Detection of Adversarial Attacks in Machine Learning-Based Wireless Communication Systems

Ewen Rai supervised by Aissa Ikhlef

***Abstract*—With wireless communication being a fundamental component of modern society, enabling easy connectivity between multiple devices, machine learning (ML), and more specifically deep learning (DL) methods such as neural networks (NN), have emerged as powerful techniques to aid in the automation of radio signal classification for these systems. Yet, recent strides to incorporate deep learning's ability to classify and detect radio signals have uncovered notable security vulnerabilities. Such shortcomings have created opportunities for adversarial attacks capable of causing significant disruptions and harm to communication networks. Despite the substantial and potentially detrimental costs associated with these attacks, there is a scarcity of research that aims to develop robust defences against adversarial attacks in radio signal processing. Therefore, this paper reintroduces [10] two highly effective adversarial attacks, namely a white-box and a black-box attack, and utilises features crafted from the Softmax layer output to classify whether an attack has taken place or not. Detection accuracies have been evaluated to be as high as 99.2% for the white-box attack, with little variation observed across different PNRs. On the other hand, the detection accuracy of the black-box attack exhibits a larger variation among different PNRs. While it can achieve a remarkably high accuracy of 99.5% at a PSR of 10dB, it drops down to 90.3% at a PSR of -10dB and down again to 65.4% at a PSR of -20dB.**

***Index Terms*—Adversarial examples, Adversarial attacks, Deep learning (DL), Machine learning (ML), Neural networks (NN), Radio signals.**

I. Introduction

THE capability to automate the recognition of modulated radio signals is a highly useful endeavour. In the past, [1] the process of classifying radio signals and recognising their modulations involved creating specialised feature extractors for specific signal types and then using either analytical or statistical methods to generate decision boundaries within low-dimensional feature spaces [2]. This process required careful manual work. As a result, the introduction of deep learning (DL) models tasked to remove the tedious nature and needless error from humans have been implemented. Although the use of deep learning models for this purpose is relatively recent, they have previously been applied to various fields, excelling particularly in image and speech recognition [3].

Deep learning [4] is a subfield of machine learning (ML) that involves the use of neural networks (deep learning models) to analyse and interpret data. Made up of multiple layers of interconnected nodes, they learn and adapt as they are fed more data. Whilst proven to be effective and efficient in numerous fields, a complex system like DL models can be vulnerable to attacks that lead to the questioning of robustness of the network and the validity of the conclusions [5]. These adversarial attacks attempt to deceive the model by adding subtle, carefully crafted perturbations to the input data, causing incorrect predictions by disrupting the feature space [5]. These perturbations can mislead the model by directing its attention to irrelevant features or patterns in the data. For example, a perturbation might cause the model to focus on a specific pixel in an image that has no bearing on the image's content, leading to an erroneous prediction. In similar fashion, classifying radio signals inherits the same complication.

Previous research has explored techniques for classifying radio signal modulations through deep learning models [2] and detecting adversarial attacks on other classification problems but there has been limited work on detecting adversarial attacks on radio signal classification [6]. Furthermore, efforts by similar research to defend against adversarial attacks have primarily involved attempting to increase the robustness of deep learning models through improved training strategies like adversarial training [7], [8]. However, these approaches have only been successful in making it harder to create adversarial examples [9], rather than eliminating the problem altogether. An alternative and relatively unexplored strategy for defence against adversarial attacks is to differentiate between adversarial inputs and legitimate ones. As a result, there is still a need for more effective methods of protection against adversarial attacks on deep learning models.

Therefore, a new method of detecting adversarial attacks is proposed in this paper. Two highly effective white-box and black-box adversarial attacks are applied [10] to a convolutional neural network (CNN) [11] using synthetic radio signals generated from the GNU radio machine learning dataset [12]. Features are found after obtaining the Softmax layer outputs, which represent the probabilities assigned by the classifier to each trained class. Some consist of statistical measures, while others exploit a weakness identified. To assess the likelihood of an adversarial attack, the features are then fed as inputs into classifiers trained on the same analysis for both adversarial and clean inputs.

A. Adversarial Attacks on Deep-Learning Based Radio Signal Classification

As this paper's content is significantly influenced by [10], it is recommended to familiarise oneself with the attacks discussed in the paper, specifically algorithm 1 and algorithm 2. Although a brief overview of the attacks will be provided, this paper will concentrate solely on the detection methods for these attacks and the results they produce.

