Hybrid Automatic Modulation Classification for Increased Robustness under White-Box Adversarial Attacks

A van der Merwe[0009−0001−5032−7090] and ASJ Helberg[0000−0001−6833−5163]

Faculty of Engineering, North-West University, Potchefstroom, South Africa
arnold1.vandermerwe@gmail.com
albert.helberg@nwu.ac.za

Abstract. Automatic modulation classification (AMC) is an important function in wireless communication systems that is used to identify the modulation type of a signal without prior knowledge. AMC has historically been done using likelihood or feature-based methods, yet recent research has focused on using deep neural networks (DNNs) as they outperform the classical methods in challenging signal channel conditions. However, deep learning (DL) based classifiers are vulnerable to adversarial attacks that can significantly deteriorate their classification performance. This paper explores the robustness of different AMC classifiers to the white-box fast gradient method (FGM) and projected gradient descent (PGD) attacks under different perturbation-to-noise ratios (PNRs) and signal-to-noise ratios (SNRs) for a noisy signal channel. The investigated AMC classifiers consist of the quasi-hybrid likelihood ratio test (QHLRT), a k-nearest neighbour (KNN) that uses higher-order cumulants, and the parameter estimation and transformation-based CNN-GRU deep neural network (PET-CGDNN). The adversarial attacks are found to have limited transferability to the QHLRT and the KNN classifiers when scaled to be imperceptible against the noise of the signal channel. Based on this finding, we propose a hybrid classifier that uses the neural rejection technique through a support vector machine (SVM) that acts as a switching mechanism to decide whether to use the KNN or PET-CGDNN to classify the modulation type. The hybrid classifier demonstrates improved robustness against attacks, while benefiting from the good performance of the DNN.

Keywords: Modulation classification · Adversarial attacks · Deep neural networks · Likelihood ratio test · K-nearest neighbours

1 Introduction

Artificial intelligence (AI) has increasingly become a foundational technology in a wide range of industries, including telecommunications, through the implementation of deep neural networks (DNNs). These DNNs are used to automate network management, optimise resource allocation, and improve signal detection

and decoding [21]. For example, AI-based algorithms can dynamically adjust signal modulation and coding schemes based on real-time channel conditions and perform anomaly detection for network security [20, 23]. In 5G and emerging 6G systems, AI is also used in beamforming and massive multiple-input multiple-output (MIMO) configuration, which highlights the increasing importance of this technology [14, 21]. Wireless communications systems have been utilising AI increasingly in their most basic functions.

In a wireless communication system, a transmitter transfers data to a receiver using modulated radio frequencies [13, pp. 24–30]. A signal modulation technique is used to encode the information by changing the properties of a radio carrier, such as its phase, amplitude, or frequency. The modulated signal is transmitted over a radio channel to a receiver, where the real world environment through which it travels can apply various negative effects to the signal. Automatic modulation classification (AMC) is utilised in these communication systems to systematically identify a signal's modulation type without prior knowledge, enabling it to decode the signal and obtain the original transmitted data.

Historically, AMC was approached using statistics and hand-crafted features through classical machine learning algorithms such as support vector machines (SVMs), k-nearest neighbours (KNNs), and decision trees. The common classical AMC approaches can be grouped as feature-based or likelihood-based [19, 27].

In the likelihood-based category, AMC is framed as a hypothesis test that determines the modulation type by calculating the likelihood function of the received signal and applying a likelihood ratio test [4, 27]. In this approach, the method tests a pool of possible modulation types, and the modulation type that returns the highest likelihood is accepted as the modulation type used. This process is known as the maximum likelihood classifier [27]. Likelihood-based techniques depend on good signal channel knowledge, and when this is lacking, a method for estimating or treating unknown channel parameters is needed [4]. When channel parameters are unknown, the alternatives of average likelihood ratio test (ALRT), generalised likelihood ratio test (GLRT), or hybrid likelihood ratio test (HLRT) can be used [19, 27]. These alternative methods treat unknown parameters either as random variables with their own probability distribution in ALRT or as deterministic unknowns in GLRT [4]. The HLRT method is a combination of ALRT and GLRT where some parameters are treated as random variables and others as deterministic unknowns.

Feature-based methods are based on the extraction and analysis of signal characteristics that can be used to determine the modulation type [19, 27]. Common features used are instantaneous features, signal constellation features, higher-order cumulants, and the cyclic spectrum [19]. Higher-order cumulants are derived from signal moments, which describe the shape of the distribution of the modulated signal. These features are typically used with a classifier or other decision-making algorithm to distinguish the modulation type, such as a decision tree, KNN, or SVM. Feature-based AMC methods are less computationally complex than likelihood-based approaches but have less accurate classification when channel knowledge is available [27].

More recently, the focus has been on using deep learning (DL) for AMC as DNNs have demonstrated good classification accuracy for this task [8, 9, 22, 23]. The DNNs are trained on a dataset that is representative of the different signal modulation types with the expected signal channel effects present. Usually, the DNNs are left to automatically determine the most relevant characteristics to identify the modulation type by receiving the raw signal samples as input [23]. With this approach, the DNNs have shown significant adaptability and performance. DL architectures that are commonly used for AMC are convolutional neural networks (CNNs), recurrent neural network (RNN), and hybrid architectures that combine CNN and RNN techniques [23]. Furthermore, transformers have also been tested for AMC with significant success [3, 26]. These DL methods outperform classical techniques for AMC and excel in complex signal channel environments where classical approaches often fail.

However, using DL methods exposes the AMC process to vulnerability through adversarial attacks [1, 15, 25]. Adversarial attacks are small perturbations that are added to the input of a model that cause the model to misclassify the input. They are often referred to as evasion attacks. These attacks may be classified as white-box, gray-box, or black-box, based on the information known about the DL model. White-box attacks have complete knowledge of the model, its parameters, and training configuration, whilst gray-box attacks have partial knowledge, and black-box attacks have no available knowledge of the model [18].

