validate, and test any ML algorithm must be representative of the data that will be observed once deployed, as further discussed in Section VI. The three critical components of dataset creation identified in this section are:

- Considering real-world effects caused by the channel environment or transmitter/receiver hardware in any simulated, captured, and augmented datasets created, either through mathematical modeling or identifying the conditions under which the degradations can be collected, to ensure the data is representative of the deployed environment,
- Correct and complete labeling through the thorough recording of metadata parameters used during the generation or collection of data, as unsupervised and semi-supervised RFML techniques remain an area for future research (discussed further in Sections III and VIII), and
- Transparency regarding the generation/collection parameters used to create the data used in existing works, to encourage reproducibility.

For clarity, general guidelines for creating an RFML dataset are provided above in Section IV-C.

V. SECURITY

While the benefits of DL are copious across modalities, it's limitations in adversarial settings have been well documented, especially in CV [75], audio recognition [76]–[78], and NLP [79]. These attacks demonstrated in other modalities serve as a prescient warning for applications of RFML and many parallels can be drawn. Though the field of adversarial RFML is still in its infancy, recent work has shown that there are unique considerations for securing RFML-based systems due to the nature of wireless propagation, pre-processing steps used to isolate and normalize signals-of-interest for input to DNNs, and the fact that wireless communications are generally quite sensitive to perturbations in the transmission. Therefore, while this section provides a brief overview of DL security in general, the focus is on the unique considerations for RFML, a Threat Model for which is provided in Figure 4 to quickly categorize the related work in the area.

When discussing general DL security, the conversation primarily revolves around Adversarial ML which concerns the development of algorithms to attack data driven models (primarily DNN) and to defend against such attacks. This topic has gained so much popularity and concern that the taxonomy used to describe this field is still being standardized [90]. This section places primary focus on the most studied attack vector in the context of RFML, *Adversarial Evasion Attacks* (Section V-A), and defenses against this attack (Section V-B). The section concludes with a brief discussion on other attack vectors, offensive security, and future work (Section V-C).

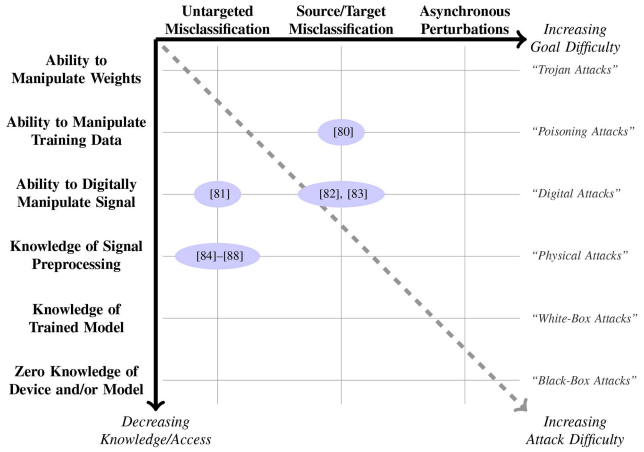

FIGURE 4. Threat Model for RFML adopted from [85], [89] and including related work.

A. ADVERSARIAL EVASION ATTACKS

As previously mentioned, Adversarial ML concerns the development of algorithms to attack data driven models (primarily DNNS) and to defend against such attacks. The topic dates back at least 15 years [91]–[95], and has broadened to include exploratory attacks that seek to learn information about (or replicate) the classifier [96] or training data [97] through limited probes on the model to observe it's input/output relationship. However, the most recent explosion in concern for the vulnerabilities of DNNs in particular is largely credited to the Fast Gradient Sign Method (FGSM) attack which showed that CV models are vulnerable to small, human imperceptible, perturbations to their input images causing misclassification [98]. This manipulation of the model's inputs to achieve a goal such as misclassification is termed an *evasion attack* and is the most widely studied sub-field of Adversarial ML, including in RFML.

Evasion attacks are most prevalent in the study of classification tasks where a key constraint is to remain *imperceptible* to the intended receiver, which is uniquely defined in the context of wireless communications. Evasion attacks can further be categorized as untargeted or targeted digital attacks, as discussed further below. While this section focuses primarily on evasion attacks on DL-based AMC systems, adversarial attacks can be applied to any RFML application discussed in Section III. That is, any application of DL for spectrum sensing discussed in Section III is susceptible to attack.

1) UNTARGETED DIGITAL ATTACKS

Untargeted digital attacks can be defined as evasion attacks in which the goal is to induce a misclassification of any kind. RFML models have been shown to be just as vulnerable to these untargeted adversarial attacks as their counterparts in CV. More specifically, both [81] and [85] showed that the FGSM attack is sufficient to completely evade AMC by a

DNN with a perturbation that is 10 dB below the actual signal. While FGSM is a computationally cheap method for creating adversarial examples, the large body of literature in adversarial ML for CV has yielded algorithms that can evade classifiers with even smaller perturbations. In [88], a more sophisticated adversarial methodology was used to carry out an attack on AMC [99]. Not only was this attack successful for a DNN, but, when the adversarial examples were input to classifiers not based on DNNs (i.e., Support Vector Machine (SVM), Decision Trees, Random Forests) the models had similar decreases in accuracy. Therefore, although adversarial ML methodologies use DNNs to craft adversarial examples, they are transferable across various classification methodologies. As a result, it can be concluded that the perturbations generated by these adversarial methodologies are not simply noise which is specific to a DNN model, they must be changing something inherent to the signal properties that are used by many methodologies for classification.

2) TARGETED DIGITAL ATTACKS

The goal of targeted digital attacks is to force a model to make a *specific* misclassification. By more closely examining *how* DNN-based AMC systems break down under evasion attacks, existing work has shown that Adversarial ML techniques take advantage of something inherent to the properties of man-made signals [82]. More specifically, because modulation formats for wireless communications are man made, they can be intuitively grouped into a hierarchical structure. For example, analog modulations, such as the Amplitude Modulation and Frequency Modulation used in older vehicle radios, are distinctly separate from digital modulations used to carry discrete symbols representing the bits of a data transmission. Within digital modulations, the formats can be hierarchically grouped into whether they represent symbols in the frequency domain (Frequency Shift Keying (FSK)), in the signal's phase (Phase Shift Keying (PSK)), or in both the signal's phase and amplitude (Quadrature Amplitude Modulation (QAM)). One would expect that a DNN would learn this intuitive grouping as well, and as a result, it would more easily confuse an analog modulation with another analog modulation than it would mislabel an analog modulation as a digital transmission. In [82], the authors used the Momentum Iterative FGSM attack to show that this is precisely the case [100]. Additionally, results presented in [82] showed that higher power adversarial perturbations are required to force misclassification to a different category of signals (i.e., from an analog to a digital) than to force misclassification to a signal belonging to the same category (i.e., from Binary Phase Shift Keying (BPSK) to Quadrature Phase Shift Keying (QPSK)).

3) RUBBISH CLASS EXAMPLES/FOOLING IMAGES

Other research has considered the ability to create examples that are classified as some target class but have no semantic meaning, using approaches such as GANs [98], [101] or Evolutionary Algorithms (EAs) [102]. Such attacks are commonly referred to as *Rubbish Class Examples* [98], *Fooling Images* [102], or, in the context of wireless communications, *Spoofing Attacks* [101]. However, no communication can occur using such an attack. Therefore, the benefits of using Spoofing Attacks are limited, and the more prevalent threat must consider how signals can be manipulated without losing their underlying semantic meaning.

4) DEFINING PERCEPTIBLE PERTURBATIONS IN WIRELESS

The main constraint on Adversarial ML techniques is generally provided as a constraint on the perturbation power: a proxy for the notion of perceptibility of the perturbation (e.g., does this perturbation affect a human observer's judgement of the image, interpretation of the audio signal content, or reading of a sentence). This notion is more easily defined in RFML as the Bit Error Rate (BER) at a receiver. More specifically, because the receiver is blind to the perturbation being applied, BER defines the perceptibility of the adversarial attack (i.e., the more obvious the perturbation, the higher the BER) [85]. In general, attacks directly transferred from CV have lower utility in wireless communications due to their large impact on the wireless transmission. That is, they yield a high BER. However, the ability to formally define a perceptible perturbation as BER has allowed recent works to create differentiable versions of the receive chain, that allow for the BER to be directly incorporated into the loss function of an adversarial attack [84], [86], yielding more sophisticated and effective threats.

B. DEFENSE

Given the advent of viable adversarial RFML approaches discussed previously and pace of research in the field, defenses must be investigated that mitigate future threats to RFML systems being deployed in high risk adversarial environments. Current methods for defending against adversarial attacks can be roughly split into two categories, discussed further in the following subsections:

i. detecting an attack is occurring in order to take counter-measures (Section V-B1), or
ii. becoming robust to attacks by increasing the power of the perturbation required to cause a misclassification (Section V-B2).

However, the vulnerabilities posed by Adversarial ML can and should be mitigated by standard security practices that secure the whole of the device and limit an adversary's access to the model's inputs and parameters, information about the model, and the pre-processing steps used, a topic further explored below in Section V-B3. Additionally, it should be noted that this section only focuses on the work that has been done specifically for RFML in the context of adversarial evasion attacks. More general surveys on adversarial attacks and defenses are provided in [91], [103].

