Zero-bias Deep Neural Network for Quickest RF Signal Surveillance

Yongxin Liu[1], Yingjie Chen[2], Jian Wang[3], Shuteng Niu[4], Dahai Liu[3], Houbing Song[3]
[1]Auburn University at Montgomery, Montgomery, AL 36117 USA
[2]Qingdao University, Qingdao, Shandong 266071 China
[3]Embry-Riddle Aeronautical University, Daytona Beach, FL 32114 USA
[4]Bowling Green State University, Bowling Green, OH 43403 USA
[1]yongxin.liu@aum.edu, [2]2018207163@qdu.edu.cn, [3]wangj1@my.erau.edu, dahai.liu@erau.edu, h.song@ieee.org
[4]sniu@bgsu.edu

***Abstract*—The Internet of Things (IoT) is reshaping modern society by allowing a decent number of RF devices to connect and share information through RF channels. However, such an open nature also brings obstacles to surveillance. For alleviation, a surveillance oracle, or a cognitive communication entity needs to identify and confirm the appearance of known or unknown signal sources in real-time. In this paper, we provide a deep learning framework for RF signal surveillance. Specifically, we jointly integrate the Deep Neural Networks (DNNs) and Quickest Detection (QD) to form a sequential signal surveillance scheme. We first analyze the latent space characteristic of neural network classification models, and then we leverage the response characteristics of DNN classifiers and propose a novel method to transform existing DNN classifiers into performance-assured binary abnormality detectors. In this way, we seamless integrate the DNNs with parametric quickest detection. Finally, we propose an enhanced Elastic Weight Consolidation (EWC) algorithm with better numerical stability for DNNs in signal surveillance system to evolve incrementally, we demonstrate that the zero-bias DNN is superior than regular DNN models considering incremental learning and decision fairness. We evaluated the proposed framework using real signal datasets and we believe this framework is helpful in developing a trustworthy IoT ecosystem.**

I. Introduction

The Internet of Things (IoT) is providing applications and services that would otherwise not be possible [1], [2]. Intelligent decision making is of great significance in IoT [3]. A typical way to implement smart decision functionality in IoT is by integrating learning-enabled components through Deep Learning (DL) and Deep Neural Networks (DNNs). One typical application of DNNs in IoT is the RF signal surveillance to either identify device type or modulation schemes [4]–[6].

Although DL and DNNs have been applied in the recognition of RF signals for device identification [7] and event surveillance [8], applying DNNs in safety-critical systems requiring assured performance is still controversial. Firstly, DNNs perform well on specifying known subjects but cannot distinguish abnormalities. Abnormal signals, such as those from unauthorized signal sources, are required to be identified accurately rather than being erroneously classified into the most likely known ones [9]. Secondly, DNN related systems lack the characteristics of timeliness assurability, while applications in safety-critical systems require making accurate decisions with both theoretical assured minimum latency and pre-defined false alarm constraints. The two obstacles impede the deployment of DL and DNNs in IoT of safety-critical systems.

For unknown event detection, the most intuitive way is to use statistical models to generate likelihood metrics and then use thresholds to distinguish whether an input is within the learned knowledge domain. However, selecting features from data and specifying statistical metrics can also be time-consuming. Existing works use deep autoencoders or Generative Adversarial Networks (GANs) as well as reconstruction loss to measure whether an input is from some known domain. However, training deep autoencoders or GAN models is more computationally expensive. Moreover, autoencoders or GAN models do not guarantee to respond with constrained false alarms or predictable behaviors [10].

The problem of timeliness assurability has been widely discussed in Quickest Detection (QD) algorithms. QD algorithms are widely applied for detecting the abrupt change of statistical parameters with the lowest latency under given false alarm constraints. Existing Quickest Detection (QD) algorithms can detect changes with minimum latency under constrained false alarms, they are neither sufficient in handling high dimension inputs nor can provide mathematically assured performance. Even though there are some methods to integrate quickest detection with DNNs, the performances of the connected systems are only measurable but not strictly assurable. We have to claim that there is a gap between machine learning and QD.

In this paper, we utilize the enhanced deep learning framework based on our previous work [11], the zero-bias DNN model, and significantly enhance it for quickest and reliable classification of wireless signals. In this DL framework, we facilitate DNNs with both explainable behaviors in distinguishing known or abnormal inputs. Besides with minimum latency under certain false alarm constraints. Furthermore, our solution efficiently transforms existing DNNs into abnormality detectors with predictable performance. The effectiveness of the

978-1-6654-4331-9/21/$31.00 ©2021 IEEE

proposed framework in handling massive signal recognition has been demonstrated. Our contributions are as follows:

- We explore the latent space characteristics of DNNs and discovered a novel method to efficiently transfer existing DNN classifiers into DNN abnormality detectors with adaptive decision boundaries.
- We provide a more stable Elastic Weight Consolidation (EWC) algorithm and show that zero-bias DNNs are more reliable than regular DNNs during incremental learning.
- We combine our zero-bias DNN model with the parametric Quickest Change Detection theory, and our validation on massive real signal detection demonstrates the effectiveness of our integral solution.

Our research offers a solution to the accurate identification of RF signals with assured performance, thus useful in promoting trustworthy IoT and deepening the understanding of deep neural networks. Besides, the successful integration of the neural network and QD enables the move from IoT to real-time control.