Algorithm 1 *White-Box Attack [10]*

Inputs:
- Input x and its label l_{true}
- The model $f(., \boldsymbol{\theta})$
- Desired perturbation accuracy ε_{acc}
- Maximum allowed perturbation norm p_{max}

Output: Adversarial perturbation of input $\boldsymbol{r}_{\mathbf{x}}$

Initialise: $\boldsymbol{\varepsilon} \leftarrow 0^{C\times 1}$
for class-index in range(C) **do**
  $\varepsilon_{max} \leftarrow p_{max}$, $\varepsilon_{min} \leftarrow 0$
  $\mathbf{r}_{norm}$ = $(\|\nabla_{\mathrm{x}} \boldsymbol{L}(\mathbf{x}, \boldsymbol{e}_{\textit{class-index}}\|_2)^{-1} \nabla_{\mathrm{x}} \boldsymbol{L}(\mathbf{x}, \boldsymbol{e}_{\textit{class-index}})$
  while $\varepsilon_{max} - \varepsilon_{min} > \varepsilon_{acc}$ **do**
    $\varepsilon_{ave} \leftarrow (\varepsilon_{max} + \varepsilon_{min})/2$
    $\mathbf{x}_{adv} \leftarrow \mathbf{x} - \varepsilon_{ave}\, \mathbf{r}_{norm}$
    if $(\mathbf{x}_{adv}) == l_{true}$ **then**
      $\varepsilon_{min} \leftarrow \varepsilon_{ave}$
    else
      $\varepsilon_{max} \leftarrow \varepsilon_{ave}$
    end if
  end while
  $[\varepsilon]_{\textit{class-index}} = \varepsilon_{max}$
end for
target-class = arg min ε and $\varepsilon^* = min\ \varepsilon$
$\mathbf{r}_{\mathbf{x}} = -\frac{\varepsilon^*}{\|\boldsymbol{\nabla}_{\mathrm{x}} \mathbf{L}(\mathbf{x}, \boldsymbol{e}_{\textit{target-class}})\|_2} \boldsymbol{\nabla}_{\mathrm{x}} \mathbf{L}(\mathbf{x}, \boldsymbol{e}_{\textit{target-class}})\|_2$

B. Brief Review of Adversarial Attacks

In general, adversarial attacks can be expressed as the following optimisation problem:

$$\begin{aligned} &\min_{r_{\mathrm{x}}} \|\mathrm{r_x}\|_p \\ s.t.\ \ &l(\mathrm{x}, \theta) \neq l(\mathrm{x} + \mathrm{r_x}, \theta),\ and \\ &\mathrm{x} + \mathrm{r_x} \in \mathcal{X} \end{aligned} \tag{1}$$

Where $||\cdot||_p$ denotes the l_p-norm and $l(\mathrm{x}, \theta)$ is the label given to an input x by a neural network with parameters θ. For wireless communications, the l_2-norm is a suitable choice as it considers the magnitude of each element of the perturbation vector.

In general, the optimisation problem (1) is difficult to solve, one reason being that it is often non-convex, making it complex to obtain a global optimal solution. As a result, suboptimal methods have been proposed. One method is named FGSM [7]. The attack is iterative and employs the loss function's gradient with respect to the input data to determine the direction in which the input must be perturbed to cause the model to make an erroneous prediction. The gradient provides information on how the loss changes as the input data is modified, and by adding a small perturbation to the input data in the gradient direction, the FGSM attack can boost the loss and lead the model to produce an incorrect prediction. FGSM finds the adversarial input $\mathrm{x}_a = \mathrm{x} + \mathrm{r_x}$ by maximising the loss $L(\theta, \mathrm{x}_a, y) \equiv L(\theta, f(\mathrm{x}_a, \theta), l(\mathrm{x}_a, \theta))$ subject to the l_∞-norm perturbation constraint $\mathrm{x}_a - \mathrm{x}_\infty \leq \epsilon$ with ϵ as the attack power.

Algorithm 2 *Black-Box Attack [10]*

Inputs:
- A random subset of input data points $\{\mathrm{x}_1, ..., \mathrm{x}_N\}$ and their corresponding labels
- The model f(., $\boldsymbol{\theta}$)
- Maximum allowed perturbation norm p_{max}

Output: UAP r

Evaluate $\mathrm{X}^{N\times p} = [n_{\mathrm{x}_1}, ..., n_{\mathrm{x}_N}]^T$
Compute the first principal direction X and denote it by $\mathrm{v}_1, i.e., \mathrm{X} = \mathrm{U}\Sigma\mathrm{V}^T\ and\ \mathrm{v}_1 = \mathrm{Ve}_1$
r = $\mathrm{p}_{max}\mathrm{v}_1$

$$\begin{aligned} \mathrm{x}_a &= \mathrm{x} + \epsilon \cdot sign(\nabla_{\mathrm{x}} L(\mathrm{x}, y)) \\ \mathrm{r_x} &= \epsilon \cdot sign(\nabla_{\mathrm{x}} L(\mathrm{x}, y)) \end{aligned} \tag{2}$$

Adversarial attacks can be categorised as either white-box or black-box [7], based on the attacker's level of information regarding the targeted model. A white-box attack occurs when the attacker possesses complete knowledge of the model, including its internal mechanisms and parameters. This is reflected in FGSM. Conversely, a black-box attack is carried out when the attacker lacks knowledge of the model's internal workings and only has access to its output.

II. Adversarial Detection Strategies

Adversarial defences are commonly presented as techniques to enhance robustness by utilising the concept of adversarial training and gradient masking [13]. Detection of adversarial attacks, however, is still an open problem [14]. The proposed detection method in this paper is based on a combination of ideas from four distinct groups of recent studies, as outlined in [15], [16]: *sample statistics*, *detector training*, *prediction inconsistency* and *feature-space detection*. Recent findings specific to radio signal modulation classification are introduced in the latter half of the section.