1) DETECTING ATTACKS

Detecting an attack can be thought of as a supplemental binary classification that determines *whether or not an example is in or out of distribution*. While more general discussion of detecting out of distribution examples is left to Section VI, two metrics are proposed in [104] for detecting adversarial attacks on wireless communications. The first uses the distribution of the Peak-to-Average Power Ratio (PAPR) of the underlying signal along with the model's classification. Since the PAPR can be used as a signature for a given modulation, the work in [104] tests whether the DNN classification and PAPR signature are in agreement on the classification; if not, then the example is assumed to be an adversarial example. This test is specific to the RFML task, AMC, but agnostic of the model used. The second test uses the distribution of the output probabilities of the DNN to determine whether an example is in or out of distribution and is therefore agnostic of the task it is applied to. However, performing statistical tests during inference can increase system complexity on an already Size, Weight, and Power (SWaP) constrained RFML system, discussed further in Section VII, which leads to increased classification latency and thus decreased bandwidths, limiting real-time sensing capabilities. If the attacker becomes aware of the statistical tests being performed, this additional check can also be incorporated into the attack and likely bypassed just as the original classification was [105]. Therefore, pushing the defense methodology into the training stage of the DNN, where the computational complexity can be handled off target and without a time constraint, is often beneficial, and is discussed in the next subsection.

2) BECOMING ROBUST TO ATTACKS

The most widely used methodology for gaining robustness is adversarial training [98], [106]–[109]. Adversarial training is simply the introduction of correctly labeled adversarial examples during training time using a known adversarial attack (such as FGSM). Another method for increasing robustness involves altering the training strategy of the DNN by lowering the input dimensionality, thereby reducing the degrees of freedom available to an attacker. Work in [110] adopted both strategies, and observed an increase the model's robustness to FGSM attacks. However, the results presented in [110] also showed that lowering the input dimensionality alone was sufficient to increase robustness to an FGSM attack. However, no work has shown that an adversarially trained classifier would be robust to *all* attack methodologies [100].

As an aside, it should also be noted that adversarial training also decreases the number of training epochs needed to reach near perfect accuracy on legitimate examples. Therefore, adversarial training is not only good for conferring robustness, but can also be used a data augmentation technique for RFML, a topic previously discussed in Section IV.

Given that many proposed defenses have been quickly proven to be inadequate, it is important to be overly cautious when evaluating a new attack methodology [111]. In addition to evaluating defenses against a large and growing list of adversarial attacks such as those available in open source libraries like Cleverhans [112], research has begun looking into provable robustness. More generally, this concept is about whether the model can be trusted on real inputs, where the inputs are distorted by some perturbation, regardless of whether the perturbations are man-made or naturally occurring. A larger discussion of such topics is deferred to Section VI.

3) MITIGATION THROUGH STANDARD SECURITY PRACTICES

Defending an RFML system from attack does not have to only revolve around adversarial ML based defenses. By using standard cybersecurity best practices, an adversary can be forced to move down the Threat Model presented in Figure 4 by limiting their knowledge of and access to the RFML system. As a result, attacks become much more difficult to successfully execute. More specifically, most adversarial attacks and defenses are proposed and evaluated in a simulated, fully digital world (a digital attack in Figure 4). However, these attacks and defenses transfer to the physical environment as well [113]. In the context of RFML, this means that the perturbation is radiated from an external transmitter. Therefore, both the transmission and perturbation are impacted by channel effects, hardware impairments at both the transmitter and receiver, and DSP pre-preprocessing techniques used before reaching the DNN for classification (a physical attack in Figure 4). All of these can serve as an impediment for an attacker, forcing them to raise their adversarial perturbation power [85], [86], [88], [114]. Additionally, so-called *white-box* attacks, which assume full knowledge of the target DNN, are generally known to be more effective than *black-box* attacks which assume close to zero knowledge about the target, regardless of modality. This is not meant to say that adversarial examples do not transfer between models, only that when transferring adversarial examples between models, a small penalty on the adversary's success is incurred. Therefore, limiting the amount of information an adversary can gather about a DNN is a critical first step in defending against attack.

C. DISCUSSION AND FUTURE WORK

As RFML is commercialized, the types of threats that draw interest is expected to expand beyond disruption of a classifier at inference time. For example, models trained on Over-the-Air (OTA) captures could unintentionally expose private information, such as the underlying bit patterns of the signals within their training dataset, creating privacy concerns. This type of attack has successfully been demonstrated in an NLP setting [115], but has been yet to be successfully replicated in the context of RFML. Furthermore, given the significant resources required to develop effective RFML systems, as discussed heavily in Sections IV and VII, once a model is deployed, model replication or imitation should be prevented in order to maintain a competitive advantage [116].

While the ability to build similar models was studied in a CR setting in [117], the goal of this attack was to jam transmissions by mimicking an existing CR. This differs greatly from the goal of building a functionally equivalent version of an adversary's model for the same task, described in [116], which have yet to be studied in the context of RFML.

Additionally, the broader community's understanding of adversarial examples including *how* to create and defend against them, *when* to inject them into a DNN (training/inference), and *why* they exist, is still rapidly evolving. While much of this discussion can be applied generally across all data modalities, RFML provides the following unique considerations that must also be studied separately:

i. the physical channels between adversary and receiver are significantly different,
ii. the perceptability of the perturbation is machine defined, not human defined, and
iii. the actions taken based on the generated knowledge are application specific.

Due to i. and ii., adversarial attacks from other modalities are of limited concern to deployed RFML systems as the wireless channel forces the adversary to increase the perturbation power to a level that significantly interferes with the primary objective of the transmission: to communicate. Ongoing work has shown that more sophisticated attacks will emerge to overcome these limitations by incorporating more expert knowledge into the adversarial process such as the channel type [118], channel coding scheme [119], or device type[120]. Another key research direction going forward is identifying the type of information being transmitted from an RFML-enabled device to increase attack effectiveness through targeted attacks on acknowledgement messages or transmission decisions, for example [117], [120]. Further, recent work in determining how the training data of RFML systems can be manipulated to cause a degradation in model performance [80], [121] motivates the study of data cleaning methodologies for RFML. Such concerns echo the discussion in Section IV surrounding the need for transparency regarding the generation and metadata parameters for publicly available RFML datasets, as well as validating said datasets before use.

VI. TRUST AND ASSURANCE

For all the RFML applications discussed in Section III, there is a desire to translate laboratory *performance* into a user-defined *mission assurance*. This is critical to not only assuring that the machine learned behaviors, which are difficult to reverse engineer, behave as expected when put to practice, but more importantly, understanding how the system will respond to unanticipated stimuli and/or recognizing that an input is outside the training set of its learned responses, as alluded to in Section V. Taking this need to an extreme, a significant amount of end-user confidence is required to give RFML systems the authority to permit autonomous weapons release [122].

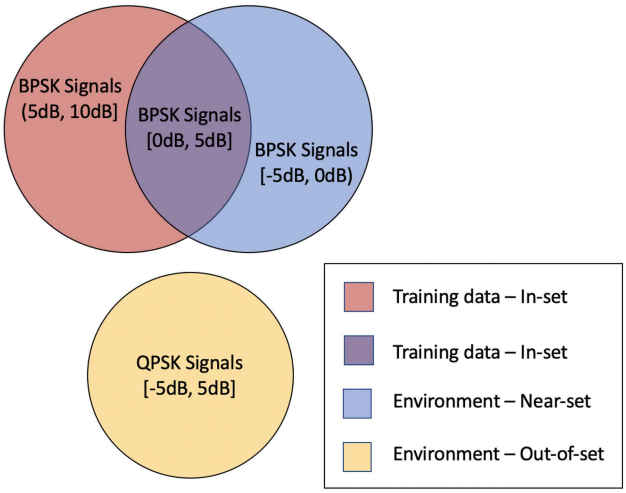

FIGURE 5. A pictorial representation of the logical relationship between in-set, near-set, and out-of-set data types for an example RFML algorithm trained on BPSK signals with SNRs between 0dB and 10dB.

For most current RFML techniques, learned behaviors are a function of correlation rather than causation. That is, algorithms are data-driven, and thereby assume that the training, validation, and test datasets used to develop said algorithm are drawn from the same distribution which will be seen once deployed. A primary concern for early adopters of RFML is how the algorithm will behave when this assumption is invalid, either due to the real-world considerations not present in the training data or adversarial attack as discussed in Sections IV, V, and VII.

In other words, we can categorize model inputs in one of three ways, illustrated in Figure 5:

- *in-set* - Those that match the distribution of the training data. Using modulation classification as an example, an in-set input is a known modulation scheme under the same channel effects, SNR, transmitter/receiver imperfections, etc. that were seen during training.
- *near-set* - Those close to the distribution of the training dataset, but not included. In our modulation classification example, near-set inputs might be a trained modulation scheme, but different channel effects or SNR. Near-set inputs may also include in-set examples which have been perturbed by an adversary using the techniques discussed in Section V.
- *out-of-set* - Those completely outside of the distribution of the training data. Completing our example, an out-of-set example would be an untrained modulation scheme.