The remainder of this paper is organized as follows: A literature review of related works is presented in Section II. We present the methodology in Section IV. Performance evaluation is presented in Section V with conclusions in Section VI.

II. Related Work

Real-time event detection is a critical function in safety-critical IoT. From the perspective of input data, we may categorize them into single-shot and sequential detection paradigms. In single-shot detection [11], event detections are performed per observation, and the past data will not be retained for future use. In contrast, the sequential detection paradigm allows accumulating information from past observations [10].

A. Single-shot unknown event detection in DNN

Event detection plays an increasingly important role in safety-critical and latency-constrained IoT, e.g., the aviation communication system. Detecting known events are straightforward, while detecting abnormal or unseen events are more difficult.

A critical problem for DL enabled signal identification systems is that classifiers only recognize pretrained data but can not deal with abnormal or unknown data. From the perspective of DL, this issue is categorized as the Open Set Recognition [12], [13] problem. An intuitive solution is to model the distribution in the latent space. In [14], the authors first trained a CNN model with a Softmax output on known data. They then remove the Softmax layer and turn the neural network into a nonlinear feature extractor. Finally, they use the DBSCAN algorithm to perform cluster analysis on the remapped features and show that the method has the potential of detecting a limited number of unknown classes. In [15], the authors provide two methods to deal with abnormalities: i) Reuse trained convolutional layers to transform inputs to feature vectors, and then use Mahalanobis distance to judge the outliers. ii) Reuse the pretrained convolutional layers to transform signals to feature vectors, and then perform k-means (k = 2) clustering to discover the groups of outliers. Another approach is to leverage the characteristics of generative models. In [16], the authors use the Generative Adversarial Network (GAN) to generate highly realistic fake data. Then they exploit the discriminator network to distinguish whether an input is from an abnormal source.

B. Sequential event detection

From the perspective of the stochastic process, a wireless communication system in different states can be described by distributions with measurable statistical properties [17]. Therefore, transitions within states cause the change of those properties. The quickest detection aims to detect the change as quickly as possible, subject to false alarm constraints [18]. Considering whether prior observations are independent of an abnormal event's appearance, the optimization scheme can be defined in different forms as in [19]. We can also categorize the quickest event detection methods into two branches: a) detecting events with known post-change distributions. b) detecting events with unknown post-change distributions. Generally, detecting known events is faster with the Cumulative Sum Control Chart (CUSUM) algorithm can be applied directly [20], [21]. A postchange distribution may not be known in some scenarios in advance, and nonparametric strategies have to be used and bring higher latency.

Quickest detection provides a performance-assured solution to detect change points (related to events) in sequential data. However, the selection of statistic metrics still depends on trial-and-error. We focus on real-time sequential detection of events, especially on integrating the quickest detection theory with deep learning to provide an automated and performance-assured solution to latency-constrained CPS.

III. Problem Definition

Suppose that we have a sequence of signal vectors denoted as:

$$\boldsymbol{SS} = \{\boldsymbol{S_1}, \ldots, \boldsymbol{S_k}, \ldots, \boldsymbol{S_n}\} \tag{1}$$

Suppose that some known or unknown events will occur at time k, our signal surveillance system is required to detect the occurrence of the known or unknown event with minimal delay.

One straightforward method is to use a DNN model $D(\cdot)$ to process $\boldsymbol{SS}$ sequentially, the goal of $D(\cdot)$ is to provide a score for each signal element to quantify whether it is from the previous known knowledge domain. From the perspective of domain adaption, feature extractors are specifically trained to fit the characteristics only within their learned tasks [22], the task-specific knowledge domain. However, the DNN model, $D(\cdot)$, can be trivial to use, firstly, we do not have good method to explain or adjust the decision threshold for $D(\cdot)$. Secondly, $D(\cdot)$ can generate false alarms or encounter miss detection, we need to find a sequential detection scheme that can aggregate evidence sequentially and provide the minimal

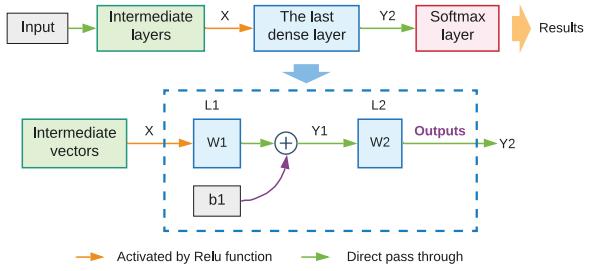


Fig. 1. Data flow of zero-bias deep neural networks.

detection latency. Finally, if the signal surveillance system is required to evolve incrementally, $D(\cdot)$ needs to be retrained frequently with large overhead as new data are emerging incrementally. Therefore, we need to develop a new DL paradigm that: a) enables the explainable and reliable event detection. b) being able to learn incrementally and adapt to operational variations.