A. Detection of Adversarial Attacks in Classification

- *Sample Statistics*: Study [17] proposes using two statistical metrics, Maximum Mean Discrepancy (MMD) and Energy Distance (ED), to detect adversarial examples. The authors argue that these metrics can capture differences in the underlying distribution between clean and adversarial inputs. However, the paper also notes that while their approach can detect adversarial examples at the group level, it cannot identify individual adversarial examples with high confidence. Study [18] explores the application of kernel density estimates (KDE) in the subspace of the last hidden layer to detect adversarial examples. Although the method shows promise in detecting such samples in datasets, it can only detect adversarial examples far from the legitimate population and as such, still faces a similar challenge of detecting individual adversarial examples. While [15] argues that statistical approaches are unlikely to be effective in detecting adversarial examples due to their inherently

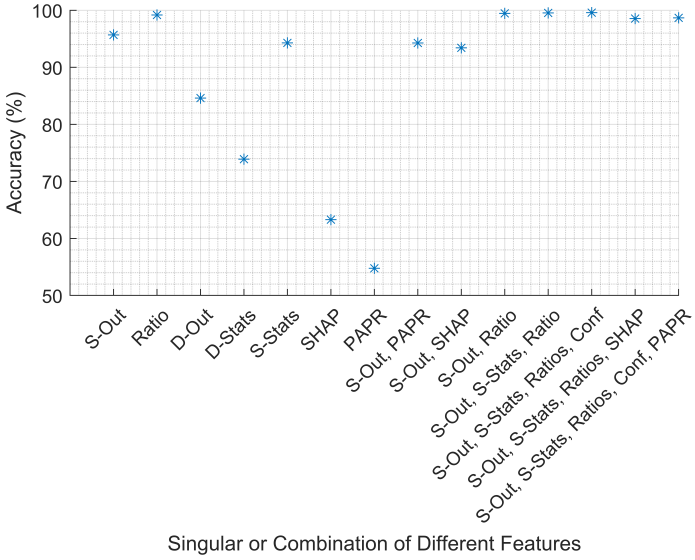

Fig. 1. Accuracy for singular or combinations of features tested on Algorithm 1 white-box attack.

Singular/Combination of Features	Accuracy (%)
Softmax Layer Output *(S-Out)*	95.68
Ratio	99.17
Dense Layer Output *(D-Out)*	84.61
Dense Layer Statistics *(D-Stats)*	73.90
Softmax Layer Statistics *(S-Stats)*	94.28
SHAP	63.30
PAPR	54.78
Softmax Layer Output, PAPR	94.27
Softmax Layer Output, SHAP	93.42
Softmax Layer Output, Ratio	99.46
S-Out, S-Stats, Ratio	99.53
S-Out, S-Stats, Ratio, Confidence Value *(Conf)*	99.58*
S-Out, S-Stats, Ratio, SHAP	98.54
S-Out, S-Stats, Ratio, Conf, PAPR	98.67

TABLE I
NUMERICAL RESULTS FROM SINGULAR OR COMBINATION OF FEATURES. (* HIGHEST ACCURACY)

unperceptive nature, this paper demonstrates the viability of statistical analysis. A less complex statistical approach than population-based statistics may be more effective in detecting adversarial examples in individual cases.

- *Detector Training*: This type of detection uses a similar concept to adversarial training. Notably [19] explores augmented classification networks with adversary detection networks. The authors propose a technique where a subnetwork branches off from the main network at some layer and produces an output indicating the probability of the input being adversarial. The detector is trained to classify network inputs into clean or adversarial examples using a binary classification dataset of clean and corresponding adversarial examples. The weights of the classification network are frozen and the detector is trained to minimise the cross-entropy of the output probability and the labels.
- *Prediction Inconsistency*: Similar to ensemble-based detection, this involves combining multiple models to measure the level of disagreement. In the case of [18], the authors utilised the training mode of dropout layers and generated numerous predictions for an input during testing. They observed that disagreements among submodel predictions are uncommon on clean inputs but prevalent on adversarial examples. Study [20] employs training an ensemble of N models that can accurately label clean examples while also disagreeing on randomly perturbed examples. During testing, the ensemble is used for both classification and adversarial detection, with the output label being the one that achieves the maximum agreement. If the agreement is too low, the example is labelled as adversarial.
- *Feature-Space Detection*: These techniques generally involve training detector models using intermediate or final layer outputs of NNs, based on the idea that the intermediate representations learned by the model during training are more robust than the final predictions [21]. The authors of [22] use the Softmax layer output values of the DNN to distinguish between adversarial and legitimate inputs. Specifically, they compare the maximum Softmax value (i.e., the predicted class probability) of an input image with the second highest Softmax value. If the difference between these two values is greater than a certain threshold, the input is classified as legitimate. However, if the difference is below the threshold, the input is classified as adversarial.