In this paper, we investigate the robustness of various AMC methods against the fast gradient method (FGM) and projected gradient descent (PGD) adversarial attacks under white-box scenarios. We compare both classical and DL-based classifiers and evaluate the impact of attacks on their performance in a noisy signal channel. Furthermore, a hybrid classifier is proposed that combines techniques from DL and classical approaches through the use of a SVM to improve the robustness of the modulation classifier. The specific AMC classifiers that will be compared are as follows:

- Parameter estimation and transformation-based CNN-GRU deep neural network (PET-CGDNN) [22]
- KNN using higher-order cumulants [17]
- Quasi-hybrid likelihood ratio test (QHLRT) [4, 7]
- Hybrid classifier that combines PET-CGDNN and KNN with cumulants

These methods were selected as well-performing examples of the different classes of AMC in order to effectively investigate the effect that adversarial attacks have on their approaches.

2 Background

2.1 System Model

A simplified wireless communication system is shown in Figure 1. This diagram illustrates a transmitter that modulates an input signal using a specific modulation type and transmits it over a signal channel. The signal channel can apply

negative effects, such as noise and fading, to the signal, which causes the signal to be altered from the original modulated signal. The AMC classifier is required to be able to identify the modulation type used without any prior information using only the noisy signal. If the modulation type is successfully classified, the signal can be demodulated, allowing the receiver to extract the transmitted information. This simplified wireless system is used in this paper.

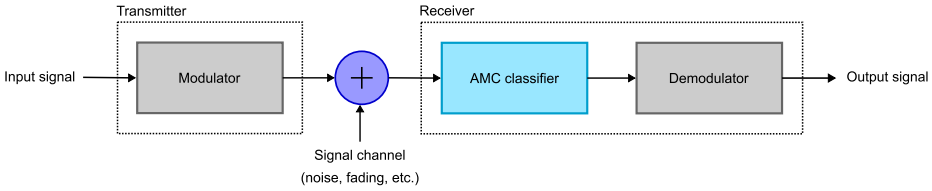


Fig. 1: Diagram showing simplified wireless communication system.

The equation for the received signal is given below, where $\mathbf{r}$ is the received signal vector such that $\mathbf{r} = [r_1, r_2, \ldots, r_K]^T$, where K is the length of the signal frame [4]. Furthermore, $\mathbf{s} = [s_1, s_2, \ldots, s_K]^T$ is the transmitted signal vector and $\mathbf{n} = [n_1, n_2, \ldots, n_K]^T$ is the signal channel noise vector. In this context, noise refers to unwanted random interference that corrupts a signal, which is often modelled as Gaussian-distributed to reflect its statistical properties in the real world. The variables α and φ represent the channel amplitude change and phase change, respectively, that is applied to the signal.

$$\mathbf{r} = \alpha e^{j\varphi}\mathbf{s} + \mathbf{n} \tag{1}$$

A commonly used metric that describes the quality of a signal is the signal-to-noise ratio (SNR) [13, pp. 275–276]. This metric quantifies the relationship between the power of the desired signal and the power of the background noise. A high SNR indicates a signal with less distortion to noise, which makes AMC easier. The SNR is commonly expressed in decibel (dB).

Two digital signal modulation techniques that are frequently used are phase-shift keying (PSK) and quadrature amplitude modulation (QAM) [13, pp. 426–449]. With PSK the phase of a carrier signal is varied, while with QAM the phase and amplitude is varied simultaneously to represent information. The modulation order M of the modulation type defines how many distinct symbols are available, with a higher order enabling a higher data transmission rate. Modulated signals are represented with in-phase (I) and quadrature (Q) components.

2.2 Related Work

Extensive research has led to new DL models for AMC [8, 9, 22, 23], where existing DL methods are tested in an AMC context to enhance classification performance. Hermawan *et al.* [8] proposed IC-AMCNet, a CNN-based architecture

with a focus on good classification performance with low latency for beyond 5G communications, making use of dropout layers and Guassian noise layers. In contrast to IC-AMCNet, Huynh-The *et al.* [9] proposed MCNet, a deeper CNN architecture with asymmetric convolutional blocks and skip connections that focuses on using spatiotemporal features to achieve good performance. Zhang *et al.* [22] introduced the PET-CGDNN model, which is a lightweight model with a low number of model parameters and that uses a parameter estimation and transformation module to compensate for phase offsets before feature extraction by CNN and gated recurrent unit (GRU) layers.

In [25], the vulnerability of DL-based signal classifiers, including protocol and modulation classifiers, to adversarial attacks was investigated. The study evaluated CNNs and RNNs against white-box, black-box and gray-box attacks. Using the attack methods fast gradient sign method (FGSM), PGD, and DeepFool they demonstrated that even low power, unsynchronised, and intermittent adversarial perturbations can significantly reduce the classification performance of the DL models. The attacks were also found to be more effective than traditional jamming attacks where only additive white Gaussian noise (AWGN) noise is transmitted because less transmit power is required. These adversarial perturbations are superimposed on the original signal and can be considered to be imperceptible since their low power is below the noise power. Furthermore, they proposed a two-step adversarial training mechanism which improved classifier robustness.

Manoj *et al.* [11] tested adversarial defense techniques on a CNN for AMC. These defense techniques aimed to improve the robustness of the CNN against adversarial attacks. Three defence techniques were used, namely randomised smoothing, hybrid projected gradient descent adversarial training, and fast adversarial training. The attacks consisted of white-box and black-box attacks such as FGSM, FGM, PGD, DeepFool and universal adversarial perturbation (UAP). They found that the defense techniques make the CNN more robust against attacks, particularly against black-box attacks. In a white-box attack scenario, the defense techniques enhance robustness to a limited extent. Minimal performance gains were observed if the attack power approached the level of noise power levels. These findings highlight that some vulnerability to adversarial attacks remains in the case that defense techniques are used.