IV. Methodology

A. The zero-bias neural network

We have discovered that the last dense layer of a DNN classifier performs the nearest neighbor matching with biases and preferabilities using cosine similarity, We also show that a DNN classifier's accuracy will not be impaired if we replace its last dense layer with a zero-bias dense layer [11], in which the decision biases and preferabilities are eliminated. We can denote its mechanism as (also in Figure 1):

$$\begin{aligned} \boldsymbol{Y_1}(\boldsymbol{X}) &= \boldsymbol{W_0}\boldsymbol{X} + \boldsymbol{b} \\ \boldsymbol{Y_2}(\boldsymbol{X}) &= cosDistance(\boldsymbol{Y_1}, \boldsymbol{W_1}) \end{aligned} \tag{2}$$

where $\boldsymbol{X}$ is the output of the prior convolution layers, a.k.a., feature vectors. $\boldsymbol{X}$ is an N_0-D vector, where N_0 denotes the number of features. $\boldsymbol{W}_0$ is an N_0 by C matrix where C denotes the feature dimension in the latent space which equals to the number of classes. $\boldsymbol{W_1}$ is a matrix to store fingerprints of different classes, namely the similarity matching layer and it is a C by C square matrix. Please be noted that in $\boldsymbol{W_1}$, each row represents a fingerprint of corresponding class whilst in $\boldsymbol{Y}_1$ each column represents a feature vector in the latent space. In short, the last dense layer is spitted into two layers, L_1 for feature embedding and L_2 for vector scaling and similarity matching. We have the first remark:

Remark 1. *Latent space of neural networks: The latent space of a neural network for the final classification is a unit hypersphere surface. We define it as the classification hypersphere surface.*

We have proved that the classification comparison in a regular neural network is the angular matching with class-specific biases and weights, while in a zero-bias neural network, the biases are eliminated and the weights are equalized to one. We assure that decisions can not be made according to biases in safety critical systems. Besides that, our previous work has demonstrated that such a modification will not impair the classification performance of DNNs [11].

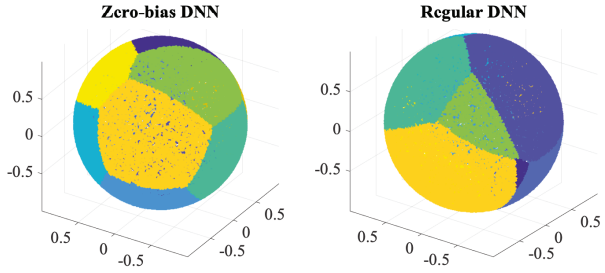


Fig. 2. Associate region of classes on MNIST dataset. Data are mapped into a 3D space using t-SNE. Colors represent different classes. Code available at https://github.com/pcwhy/NeuralDBVis

To demonstrate this characteristic, we use a hand-written digit classification model as in [23] and we convert it to a zero-bias neural network and retrain it. Next, we then generate random points that uniformly cover the classification hypersphere surfaces of the two models and associate the random points with their nearest class fingerprint. Finally, we use the t-SNE [24] algorithm to remap the class fingerprints into a 3D hypersphere and visualize the association region of each class as in Figure 2.

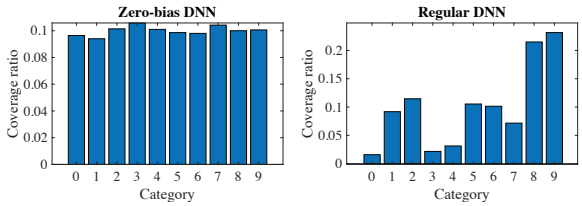


Fig. 3. Compare of classification hypersphere ratio of on MNIST dataset.

As depicted, in the zero-bias DNN, the classification hypersphere is uniformly divided into subregions for different classes, the subregions are are not uniform when it is in the regular DNN. A more explicit numerical comparison is given in Figure 3. The DNN model in signal surveillance system needs to treat the input signals without biases and preferences, and thus the zero-bias neural networks can perform much better than the regular neural networks in distinguishing abnormalities (identifying the unknown input data as summarized in Table I). Despite that one-class SVM is slightly better, it requires an optimal threshold case by case.

Therefore, we believe the zero-bias neural network is will be a better tool to enable the intelligent surveillance of RF signals in IoT.

B. Zero-bias neural network for unsupervised and adaptive abnormality detection

In the zero-bias neural network, the comparison between class fingerprints and mapped data in the feature space is fair without bias and weights. Naturally, we could assume that:

Remark 2. *For each fingerprint in the classification hypersphere, there will be a cut-off cosine similarity value that*

TABLE I
PERFORMANCE OF ABNORMALITY DETECTORS.

Metric	One-class SVM [1]	Zero-bias DNN [1]	Regular DNN [1]
False Positive	0.19	**0.2**	0.2
False Negative	0.05	**0.05**	0.28

[1] We set a threshold value on the maximum matching score of each input, and the threshold is set according to the maximum margin of separation as in [11].

separates the feature vectors of known and abnormal data. We define this value as the cut-off distance of this fingerprint.

To verify this assumption, we use an aircraft ADS-B signal dataset [25] with the corresponding zero-bias DNN signal emitter identification model in Figure 4.

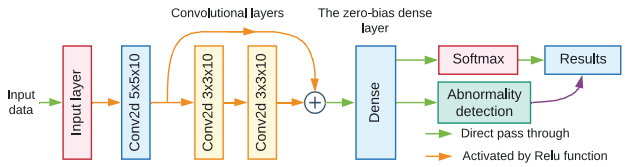


Fig. 4. Deep neural network architecture [26].