B. Detection of Adversarial Attacks within Radio Signal Classification

- *Local Intrinsic Dimensionality and Constellation Diagram*: A detection method focused on multifeatured fusion [23] employs techniques of local intrinsic dimensionality (LID) and constellation diagrams (CD). The LID values of normalised examples using min-max calculation are calculated for each layer of the deep neural network model. The difference in LID values between adversarial and clean examples becomes increasingly distinct as the layer depth increases. For comparing the distribution of constellation diagram points between clean and corresponding adversarial examples, the constellation diagram of certain modulation types is used. It is found that the distribution of constellation diagram points is more spread out in adversarial examples compared to normal examples. The density value and the distance value of the constellation diagram points serve as features for identifying adversarial examples. Combining both LID and CD features using a multifeatured detection method leads to superior detection outcomes
- *Peak-to-Average Power Ratio and Kolmogorov-Smirnov Test*: A different approach by [24] proposes a defence technique that utilises Peak-to-Average Power Ratio (PAPR) in conjunction with a two-sample Kolmogorov-Smirnov (KS) test to identify adversarial examples. PAPR is a metric that quantifies the ratio between peak power and average power of a signal. Adversarial attacks can modify signal amplitude, which can considerably impact

Class	Clean	Attacked
1*	**0.562556565**	**0.347342193**
2	4.72E-08	5.67E-07
3	4.49E-18	4.08E-18
4	0.00014286	2.84E-05
5	0.00650708	**0.347477347**
6	8.02E-05	0.003821981
7	3.54E-13	3.27E-16
8	0.003896232	0.000202076
9	0.001270114	3.36E-05
10	0.425535053	0.300854862
11	1.18E-05	0.000238964

TABLE II
SOFTMAX OUTPUT EXAMPLE OF AN INPUT DATA THAT IS LABELLED AS CLASS 1 FOR THE ALGORITHM 1 WHITE-BOX ATTACK. THE SOFTMAX OUTPUT REVEALS THE HIGHEST VALUE FOR THE CLEAN SAMPLE ALSO CORRESPONDS TO CLASS 1, WHILE THE SECOND-HIGHEST VALUE FOR THE PERTURBED SAMPLE IS ALSO ASSIGNED TO CLASS 1. (*DENOTES CORRECT CLASS, BOLD SIGNIFIES THE IMPORTANT VALUES)

PAPR values. The KS test is a statistical technique which calculates the confidence (p-value) that two sets of statistics are derived from the same distribution and hence, the PAPR values for the clean and adversarial examples are compared to evaluate if they belong to the same distribution. The results indicated that KS test with PAPR is effective in detecting adversarial examples, particularly for lower-order modulations such as BPSK and QPSK. However, the confidence of detecting adversarial examples decreases for higher-order modulations like 16QAM, which require more than 1024 samples to generate reliable PAPR statistics. The authors also cautioned that this defence method may not be robust in an over-the-air setting where channel conditions are unknown.

III. System Model

To guarantee the reproducibility of the research, both the model and dataset employed in this study are publicly accessible and is previously used in [10]. The synthetic dataset was produced by Deepsig using GNU radio and encompasses 11 different modulations [11]: 8PSK, AM-DSB, AM-SSB, BPSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK, and WBFM. For each modulation, 1000 data inputs were created for 20 various SNR levels that range from -20 dB to 18 dB with 2 dB increments. Each SNR level for every modulation includes a 2 × 128 vector that comprises 128 in-phase and 128 quadrature components of a signal. Using TensorFlow 2.0, the convolutional neural network, VT-CNN2, was trained with half of the data, while the remaining half was used for testing purposes.

Note: “VT-CNN2”, “the NN” and “the model” are all used interchangeably henceforth.

A. White-Box Adversarial Attack Algorithm [10]

Algorithm 1 builds upon the Fast Gradient Method (FGM) [7] attack and is designed to generate finely tuned adversarial perturbations. Similar to the FGSM, FGM attacks are relatively easy to implement and computationally efficient, but they

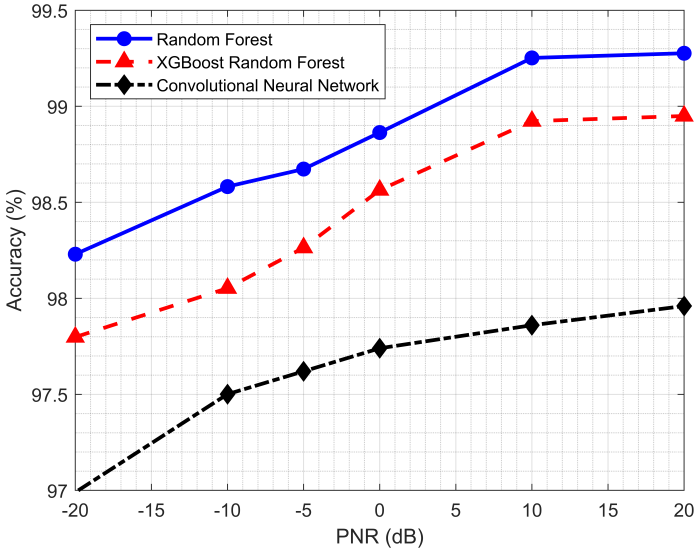

Fig. 2. Average accuracy of determining the Algorithm 1 white-box attack for varying levels of PNR and classification algorithms.