Zhang *et al.* [24] proposed a countermeasure to defend DL-based classifiers against adversarial attacks. They use the neural rejection technique that uses the final feature layer activations of a CNN classifier with a one-vs-all SVM to detect and reject adversarial input. Furthermore, they use label smoothing (LS) and Gaussian noise augmentation (GNA) to form the LS-GNA based neural rejection system. This combination increases robustness against FGM adversarial attacks, which forces attacks to use more power in order to be successful. The countermeasure technique that they introduced forms the basis of the hybrid classifier that we propose in this paper.

2.3 Adversarial Attacks

Fast Gradient Method One of the simplest and most effective white-box adversarial attacks is the FGSM attack [6]. This attack simply adds a perturbation to the input of a model which is in the direction of the gradient of the DNN's loss function. By taking a step against the gradient, the model is more uncertain of which class the input is, and therefore the input can move closer to the decision boundary separating different classes. The added perturbation is scaled by ϵ where a higher value creates a more significant perturbation and has a higher likelihood of causing the model to misclassify the input.

In this paper, the FGM attack is used which is a variant of the FGSM attack [5]. The FGM attack uses the L_2 norm instead of the L_∞ norm as in the FGSM attack. Mathematically, the FGM can be described as in the following equation, where $\boldsymbol{x}$ is the input, ϵ is the scaling factor, L is the loss function of the model, $\boldsymbol{\theta}$ are the parameters of the model and y is the true class label [5]. Furthermore, $\nabla_{\boldsymbol{x}}$ gives the gradient with respect to the input.

$$\boldsymbol{x}_{adversarial} = \boldsymbol{x} + \epsilon \cdot \frac{\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \boldsymbol{x}, y)}{\|\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \boldsymbol{x}, y)\|_2} \tag{2}$$

Projected Gradient Descent The PGD attack is another frequently used white-box adversarial attack in AMC. This attack can be seen as an extension of the FGM attack, where instead of only adding a fixed step against the model gradient to the input, multiple steps are taken against the model's gradient [10]. The overall size of the adversarial attack is still restricted by the scaling factor ϵ. The PGD attack is more sophisticated than the FGM and FGSM attacks due to its multistep nature which enables it to be more effective in causing the model to misclassify the input.

The equation for the PGD attack is given below when using the L_2 norm. In this equation, $\boldsymbol{x}^{t+1}$ is the new input with the added perturbation, $\boldsymbol{x}^t$ is the current input with possible perturbation already added, $\prod_{\boldsymbol{x}+S_\epsilon}$ is the projection operator that constrains the attack to the perturbation constraint set S_ϵ, α is the attack step size, and the rest of the variables are as in Equation 2 [10]. The projection operator can be visualised as a L_2 norm ball centred around the original input $\boldsymbol{x}$ with a radius of ϵ. Typically, when setting up the attack the number of steps that will be taken is already defined.

$$\boldsymbol{x}^{t+1} = \prod_{\boldsymbol{x}+S_\epsilon} \left(\boldsymbol{x}^t + \alpha \cdot \frac{\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \boldsymbol{x}, y)}{\|\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \boldsymbol{x}, y)\|_2} \right) \tag{3}$$

Perturbation-to-noise Ratio To relate adversarial attacks to an AMC context, the metrics perturbation-to-noise ratio (PNR) and perturbation-to-signal ratio (PSR) are used [15]. Similar to the SNR, the PNR metric describes the ratio of the power of the perturbation to the noise power. The same concept applies to the PSR. The relationship between SNR, PNR and PSR is provided below [15].

$$\text{PNR [dB]} = \text{PSR [dB]} + \text{SNR [dB]} \tag{4}$$

A critical aspect of the PNR is that an adversarial attack can be considered imperceptible if the PNR is at 0 dB or below. At these values of PNR the noise is at the same power level or higher, and thus the attack is obscured by the noise.

The FGM and PGD attack strengths are scaled using ϵ. The required ϵ for a specific PNR can be calculated using the following equation if the L_2 norm is used [24]. In this equation $\boldsymbol{x}$ is the received complex signal consisting of I and Q components.

$$\epsilon = \sqrt{\text{PNR} \cdot \|\boldsymbol{x}\|_2^2 / (\text{SNR} + 1)} \tag{5}$$

2.4 Classifier Models

PET-CGDNN The PET-CGDNN model is a lightweight and efficient hybrid DL model for AMC [22]. It uses domain knowledge through a phase parameter estimator and a transformation module to mitigate phase offset distortions prior to classification. The model combines CNN layers for spatial feature extraction and a GRU layer for temporal feature extraction, achieving good classification accuracy with relatively few parameters. The architecture for the PET-CGDNN model is provided in Figure 2.

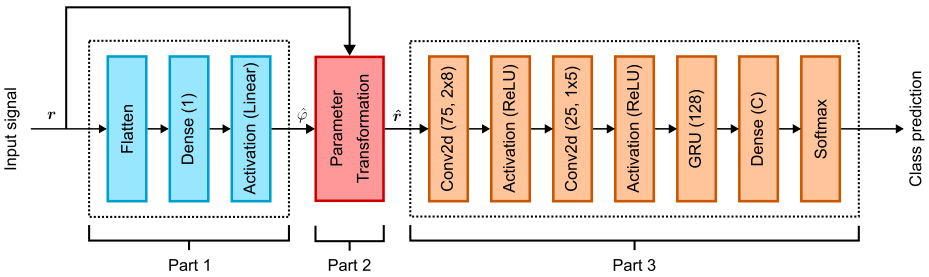


Fig. 2: PET-CGDNN architecture.
Source: Adapted from [22]

There are three distinct parts to the model architecture [22]. Part 1 contains the phase parameter estimator, which is designed to estimate the phase offset applied to the received signal through the signal channel. This is achieved by making use of a layer that flattens the two-dimensional input vector and a fully connected layer with a linear activation function. The output of this set of layers is an estimated phase offset $\hat{\varphi}$.