We first train the network in different stages, we then use the t-SNE algorithm to visualize the class fingerprints, feature vectors from known and abnormal data in the classification hypersphere as in Figure 5. We can find several important features:

- The size of clusters for different classes are gradually becoming smaller.
- The abnormalities are gradually becoming more distinctively separated from the known data. We can depict the relation of feature vectors from regular data and abnormalities as in Figure 6.
- The abnormalities (signals from unknown RF emitters) distribute randomly throughout the classification hypersphere.

For a given DNN model with the zero-bias dense layer, we follow these steps to find the cut-off distance of each class fingerprint:

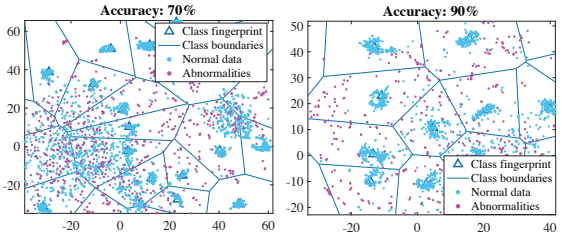


Fig. 5. Class fingerprints, feature vectors of known and abnormal data in the classification hypersphere in the zero-bias neural network for signal identification

Step 1: The training set is utilized to learn the boundaries of known classes while the validation set will be mixed with abnormal data ($\boldsymbol{A_0}$) to measure the performance of converted abnormality detector.

Step 2: We pass accurately classified data of ith known class from the training set, denoted as $\boldsymbol{KX}_i$, through layers of the DNN model and obtain the compressed feature vectors before fingerprint matching, denoted as:

$$\boldsymbol{Y}_1[F_{n-1}(\boldsymbol{KX}_i)] = \boldsymbol{W}_0 F_{n-1}(\boldsymbol{KX}_i) + \boldsymbol{b} \tag{3}$$

Where $\boldsymbol{W}_0$ and $\boldsymbol{b}$ are defined in Equation 2, $F(\cdot)_{n-1}$ denotes all network layers before the fingerprint matching. $\boldsymbol{Y}_1[F_{n-1}(\boldsymbol{KX}_i)]$ denotes feature vectors of accurately classified data in $\boldsymbol{KX}_i$.

Step 3: Calculate the centroid $\boldsymbol{c}_0^i$ of $\boldsymbol{KX}_i$ as:

$$\boldsymbol{c}_0^i = mean(\boldsymbol{Y}_1[F_{n-1}(\boldsymbol{KX}_i)]) \tag{4}$$

Step 4: Calculate all the cosine distances between the $\boldsymbol{c}_0^i$ to all accurately classified feature vectors. We then use the greatest cosine distance value as the cut-off distance, CO_i, for the ith known class.

Step 5: Abnormality detection using cut-off boundaries on input data $\boldsymbol{X}$) is formally defined as:

$$D(\boldsymbol{X}) = \begin{cases} 1 & \exists\ i,\ cosineDistance[\boldsymbol{Y}_1, \boldsymbol{c}_0^i] \le CO_i \\ 0 & \text{Otherwise} \end{cases} \tag{5}$$

These steps convert a zero-bias DNN into an abnormality detector with binary output. In this binary abnormality detector, we do not need to specifically adjust its decision threshold compared with our previous method in [11]. According to our observation, as long as the zero-bias DNN is well-trained before conversion, very few anomaly data may have a cosine distance less than the cut-off distance.

From the perspective of signal surveillance, we may randomly encounter known or abnormal signals. Therefore, the output of the binary abnormality detector can also be regarded as a random sequence, in which when the signals are from some known events, the output of the abnormality detector follows a Bernoulli distribution: [27]:

$$P_0(I_k) = FPR^{I_k}(1 - FPR)^{1-I_k} \tag{6}$$

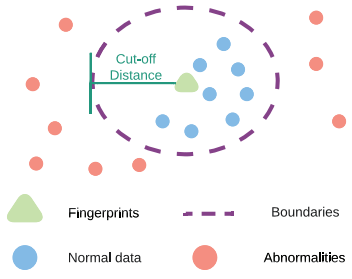


Fig. 6. Class cut-off distance for distinguishing known and abnormal feature vectors in the classification hypersphere of zero-bias DNN.

where $I_k \in \{0, 1\}$ is the output of the binary abnormality detector with $I_k = D(\boldsymbol{X_k})$. FPR is the false positive rate of the binary abnormality detector.

When we encounter some abnormal events, the output of the binary abnormality will become:

$$P_1(I_k) = (1 - FNR)^{I_k} FNR^{1-I_k} \tag{7}$$

$$= (TPR)^{I_k} (1 - TPR)^{1-I_k} \tag{8}$$

where TNR and TPR are the true negative and true positive rates of the binary abnormality detector.

Relations of FPR, TPR of the binary abnormality detector and the training accuracy of zero-bias DNN before conversion are depicted in Figure 7. These relations can be quantified using two linear models on both MNIST [23] and our ADS-B signal dataset [25], [26]. They are:

$$FPR = 1 - ACC \quad R^2 = 0.85 \tag{9}$$

$$TPR = 0.2 + 0.77 \cdot ACC \quad R^2 = 0.89 \tag{10}$$

Therefore, we can directly use the training accuracy of the zero-bias DNN model as an predictor for FNR and TPR of the converted binary abnormality detector.