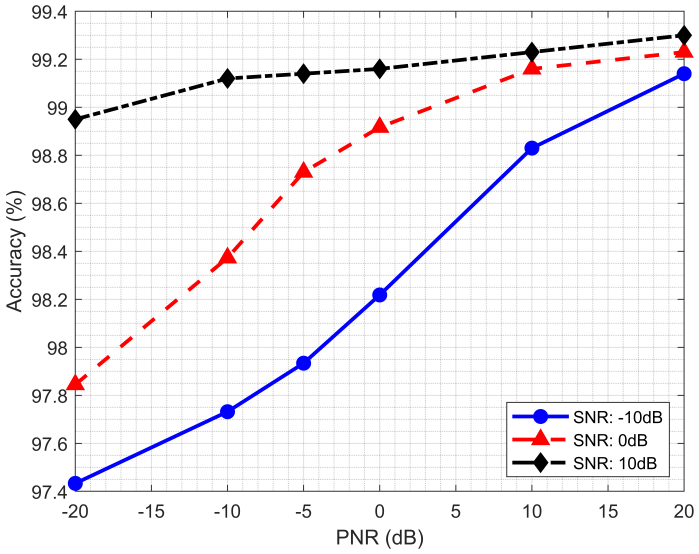

Fig. 3. Random Forest classifier accuracy of determining the Algorithm 1 white-box attack for varying levels of PNR at different SNR values.

tend to create relatively large and noticeable perturbations. Consequently, several methods have already been proposed to detect these attacks. Algorithm 1 overcomes two significant limitations of FGM. FGM sets the scaling factor, alpha, to expand the perturbation to the boundary of a norm ball surrounding the input. In contrast, Algorithm 1 employs a bisection search to determine the exact scaling factor necessary to guarantee misclassification while remaining within the perturbation norm constraint. Moreover, Algorithm 1 evaluates all feasible targeted attacks and selects the one requiring the least amount of perturbation to achieve misclassification. Despite these enhancements, Algorithm 1 still inherits the same computational efficiency as the FGM attack.

B. Universal Adversarial Perturbation Algorithm (Black-Box) [10]

Algorithm 2 is a method for creating a universal adversarial perturbation (UAP) that has a relatively low level of complex-

Class	Clean	Attacked
1	0.255118	2.98E-27
2	7.72E-10	0
3	3.63E-17	0
4	0.000932	1.81E-16
5	7.48E-07	0
6	3.93E-09	0
7	2.34E-08	0.012984
8	0.217256	0.18173
9	0.233255	0.805286
10	0.293439	2.10E-25
11	6.66E-10	0

TABLE III
SOFTMAX OUTPUT EXAMPLE FOR ALGORITHM 2 BLACK-BOX ATTACK. NOTABLY FOR ATTACKED THERE ARE SEVERAL ZERO VALUES.

UAP	PSR(dB)	-20	-10	0	10
ICM	ICM (%)	57.2	88.5	96.8	98.3
	CCM (%)	58.7	87.9	96.3	99.1
	MIX (%)	58.4	88.8	97.3	98.6
CCM	ICM (%)	62.1	89.4	96.6	98.5
	CCM (%)	62.1	90.3	97.2	99.2
	MIX (%)	60.8	89.9	95.1	98.4
MIX	ICM (%)	61.5	91.3	97.5	98.7
	CCM (%)	59.6	90.2	97.8	99.1
	ICM (%)	63.5	90.4	97.5	99.2

TABLE IV
ACCURACY FOR DETECTING THE ALGORITHM 2 BLACK-BOX ATTACK AT SNR = 10DB. DIFFERENT SETS OF INPUT DATA WERE USED TO GENERATE THE UAP AND IT WAS THEN APPLIED TO DIFFERENT GROUPS.

ity. This technique is based on principal component analysis (PCA) and is capable of deceiving a machine learning model with a high degree of certainty, regardless of the input data that the model is processing. To utilise this algorithm, a subset of input data points, a trained machine learning model, and a maximum allowable perturbation norm are required as input. For each input data, the algorithm computes the perturbation direction and these directions are combined and stacked into a matrix. The matrix is then decomposed using singular value decomposition (SVD) and the first principal direction is obtained. This principal direction represents the most significant variability in the perturbation directions. Multiple universal adversarial perturbations are computed, and the one that has the most significant potential (highest accuracy) to deceive the model is selected.

IV. PROPOSED METHOD OF DETECTION

A. Classification Algorithms and DNN

Initially, various classification algorithms were evaluated for their detection capability, such as k-nearest neighbour and decision tree. The top three performing algorithms were selected based on the analysis of their accuracy. Random Forest (RF) and XGBoost applied Random Forest were the two classification algorithms that achieved the highest accuracy. In addition to these, a simple deep CNN comprising several dense layers was also employed to showcase the simplicity of detection. Optimisation techniques (hyperparameter tuning) were also applied to the classification algorithms.

