

2022

Defensive Distillation-Based Adversarial Attack Mitigation Method for Channel Estimation Using Deep Learning Models in Next-Generation Wireless Networks

Ferhat Ozgur Catak

Murat Kuzlu

Old Dominion University, mkuzlu@odu.edu

Evren Catak

Umit Cali

Ozgur Guler

Follow this and additional works at: https://digitalcommons.odu.edu/engtech_fac_pubs



Part of the [Electrical and Computer Engineering Commons](#), and the [Information Security Commons](#)

Original Publication Citation

Catak, F. O., Kuzlu, M., Catak, E., Cali, U., & Guler, O. (2022). Defensive distillation-based adversarial attack mitigation method for channel estimation using deep learning models in next-generation wireless networks. *IEEE Access*, 10, 98191-98203. <https://doi.org/10.1109/ACCESS.2022.3206385>

This Article is brought to you for free and open access by the Engineering Technology at ODU Digital Commons. It has been accepted for inclusion in Engineering Technology Faculty Publications by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.

Received 19 August 2022, accepted 8 September 2022, date of publication 12 September 2022,
date of current version 22 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3206385

RESEARCH ARTICLE

Defensive Distillation-Based Adversarial Attack Mitigation Method for Channel Estimation Using Deep Learning Models in Next-Generation Wireless Networks

FERHAT OZGUR CATAK¹, (Member, IEEE), MURAT KUZLU², (Senior Member, IEEE),
EVREN CATAK³, (Member, IEEE), UMIT CALI⁴, (Member, IEEE), AND OZGUR GULER⁵

¹Department of Electrical Engineering and Computer Science, University of Stavanger, Rogaland, 4021 Stavanger, Norway

²Department of Engineering Technology, Old Dominion University, Norfolk, VA 23529, USA

³4034 Stavanger, Norway

⁴Department of Electric Power Engineering, Norwegian University of Science and Technology, 7034 Trondheim, Norway

⁵eKare Inc., Fairfax, VA 22031, USA

Corresponding author: Ferhat Ozgur Catak (f.ozgur.catak@uis.no)

This work was supported in part by the Commonwealth Cyber Initiative, an Investment in the Advancement of Cyber Research and Development, Innovation, and Workforce Development in Virginia. For more information about CCI, visit (www.cyberinitiative.org).


ABSTRACT Future wireless networks (5G and beyond), also known as Next Generation or NextG, are the vision of forthcoming cellular systems, connecting billions of devices and people together. In the last decades, cellular networks have dramatically grown with advanced telecommunication technologies for high-speed data transmission, high cell capacity, and low latency. The main goal of those technologies is to support a wide range of new applications, such as virtual reality, metaverse, telehealth, online education, autonomous and flying vehicles, smart cities, smart grids, advanced manufacturing, and many more. The key motivation of NextG networks is to meet the high demand for those applications by improving and optimizing network functions. Artificial Intelligence (AI) has a high potential to achieve these requirements by being integrated into applications throughout all network layers. However, the security concerns on network functions of NextG using AI-based models, i.e., model poisoning, have not been investigated deeply. It is crucial to protect the next-generation cellular networks against cybersecurity threats, especially adversarial attacks. Therefore, it needs to design efficient mitigation techniques and secure solutions for NextG networks using AI-based methods. This paper proposes a comprehensive vulnerability analysis of deep learning (DL)-based channel estimation models trained with the dataset obtained from MATLAB's 5G toolbox for adversarial attacks and defensive distillation-based mitigation methods. The adversarial attacks produce faulty results by manipulating trained DL-based models for channel estimation in NextG networks while mitigation methods can make models more robust against adversarial attacks. This paper also presents the performance of the proposed defensive distillation mitigation method for each adversarial attack. The results indicate that the proposed mitigation method can defend the DL-based channel estimation models against adversarial attacks in NextG networks.

INDEX TERMS Trustworthy AI, security, next-generation networks, adversarial machine learning, model poisoning, channel estimation.