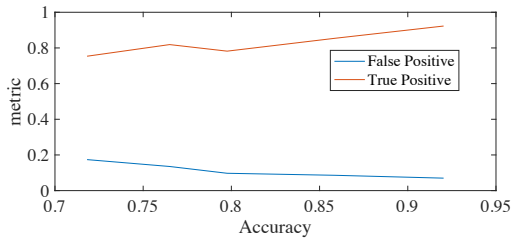


Fig. 7. Performance of the converted abnormality detector.

C. Sequential event detection with zero-bias neural network

As we have converted the regular zero-bias DNN model into a binary abnormality detector and have formulated the behavior of this model using two Bernoulli Distributions with predictable parameters. We can then define the event detection problem as a sequential statistical test scheme using the CUSUM algorithm.

First, a likelihood ratio test is employed to sequentially process the observed data at each timestamp k, denoted as:

$$g(k) = ln(\frac{P_1(\boldsymbol{S}_k)}{P_0(\boldsymbol{S}_k)}) \tag{11}$$

where $g(k)$ is a sufficiency metric, $P_0(\cdot)$, $P_1(\cdot)$ denotes the probabilistic density functions of abnormal and abnormal states, respectively. A constrained cumulative sum of sufficiency metrics is used as an indicator, denoted as:

$$s(k) = max(0, s(k-1) + g(k)) \tag{12}$$

An alarm will be sent once $s(k)$ is greater than a predefined threshold, h_{CUSUM}, and this alarm will indicate that some unknown events are happening. The CUSUM algorithm has been proved to provide the lowest worst-case detection latency at specific false alarm intervals [21], [28]. Therefore, if our solution is applied, the detection threshold, h_{CUSUM}, is the only parameter that needs to be specified.

D. Light-weight incremental learning algorithm for zero-bias neural network

A benefit of zero-bias DNN enabled binary abnormality detector is that incremental learning can be implemented to facilitate the model to evolve in its lifecycle. Incremental learning enables a neural network to classify new targets without needing to retrain from scratch. In our research, specific events can be learned as new classes to be recognized directly. For both zero-bias DNN and conventional DNN classifiers:

Remark 3 (Incremental learning on the classification hypersphere)**.** *To enable a neural network to recognize a new class, we only need to place its class fingerprint on the classification hypersphere and fine-tune the old fingerprints' directions when necessary.*

We also have:

- For a specific new class, as long as the previous layers have extracted sufficient distinctive features, we do not need to retrain the previous layers.
- For new classes, we need to insert new fingerprints and then adjust the old fingerprints when necessary.

To adjust an old fingerprint, we need to identify which parameter (or dimension) is critical to the classification accuracy. According to the Elastic Weight Consolidation (EWC) [29], the Fisher Information Matrix is used to model the importance of parameters as:

$$\boldsymbol{F_\Omega} = [\frac{\partial log(P(\boldsymbol{X}_{CV}|\boldsymbol{\Omega}))}{\partial \boldsymbol{\Omega}}][\frac{\partial log(P(\boldsymbol{X}_{CV}|\boldsymbol{\Omega}))}{\partial \boldsymbol{\Omega}}]^T$$

$$P(\boldsymbol{X}_{CV}|\boldsymbol{\Omega}) \approx \overline{\boldsymbol{Y}_{Softmax}}(\boldsymbol{X}_{CV}|\boldsymbol{\Omega}) \tag{13}$$

Where $\overline{\boldsymbol{Y}_{Softmax}}(\boldsymbol{X}_{CV}|\boldsymbol{\Omega})$ denotes the averaged outputs of Softmax layer on validation set $\boldsymbol{X}_{CV}$ given parameter set $\boldsymbol{\Omega}$, it approximates the posterior probability $P(\boldsymbol{X}_{CV}|\boldsymbol{\Omega})$. $\boldsymbol{F_\Omega}$ denotes the Fisher information matrix of the current task. In our experiment, we further apply an exponential function to the Fisher Information to increase the numerical stability as:

$$\boldsymbol{F_\Omega} := \exp(\boldsymbol{F_\Omega}) \tag{14}$$

Intuitively, the importance of a parameter is equivalent to the square of its gradient with respect to the logarithm of Softmax function.

Knowing the importance of existing parameters, we can define an integral loss function for incremental learning as:

$$F_1(\boldsymbol{\Omega}) = \frac{\lambda_1}{2} \sum_i [\boldsymbol{F}_{\boldsymbol{\Omega}^*} \cdot (\boldsymbol{\Omega} - \boldsymbol{\Omega}^*)^2]$$

$$L(\boldsymbol{\Omega}) = (L_2(\boldsymbol{\Omega}) + F_1(\boldsymbol{\Omega})) \cdot \boldsymbol{G_m} \tag{15}$$

Where $F_1(\boldsymbol{\Omega})$ denotes the Fisher Loss with respect to old tasks (a.k.a., task-1). $\boldsymbol{\Omega}^*$ denotes the loss function and model parameters on task-1. $L_2(\boldsymbol{\Omega})$ and $\boldsymbol{\Omega}$ denote the raw loss function on Task-2 and the new model parameters. λ_1 denotes

the importance of task-1. Intuitively, this integral loss function additionally penalizes the change of critical parameters. $\boldsymbol{G_m}$ is a mask matrix to control which parameter is locked or unlocked. The value of each element can only be zero or one.

Given a neural network trained on Task-1 (DNN_1), incremental learning on Task-2 is performed as follows:

Step 1: Store all learnable parameters of DNN_1 as $\boldsymbol{\Omega^*}$ and calculate their importance matrix $\boldsymbol{F_{\Omega^*}}$.