B. Features

Numerous features were explored and merged to determine the optimal combination that would yield the highest accuracy. Ultimately, the selected features that garnered the highest accuracy utilised the Softmax output. The Softmax layer outputs a vector of probabilities (the likelihood that it can be a certain class) for each class (modulation type) for every data input. By training the classifiers using features extracted from the Softmax output of clean inputs and inputs that have undergone an adversarial attack, the classifiers are able to determine whether a given input has been adversely affected. This approach enables the detection of adversarial attacks on specific inputs, thereby allowing for the identification of whether a perturbation has been introduced. Figure 1 sheds more light on the accuracy of several amalgamations.

Features Tested:

- *Dense Values*: Corresponds to the penultimate layer's output.
- *Dense Statistics*: The Dense layer's output is subjected to: Variance, Skewness and Kurtosis.
- *PAPR*: Refers to Peak-to-Average-Power Ratio previously mentioned in [24].
- *SHAP*: Shapely Additive Explanations [25]. SHAP signatures capture the contribution of each input feature to the model's output.

Features Chosen:

- *Softmax Values*
- *Ratio*: Refers to the exploit found in Algorithm 1, where for the majority of the cases the second highest Softmax value within the vector output often corresponded to the correct class (Table II). The ratio is the division between the second highest value and the highest value.
- *Softmax Statistics*: The Softmax outputs are subjected to: Variance, Skewness and Kurtosis.
- *Confidence Score*: The difference between the Softmax output's max values and the sum of the rest.

C. Approach and Experimental Scenarios

Both adversarial attacks were generated using the testing data for the VT-CNN2. More specifically, after using half of the data to train the VT-CNN2, the remaining half was used to test the model's accuracy and for our further analysis. The data inputs that were correctly classified by the model were then selected to undergo an adversarial attack. We also consider the alternative.

Hence, for each attack, two scenarios are examined: the impact of the attack on data inputs that the model cannot correctly classify, which will be referred to as ICM (incorrectly classified modulations), and the impact of the attack on data inputs that the model can classify correctly, denoted as CCM (correctly classified modulations).

For the white-box attack it is simple: if a data input that the model could not classify correctly (ICM) was used, the resulting output would remain relatively unchanged. Since

the white-box attack generates the minimum possible perturbation to cause misclassification of a given data input, the perturbation required to misclassify is theoretically null, as the misclassification occurs regardless of the attack. In practice, the attack does create a minute perturbation, however, this perturbation is deemed negligible as it does not alter the data considerably (signal characteristics such as in-phase and quadrature components and PAPR are all very closely identical) and, as a result, has no significant impact on the Softmax output. However, for CCM inputs, perturbations strong enough to misclassify were generated for each input.

Conversely, for the black-box attack, the UAP generated uses a random subset of data inputs. The perturbation that yields the highest misclassification accuracy from multiple random subsets of data inputs is chosen as the UAP to be applied to all data inputs. Although the real-world will use a random subset of data inputs that comprises of both CCM and ICM inputs to generate a UAP, all possibilities are examined in this paper.

UAPs are generated in three different scenarios: by solely using CCM inputs, solely using ICM inputs, and by randomly combining both ICM and CCM inputs. After the UAP from each group was generated, it was applied to three distinct types of data inputs: only CCM inputs, only ICM inputs, and a combination of both CCM and ICM inputs (MIX).

By conducting experiments with these scenarios, we aim to demonstrate the detection method's ability to identify perturbations that cause misclassification of inputs that were initially classified correctly, assess the effectiveness of the detection method when dealing with inputs that are already difficult for the neural network to classify accurately, and simulate real-world conditions by leveraging a combination of both CCM and ICM inputs.

The aim is to provide a more comprehensive understanding of the effects of the attack and the robustness of the detection method.

V. Results and Discussion

Features: The results illustrated in Figure 1 demonstrate that the ratio is the singular feature that yields the highest accuracy, followed by the Softmax layer output and the statistics applied to the Softmax layer outputs. On the other hand, PAPR, SHAP values and the penultimate layer outputs do not yield results that are anywhere close in accuracy.

Upon closer inspection of the PAPR for both clean and adversarial inputs, no noticeable changes were observed, besides limited variance. The perturbations generated by the white-box attack are small enough to not produce significant changes in the power. Moreover, the Dense layer output did not exhibit the same level of exploitation as the Softmax layer output and hence did not yield a high accuracy. The SHAP values were deemed inconclusive as the computational power required to produce these values was very high. As the number of classes and data input size increases, more SHAP values are needed to be evaluated. Further work is needed to optimise this feature. Ultimately, both attacks contained vulnerabilities in the Softmax outputs that could be exploited to detect adversarial attacks.