Data-driven RFML systems will behave as expected on *in-set* inputs, but are unpredictable given *near-set* input values, and necessarily incorrect in processing all *out-of-set* input values. This points to a critical need for approaches to both rigorously assure "the safety and functional correctness" of RFML systems throughout deployment and explain or understand the behavior of RFML systems [123], [124].

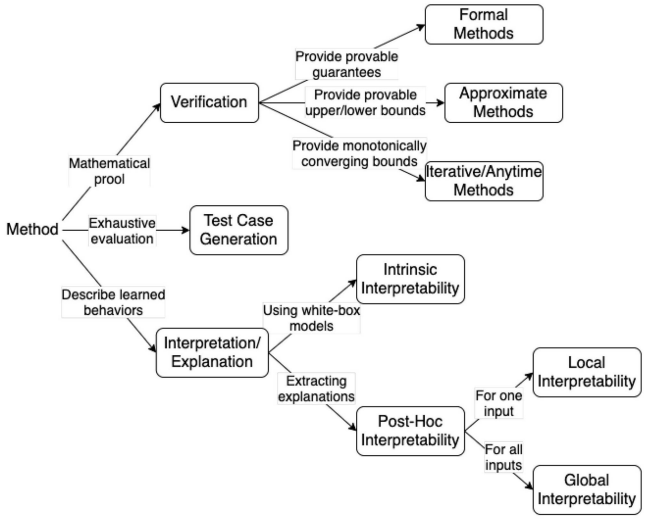

FIGURE 6. Overview of trust and assurance methods used in the literature, but yet to be fully explored in the context of RFML.

These concerns are not unique to RFML [125], but have yet to be adequately addressed in RFML literature. Therefore, this section will focus on the very young body of work in testing, verification, and interpretation of general ML systems which could be explored for use in the RF domain, and will discuss the pros and cons of the various approaches. These works can broadly be categorized into three research areas which are also shown in Figure 6:

- *verification* methods which provide mathematical proof that a desired property holds for a trained model,
- *testing* methods which aim to exhaustively evaluate a trained model to identify flaws, and
- *interpretation/explanation* techniques which include methods to describe and/or quantify a trained model's learned behaviors in a human-understandable format, such as decision/model explanations or uncertainty/reliability metrics.

A. VERIFICATION

Beginning with the most rigorous approaches, current methods for verifying ML algorithms apply formal methods such as constraint solving [126]–[128], global optimization [129], search-based methods [130], and over approximation [131] to provide provable guarantees about their behavior when provided with in-set, near-set, out-of-set, and even adversarial examples. While verification methods provide deterministic or statistical guarantees of the robustness of previously trained models, they are also typically NP (non-deterministic polynomial time) hard or extremely computationally expensive and time intensive. As a result, none have been able to successfully scale to the state-of-the-art DNNs used today.

Towards scalable DL verification methods, a promising path forward is that of approximate or iterative/anytime methods which provide provable upper/lower bounds or monotonically converging bounds [128]–[131]. However, future work is needed to improve these methods, in order to yield tighter and more useful bounds on the robustness of trained models.

B. TESTING

Traditionally, DL researchers and engineers rely on a held-out test set, which remains unseen throughout the training and model selection process, to provide an estimate of a trained model's performance [132]. This computationally efficient method provides a good estimate of how the model will perform on in-set data, but fails to identify how the model will perform on near-set or out-of-set data.

In an effort to strike a balance between computational efficiency and rigorousness, there is a growing body of work adapting and applying software testing and debugging techniques to more thoroughly test ML and DL algorithms. These approaches generate test cases using methods such as concolic testing [133], [134], mutation testing [135], differential analysis [136], or even adversarial methods [99], which is typically guided by a user-selected coverage metric. Some of the most popular coverage metrics used have included neuron or layer coverage [137], [138] and modified condition/decision coverage [133]. The aim is to generate a set of test cases/inputs which provide sufficient coverage of the trained model, dictated by a user-selected threshold.

Though test case generation may provide more assurance than traditional ML and DL testing practices and are typically more computationally efficient than verification methods, there are a number of drawbacks which should be addressed. First and foremost, like traditional software testing methods, ML testing methods can only identify a lack of robustness, and cannot ensure robustness. In the same vein, the effectiveness of the testing method is highly dependent upon the coverage metrics and thresholds used, both of which are chosen by the user. With some coverage metrics and thresholds, testing methods may be just as computationally expensive and time consuming in comparison to approximate verification methods. Ultimately, while there is certainly value in more effective testing techniques, future work will likely need to focus on RFML verification over RFML testing, in order to effectively mitigate against adversarial attack and provide assured performance [139].

C. INTERPRETATION/EXPLANATION

In contrast, the aim of interpretation/explanation methods is to address the challenge of Human-Machine Interaction (HMI) by "enabl[ing] users to understand how the data is processed and supports awareness of possible bias and systems malfunctions" [140]. In other words, HMI becomes more feasible if the model/decision is better understood by the end user. Approaches to interpret or explain black-box ML models such as deep NNs and/or their decisions can broadly be categorized into two groups.

The first group of approaches provide *intrinsic interpretability* by using inherently more interpretable models either from the offset or extracted from a black box

model [141]. Examples of such models include decision trees [25], [142], attention mechanisms [143], clustering algorithms, or linear/Bayesian classifiers [144]. While these methods are typically the most straightforward and provide the most useful model/decision explanations, inherently interpretable models are typically less expressive than black-box models such as deep NNs, and therefore do not provide the same level of performance.

The second group of approaches provide *post-hoc interpretability* by extracting decision/model explanations from black-box models or through model exploration [140], [141]. Post-hoc interpretability methods can be further broken down into *local interpretability* methods and *global interpretability* methods. Local interpretability methods aim to provide an explanation for why and/or how a black box model made the decision it made for a given example input. These instance-level explanations can be aggregated over a group of example inputs to draw larger conclusions about a model's knowledge. Meanwhile global interpretability methods focus on increasing the transparency of black-box models by "inspecting the structures and parameters" in an effort to understand the scope of the model's knowledge more directly [141].

Local interpretability methods typically utilize some form of visualization to describe the network's response to the input such as heatmaps, which indicate which portions of the example input contributed most to the network's decision [145], [146]. Popular and successful local interpretability methods in the image processing domain include backpropagation techniques such as layerwise relevance propagation, Taylor decomposition, and GradCAM [145], [147], [148], saliency mapping [146], and deconvolutional networks [149]. However, transitioning these methods to the RF domain has proven challenging, as raw RF data is more difficult to visualize, especially in the intermediate layers of a DNN. Therefore, a more promising local interpretability method for use in the RF domain is the use of uncertainty metrics to accurately quantify a model's confidence in any given decision, and could be used to identify unpredictability due to adversarial attack or operating environments [150], [151].

Global interpretability methods focus less on visualization techniques due to the large number of parameters in DNN models, but have been explored through approaches such as activation maximization and partial dependence [146], [152]–[154]. More common is the use of metrics such as feature importance [155], [156], sensitivity [157], [158], and mutual information [159].

The primary challenge shared amongst both local and global interpretability methods is that there are no universal definitions for terms such as trust, interpretability, assurance, and explanation in the deep learning literature. Furthermore, the concept of interpretability is highly dependent on the end user and their technical background [160]. For example, some argue that while global interpretability methods are useful to the DL expert who understands the inner-workings of a black-box model, local interpretability methods are more tangible, intuitive, and provide more benefit to the end user. Furthermore, trust, interpretability, assurance, and explanation are largely gauged qualitatively rather than quantitatively, and therefore are hard to compare and evaluate across approaches [140].

Additional challenges to DL interpretations include, but are not limited to [125].

- How to accurately characterize and/or classify out-of-set examples. This is one area where uncertainty metrics would likely be more useful than visualization based explanation methods
- Producing consistent explanations for similar inputs
- Producing explanations without significant computational overhead
- DNN produce an overwhelming amount of highly complex and interdependent data that is difficult to visualize, describe, and/or explain in a helpful manner. The abstract nature of RF data only exacerbates this challenge.

D. DISCUSSION

Trust/assurance in RFML systems will likely require some form of both verification method in conjunction with interpretation/explanation methods [161], in order to provide designer, administrator, operator, and end-user confidence in a model's decision-making capabilities both before and during deployment. As discussed above, interpretation/explanation methods provide the user with an intuitive and/or quantifiable level of confidence in a model's decision, improving their understanding of and trust in the system. While this understanding and trust is critical to HMI, assured RFML suitable for use in safety-critical systems, such as self-driving cars and military systems, will require the rigorous guarantees that verification provides. Furthermore, verification methods can ensure that these safety-critical systems are robust to the increasingly sophisticated adversarial attacks discussed in Section V.