Step 2: Generate the initial fingerprint of each new class by averaging their feature vectors.

Step 3: Concatenate initial fingerprints into the last dense layer or zero-bias dense layer.

Step 4: Lock the weights of previous layers and calculate the importance of parameters of old fingerprints. The importance of newly concatenated fingerprints is set to zeros; thus, we could allow them to learn freely.

Step 5: Use loss function as in Equation (15) and a training set of Task-2 to perform network training.

Notably, we do not need to retain old training data to learn a new task, and such a benefit is critical for DNN models in practical scenarios.

V. Evaluation and Discussion

A. Evaluation dataset

Our dataset is available in [25], we use the wide-spreading signals from Automatic Dependent Surveillance-Broadcast (ADS-B) signals [30], which provides a great variety of signals from commercial aircraft's RF transponders with labels. Specifically, each aircraft use transponders at 1090MHz to broadcast its flight information to the Air Traffic Control (ATC) center. The integrity and trustworthiness of ADS-B messages are critical to aviation safety. However, the ADS-B system does not contain cryptographic identity verification mechanisms and thus is vulnerable to identity spoofing attacks. Our previous works [11], [26] have shown that the responses of the zero-bias DNN to known (learned) aircraft and unknown sources (also from unknown aircraft) can be modeled by different probability distributions. Here we define the appearance of unknown aircraft's signals as abnormal events, we can use the framework in this paper to design a sequential event detector to aggregate warnings and identify the adversaries who use fake IDs.

From the perspective of DL, the input is the raw signal collected by a Software Defined Radio Receiver (USRP B210) and the DNN is trained to identify the known aircraft through their signals. As in our previous work [11], [26], we take the first 1024 samples from each signal record and extract pseudo noise, magnitude-frequency, and phase-frequency features. The extracted features of each signal record are then packed into a 32 by 32 by 3 tensor. The architecture of our DNN model is depicted in Figure 4 with a description of the dataset in Table II. After training to recognize known aircraft, the zero-bias DNN model is then converted to a binary abnormality detector as in Section IV-B.

TABLE II
Description of Dataset

Usage	Description
Training	60% of signal records from 28 aircraft.
Test	40% of signal records from 28 aircraft.
Normal data	The test set.
Abnormal data	Signal records from the remaining 100 aircraft.

B. Quickest abnormal event detection

The converted binary abnormality detector can be utilized for abnormal event detection with very low latency as a result of both high true positive and low false positive rates. To further evaluate our proposed method, we first define a quality metric, $Q = \frac{TPR}{FPR}$, for the binary abnormality detector. Then, we can use numerical simulation to evaluate the performance of zero-bias DNN under different Q values and different sequential detection algorithms: CUSUM [21], EWMA (Exponentially Weighted Moving Average [31]) and sliding window [32]. We simulate the possible values of h_{GLR}, FPR, and TPR that a binary abnormality detector can encounter with $TPR \in [0.6, 0.99]$, $FPR = 0.4$, $Q \in [1.625, 2.25]$. We configure three sequential detection algorithms as follows:

- *CUSUM:* we set the event detection threshold $h_{CUSUM} \in [2.0, 20.0]$.
- *EWMA:* we set $\lambda = 0.15$ and $L \in [3.0, 4.0]$.
- *Sliding window:* we set the length of window $L \in [50, 300]$ with a threshold 0.7.

We first compare the best and the worst detection delay of the three sequential event detection method in Figure 8. We found that considering the best case, the detection delays of EWMA and CUSUM algorithms are close while in the worst case the CUSUM algorithm performs better than EWMA and sliding window. The averaged detection delays as well

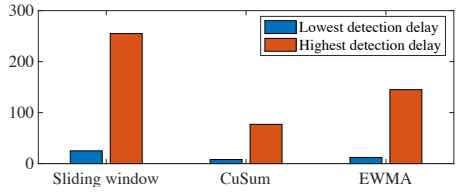


Fig. 8. The best and the worst case detection delay

as its range are compared in Figure 9 and 10. Although EWMA algorithm seems to achieve the best performance in the averaged detection delay, the ranges of detection delay in Figure 10 reveal that the EWMA algorithm is not very stable when the Q value is not sufficiently large. As predicted, the sliding window algorithm always has the worst performance.

C. Incremental learning

To evaluate our incremental learning mechanism, we separate our data set into two parts, namely *task-1* and *task-2*, respectively. We first train the zero-bias DNN on *task-1*

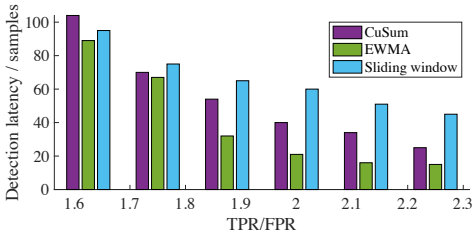


Fig. 9. The averaged detection delay.

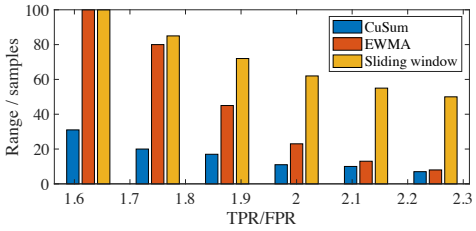


Fig. 10. The range of detection delay.

and use continual learning mechanisms to let our network recognize wireless transmitters in *task-2*.