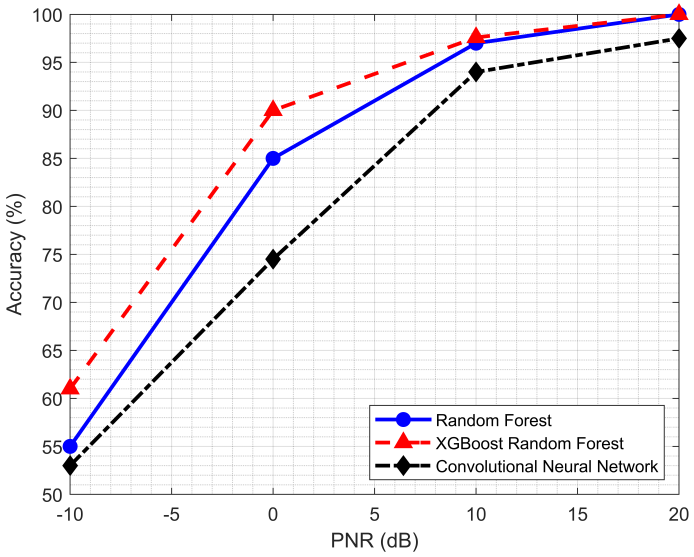

Fig. 4. Best performing classification algorithms and one neural network for varying PNR values at SNR = 10dB.

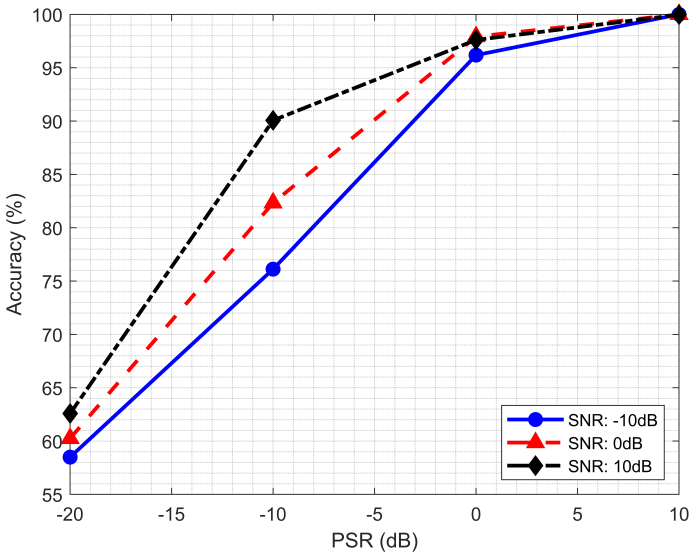

Fig. 5. XGBoost Random Forest accuracy of varying PSR values for various SNR values.

Algorithm 1 White-Box Attack: The results of the white-box attack demonstrate a high level of promise across different PNR ranges, with an average accuracy of approximately 98.2%. The detection accuracy ranges from a high of 99.2% to a low of 97.4% (as depicted in Figure 2 and Figure 3). The ratio feature is largely responsible for this accuracy, as subsequent features only contribute a marginal increase of 0.4% (Table I). Additionally, although the accuracy does improve with increasing PNR, the relationship is not linear and eventually saturates. The cause may be largely attributed to the optimisation methods used for the classification algorithms or the adversarial examples being situated closely to the decision boundaries.

Upon further examination, it becomes apparent that as the PNR increases, the majority of the data inputs are accurately classified as either clean or adversarial. However, for certain PNR values, some data inputs may be incorrectly classified, even if they are classified correctly for other PNR values. For

example, the correct classification of a data input as adversarial with a PNR value of 5dB does not guarantee the correct classification of the same data input with a PNR value of 10dB. While this observation is based on a small number of values, it suggests that additional tuning is necessary.

Algorithm 2 Black-Box Attack: Results for various scenarios are presented in Figure 4, Figure 5, and Table IV. The MIX-MIX case in Table IV corresponds to the real-world example illustrated in Figures 4 and 5. The black-box detection method, which employs the same features as the white-box detection method, achieves high detection accuracy for perturbations above PSR = 0dB. The XGBoost Random Forest algorithm proves its superiority by even providing reasonably reliable results at a low PSR of -10dB. The detection accuracies for the various scenarios are all relatively similar (Table IV). This is primarily due to the information presented in Table III, which shows the Softmax outputs for data inputs that have undergone the black-box attack. The results reveal that the Softmax outputs for adversely affected data inputs contain numerous zero values, indicating that the probability that it is a certain class is null. This pattern is noticeable across most input data that has undergone an attack, making it easily detectable. This detection method appears to show some robustness due to the small variations within Table IV.

Ultimately, the results suggest that the method is capable of identifying whether a perturbation has been added irrespective of whether the model can classify the data input correctly or not, and as such, can detect the occurrence of an attack.

Total Accuracy: To determine the total accuracy, let us assume this detection method has been implemented into a system in conjunction with the model to detect adversarial inputs. If we define the total accuracy as the accuracy of detecting a perturbation, then we have two accuracies for each attack.

In the white-box attack, perturbations are only added to data inputs that the model can predict accurately. Consequently, the total accuracy would rely on the accuracy of the model multiplied by the accuracy of the detection method.

On the other hand, for the black-box attack, perturbations are added regardless of the model's ability to classify the modulation type accurately. Therefore, the total accuracy would be the accuracy of the detection method alone.