VII. OPERATIONAL CONSIDERATIONS

Though early adoption of RFML systems has already taken place in a variety of military systems [2], [52], [162], [163], a broader interest is expected in the roll-out of commercial cellular [164]–[166], IoT [41], [167]–[169], and satellite communications systems [170]–[172]. While the prior section addressed concerns of providing user-defined mission assurance, this section evaluates the practical size, weight, and power (SWaP) constraints encountered in the transition to real systems, highlighting that the requisite hardware and algorithmic technologies for RFML deployment are well under way. More specifically, given the low processing and storage requirements for RFML algorithms, compared to CV algorithms, and current availability of RF sensors on board low-SWaP mobile devices such as cell phones, the barriers to entry for deployed RFML algorithms are primarily the cost of training data, decision-making infrastructure, and trust/assurance (discussed previously in Section VI).

A. SIZE, WEIGHT, AND POWER (SWaP)

Many DL techniques employ significant computing infrastructures during their training phases which makes training in the field infeasible [173]. When considering deployment, we are most concerned with a DL algorithm's computational requirements post-training, when attempting to process incoming data inputs. Current state-of-the-art RFML techniques often utilize NNs significantly smaller than CV techniques, with 2-3 orders of magnitude fewer trainable parameters, boding well for deployment on low SWaP devices. Further, RF sample frames occur on the order of 1 kHz compared to image inputs that might occur on the order of 1 Hz in inexpensive commercial devices. Therefore, the evaluation time of a NN processing raw IQ data must meet more stringent real-time requirements than a NN processing images.

In an effort to further reduce processing requirements, some RFML implementations have also embedded traditional signal processing techniques such as Fourier and wavelet transforms, cyclostationary feature estimators, and other expert features directly into the NN [170], [174], [175]. Meanwhile, other research has focused on reduced precision implementations of NNs, enabling a path towards real-time implementation [176]–[178]. However, reducing real-time computational resources to mobile systems remains a challenge that must be overcome, especially if online learning techniques are to be developed for future RFML systems [179], [180].

Given the highly effective miniaturization of digital electronics, a deployed system's weight is primarily driven by it's power consumption and the associated batteries or heatsinks [181]. In a spectrum situational awareness system, the instantaneous bandwidth of the spectrum analyzed, the density of signals within the environment (affecting the number of calls to an RFML algorithm), implementation in hardware vs. software, and the environment where the device is used will all contribute to the system power usage. Real-time signal detection [182], signal characterization [52], and SEI [41] systems have already been achieved, either through the assumption of vehicle power or a tightly regulated and small duty cycle, showing the feasibility of using these algorithms in current mobile systems. Further, the use of wake-up circuits for periodic/event-triggered execution of an RFML function can be used to further reduce average power draw, permitting the use of such techniques in extreme low-power applications such as the IoT [41]. Finally, the integration of RFML processing with energy harvesting techniques are of particular interest for battery-powered IoT and solar-powered satellites, but have yet to be investigated.

B. COST

Beyond SWaP, cost is typically considered the next most important operational consideration. Because the quality of the training data drives the overall functionality of an RFML system and often requires human-intensive labeling and/or pre-processing [2], as discussed previously in Section IV, the primary cost drivers of current RFML systems are the curation of datasets used for training/evaluation/testing, the training hardware, and the RF hardware to be deployed.

The cost of the training process itself is in part driven by power consumption [183], and in part driven by the purchase of parallelized processors such as Graphical Processing Units (GPUs), Tensor Processing Units (TPUs), or other special purpose hardware. While the purchase of specialized hardware is typically a one-time expense, current training approaches for most RFML algorithms require complete retraining of the underlying model when new training data is added, as online, unsupervised, semi-supervised, and transfer learning techniques have yet to successfully be employed. As a result, power requirements for maintaining RFML models can be high. Improvements in online, incremental, and transfer learning approaches are necessary, not only to learn behaviors associated with new signals or environmental changes, but to minimize re-training to when performance degrades.

The cost of the RF hardware is dependent upon the quality, and will impact the performance of the RFML algorithm and resultant learned behaviors, as discussed in Section IV. For example, SEI algorithms are better at differentiating between low-cost sensors, such as those used in the IoT, than between high-cost sensors, due to more significant variations during manufacturing. Conversely, the peak accuracy of an AMC algorithm will decrease, as the model is forced to generalize it's learned behavior across the imperfections present in low-cost hardware [24]. Such phenomenon are not confined to the transmit side of the RF chain, and can be induced low-quality receiver hardware as well. Therefore, RFML system design must consider impact of the cost of the RF hardware on each side of the communications link in the creation and/or expansion of the training datasets and in the expected performance of the RFML system.

C. APPLICATION DEPENDENCIES

As alluded to in Sections III and IV, the scale and scope of different applications can lead to vastly different hardware and SWaP requirements. For example, a Raspberry Pi 0 has been shown to be suitable for performing event-triggered packet-based SEI for IoT networks [41], but much larger systems are needed to realize real-time 5 GHz instantaneous continuous spectrum monitoring systems [2]. For most RFML applications, decisions must be made locally due to bandwidth and time constraints [184]. As a result, environmental effects on the hardware must also be considered, in addition to the SWaP requirements required to execute RFML algorithms on varying devices. For example, when deploying RFML algorithms aboard small spacecraft which are impacted by radiation-induced single event upsets [185], [186], without the addition of radiation shielding and/or extensive mitigation strategies, the performance of the ML structures fail to achieve the necessary performance to be practically useful [187]–[192].

Broader dependencies include harnessing the more rapid decision making of RFML. More specifically, many of the applications discussed in Section III cite rapid decision making as a benefit of using a DL-based approach over traditional approaches. However, additional work is required to make the outputs of such RFML systems fully actionable.

D. DISCUSSION

While RFML algorithms have already begun to make their way onto deployed military systems, it is expected that RFML will become a vital component of future commercial cellular, IoT, and satellite communications systems in the near future. The scale and scope of the different RFML applications to be deployed will lead to different hardware, SWaP, and bandwidth requirements which will need to be considered. However, the discussion above highlights the feasibility of using current state-of-the-art RFML techniques on even low SWaP-C devices, given the use of significantly smaller models and the speed with which raw RF data can be collected and batched for processing.

The quality of these potential deployed RFML systems is largely dictated by the quality of the training data in how closely it resembles the data observed during deployment. Therefore, the collection and labeling of training data is also one of the largest cost drivers, in addition to the training hardware (i.e., GPUs and TPUs), power consumed during training process, and RF hardware. However, the development of online, unsupervised, and/or semi-supervised learning techniques will mitigate these costs to some extent by limiting the amount of model re-training that must occur when hardware or environments change.

VIII. CHALLENGES AND FUTURE WORK

RFML is a rapidly growing area of research in DL technologies, and has demonstrated particular success in improving and automating spectrum situational awareness applications, as well as supporting the next-generation of CR and cellular communications applications. However, while research into the application of DL technologies to RFML is accelerating, there is still a lack of works which holistically look at all of the considerations for making these systems deployable in real-world applications. As a result, RFML lags significantly behind more mature deep learning technologies such as CV and NLP.

This paper has provided a summary of a so-called RFML "Ecosystem" of research concerns that need to be addressed before RFML can be considered sufficiently mature for widespread deployment in commercial and military applications. In particular, this paper has addressed the fundamental concerns of dataset creation (Section IV), security (Section V), trust and assurance (Section VI), and real-world operations (Section VII). For each element of the ecosystem, an overview of the topic was provided, the primary research areas were identified with examples of existing works, and directions for future research were discussed.

As made clear by significant overlap in discussion between the sections above, RFML-based systems must holistically consider the different aspects of the described "ecosystem" in order to successfully be deployed in real-world systems, with several challenges and areas for future research remaining under-developed. Each of the following challenges and areas for future research were highlighted in at least one of the previous sections, but in many cases, span multiple of the major research thrusts identified as part of the *RFML ecosystem* presented in the introduction to this work and are amongst the most salient needs that must be addressed. Namely, the following subsections discuss the need for online and transfer learning techniques, robust confidence metrics in RFML-derived decisions to improve human-machine interaction and trust, and real-time processing improvements, based on the collective lessons learned to date.

A. ONLINE AND TRANSFER LEARNING TECHNIQUES

Current RFML systems predominately utilize supervised learning solutions in which the training process is performed offline, before deployment, and the learned model remains fixed during deployment. The inflexibility of these systems means that, while they are appropriate for the conditions assumed during offline training, they are largely not adaptable to changes in the propagation environment and transmitter/receiver hardware. Given the fluidity of modern communication environments, this rigidness greatly limits widespread adoption of RFML solutions. Additionally, many RF systems offer the potential for multiple apertures/nodes whose spectrum observations can be integrated to gain a larger system picture.

As previously discussed in both Sections IV and VII, research and development is needed to allow for online learning and transferring learned behaviors between platforms. However, such solutions must consider that the behaviors learned at one node will be influenced by their RF hardware, which is distinct and possibly vastly different from a second node. Therefore, any behaviors learned in one environment may be distinct from another. As a result, any use of online, incremental, and/or transfer learning techniques also poses the risk that any learned or transferred behaviors may misrepresent or bias the outcomes at each node and must be intelligently handled.