The phase offset is corrected through the parametric inverse transformation in Part 2, as given in Equation 6, where the complex signal received $\boldsymbol{r}$ has an

estimated phase offset $\hat{\varphi}$ [22]. The $\Re$ operator gives the real part of the complex signal, while the $\Im$ operator gives the imaginary part.

$$\hat{r}[n] = r[n]e^{-j\hat{\varphi}} = \begin{bmatrix} \Re\{r[n]\}\cos\hat{\varphi} + \Im\{r[n]\}\sin\hat{\varphi} \\ \Im\{r[n]\}\cos\hat{\varphi} - \Re\{r[n]\}\sin\hat{\varphi} \end{bmatrix} \tag{6}$$

Part 3 contains the feature extraction and classification required for AMC. Firstly, spatial feature extraction is done through two consecutive 2D convolutional layers, where the first layer applies 75 filters with a kernel size of (8×2) and the second layer applies 25 filters with a kernel size of (5×1). Each convolutional layer is followed with the rectified linear unit (ReLU) activation function. After extraction of spatial features, the temporal features are extracted through a GRU layer with a size of 128. A fully connected layer with the number of possible modulation types as the number of nodes is used after the GRU layer. The final output is a softmax layer that obtains the predicted modulation type.

QHLRT The QHLRT is a likelihood-based approach to AMC which uses the statistical likelihood of each modulation type to determine the used modulation type [4, 7]. It is a more computationally efficient variant of the HLRT, which uses maximum likelihood (ML) estimates of unknown signal channel parameters. The QHLRT uses method-of-moments (MoM) estimators for the channel amplitude, phase, and noise, thus significantly reducing computational complexity while having similar performance to the HLRT. The likelihood function in QHLRT is computed by averaging over unknown constellation points and substituting MoM estimates of the unknown parameters into the expression. The QHLRT strikes a balance between classification accuracy and implementation complexity.

Moments are used for the QHLRT to determine unknown channel parameters. M_{pq} denotes the (p, q)-th order moment, computed as given below, where $\boldsymbol{r}$ is the complex signal, $\boldsymbol{r}^*$ is the complex conjugate of $\boldsymbol{r}$ and $E[\cdot]$ is the statistical expected or average value.

$$M_{pq} = E[\boldsymbol{r}^{p-q}\boldsymbol{r}^{*q}] \tag{7}$$

The parameter $b^{(i)}$ is used in the estimate of the channel parameters and can be calculated as follows, where $s_k^{(i)}$ are the noiseless signal constellation points of a specific modulation type.

$$b^{(i)} = \frac{E[|s_k^{(i)}|^4]}{\left(E[|s_k^{(i)}|^2]\right)^2} \tag{8}$$

The parameter $b^{(i)}$ for the PSK and QAM modulation types used in this paper is provided in Table 1. Note these values are specific to the modulation order and constellation shape, thus different configurations of the same order QAM can result in different $b^{(i)}$ values. The provided values reflect MATLAB R2023b's default configuration for the different modulation types.

Table 1: Parameter $b^{(i)}$ for different modulation orders of PSK and QAM.

Modulation type	**Value for** $b(i)$
M-PSK	1
8-QAM	1.444
16-QAM	1.32
32-QAM	1.31
64-QAM	1.381
128-QAM	1.342

The MoM estimators for the channel amplitude α, phase φ and noise N are provided in the following equations, where the estimated moments $\hat{M}_{pq}$ are primarily used alongside the modulation order M_i [4]. These estimates are calculated for each modulation type. The channel phase equation used is dependent on whether PSK or QAM is being used for the likelihood function. The operator arg gives the angle between the real and imaginary axes of a complex number.

$$\hat{\alpha}^{(i)} = \left(\frac{\hat{M}_{42} - 2\hat{M}_{21}^2}{b^{(i)} - 2} \right)^{1/4} \left(E[|s_k^{(i)}|^2] \right)^{-1/2} \tag{9}$$

$$\hat{N}^{(i)} = \hat{M}_{21} - \left(\frac{\hat{M}_{42} - 2\hat{M}_{21}^2}{b^{(i)} - 2} \right)^{1/2} \tag{10}$$

$$\hat{\phi}^{(i)}_{\text{M-PSK}} = M_i^{-1} \arg \left(\sum_{k=1}^{K} r_k^{M_i} \right) \tag{11}$$

$$\hat{\phi}^{(i)}_{\text{M-QAM}} = 4^{-1} \arg \left(-\sum_{k=1}^{K} r_k^4 \right) \tag{12}$$

Using MoM estimates for the channel parameters, the likelihood of the received signal frame corresponding to a specific modulation type is expressed in Equation 13 [4]. In this equation, the average difference between each received signal symbol r_k and the signal constellation points s_m for the specific modulation type is used as a basis to determine the likelihood. The likelihood $\text{LF}^{(i)}$ of each modulation type is calculated for the pool of possible modulation types. The channel parameters are assumed to be constant over the interval that the signal frame is received. The modulation type that returns the highest likelihood is accepted as the identified modulation type.

$$\text{LF}^{(i)}(\mathbf{r}) = \prod_{k=1}^{K} \frac{1}{M_i} \sum_{m=1}^{M_i} \frac{1}{\pi \hat{N}^{(i)}} \exp \left(-\frac{1}{\hat{N}^{(i)}} \left| r_k - \hat{\alpha} e^{j\hat{\phi}} s_m \right|^2 \right) \tag{13}$$