1) Numerical stability: We compare the numerical stability of Fisher Loss during incremental learning. The results in Figure 11 demonstrate that without applying the exponential function as in Equation (14), the Fisher Loss is numerically unstable and gradually vanishes to zero (depicted by dashed lines). When Fisher Loss becomes zero, the incremental learning algorithm can no longer penalize the neural network for forgetting the old tasks. In contrast, if the exponential function is applied, the Fisher Loss never vanishes to zero and prevents catastrophic forgetting. As incremental learning procedures, the Fisher Loss gradually converges to a nonzero constant value. The results indicate that the zero-bias layer has a smoother converging characteristic than the regular dense layer.

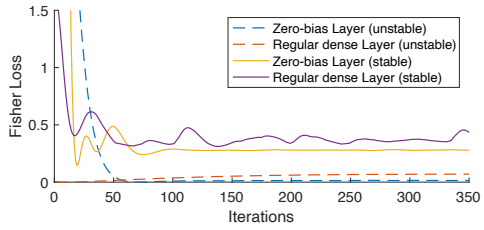
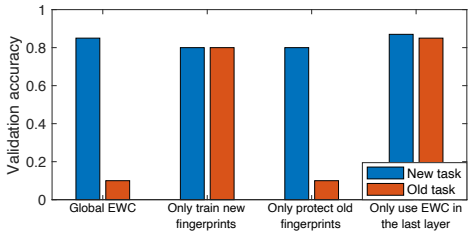


Fig. 11. Compare of numerical stability during continual learning

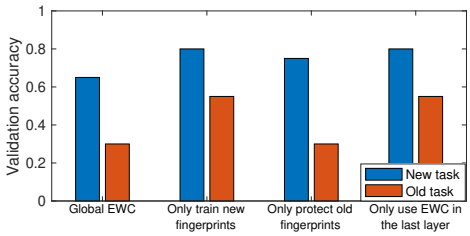
2) Comparison of incremental learning approaches: This subsection will compare other incremental learning approaches with our method. The descriptions of all these approaches are given in Table III. We aim to compare the effect of EWC as well as other network knowledge protection methods. Please be noted that during the incremental learning, $L2$ regularization factors for the regular dense layer and zero-bias layer are all set to 0 and 0.025, respectively.

TABLE III
COMPARED APPROACHES FOR CONTINUAL LEARNING

Approaches	Lock zero-bias layer	Elastic Weight Consolidation	Locked prior layers
Global EWC	No	Globally	No
Only train new Fingerprints	Lock old fingerprints	No	Yes
Only protect old Fingerprints	Lock old fingerprints	Yes	No
Only use EWC in the last layer	No	Only in the last layer.	Yes



(a) Zero-bias layer



(b) Regular dense layer

Fig. 12. Performance comparison of zero-bias layer and regular dense layer DNNs for incremental learning

The results are given in Figure 12, with the following highlights:

1) In Global EWC, catastrophic forgetting is not prevented. Besides, the zero-bias layer retains far less knowledge from previous tasks.
2) Only training new fingerprints and locking all old weights in the network can help retaining knowledge from previous tasks. This phenomenon indicates that the prior layers have already extracted useful features for the final classification. Moreover, the zero-using bias layer's performance indicates that it can enable prior neural network layers to discover better features without relying on biases and weights. Please be noted that this scenario also prevents the fine-tuning of existing fingerprints even if they are in sub-optimal directions.
3) Only protecting old fingerprints does not seem to be helpful. The new task will destroy all useful feature extractors in prior layers.
4) Applying EWC only in the last layer provides the most promising results. Notably, the neural networks with the zero-bias layer still outperform regular neural networks. This fact explains that EWC tries to protect old fingerprints from changing erroneously (forgetting) and enables

fine-tuning.

VI. Conclusion

In this paper, we significantly extend the analysis of our previously proposed zero-bias DNN and combine it with the Quickest Detection algorithms to detect abnormalities and time-dependent abnormal events in IoT with the lowest assured latency. We first analyze the zero-bias DNN and show that zero-bias DNN is superior than regular DNN for RF signal surveillance. We then propose a novel solution to convert zero-bias DNN classifiers into performance-assured binary abnormality detectors. We model the converted abnormality detectors using Bernoulli distribution, which perfectly adapts to the CUSUM based Quickest Detection scheme. In this Quickest Detection scheme, the theoretically assured lowest abnormal event detection delay is provided with predictable false alarms. Finally, to facilitate DNN for RF signal surveillance to evolve incrementally, we propose a more stable EWC algorithm and shown that zero-bias DNN is more reliable than regular DNN under incremental learning. The framework is evaluated using both massive signal records from real-world aviation communication systems and simulated data. In the future, we will explore the incremental learning capability of zero-bias DNN.

Acknowledgment

This research was partially supported by the Center for Advanced Transportation Mobility (CATM), USDOT under Grant No. 69A3551747125 and the National Science Foundation under Grant No. 1956193.

References

[1] Y. Sun, H. Song, A. J. Jara, and R. Bie, "Internet of things and big data analytics for smart and connected communities," *IEEE Access*, vol. 4, pp. 766–773, 2016.