The advantage of only detecting the presence of a perturbation is that even in scenarios where an adversarial attack has occurred, and the perturbation somehow manages to classify a previously misclassified input data to its correct class, this method can still detect the anomaly. This is because the focus of the approach is solely on detecting variations in the Softmax output, which has now been identified as being caused by the presence of a perturbation.

VI. Conclusion and Future Work

It has been demonstrated that by utilising the output of the final layer, the adversarial attacks generated by [10] can be detected effectively. Despite their effectiveness in generating adversarial attacks on radio signals, these algorithms forge subtle hints in the Softmax layer that can be exploited. The white-box attack detection method is highly effective in detecting attacks, irrespective of the power of the perturbation, with little effects shown from varying the PNR. Similarly, the same detection method applied to the black-box attack yields excellent results at higher PSRs and promising outcomes even at lower values, however, there is potential for additional improvement in achieving better accuracy at lower PSRs.

To further showcase the efficacy of feature-space detection methods, it would be beneficial to explore different setups, such as multi-antenna configurations, incorporating natural radio signals instead of synthetic data, and the inclusion of additional attacks. Additionally, testing the shift-invariant properties of UAPs highlighted in [10] to gauge whether this detection approach is valid for other NNs would also be insightful.

Ultimately, it has been demonstrated that hidden features within the layers can aid in identifying the happenings of an attack.

Acknowledgment

I would like to thank my supervisor, Dr Aissa Ikhlef, Associate Professor in the Department of Engineering at Durham University for his advice and guidance. I would also like to thank the authors of [10] Meysam Sadeghi and Erik G. Larsson, as my work of detecting adversarial attacks was heavily influenced by their paper.

References

[1] J. Chen, H. Cui, S. Miao, C. Wu, H. Zheng, S. Zheng, L. Huang, and Q. Xuan, "Fem: Feature extraction and mapping for radio modulation classification," *Phys. Commun.*, vol. 45, p. 101279, 2021.

[2] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[5] X. Yuan, P. He, Q. Zhu, R. Bha, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, p. 2805–2824, 2019.

[6] S. Kokalj-Filipovic, R. Miller, and G. Vanhoy, "Adversarial examples in rf deep learning: Detection and physical robustness," in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2019, pp. 1–5.

[7] S. J. S. C. Goodfellow, I.J., "Explaining and harnessing adversarial examples," *Proceedings of International Conference on Learning Representation (ICLR)*, p. 1–11, 2015.

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

[9] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," 04 2017, pp. 506–519.

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

[11] T. O'Shea and N. West, "Radio machine learning dataset generation with gnu radio," *Proceedings of the GNU Radio Conference*, vol. 1, no. 1, 2016. [Online]. Available: https://pubs.gnuradio.org/index.php/grcon/article/view/11

[12] Deepsig, "Radioml 2016.10a," https://www.deepsig.ai/datasets, released at the 6th Annual GNU Radio Conference.

[13] N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman, "Sok: Security and privacy in machine learning," in *2018 IEEE European Symposium on Security and Privacy (EuroSP)*, 2018, pp. 399–414.

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

[15] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *CoRR*, vol. abs/1704.01155, 2017. [Online]. Available: http://arxiv.org/abs/1704.01155

[16] N. Drenkow, N. Fendley, and P. Burlina, "Attack agnostic detection of adversarial examples via random subspace analysis," *CoRR*, vol. abs/2012.06405, 2020. [Online]. Available: https://arxiv.org/abs/2012.06405

[17] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. D. McDaniel, "On the (statistical) detection of adversarial examples," *CoRR*, vol. abs/1702.06280, 2017. [Online]. Available: http://arxiv.org/abs/1702.06280

[18] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," 2017.

[19] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," 2017.

[20] A. Bagnall, R. C. Bunescu, and G. Stewart, "Training ensembles to detect adversarial examples," *CoRR*, vol. abs/1712.04006, 2017. [Online]. Available: http://arxiv.org/abs/1712.04006

[21] F. Carrara, F. Falchi, R. Caldelli, R. F. G. Amato, and R. Beccarelli, "Detecting adversarial examples in deep neural networks," in *2017 16th International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2017, pp. 1–7.

[22] H. Kwon, Y. Kim, H. Yoon, and D. Choi, "Classification score approach for detecting adversarial example in deep neural network," *Multimedia Tools and Applications*, vol. 80, pp. 1–22, 03 2021.

[23] D. Xu, H. Yang, C. Gu, Z. Chen, Q. Xuan, and X. Yang, "Adversarial examples detection of radio signals based on multifeature fusion," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 12, pp. 3607–3611, 2021.

[24] S. Kokalj-Filipovic, R. Miller, and G. Vanhoy, "Adversarial examples in rf deep learning: Detection and physical robustness," in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2019, pp. 1–5.

[25] G. Fidel, R. Bitton, and A. Shabtai, "When explainability meets adversarial learning: Detecting adversarial examples using SHAP signatures," *CoRR*, vol. abs/1909.03418, 2019. [Online]. Available: http://arxiv.org/abs/1909.03418