I. INTRODUCTION

A. PREAMBLE

In the last decade, the next-generation networks deployed on cellular networks (i.e., 5G and beyond) are undergoing

The associate editor coordinating the review of this manuscript and approving it for publication was Miguel López-Benítez .

a major revolution along with advanced telecommunication technologies for high-speed data transmission, high cell capacity, and low latency. Each network has its own focus, i.e., 5G: deliver higher multi-Gbps peak data speeds, ultra-low latency, 6G: embed artificial intelligence. NextG networks require a high-cost investment and research to meet infrastructure, computing, security, and privacy requirements.

These technologies will enable the next data communications and networking era by connecting everyone to a world in which everything is connected. The main goal of those technologies is to support a wide range of new applications, such as Augmented reality (AR), Virtual reality (VR), metaverse, telehealth, education, autonomous and flying vehicles, smart cities, and smart grids, and advanced manufacturing. They will create new opportunities for industry to improve visibility, enhance operational efficiency, and accelerate automation [1]. It is expected that next-generation networks must simultaneously provide high data speed, ultra-low latency, and high reliability to support services for those applications [2]. Artificial Intelligence (AI) plays a crucial role in achieving these requirements by being integrated into applications throughout all levels of the network. AI is one of the key drivers for next-generation wireless networks to improve network applications' efficiency, latency, and reliability [3]. AI is also applied to channel estimation applications, which is one of the fundamental prerequisites in wireless networks. The traditional channel estimation methods are extremely complex and low accurate due to the multi-dimensional data structure and the nonlinear characteristics of the channel. Therefore, DL-based channel estimation models have been used in next-generation networks to address the traditional channel estimation. However, DL-based channel estimation models can be vulnerable to adversarial machine learning (ML) attacks. A secure scheme is crucial for DL-based channel estimation models used in next-generation networks and security and vulnerability issues. DL-based models in the next-generation wireless communication systems should be evaluated before deploying them to the production environments in terms of vulnerability, risk assessment, and security threat.

B. RELATED WORKS

The main goal of NextG networks is to provide very high data rates (Tbps) and extremely low latency (less than milliseconds) with a high cell capacity (10 million devices for every square kilometer) [4], [5]. The key of the next-generation networks is to use new technologies, such as millimeter wave (mmWave), massive multiple-input multiple-output (massive MIMO), and AI. mmWave is essential for those networks, which provides a high capacity, throughput, and very-low latency in frequency bands above 24 GHz. Massive MIMO is an advanced version of MIMO, which includes a group of antennas at both the transmitter and receiver sides. This method provides better throughput and spectrum efficiency in wireless communication. AI-based algorithms have been used to improve network performance and efficiency. This study focuses on DL-based channel estimation models in next-generation wireless networks and their vulnerabilities. In the literature, these topics have already been studied with and without vulnerability concerns [6], [7], [8], [9], [10], [11]. The authors in [6] reviewed AI-empowered wireless networks and the role of AI in deploying and optimizing next-generation architectures in terms of operations.

It indicated that AI-based models have already been used to train the transmitter, receiver, and channel as an auto-encoder. This allows the transmitter and receiver to be optimized mutually. The study also indicated that next-generation networks would differ from current ones, such as network infrastructures, wireless access technologies, computing, application types, etc. The authors in [12] reviewed DL-based solutions in next-generation networks, focusing on physical layer applications of cellular networks from massive MIMO, reconfigurable intelligent surface (RIS), and multi-carrier (MC) waveform. It also emphasized the AI-based solutions' contribution to improving network performance. The authors in [13] and [14] proposed a robust channel estimation framework using the fast and flexible denoising convolutional neural network (FFDNet) and deep convolutional neural networks (CNNs) for mmWave MIMO. Both proposed methods can deal with a wide range of signal-to-noise ratio (SNR) levels with a flexible noise level map and offer better performance for channel estimators in terms of accuracy. DL-based algorithms significantly improve the overall system performance for next-generation wireless networks. Fortunately, several research groups in the wireless research community study the main potential security issue related to AI-based algorithms, i.e., model poisoning [15], [16]. The authors in [17] and [18] provided a comprehensive review of NextG wireless networks in terms of opportunities and security and privacy challenges, as well as proposed solutions for NextG networks. Several studies also present robust frameworks focusing on detecting adversarial attacks accurately. The authors in [19] proposed a framework to detect adversarial attacks for industrial artificial intelligence systems (IAISs), called DeSVig, i.e., decentralized swift vigilance framework. According to the results, the proposed framework can detect adversarial attacks, such as DeepFool and FGSM, with high accuracy and low delay. The authors also stated that the DeSVig framework provides better performance than current state-of-art defense approaches in terms of robustness, efficiency, and scalability based on experimental results.