For computational purposes and to improve numerical stability, Equation 13 is transformed to the following log-likelihood format. Logarithms are strictly increasing functions, and thus maximising the log-likelihood function is equivalent to maximising the likelihood function in Equation 13.

$$\log\left(\mathrm{LF}^{(i)}(\mathbf{r})\right) = \sum_{k=1}^{K} \log\left(\sum_{m=1}^{M_i} \exp\left(-\frac{1}{\hat{N}^{(i)}}\left|r_k - \hat{\alpha}e^{j\hat{\phi}}s_m\right|^2\right)\right) - \log\left(M_i \pi \hat{N}^{(i)}\right) \tag{14}$$

KNN with Cumulants A classical approach to AMC is using a KNN with handcrafted features to uniquely recognise various modulation types [27]. A KNN is a supervised machine learning algorithm that operates on the principle of instance-based learning. Classification is performed by identifying the k training samples closest to a given input based on a distance metric, which is typically Euclidean distance. The predicted class is determined by a majority vote among the k nearest neighbours. The performance of a KNN is dependent on the features that it uses for classification.

In the context of AMC, higher-order cumulants are frequently used as discriminative features for different signal modulation types [27]. These cumulants are sensitive to the amplitude and phase characteristics that are distinct between the different modulation types, especially for PSK and QAM. By arranging the cumulants into a feature vector, it can be used by an algorithm such as KNN for classification. Cumulants are derived using moments, where the equation for a moment was provided earlier in Equation 7. For the sake of brevity, the equations for the higher-order cumulants are not provided in this paper, but they can be found in [17].

3 Methodology

3.1 Hybrid Classifier Design

We propose a hybrid classification approach that combines the strengths of both DL and classical approaches to AMC to improve the robustness of AMC under adversarial conditions, While DNNs typically have better classification performance on signals without attacks due to their ability to learn complex features, they are vulnerable to adversarial perturbations. In contrast, classical approaches to AMC such as the QHLRT and KNN with higher-order cumulants have greater robustness to attacks, although with lower performance on signals without adversarial perturbations. This observation forms the basis for the design of the hybrid classifier and is further explored in Section 4.

We take advantage of both approaches by introducing a hybrid classifier that uses a SVM as a switching mechanism to decide whether to use the classical classifier or the DL classifier. The SVM is trained to distinguish whether the input to the classifier is adversarial or not, in an approach known as the neural rejection technique [24]. The hybrid classifier takes an input into the DNN, where the last feature layer's output serves as input for the one-vs-all SVM classifier. The SVM generates decision scores for all the possible classes using this input. Decision scores below a specified threshold Θ are seen as adversarial. This

approach is based on the concept that the outputs of neurons in the DNN become larger during propagation over the layers when the DNN has an adversarial input [24]. The SVM's decision routes the input to either the DNN in the case where the input is not attacked or to the classical classifier if the input is adversarial.

3.2 Dataset

MATLAB R2023b [16] was used to create a synthetic dataset simulating an AWGN channel for training and testing the various chosen AMC classifiers. The AWGN channel is a mathematically well-described channel that is often used in telecommunications for benchmarking. Signal frames of the modulation types (i.e. the classification classes) for BPSK, QPSK, 8-PSK, 8-QAM, 16-QAM, 32-QAM, 64-QAM and 128-QAM were generated. Unit average power was used for the QAM modulated signal frames. The SNR for the signal frames varied from -30 to 30 dB in increments of 2 dB, where for each modulation type there were 2000 frames at each SNR and each frame consisted of 2048 I/Q symbols, with the input shape of 2×2048. The frame length was chosen to be sufficiently large, ensuring that the AMC methods were constrained by classification capabilities rather than frame length. The generated frames were divided into a training set, a validation set, and a test set with a ratio of 6:2:2, respectively, where the different modulation types were equally divided to maintain class balance. A reduced training set, comprising a third of the original, was selected for the training of the hybrid classifier SVM.

3.3 Training Configuration

PET-CGDNN The PET-CGDNN model was implemented using the Keras framework with TensorFlow to train and evaluate the model. The full training set and validation set were used to train and hyperparameter tune the model. An Adam optimiser was used along with a learning rate scheduler which reduced the learning rate if a plateau occurred on the reported validation loss. The maximum number of epochs was constrained to 500 with early stopping if the model did not improve on its lowest validation loss after 30 epochs. The epoch which had the lowest validation loss was saved as the final best model.

Optuna, using the Bayesian optimisation tree-structured Parzen estimator algorithm, was utilized for hyperparameter tuning to determine the best parameters for the AWGN dataset [2]. Specifically, the model training parameters of the initial learning rate, batch size, learning rate scheduler's patience and the learning rate scheduler's factor were tuned. The trial that returned the lowest loss on the validation set was selected to have the best parameters. The results of the hyperparameter tuning after 40 trials are provided in Table 2.

KNN with Cumulants The KNN was set up to use 5 neighbours ($K = 5$) for classification. The full training set regardless of SNR was used to construct the feature space for the KNN. Specifically, the higher-order cumulants C_{40}, C_{41}, C_{42}, C_{60}, C_{61}, C_{62}, and C_{63} were used as the discriminative features.

Table 2: Hyperparameter tuning results for PET-CGDNN

Hyperparameter	**Search space boundaries**	**Optimised value**
Initial learning rate	$[1 \times 10^{-4}, 1 \times 10^{-1}]$	3.04×10^{-4}
Batch size	$[32, 512]$	438
Reduce learning rate patience	$[3, 20]$	8
Reduce learning rate factor	$[0.01, 0.9]$	0.8074

Table 3: Hyperparameter tuning results for SVM of hybrid classifier

Hyperparameter	**Search space**	**Optimised value**
C	$0.1, 1, 10, 100$	10
Gamma	$\text{scale}, 0.001, 0.01, 0.1, 1$	0.01

Hybrid Classifier The hybrid classifier makes use of a one-vs-all SVM classifier, which is defined with the parameters C and gamma. The C parameter is a regularisation parameter that controls the trade-off between a smooth decision boundary and classification error, while the gamma parameter controls the influence that a single sample in the training set has on the decision boundary. Furthermore, the radial basis function (RBF) was used as the kernel for the SVM. Grid search with three-fold cross-validation set was used to find the optimal hyperparameters for the SVM using the reduced training set. Note that the SVM is solely trained on signals without adversarial perturbations. Table 3 contains the results of the hyperparameter tuning. The value of "scale" means that the gamma parameter was automatically calculated as 1/(number of features $\times$ variance of input).