B. HUMAN-MACHINE INTERACTION AND END-USER CONFIDENCE

While DL technologies have shown the ability to solve complex and hard to model problems, both within RFML and other application spaces, the black-box nature of their decision making process hampers their widespread adoption. In particular, while DL systems can provide decisions to the user, they typically do not provide a good justification or confidence in their decision to the end-user, limiting the utility of the system outputs, as touched on in Section VII.

Beyond trusting individual decisions, additional work is also needed to help the end-user understand the limits of the

learned behaviors, how to shape and/or optimize the system, and how to visualize and/or verify whether the machine should be trusted, a topic further explored in Section VI. In the same vein, additional work is required to identify in real-time if the current inputs are representative of the training data, using methods for identifying covariance shift or out-of-distribution examples. Such methods are not only needed in order to provide assured performance, but to begin ruggedizing the decision chain against spoofing and other adversarial techniques, as discussed in more detail in Section V.

C. REAL-TIME PROCESSING CAPABILITIES

The widespread availability and adoption of GPUs have vastly accelerated the research and deployment of DL-based image processing applications. While GPUs have certainly accelerated RFML-based applications as well, the sequential time-series nature of RF data may require novel hardware processing architectures to facilitate further acceleration of these data types. In particular, recent research has demonstrated the applicability of FPGA-based implementations that greatly accelerate sequential data streams and may prove fruitful for real-time RFML processing [193]. The ability to process RF data and make decisions on a sample-by-sample basis allows for quicker, more agile, decision making which is incredibly important for the RFML application spaces considered in this work [194], as previously discussed in Section VII.

REFERENCES

[1] M. E. Morocho-Cayamcela, H. Lee, and W. Lim, "Machine learning for 5G/B5G mobile and wireless communications: Potential, limitations, and future directions," *IEEE Access*, vol. 7, pp. 137184–137206, 2019.

[2] T. Rondeau. *Radio Frequency Machine Learning Systems (RFMLS)*. Accessed: Aug. 2017. [Online]. Available: https://www.darpa.mil/program/radio-frequency-machine-learning-systems

[3] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 4th Quart., 2018.

[4] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 4th Quart., 2019.

[5] M. A. Alsheikh, S. Lin, D. Niyato, and H. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1996–2018, 4th Quart., 2014.

[6] M. Di and E. M. Joo, "A survey of machine learning in wireless sensor networks from networking and application perspectives," in *Proc. Int. Conf. Inf. Commun. Signal Process.*, Dec. 2007, pp. 1–5.

[7] R. Boutaba *et al.*, "A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities," *J. Internet Services Appl.*, vol. 9, no. 1, p. 16, 2018.

[8] J. Xie *et al.*, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 393–430, 1st Quart., 2019.

[9] H. K. Jhajj, R. Garg, and N. Saluja, "Aspects of machine learning in cognitive radio networks," in *Progress in Advanced Computing and Intelligent Engineering*, K. Saeed, N. Chaki, B. Pati, S. Bakshi, and D. P. Mohapatra, Eds. Singapore: Springer, 2018, pp. 553–559.

[10] M. Alshawaqfeh, X. Wang, A. R. Ekti, M. Z. Shakir, K. Qaraqe, and E. Serpedin, "A survey of machine learning algorithms and their applications in cognitive radio," in *Proc. Int. Conf. Cogn. Radio Oriented Wireless Netw.*, 2015, pp. 790–801.

[11] C. Clancy, J. Hecker, E. Stuntebeck, and T. O'Shea, "Applications of machine learning to cognitive radio networks," *IEEE Wireless Commun.*, vol. 14, no. 4, pp. 47–52, Aug. 2007.

[12] G. P. Joshi, S. Y. Nam, and S. W. Kim, "Cognitive radio wireless sensor networks: Applications, challenges and research trends," *Sensors*, vol. 13, no. 9, pp. 11196–11228, 2013.

[13] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1136–1159, 3rd Quart., 2013.

[14] M. A. Hossain, R. M. Noor, K. A. Yau, S. R. Azzuhri, M. R. Z'aba, and I. Ahmedy, "Comprehensive survey of machine learning approaches in cognitive radio-based vehicular ad hoc networks," *IEEE Access*, vol. 8, pp. 78054–78108, 2020.

[15] T. Erpek, T. J. O'Shea, Y. E. Sagduyu, Y. Shi, and T. C. Clancy, *Deep Learning for Wireless Communications*. Cham, Switzerland: Springer, 2020, pp. 223–266. [Online]. Available: https://doi.org/10.1007/978-3-030-31764-5_9

[16] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[17] O. Simeone, "A very brief introduction to machine learning with applications to communication systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 648–664, Dec. 2018.

[18] T. J. O'Shea and N. West, "Radio machine learning dataset generation with GNU radio," in *Proc. GNU Radio Conf.*, vol. 1, 2016, pp. 1–6.

[19] J. Sen, "Security and privacy challenges in cognitive wireless sensor networks," in *Cognitive Radio Technology Applications for Wireless and Mobile Ad hoc Networks*. Hershey, PA, USA: IGI Global, 2013, pp. 194–232.

[20] F. Humaira, M. S. Islam, F. N. Nur, and K. A. Hussain, "A comprehensive study on machine learning algorithms for wireless sensor network security," in *Proc. Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, 2020, pp. 1–6.

[21] T. Tuukkanen and J. Anteroinen, "Framework to develop military operational understanding of cognitive radio," in *Proc. Int. Conf. Military Commun. Inf. Syst. (ICMCIS)*, May 2015, pp. 1–9.

[22] S. K. Sharma, T. E. Bogale, S. Chatzinotas, B. Ottersten, L. B. Le, and X. Wang, "Cognitive radio techniques under practical imperfections: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 1858–1884, 4th Quart., 2015.

[23] 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.

[24] S. C. Hauser, W. C. Headley, and A. J. Michaels, "Signal detection effects on deep neural networks utilizing raw IQ for modulation classification," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Oct. 2017, pp. 121–127.

[25] W. H. Clark *et al.*, "Developing RFML intuition: An automatic modulation classification architecture case study," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Oct. 2019, pp. 292–298.

[26] L. J. Wong, P. D. White, W. C. Headley, and A. J. Michaels, "Distributed automatic modulation classification with compressed data," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Oct. 2019, pp. 299–304.

[27] Y. Wu, X. Li, and J. Fang, "A deep learning approach for modulation recognition via exploiting temporal correlations," in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2018, pp. 1–5.

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

[29] A. Vila *et al.*, "Deep and ensemble learning to win the Army RCO AI signal classification challenge," in *Proc. 18th Python Sci. Conf.*, 2019, pp. 21–26.

[30] P. Wang and M. Vindiola, "Data augmentation for blind signal classification," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, 2019, pp. 149–154.

[31] T. J. O'Shea, T. Roy, and T. Erpek, "Spectral detection and localization of radio events with learned convolutional neural features," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2017, pp. 331–335.

[32] T. O'Shea, T. Roy, and T. C. Clancy, "Learning robust general radio signal detection using computer vision methods," in *Proc. Asilomar Conf. Signals Syst. Comput.*, Oct. 2017, pp. 829–832.

[33] J. Gao, X. Yi, C. Zhong, X. Chen, and Z. Zhang, "Deep learning for spectrum sensing," *IEEE Wireless Commun. Lett.*, vol. 8, no. 6, pp. 1727–1730, Dec. 2019.

[34] P. D. White, R. M. Buehrer, and W. C. Headley, "FHSS signal separation using constrained clustering," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, 2019, pp. 159–164.

[35] Q. Peng, A. Gilman, N. Vasconcelos, P. C. Cosman, and L. B. Milstein, "Robust deep sensing through transfer learning in cognitive radio," *IEEE Wireless Commun. Lett.*, vol. 9, no. 1, pp. 38–41, Jan. 2020.

[36] Z. Ye, A. Gilman, Q. Peng, K. Levick, P. Cosman, and L. Milstein, "Comparison of neural network architectures for spectrum sensing," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6.

[37] Q. Cheng, Z. Shi, D. N. Nguyen, and E. Dutkiewicz, "Non-cooperative OFDM spectrum sensing using deep learning," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, Feb. 2020, pp. 704–708.

[38] X. Meng, H. Inaltekin, and B. Krongold, "End-to-end deep learning-based compressive spectrum sensing in cognitive radio networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[39] L. J. Wong, W. C. Headley, S. Andrews, R. M. Gerdes, and A. J. Michaels, "Clustering learned CNN features from raw I/Q data for emitter identification," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Oct. 2018, pp. 26–33.

[40] L. J. Wong, W. C. Headley, and A. J. Michaels, "Specific emitter identification using convolutional neural network-based IQ imbalance estimators," *IEEE Access*, vol. 7, pp. 33544–33555, 2019.

[41] J. M. McGinthy, L. J. Wong, and A. J. Michaels, "Groundwork for neural network-based specific emitter identification authentication for IoT," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6429–6440, Aug. 2019.