C. PURPOSE AND CONTRIBUTIONS

The channel estimation is one of the most challenging topics in 5G and beyond networks due to the difficulties of finding the correlation between many resources, system parameters, and dynamic communication channel characteristics by using existing techniques. Therefore, sophisticated AI-based algorithms can help to model the highly nonlinear correlations and estimate the channel characteristics [20]. In our recent papers [21] and [22], adversarial attacks and mitigation methods have been investigated along with the proposed framework for mmWave beamforming prediction models in next-generation networks. This study provides a comprehensive vulnerability analysis of deep learning (DL)-based channel estimation models trained with the dataset obtained from MATLAB's 5G toolbox for adversarial attacks and defensive distillation-based mitigation methods. It also implements widely used adversarial attacks from the Fast Gradient

Sign Method (FGSM), Basic Iterative Method (BIM), Projected Gradient Descent (PGD), Momentum Iterative Method (MIM), to Carlini & Wagner (C&W) as well as a defensive distillation-based mitigation method for DL-based models. The results showed that DL-based models used in these networks are vulnerable to adversarial attacks, while the models can be more secure against adversarial attacks through the proposed mitigation method. The source code is available from GitHub.¹

The scope of this study is limited to one of the 5G physical layer applications, i.e., DL-based channel estimation, its vulnerability analysis under selected adversarial attacks, and the proposed defensive distillation mitigation method. There are also other attack types, like the CW attack, which computes intensively and requires more iterations than traditional methods. In this study, we use a less compute-intensive and more efficient way to create adversarial examples.

II. PRELIMINARIES

This section presents a brief overview of the channel estimation and the adversarial ML attacks, such as FGSM, BIM, PGD, MIM, and C&W, along with defensive distillation-based mitigation. Dataset description and scenarios are also given with the selected performance metrics to evaluate the models' performance under normal and attack conditions.

A. CHANNEL ESTIMATION FOR COMMUNICATION SYSTEM

In a wireless communication system, the channel characteristic presents the communication link properties between transmitter and receiver. It is also known as channel state information (CSI). The signal is transmitted through a communication channel. i.e., medium, the transmitted signal is received as a distortion and noise added. It is needed to decode the received signal and remove the unwanted signal, i.e., distortion and noise added by the channel, from the received signal. To identify the channel characteristics is the first process to achieve that, which is called *channel estimation* process. The received signal is attenuated by a factor h_0 and delayed by a specific time τ_0 . h_0 depends on the propagation medium, frequency, T_x/R_x gains, while τ_0 depends on the speed of an electromagnetic wave in the medium.

It is assumed that $x(t)$ presents the transmitted signal, while $y(t)$ presents the received signal. When $x(t)$ is transmitted through a communication channel, i.e., air, the signal is distorted, and noise is added to the transmitted signal. As a result, the received signal $y(t)$ is not the same as the transmitted signal $x(t)$. Received signal $y(t)$ is shown as:

$$y(t) = h_0 * x(t - \tau_0) \quad (1)$$

However, the received signal comprises several reflected and scattered paths, i.e., multiple paths, with different attenuation and delay. The composed received signal is

shown as:

$$y(t) = \sum_{l=0}^l h_l * x(t - \tau_l) \quad (2)$$

where l is the specific path/tap at a time.