The SVM threshold Θ was set based on rejecting a specific percentage of training samples. The value of this percentage can be selected to use the classical classifier or the DL classifier more frequently, where a higher percentile causes the classical classifier to be used more often as more samples would be seen as adversarial. This threshold Θ was set so that 30% of the training samples were seen as adversarial. The investigation of the different percentiles and the influence they have on the performance of the hybrid classifier is provided in Section 4.

3.4 Generation of adversarial attacks

Using the trained PET-CGDNN model, white-box attacks were generated on the test set and validation set using the attack methods FGM and PGD. Equation 5 was used to scale the attack strengths for specific PNR values, which constrained the attack methods to use the L_2 norm. The PNR was varied from -40 to 0 dB in increments of 2 dB for all of the SNRs in the test and validation set. For the PGD attack, the attack step size was set to be 0.1 times the required epsilon for the specified PNR with a maximum number of 100 iterations. The adversarial robustness toolbox (ART) was used to create the adversarial attacks in Python [12].

4 Results & Discussion

Each classifier was evaluated three times on the test set using different configurations, measuring classification accuracy for each SNR and PNR. The classifiers were tested on the test set without attacks, the test set attacked with FGM attacks, and the test set attacked with PGD attacks.

Figure 3 provides the classification accuracy on the test set without attacks and on the FGM attacked test set when the PNR is at 0 dB, that is the perturbation power is at the same level as the noise power. It is evident that the DL-based classifier, the PET-CGDNN, outperforms the classical classifiers of QHLRT and KNN on the test set without any adversarial perturbations. Furthermore, the KNN outperforms the QHLRT as the QHLRT struggles with classifying nested QAM constellations, for example between 64-QAM and 16-QAM, at higher SNRs. Adding the FGM attacks affects the accuracy of the PET-CGDNN the most, while the KNN and QHLRT are largely unaffected. The transferability of the FGM attacks to the features that the QHLRT and KNN utilise is low.

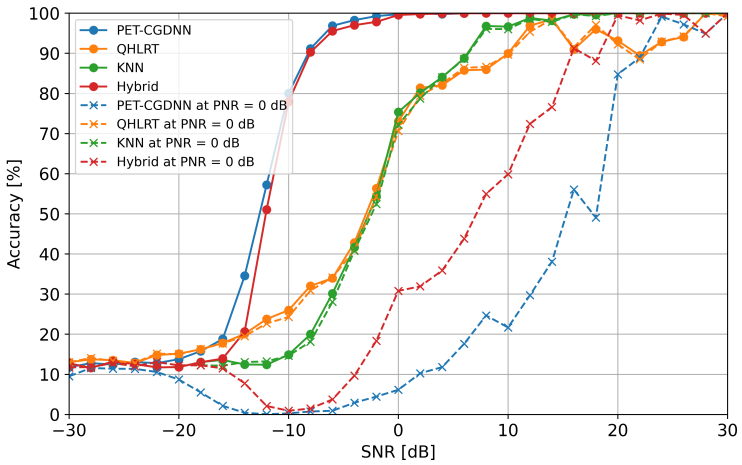
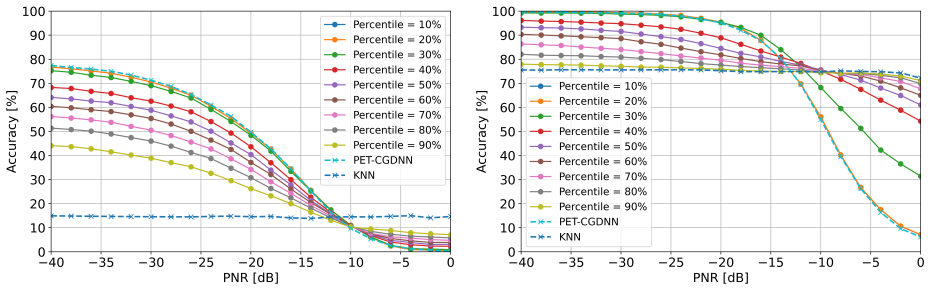


Fig. 3: Classification accuracy of different classifiers across SNR on FGM attacked signals and signals without attacks.

Figure 3 shows that the DL-based classifier performs better in the case of signals without FGM perturbations, but when perturbations are added, its performance decreases significantly. This led to the idea of a hybrid classifier that can benefit from the advantages of both approaches. The results indicate that KNN's classification performance is more stable than that of the QHLRT, therefore it was selected to be part of the hybrid classifier.

The hybrid classifier makes use of a SVM for its switching mechanism to decide whether to use the KNN or PET-CGDNN for classification. The threshold

Θ for the SVM is set for different percentiles of training samples that are classified as adversarial. The hybrid classifier was tested on the FGM attacked validation set to find the optimal percentile to choose the threshold of the SVM. Figure 4 shows the performance of the hybrid classifier when different percentiles are used for a SNR of -10 dB and 0 dB as the PNR varies. The hybrid classifier leans more towards using the KNN as the percentile of samples considered adversarial by the SVM increases, while with low percentiles it leans towards the PET-CGDNN.