[2] H. Song, D. B. Rawat, S. Jeschke, and C. Brecher, *Cyber-physical systems: foundations, principles and applications*. Elsevier, 2016.

[3] L. Wang, W. Song, Y. Lan, H. Wanga, X. Yue, X. Yin, E. Luo, B. Zhang, Y. Lu, and Y. Tang, "A smart droplet detection approach with vision sensing technique for agricultural aviation application," *IEEE Sensors Journal*, pp. 1–1, 2021.

[4] I. Butun, P. Österberg, and H. Song, "Security of the internet of things: Vulnerabilities, attacks, and countermeasures," *IEEE Communications Surveys Tutorials*, vol. 22, no. 1, pp. 616–644, 2020.

[5] J. Wang, Y. Liu, S. Niu, and H. Song, "Reinforcement learning optimized throughput for 5g enhanced swarm uas networking," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[6] Y. Liu, J. Wang, J. Li, S. Niu, and H. Song, "Class-incremental learning for wireless device identification in iot," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[7] ——, "Machine learning for the detection and identification of internet of things (iot) devices: A survey," 2021.

[8] J. Wang, Y. Liu, and H. Song, "Counter-Unmanned Aircraft System (s)(C-UAS): State of the Art, Challenges, and Future Trends," *IEEE Aerospace and Electronic Systems Magazine*, vol. 36, no. 3, pp. 4–29, 2021.

[9] Y. Jiang, M. Wang, X. Jiao, H. Song, H. Kong, R. Wang, Y. Liu, J. Wang, and J. Sun, "Uncertainty theory based reliability-centric cyber-physical system design," in *2019 International Conference on Internet of Things (iThings) and IEEE GreenCom/CPSCom/SmartData*. IEEE, 2019, pp. 208–215.

[10] P. Perera and V. M. Patel, "Efficient and low latency detection of intruders in mobile active authentication," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 6, pp. 1392–1405, 2017.

[11] Y. Liu, J. Wang, J. Li, H. Song, T. Yang, S. Niu, and Z. Ming, "Zero-bias deep learning for accurate identification of internet-of-things (iot) devices," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2627–2634, 2021.

[12] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1757–1772, 2012.

[13] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1563–1572.

[14] 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 *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE, 2018, pp. 26–33.

[15] Y. Shi, K. Davaslioglu, Y. E. Sagduyu, W. C. Headley, M. Fowler, and G. Green, "Deep Learning for RF Signal Classification in Unknown and Dynamic Spectrum Environments," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 2019, pp. 1–10.

[16] D. Roy, T. Mukherjee, M. Chatterjee, E. Blasch, and E. Pasiliao, "Rfal: Adversarial learning for rf transmitter identification and classification," *IEEE Transactions on Cognitive Communications and Networking*, 2019.

[17] L. Lai, Y. Fan, and H. V. Poor, "Quickest Detection in Cognitive Radio: A Sequential Change Detection Framework," in *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, 2008, pp. 1–5.

[18] H. Poor and O. Hadjiliadis, *Quickest detection*. United Kingdom: Cambridge University Press, Jan. 2008, vol. 9780521621045.

[19] P. Johnson, J. Moriarty, and G. Peskir, "Detecting changes in real-time data: a users' guide to optimal detection," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 375, no. 2100, p. 20160298, 2017. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2016.0298

[20] M. Basseville, I. V. Nikiforov *et al.*, *Detection of abrupt changes: theory and application*. prentice Hall Englewood Cliffs, 1993, vol. 104.

[21] P. Granjon, "The CuSum algorithm-a small review," 2013.

[22] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[23] "Create simple deep learning network for classification - MATLAB & Simulink example," https://www.mathworks.com/help/deeplearning/ug/create-simple-deep-learning-network-for-classification.html, (Accessed on 07/12/2021).

[24] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[25] Y. Liu, J. Wang, H. Song, S. Niu, and Y. Thomas, "A 24-hour signal recording dataset with labels for cybersecurity and IoT," 2020. [Online]. Available: http://dx.doi.org/10.21227/gt9v-kz32

[26] Y. Liu, J. Wang, S. Niu, and H. Song, "Deep learning enabled reliable identity verification and spoofing detection," in *Wireless Algorithms, Systems, and Applications*, D. Yu, F. Dressler, and J. Yu, Eds. Cham: Springer International Publishing, 2020, pp. 333–345.

[27] E. W. Weisstein, "Bernoulli distribution," *https://mathworld. wolfram. com/*, 2002.

[28] L. Xie, S. Zou, Y. Xie, and V. V. Veeravalli, "Sequential (Quickest) Change Detection: Classical Results and New Directions," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 2, pp. 494–514, 2021.

[29] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[30] J. Sun, "An open-access book about decoding Mode-S and ADS-B data," https://mode-s.org/, May 2017.

[31] N. Ye, S. Vilbert, and Q. Chen, "Computer intrusion detection through ewma for autocorrelated and uncorrelated data," *IEEE Transactions on Reliability*, vol. 52, no. 1, pp. 75–82, 2003.

[32] C.-H. Lee, C.-R. Lin, and M.-S. Chen, "Sliding-window filtering: An efficient algorithm for incremental mining," ser. CIKM '01. New York, NY, USA: Association for Computing Machinery, 2001, p. 263–270. [Online]. Available: https://doi.org/10.1145/502585.502630