The mobility causes Doppler frequency shift, i.e., the change in the wavelength or frequency of the waves as to the observer being in motion with respect to the wave source. Doppler effect plays an important role in telecommunications and computations of signal path loss and fading due to multipath propagation. In addition, the channel characteristics, i.e., h_0 and τ_0 , can also change over time due to the mobility of the one of communication sides, and shown as h_l^t and τ_l^t . The channel can be characterized by a number of paths/taps, the dependence of channel coefficients, and delay in time. The final received signal with the Doppler effect can be shown at a specific time as:

$$y(t) = \sum_{l=0}^l h_l^t * x(t - \tau_l^t) \quad (3)$$

The channel estimation plays an important part in wireless communications for increasing the capacity and the overall system performance. There is a high demand for new wireless networks, higher data rates, better quality of service, and higher network capacity. Therefore, new promising technologies are needed to meet these requirements. A migration from Single Input Single Output (SISO) to Multiple Input Multiple Output (MIMO) antenna technology has started with NextG networks. The channel estimation is the core of next-generation communication systems, i.e., 5G and beyond, performed in different ways for SISO and MIMO approaches at the receiver side. The channel estimation algorithm can be classified into three main categories, i.e., blind channel estimation, semi-blind channel estimation, and training-based estimation [23]. The training-based estimation among them is widely used in communication systems. The general approach of the channel estimation is to insert known reference symbols, i.e., pilots, into the transmitted signal and then interpolate the channel response based on these known pilot symbols. The process works in the following steps: (1) develop a mathematical model to correlate the transmitted and received signals using channel characteristics, (2) embed a predefined signal, i.e., pilot signal, into the transmitted signal, (3) transmit the signal through the channel, (4) receive transmitted signal as a distorted and/or noise added through the channel, (5) decode the pilot signal from the received signal, (6) compare the transmitted and the received signals, and (7) find the correlation between the transmitted and the received signals.

There have been many efforts regarding channel estimation algorithms using different approaches in the literature. However, it is still a challenging problem due to the computational complexity degree of algorithms and an enormous amount of mathematical operations, and channel estimation accuracy at low. The equalization method is typically used

¹<https://github.com/ocatak/6g-channel-estimation-dataset>

to reduce the complexity and render the frequency response at the receiver side [24]. With the introducing the machine learning methods to 5G and beyond communication systems, the performance of the channel estimation algorithm has been improved in terms of the degree of low computational complexity and channel estimation accuracy compared to conventional channel estimation algorithms [25]. In addition, the nature of deep learning-based algorithms can also save a significant computational power for complex analysis needed in channel estimation algorithms [26]. However, it can still be questionable of the feasibility of using machine learning methods in channel estimation. The study [27] presented several deep learning-based channel estimation algorithms, i.e., fully-connected deep neural network (FDNN), Convolutional Neural Network (CNN), and bidirectional long short-term memory (bi-LSTM), with different scenarios of fading multi-path channel models for 5G networks. According to the results, three presented deep learning-based algorithms reduced the channel estimation error and bit error ratio and were robust to the changes in the Doppler frequency. However, bi-LSTM among them provided the most significant reduction in channel estimation error. The authors in [28] also proposed a CNN combined with a projected gradient descent algorithm to demonstrate the feasibility of using machine learning methods in channel estimation.

A channel model is a representation of the channel that a transmitted signal follows to the receiver. In the simulation environment, the channel model is typically classified into two categories, i.e., clustered delay line (CDL) model and tapped delay line (TDL) channel model. A CDL is used to model the channel when the received signal consists of multiple delayed clusters. Each cluster contains multipath components with the same delay but slight variations for angles of departure and arrival, i.e., MIMO. On the other hand, a TDL model is defined as simplified evaluations of CDL, i.e., non-MIMO evaluations or SISO. These channel models are defined well in the technical report released by 3GPP, i.e., the 3rd Generation Partnership Project [29]. According to this report, CDL/TDL models are defined in the frequency range from 0.5 GHz to 100 GHz with a maximum bandwidth of 2 GHz. For CDL/TDL models, five different channel profile models are constructed, i.e., A, B, and C for non-line-of-sight (NLOS) propagation, while D and E for line-of-sight (LOS) propagation. Power, delay and angular information are used to define CDL models, while power, delay, and Doppler spectrum information are used for TDL models in the technical report released by 3GPP.

B. CONVOLUTIONAL NEURAL NETWORKS

The convolutional neural network (CNN) is a neural network that has shown to be very successful for image recognition [30], [31], [32]. Compared to the fully-connected neural network, CNN can extract all the information with a lower number of parameters. The main idea of CNN is that we can locate the structure of an image by the convolution operation. Suppose the image \mathbf{x} is a two-dimensional matrix.