[42] T. J. O'Shea, T. Roy, N. West, and B. C. Hilburn, "Physical layer communications system design over-the-air using adversarial networks," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, 2018, pp. 529–532.

[43] T. J. O'Shea, T. Roy, and N. West, "Approximating the void: Learning stochastic channel models from observation with variational generative adversarial networks," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, Feb. 2019, pp. 681–686.

[44] K. Davaslioglu and Y. E. Sagduyu, "Generative adversarial learning for spectrum sensing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–6.

[45] A. Zubow, S. Bayhan, P. Gawłowicz, and F. Dressler, "DeepTxFinder: Multiple transmitter localization by deep learning in crowdsourced spectrum sensing," in *Proc. 29th Int. Conf. Comput.Commun. Netw. (ICCCN)*, Aug. 2020, pp. 1–8.

[46] J. Yu, H. M. Saad, and R. M. Buehrer, "Centimeter-level indoor localization using channel state information with recurrent neural networks," in *Proc. IEEE/ION Position Location Navigation Symp. (PLANS)*, 2020, pp. 1317–1323.

[47] R. Elbakly, H. Aly, and M. Youssef, "TrueStory: Accurate and robust RF-based floor estimation for challenging indoor environments," *IEEE Sensors J.*, vol. 18, no. 24, pp. 10115–10124, Dec. 2018.

[48] M. I. AlHajri, N. T. Ali, and R. M. Shubair, "Indoor localization for IoT using adaptive feature selection: A cascaded machine learning approach," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2306–2310, Nov. 2019.

[49] N. Tandiya, A. Jauhar, V. Marojevic, and J. H. Reed, "Deep predictive coding neural network for RF anomaly detection in wireless networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2018, pp. 1–6.

[50] T. J. O'Shea, T. C. Clancy, and R. W. McGwier, "Recurrent neural radio anomaly detection," 2016. [Online]. Available: arXiv:1611.00301.

[51] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: Classical approaches and new trends," *IET Commun.*, vol. 1, no. 2, pp. 137–156, 2007.

[52] *SignalEye AI Software for Automated Signal Classification—General Dynamics*. Accessed: Feb. 2019. [Online]. Available: https://gdmissionsystems.com/products/electronic-warfare/signaleye

[53] K. Chowdhury, S. Ioannidis, and T. Melodia, "Deep learning for RF signal classification and fingerprinting," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, 2019.

[54] A. C. Polak, S. Dolatshahi, and D. L. Goeckel, "Identifying wireless users via transmitter imperfections," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 7, pp. 1469–1479, Aug. 2011.

[55] P. Tilghman, "Will rule the airwaves: A DARPA grand challenge seeks autonomous radios to manage the wireless spectrum," *IEEE Spectr.*, vol. 56, no. 6, pp. 28–33, Jun. 2019.

[56] A. Bacak and H. Çelebi, "Practical considerations for RSS RF fingerprinting based indoor localization systems," in *Proc. Signal Process. Commun. Appl. Conf. (SIU)*, Apr. 2014, pp. 497–500.

[57] W. H. Clark, S. Hauser, W. C. Headley, and A. J. Michaels, "Training data augmentation for deep learning radio frequency systems," *J. Defense Model. Simulat.*, vol. 18, no. 3, pp. 217–237, 2021. [Online]. Available: https://doi.org/10.1177/1548512921991245

[58] B. Clark. (Sep. 2016). *Efficient Waveform Spectrum Aggregation for Algorithm Verification and Validation*. [Online]. Available: https://gnuradio.org/grcon-2016/talks/

[59] J. Gaeddert. *Liquid-dsp*. Accessed: Sep. 2019. [Online]. Available: https://github.com/jgaeddert/liquid-dsp

[60] *MATLAB*, MathWorks, Natick, MA, USA, 2021.

[61] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3234–3243.

[62] A. N. Mody *et al.*, "Recent advances in cognitive communications," *IEEE Commun. Mag.*, vol. 45, no. 10, pp. 54–61, Oct. 2007.

[63] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep learning for RF device fingerprinting in cognitive communication networks," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 160–167, Feb. 2018.

[64] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "ORACLE: Optimized radio classification through convolutional neural networks," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 370–378.

[65] D. Adesina, J. Bassey, and L. Qian, "Practical radio frequency learning for future wireless communication systems," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Nov. 2019, pp. 311–317.

[66] D. M. Le Vine. (Sep. 2013). *Review of Measurements of the RF Spectrum of Radiation From Lightning*. [Online]. Available: https://ntrs.nasa.gov/citations/19870001225

[67] L. H. Pederick and M. A. Cervera, "Modeling the interference environment in the HF band," *Radio Sci.*, vol. 51, no. 2, pp. 82–90, 2016. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2015RS005856

[68] B. Hilburn, N. West, T. O'Shea, and T. Roy, "SigMF: The signal metadata format," in *Proc. GNU Radio Conf.*, vol. 3, 2018.

[69] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[70] J. Torres-Sospedra. (May 24, 2020). *UJIIndoorLoc Database*. [Online]. Available: http://geotec.uji.es/2014/10/03/ujiindoorloc-database/

[71] Deepsig. (May 21, 2020). *RF Datasets for Machine Learning*. [Online]. Available: https://www.deepsig.io/datasets

[72] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury. (May 21, 2020). *Datasets for RF Fingerprinting of Bit-Similar USRP X310 Radios*. [Online]. Available: http://www.genesys-lab.org/oracle

[73] (2021). *Machine Learning For Communications Emerging Technologies Initiative*. [Online]. Available: https://mlc.committees.comsoc.org/datasets/

[74] J. Torres-Sospedra *et al.*, "UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems," in *Proc. Int. Conf. Indoor Positioning Indoor Navigation (IPIN)*, 2014, pp. 261–270.

[75] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[76] Y. Qin, N. Carlini, I. Goodfellow, G. Cottrell, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," 2019. [Online]. Available: arXiv:1903.10346.

[77] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *Proc. IEEE Security Privacy Workshops (SPW)*, May 2018, pp. 1–7.

[78] R. Taori, A. Kamsetty, B. Chu, and N. Vemuri, "Targeted adversarial examples for black box audio systems," in *Proc. IEEE Security Privacy Workshops (SPW)*, May 2019, pp. 15–20.

[79] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, "Adversarial attacks on deep-learning models in natural language processing: A survey," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 3, p. 24, Apr. 2020. [Online]. Available: https://doi.org/10.1145/3374217

[80] K. Davaslioglu and Y. E. Sagduyu, "Trojan attacks on wireless signal classification with adversarial machine learning," in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw. (DySPAN)*, Nov. 2019, pp. 1–6.

[81] M. Sadeghi and E. G. Larsson, "Adversarial attacks on deep-learning based radio signal classification," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 213–216, Feb. 2019.

[82] S. Bair, M. Delvecchio, B. Flowers, A. J. Michaels, and W. C. Headley, "On the limitations of targeted adversarial evasion attacks against deep learning enabled modulation recognition," in *Proc. ACM Workshop Wireless Security Mach. Learn. (WiseML)*, May 2019, pp. 25–30.

[83] S. Kokalj-Filipovic, R. Miller, and J. Morman, "Targeted adversarial examples against RF deep classifiers," in *Proc. ACM Workshop Wireless Security Mach. Learn.*, 2019, pp. 6–11. [Online]. Available: https://doi.org/10.1145/3324921.3328792

[84] B. Flowers, R. M. Buehrer, and W. C. Headley, "Communications aware adversarial residual networks for over the air evasion attacks," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Nov. 2019, pp. 133–140.

[85] B. Flowers, R. M. Buehrer, and W. C. Headley, "Evaluating adversarial evasion attacks in the context of wireless communications," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1102–1113, 2020.

[86] M. Z. Hameed, A. Gyorgy, and D. Gunduz, "Communication without interception: Defense against deep-learning-based modulation detection," 2019. [Online]. Available: arXiv:1902.10674.

[87] M. Sadeghi and E. G. Larsson, "Physical adversarial attacks against end-to-end autoencoder communication systems," 2019. [Online]. Available: http://arxiv.org/abs/1902.08391.

[88] M. Usama, M. Asim, J. Qadir, A. Al-Fuqaha, and M. A. Imran, "Adversarial machine learning attack on modulation classification," in *Proc. U.K./China Emerg. Technol. (UCET)*, Aug. 2019, pp. 1–4.

[89] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Security Privacy (EuroSP)*, 2016, pp. 372–387.

[90] E. Tabassi, K. J. Burns, M. Hadjimichael, A. D. Molina-Markham, and J. T. Sexton, "A taxonomy and terminology of adversarial machine learning," Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Rep. NISTIR 8269, 2019.

[91] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," 2018. [Online]. Available: arXiv:1810.00069.

[92] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proc. 4th ACM Workshop Security Artif. Intell.*, 2011, pp. 43–58. [Online]. Available: http://doi.acm.org/10.1145/2046684.2046692

[93] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proc. ACM Symp. Inf. Comput. Commun. Security*, 2006, pp. 16–25.