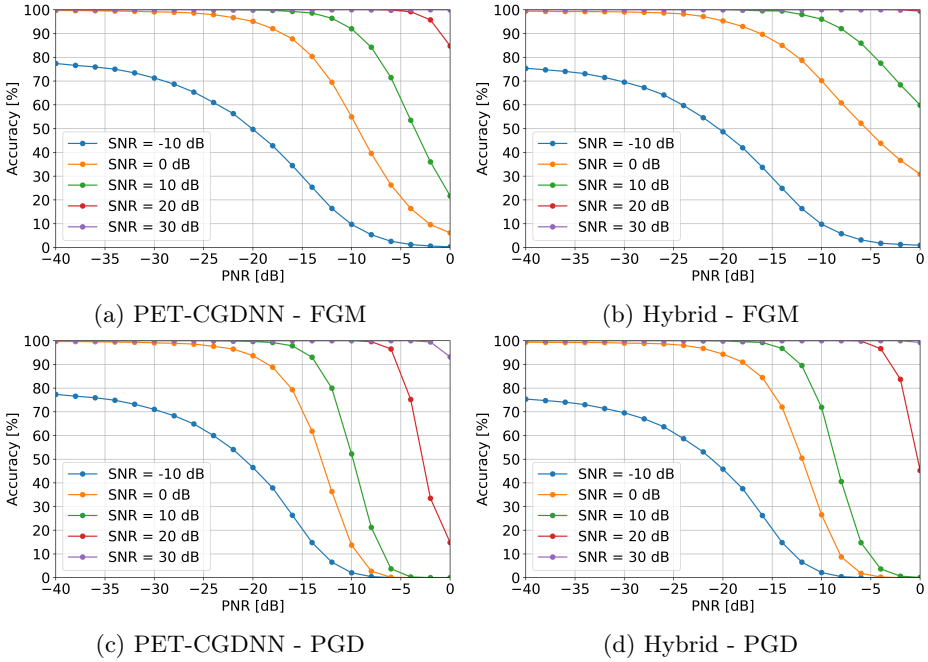
(a) SNR = -10 dB (b) SNR = 0 dB

Fig. 4: Classification accuracy using thresholds from specific percentiles for hybrid classifier on FGM attacked signals.

The best percentile to determine the threshold for the hybrid classifier was determined using F1 scores for the different configurations of both the PGD and the FGM attacks using the attacked validation sets. The score was calculated over all PNRs for the SNRs of -20, -10, 0, 10, and 20 dB using the percentiles from 10% to 90% in increments of 10%. In general, a higher SNR results in better F1 scores for higher percentiles, as the KNN demonstrates better performance. The only exception is at -20 dB SNR as at that point the random guessing from the KNN outperforms the PET-CGDNN. The best percentile would increase as the SNR increases. The percentile of 30% was selected for the hybrid classifier.

Figure 3 demonstrates the hybrid classifier's increased robustness to the FGM attack versus the PET-CGDNN. In the case where no FGM attacks are present, the hybrid classifier has almost the same performance as PET-CGDNN.

Figure 5 shows the PET-CGDNN and hybrid classifier as the PNR varies for the PGD attacks and the FGM attacks. The KNN and QHLRT is not shown as they show little change as the PNR varies. As the PNR increases, both the hybrid classifier and PET-CGDNN model's classification accuracy decreases. However, analysing the different fixed SNR lines for both classifiers shows that the hybrid classifier is less affected than the PET-CGDNN. It is evident that the PGD attack is more effective in reducing classifier performance than the FGM attack since the PGD takes multiple steps when generating the attack while the FGM only takes one step. These findings show that the hybrid classifier is more robust



(a) PET-CGDNN - FGM

(b) Hybrid - FGM

(c) PET-CGDNN - PGD

(d) Hybrid - PGD

Fig. 5: Classification accuracy of PET-CGDNN and hybrid classifier for specific SNRs and varying PNRs on attacked signals.

than PET-CGDNN and retains its performance on signals without adversarial attacks.

5 Conclusion

The PET-CGDNN model was trained on an AWGN dataset and used to generate white-box FGM and PGD attacks with varying PNRs. The QHLRT, KNN with higher-order cumulants, and PET-CGDNN was evaluated on signals without attacks, FGM attacked signals, and PGD attacked signals. Attacks using PGD and FGM were found to have limited transferability to the QHLRT and the KNN, when the PNR is at 0 dB or below. By integrating the KNN with the PET-CGDNN, a hybrid classifier was developed that uses a SVM as a selection mechanism. The hybrid classifier demonstrates strong performance on signals without adversarial perturbations and increased robustness to FGM and PGD attacks compared to the PET-CGDNN.

Acknowledgments. The authors gratefully acknowledge the financial support of this study by the Telkom Centre of Excellence (CoE) at the North-West University.