[94] M. Barreno, B. Nelson, A. D. Joseph, and J. Tygar, "The security of machine learning," *Mach. Learn.*, vol. 81, no. 2, pp. 121–148, 2010.

[95] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognit.*, vol. 84, pp. 317–331, Dec. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320318302565

[96] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIs," in *Proc. 25th USENIX Conf. Security Symp.*, 2016, pp. 601–618. [Online]. Available: http://dl.acm.org/citation.cfm?id=3241094.3241142

[97] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Security Privacy (SP)*, 2017, pp. 3–18.

[98] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015. [Online]. Available: http://arxiv.org/abs/1412.6572.

[99] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Security Privacy (SP)*, 2017, pp. 39–57.

[100] Y. Dong *et al.*, "Boosting adversarial attacks with momentum," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9185–9193.

[101] Y. Shi, K. Davaslioglu, and Y. E. Sagduyu, "Generative adversarial network for wireless signal spoofing," in *Proc. ACM Workshop Wireless Security Mach. Learn.*, 2019, pp. 55–60. [Online]. Available: https://doi.org/10.1145/3324921.3329695

[102] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 427–436.

[103] Z. Akhtar and D. Dasgupta. (2019). *A Brief Survey of Adversarial Machine Learning and Defense Strategies*. [Online]. Available: https://www.memphis.edu/cs/research/tech_reports/tr-cs-19-002.pdf

[104] S. Kokalj-Filipovic, R. Miller, and G. Vanhoy, "Adversarial examples in RF deep learning: Detection and physical robustness," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2019, pp. 1–5.

[105] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. 10th ACM Workshop Artif. Intell. Security*, 2017, pp. 3–14. [Online]. Available: https://doi.org/10.1145/3128572.3140444

[106] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," 2016. [Online]. Available: http://arxiv.org/abs/1611.01236.

[107] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017. [Online]. Available: arXiv:1706.06083.

[108] A. Shafahi *et al.*, "Adversarial training for free!" 2019. [Online]. Available: http://arxiv.org/abs/1904.12843.

[109] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. D. McDaniel, "Ensemble adversarial training: Attacks and defenses," 2017. [Online]. Available: arXiv:1705.07204.

[110] S. Kokalj-Filipovic, R. Miller, N. Chang, and C. L. Lau, "Mitigation of adversarial examples in RF deep classifiers utilizing autoencoder pre-training," in *Proc. Int. Conf. Military Commun. Inf. Syst. (ICMCIS)*, 2019, pp. 1–6.

[111] N. Carlini *et al.*, "On evaluating adversarial robustness," 2019. [Online]. Available: arXiv:1902.06705.

[112] N. Papernot *et al.*, "Cleverhans v2.0.0: An adversarial machine learning library," 2016. [Online]. Available: arXiv:1610.00768.

[113] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2016. [Online]. Available: http://arxiv.org/abs/1607.02533.

[114] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "Over-the-air adversarial attacks on deep learning based modulation classifier over wireless channels," 2020. [Online]. Available: arXiv:2002.02400.

[115] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," 2018. [Online]. Available: https://arxiv.org/abs/1802.08232.

[116] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot, "High accuracy and high fidelity extraction of neural networks," in *Proc. 29th USENIX Security Symp. (USENIX Security)*, 2020, pp. 1345–1362.

[117] T. Erpek, Y. E. Sagduyu, and Y. Shi, "Deep learning for launching and mitigating wireless jamming attacks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 1, pp. 2–14, Mar. 2019.

[118] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "Channel-aware adversarial attacks against deep learning-based wireless signal classifiers," 2020. [Online]. Available: arXiv:2005.05321.

[119] M. DelVecchio, B. Flowers, and W. C. Headley, "Effects of forward error correction on communications aware evasion attacks," 2020. [Online]. Available: arXiv:2005.13123.

[120] F. Restuccia *et al.*, "Hacking the waveform: Generalized wireless adversarial deep learning," 2020. [Online]. Available: arXiv:2005.02270.

[121] Y. Shi, T. Erpek, Y. E. Sagduyu, and J. H. Li, "Spectrum data poisoning with adversarial deep learning," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, 2018, pp. 407–412.

[122] E. B. Kania, *'AI Weapons' in China's Military Innovation*, Brookings Inst., Washington, DC, USA, 2020. [Online]. Available: https://www.brookings.edu/research/ai-weapons-in-chinas-military-innovation/

[123] S. Neema. *Assured Autonomy*. Accessed: Feb. 2020. [Online]. Available: https://www.darpa.mil/program/assured-autonomy

[124] M. Turek. *Explainable Artificial Intelligence (XAI)*. Accessed: Jun. 2019. [Online]. Available: https://www.darpa.mil/program/explainable-artificial-intelligence

[125] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, p. 832, 2019.

[126] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient SMT solver for verifying deep neural networks," in *Proc. Int. Conf. Comput. Aided Verification*, 2017, pp. 97–117.

[127] R. R. Bunel, I. Turkaslan, P. Torr, P. Kohli, and P. K. Mudigonda, "A unified view of piecewise linear neural network verification," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Assoc., 2018, pp. 4790–4799.

[128] K. Dvijotham, R. Stanforth, S. Gowal, T. A. Mann, and P. Kohli, "A dual approach to scalable verification of deep networks," in *Proc. UAI*, vol. 1, 2018, p. 2.

[129] W. Ruan, M. Wu, Y. Sun, X. Huang, D. Kroening, and M. Kwiatkowska, "Global robustness evaluation of deep neural networks with provable guarantees for the Hamming distance," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 5944–5952.

[130] M. Wu, M. Wicker, W. Ruan, X. Huang, and M. Kwiatkowska, "A game-based approximate verification of deep neural networks with provable guarantees," *Theor. Comput. Sci.*, vol. 807, pp. 298–329, Feb. 2020.

[131] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev, "AI2: Safety and robustness certification of neural networks with abstract interpretation," in *Proc. IEEE Symp. Security Privacy (SP)*, May 2018, pp. 3–18.

[132] M. Kuhn and K. Johnson, *Applied Predictive Modeling*, vol. 26. New York, NY, USA: Springer, 2013.

[133] Y. Sun, X. Huang, D. Kroening, J. Sharp, M. Hill, and R. Ashmore, "Testing deep neural networks," 2018. [Online]. Available: arXiv:1803.04792.

[134] Y. Sun, M. Wu, W. Ruan, X. Huang, M. Kwiatkowska, and D. Kroening, "Concolic testing for deep neural networks," in *Proc. 33rd ACM/IEEE Int. Conf. Autom. Softw. Eng.*, 2018, pp. 109–119.

[135] L. Ma *et al.*, "DeepGauge: Multi-granularity testing criteria for deep learning systems," in *Proc. 33rd ACM/IEEE Int. Conf. Autom. Softw. Eng.*, 2018, pp. 120–131.

[136] S. Ma, Y. Liu, W. Lee, X. Zhang, and A. Grama, "MODE: Automated neural network model debugging via state differential analysis and input selection," in *Proc. 26th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, 2018, pp. 175–186.

[137] K. Pei, Y. Cao, J. Yang, and S. Jana, "DeepXplore: Automated whitebox testing of deep learning systems," in *Proc. 26th Symp. Oper. Syst. Principles*, 2017, pp. 1–18.

[138] L. Ma *et al.*, "DeepMutation: Mutation testing of deep learning systems," in *IEEE Int. Symp. Softw. Rel. Eng. (ISSRE)*, 2018, pp. 100–111.

[139] I. Goodfellow and N. Papernot. (Jun. 2017). *The Challenge of Verification and Testing of Machine Learning*. [Online]. Available: http://www.cleverhans.io/security/privacy/ml/2017/06/14/verification.html

[140] S. Mohseni, N. Zarei, and E. D. Ragan, "A survey of evaluation methods and measures for interpretable machine learning," 2018. [Online]. Available: arXiv:1811.11839.

[141] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Commun. ACM*, vol. 63, no. 1, pp. 68–77, 2019.

[142] O. Bastani, C. Kim, and H. Bastani, "Interpretability via model extraction," 2017. [Online]. Available: arXiv:1706.09773.

[143] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014. [Online]. Available: arXiv:1409.0473.

[144] C. Molnar. (2019). *Interpretable Machine Learning*. [Online]. Available: https://christophm.github.io/interpretable-ml-book/

[145] S. Bach, A. Binder, G. Montavon, F. Klauschen, K. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS One*, vol. 10, no. 7, 2015, Art. no. e0130140.

[146] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," 2013. [Online]. Available: arXiv:1312.6034.

[147] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," 2014. [Online]. Available: arXiv:1412.6806.

[148] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.

[149] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[150] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, Dept. Eng., Univ. Cambridge, Cambridge, U.K., 2016.

[151] S. Jha *et al.*, "Attribution-based confidence metric for deep neural networks," in *Advances in Neural Information Processing Systems*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Assoc., Inc., 2019, pp. 11826–11837. [Online]. Available: http://papers.nips.cc/paper/9355-attribution-based-confidence-metric-for-deep-neural-networks.pdf

[152] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," 2015. [Online]. Available: arXiv:1506.06579.

[153] G. Hooker, "Discovering additive structure in black box functions," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2004, pp. 575–580.

[154] J. Krause, A. Perer, and K. Ng, "Interacting with predictions: Visual inspection of black-box machine learning models," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2016, pp. 5686–5697.

[155] A. Zien, N. Krämer, S. Sonnenburg, and G. Rätsch, "The feature importance ranking measure," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Disc. Databases*, 2009, pp. 694–709.

[156] M. M. C. Vidovic, N. Görnitz, K. Müller, and M. Kloft, "Feature importance measure for non-linear learning algorithms," 2016. [Online]. Available: arXiv:1611.07567.

[157] A. Saltelli, "Sensitivity analysis for importance assessment," *Risk Anal.*, vol. 22, no. 3, pp. 579–590, 2002.

[158] J. D. Olden and D. A. Jackson, "Illuminating the 'black box': A randomization approach for understanding variable contributions in artificial neural networks," *Ecol. Model.*, vol. 154, nos. 1–2, pp. 135–150, 2002.

[159] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," 2017. [Online]. Available: arXiv:1703.00810.

[160] Z. C. Lipton, "The mythos of model interpretability," *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[161] X. Huang *et al.*, "A survey of safety and trustworthiness of deep neural networks," 2018. [Online]. Available: arXiv:1812.08342.

[162] D. Roy, T. Mukherjee, and M. Chatterjee, "Machine learning in adversarial RF environments," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 82–87, May 2019.

[163] "Enabling AI research for 5G networks with NI SDR," Austin, TX, USA, Nat. Instrum., White Paper, 2019. [Online]. Available: https://www.ni.com/en-us/innovations/white-papers/19/enabling-ai-research-for-5g-with-sdr-platform.html

[164] E. Balevi and R. D. Gitlin, "Unsupervised machine learning in 5G networks for low latency communications," in *Proc. IEEE Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2017, pp. 1–2.

[165] T. Ma, F. Hu, and M. Ma, "Fast and efficient physical layer authentication for 5G HetNet handover," in *Proc. Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2017, pp. 1–3.

[166] V. P. Kafle, Y. Fukushima, P. Martinez-Julia, and T. Miyazawa, "Consideration on automation of 5G network slicing with machine learning," in *Proc. ITU Kaleidoscope Mach. Learn. 5G Future (ITU K)*, Nov. 2018, pp. 1–8.

[167] M. I. AlHajri, N. T. Ali, and R. M. Shubair, "Classification of indoor environments for IoT applications: A machine learning approach," *IEEE Antennas Wireless Propag. Lett.*, vol. 17, no. 12, pp. 2164–2168, Dec. 2018.

[168] B. Chatterjee, D. Das, S. Maity, and S. Sen, "RF-PUF: Enhancing IoT security through authentication of wireless nodes using *in − situ* machine learning," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 388–398, Feb. 2019.

[169] A. Guerra-Manzanares, H. Bahsi, and S. Nõmm, "Hybrid feature selection models for machine learning based botnet detection in IoT networks," in *Proc. Int. Conf. Cyberworlds (CW)*, Oct. 2019, pp. 324–327.

[170] Y. Liu, Y. J. Morton, and Y. Jiao, "Application of machine learning to the characterization of GPS L1 ionospheric amplitude scintillation," in *Proc. IEEE/ION Position Location Navigation Symp. (PLANS)*, Apr. 2018, pp. 1159–1166.

[171] G. Liu, R. Zhang, C. Wang, and L. Liu, "Synchronization-free GPS spoofing detection with crowdsourced air traffic control data," in *Proc. IEEE Int. Conf. Mobile Data Manage. (MDM)*, Jun. 2019, pp. 260–268.

[172] D. I. Moody, D. A. Smith, T. E. Light, M. J. Heavner, T. D. Hamlin, and D. M. Suszcynsky, "Signal classification of satellite-based recordings of radiofrequency (RF) transients using data-adaptive dictionaries," in *Proc. Asilomar Conf. Signals Syst. Comput.*, Nov. 2013, pp. 1291–1295.

[173] Nvidia. (2018). *DGX-2 Datasheet*. [Online]. Available: https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/dgx-2/dgx-2-print-datasheet-738070-nvidia-a4-web-uk.pdf

[174] F. Altiparmak, F. C. Akyon, E. Ozmen, F. Cogun, and A. Bayri, "Towards cognitive sensing: Radar function classification using multitask learning," in *Proc. Signal Process. Commun. Appl. Conf. (SIU)*, Apr. 2019, pp. 1–4.

[175] R. M. Bowen, F. Sahin, A. Radomski, and D. Sarosky, "Embedded one-class classification on RF generator using mixture of Gaussians," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Oct. 2014, pp. 2657–2662.

[176] V. Camus, L. Mei, C. Enz, and M. Verhelst, "Review and benchmarking of precision-scalable multiply-accumulate unit architectures for embedded neural-network processing," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 4, pp. 697–711, Dec. 2019.

[177] S. Fox, J. Faraone, D. Boland, K. Vissers, and P. H. W. Leong, "Training deep neural networks in low-precision with high accuracy using FPGAs," in *Proc. Int. Conf. Field-Programmable Technol. (ICFPT)*, Dec. 2019, pp. 1–9.

[178] I. Colbert, K. Kreutz-Delgado, and S. Das, "AX-DBN: An approximate computing framework for the design of low-power discriminative deep belief networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–9.

[179] Y. Gwon, S. Dastangoo, C. Fossa, and H. T. Kung, "Fast online learning of antijamming and jamming strategies," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.

[180] L. H. Nguyen and T. D. Tran, "Separation of radio-frequency interference from SAR signals via dictionary learning," in *Proc. IEEE Radar Conf. (RadarConf)*, Apr. 2018, pp. 908–913.

[181] M. A. Hannan, M. M. Hoque, A. Hussain, Y. Yusof, and P. J. Ker, "State-of-the-art and energy management system of lithium-ion batteries in electric vehicle applications: Issues and recommendations," *IEEE Access*, vol. 6, pp. 19362–19378, 2018.

[182] K. Vinsen, S. Foster, and R. Dodson, "Using machine learning for the detection of radio frequency interference," in *Proc. URSI Asia-Pacific Radio Sci. Conf. (AP-RASC)*, Mar. 2019, pp. 1–4.

[183] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," 2019. [Online]. Available: arXiv:1906.02243.

[184] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, "Detection and classification of UAVs using RF fingerprints in the presence of Wi-Fi and Bluetooth interference," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 60–76, 2020.

[185] A. P. Arechiga and A. J. Michaels, "The effect of weight errors on neural networks," in *Proc. IEEE Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2018, pp. 190–196.

[186] A. P. Arechiga and A. J. Michaels, "The robustness of modern deep learning architectures against single event upset errors," in *Proc. IEEE High Perform. Extreme Comput. Conf. (HPEC)*, Sep. 2018, pp. 1–6.

[187] G. Li *et al.*, "Understanding error propagation in deep learning neural network (DNN) accelerators and applications," in *Proc. Int. Conf. High Perform. Comput. Netw. Storage Anal.*, 2017, pp. 1–12. [Online]. Available: https://doi.org/10.1145/3126908.3126964

[188] E. Altland *et al.*, "Quantifying degradations of convolutional neural networks in space environments," in *Proc. IEEE Cogn. Commun. Aerosp. Appl. Workshop (CCAAW)*, Jun. 2019, pp. 1–7.

[189] B. Reagen *et al.*, "Ares: A framework for quantifying the resilience of deep neural networks," in *Proc. ACM/ESDA/IEEE Design Autom. Conf. (DAC)*, Jun. 2018, pp. 1–6.

[190] Z. Yan, Y. Shi, W. Li-Ao, M. Hashimoto, X. Zhou, and C. Zhuo, "When single event upset meets deep neural networks: Observations, explorations, and remedies," 2019. [Online]. Available: arXiv:1909.04697.

[191] M. A. Neggaz, I. Alouani, P. R. Lorenzo, and S. Niar, "A reliability study on CNNs for critical embedded systems," in *Proc. IEEE Int. Conf. Comput. Design (ICCD)*, Oct. 2018, pp. 476–479.

[192] E. Ozen and A. Orailoglu, "Sanity-Check: Boosting the reliability of safety-critical deep neural network applications," in *Proc. IEEE Asian Test Symp. (ATS)*, Dec. 2019, pp. 7–75.

[193] S. Soltani, Y. E. Sagduyu, R. Hasan, K. Davaslioglu, H. Deng, and T. Erpek, "Real-time and embedded deep learning on FPGA for RF signal classification," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, 2019, pp. 1–6.

[194] M. Moore, W. H. Clark, R. M. Buehrer, and W. C. Headley, "When is enough enough? 'Just enough' decision making with recurrent neural networks for radio frequency machine learning," in *Proc. IEEE 39th Int. Perform. Comput. Commun. Conf. (IPCCC)*, 2020, pp. 1–7.