References

1. Adesina, D., Hsieh, C.C., Sagduyu, Y.E., Qian, L.: Adversarial Machine Learning in Wireless Communications Using RF Data: A Review. IEEE Communications Surveys & Tutorials **25**(1), 77–100 (2023). https://doi.org/10.1109/COMST.2022.3205184
2. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A Next-Generation Hyperparameter Optimization Framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2019)
3. Cai, J., Gan, F., Cao, X., Liu, W.: Signal Modulation Classification Based on the Transformer Network. IEEE Transactions on Cognitive Communications and Networking **8**(3), 1348–1357 (Sep 2022). https://doi.org/10.1109/TCCN.2022.3176640
4. Dobre, O.A., Hameed, F.: Likelihood-Based Algorithms for Linear Digital Modulation Classification in Fading Channels. In: 2006 Canadian Conference on Electrical and Computer Engineering. pp. 1347–1350 (May 2006). https://doi.org/10.1109/CCECE.2006.277525
5. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting Adversarial Attacks with Momentum. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9185–9193 (Jun 2018). https://doi.org/10.1109/CVPR.2018.00957
6. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples (Mar 2015). https://doi.org/10.48550/arXiv.1412.6572
7. Hameed, F., Dobre, O.A., Popescu, D.C.: On the Likelihood-Based Approach to Modulation Classification. IEEE Transactions on Wireless Communications **8**(12), 5884–5892 (Dec 2009). https://doi.org/10.1109/TWC.2009.12.080883
8. Hermawan, A.P., Ginanjar, R.R., Kim, D.S., Lee, J.M.: CNN-Based Automatic Modulation Classification for Beyond 5G Communications. IEEE Communications Letters **24**(5), 1038–1041 (May 2020). https://doi.org/10.1109/LCOMM.2020.2970922
9. Huynh-The, T., Hua, C.H., Pham, Q.V., Kim, D.S.: MCNet: An Efficient CNN Architecture for Robust Automatic Modulation Classification. IEEE Communications Letters **24**(4), 811–815 (Apr 2020). https://doi.org/10.1109/LCOMM.2020.2968030
10. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards Deep Learning Models Resistant to Adversarial Attacks. In: 2018 International Conference on Learning Representations (Feb 2018)
11. Manoj, B.R., Santos, P.M., Sadeghi, M., Larsson, E.G.: Toward Robust Networks against Adversarial Attacks for Radio Signal Modulation Classification. In: 2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC). pp. 1–5 (Jul 2022). https://doi.org/10.1109/SPAWC51304.2022.9833926
12. Nicolae, M.I., Sinn, M., Tran, M.N., Buesser, B., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., Molloy, I.M., Edwards, B.: Adversarial Robustness Toolbox v1.2.0 (Nov 2019). https://doi.org/10.48550/arXiv.1807.01069
13. Proakis, J., Salehi, M.: Fundamentals of Communication Systems. Pearson, Boston, 2 edn. (Jun 2013)
14. Roy, D., Salehi, B., Banou, S., Mohanti, S., Reus-Muns, G., Belgiovine, M., Ganesh, P., Dick, C., Chowdhury, K.: Going beyond RF: A Survey on How AI-enabled

Multimodal Beamforming Will Shape the NextG Standard. Computer Networks **228**, 109729 (Jun 2023). https://doi.org/10.1016/j.comnet.2023.109729
15. Sadeghi, M., Larsson, E.G.: Adversarial Attacks on Deep-Learning Based Radio Signal Classification. IEEE Wireless Communications Letters **8**(1), 213–216 (Feb 2019). https://doi.org/10.1109/LWC.2018.2867459
16. The MathWorks Inc.: MATLAB version: 23.2.0 (R2023b). Natick, Massachusetts, United States (2023), https://www.mathworks.com
17. Venkata Subbarao, M., Samundiswary, P.: Performance Analysis of Modulation Recognition in Multipath Fading Channels using Pattern Recognition Classifiers. Wireless Pers Commun **115**(1), 129–151 (Nov 2020). https://doi.org/10.1007/s11277-020-07564-z
18. Wang, Y., Sun, T., Li, S., Yuan, X., Ni, W., Hossain, E., Vincent Poor, H.: Adversarial Attacks and Defenses in Machine Learning-Empowered Communication Systems and Networks: A Contemporary Survey. IEEE Communications Surveys & Tutorials **25**(4), 2245–2298 (2023). https://doi.org/10.1109/COMST.2023.3319492
19. Xie, S., Ye, J.: Overview of Automatic Modulation Recognition Methods. In: 2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE). pp. 1–7 (Apr 2023). https://doi.org/10.1109/ICDCECE57866.2023.10150889
20. Zhang, C., Costa-Pérez, X., Patras, P.: Adversarial Attacks Against Deep Learning-Based Network Intrusion Detection Systems and Defense Mechanisms. IEEE/ACM Transactions on Networking **30**(3), 1294–1311 (Jun 2022). https://doi.org/10.1109/TNET.2021.3137084
21. Zhang, C., Patras, P., Haddadi, H.: Deep Learning in Mobile and Wireless Networking: A Survey. IEEE Communications Surveys & Tutorials **21**(3), 2224–2287 (2019). https://doi.org/10.1109/COMST.2019.2904897
22. Zhang, F., Luo, C., Xu, J., Luo, Y.: An Efficient Deep Learning Model for Automatic Modulation Recognition Based on Parameter Estimation and Transformation. IEEE Communications Letters **25**(10), 3287–3290 (Oct 2021). https://doi.org/10.1109/LCOMM.2021.3102656
23. Zhang, F., Luo, C., Xu, J., Luo, Y., Zheng, F.C.: Deep Learning Based Automatic Modulation Recognition: Models, Datasets, and Challenges. Digital Signal Processing **129**, 103650 (Sep 2022). https://doi.org/10.1016/j.dsp.2022.103650
24. Zhang, L., Lambotharan, S., Zheng, G., AsSadhan, B., Roli, F.: Countermeasures Against Adversarial Examples in Radio Signal Classification. IEEE Wireless Communications Letters **10**(8), 1830–1834 (Aug 2021). https://doi.org/10.1109/LWC.2021.3083099
25. Zhang, W., Krunz, M., Ditzler, G.: Stealthy Adversarial Attacks on Machine Learning-Based Classifiers of Wireless Signals. IEEE Transactions on Machine Learning in Communications and Networking **2**, 261–279 (2024). https://doi.org/10.1109/TMLCN.2024.3366161
26. Zheng, Q., Zhao, P., Wang, H., Elhanashi, A., Saponara, S.: Fine-Grained Modulation Classification Using Multi-Scale Radio Transformer With Dual-Channel Representation. IEEE Communications Letters **26**(6), 1298–1302 (Jun 2022). https://doi.org/10.1109/LCOMM.2022.3145647
27. Zhu, Z., Nandi, A.K.: Automatic Modulation Classification: Principles, Algorithms and Applications. Wiley (2015)