The convolution operation between the image \mathbf{x} and a filter \mathbf{W} is defined by

$$\mathbf{y} = \mathbf{W} * \mathbf{x} = \sum_{i=1}^W \sum_{j=1}^H \mathbf{W}_{i,j} \mathbf{x}_{i-s,j-s}, \quad (4)$$

where W and H are the width and height of the image \mathbf{x} , respectively, and s is the number of strides, which is the distance between two adjacent positions.

The CNN is composed of several types of layers. The convolution layer is the most critical layer of the CNN, consisting of several filters. Each filter extracts a particular type of feature from an input image. The pooling layer is a down-sampling layer, which reduces the size of the convolution output. Each pooling operation replaces several adjacent values with the maximal value or the mean value. The fully-connected layer is a standard neural network layer that combines all the features extracted by the convolution layer. The softmax layer is a classification layer to classify the input data.

The input image is a two-dimensional matrix. The filter in the convolution layer extracts a particular type of feature from the input image. For example, the leftmost filter extracts horizontal lines, and the middle filter extracts diagonal lines. The output of the convolution layer is then sent to the pooling layer, which reduces the size of the data. The output of the pooling layer is then sent to the fully-connected layer, which combines all the features extracted by the convolution layer. The output of the fully-connected layer is then sent to the softmax layer, which classifies the data.

C. ADVERSARIAL ATTACKS

ML-based models are trained to automatically learn the underlying patterns and correlations in data by using algorithms. Once an ML-based model is trained, it can be used to predict the patterns in new data. The accuracy of the trained model is essential to achieving a high performance, which can also be called as a generalization. However, the trained model can be manipulated by adding noise to the data, i.e., targeted and non-targeted adversarial ML attacks. The adversarial ML attacks are generated by adding a perturbation to a legitimate data point, i.e., an adversarial example generated craftily input with a slight difference, to fool the ML-based models. In such attacks, the attacker does not change training instances and tries to make some small input instances perturbations to make this new input instance safe in the model's inference period. The existing defenses and adversarial attacks for images can be applied to attack and defend on other fields [33], [34], [35]. The cleverly-designed adversarial examples can fool the deep neural networks with high success rates on the test images. The adversarial examples can also be transferred from one model to another model. There are various kinds of adversarial ML attacks, such as evasion attacks, data poisoning attacks, and model inversion attacks [36]. An evasion attack aims to cause the ML-based models to classify improperly the adversarial examples as

legitimate data points, i.e., targeted and non-targeted evasion attacks. Targeted attacks aim to force the models to classify the adversarial example as a specific target class. Non-targeted attacks aim to push the models to classify the adversarial example as any class other than the ground truth. Data poisoning aims to generate malicious data points to train the ML-based models to find the desired output. It can be applied to the training data, which causes the ML-based models to produce the desired outcome. Model inversion aims to generate new data points close to the original data points to find the sensitive information of the specific data points. In this study, we focus on this kind of adversarial attack. Taking channel estimation CNN model as an example, here, we use $h(\mathbf{x}, \omega) : \mathbb{R}^{m \times n} \mapsto \mathbb{R}^{m \times n}$ to denote the channel estimation CNN model, where ω is the parameters of the channel estimation CNN model, and \mathbf{x} is the input data. A targeted adversarial attack aims to generate an adversarial example \mathbf{x}' from a legitimate example \mathbf{x} to fool the channel estimation CNN model to produce the desired output. The attacker uses the lowest possible budget to corrupt the inputs, aiming to increase the distance (i.e., MSE) between the model's prediction and the real channel. Therefore σ is calculated as

$$\sigma^* = \arg \max_{\|\sigma\|_p \leq \epsilon} \ell(\omega, \mathbf{x} + \sigma, \mathbf{y}) \quad (5)$$

where $\mathbf{y} \in \mathbb{R}^{m \times n}$ is the label (i.e., channel information), and p is the norm value and it can be 0, 1, 2, ∞ .

Figure 1 shows a typical adversarial ML-based adversarial sample generation procedure.

These adversarial attack types are given as follows.

1) FGSM

Fast Gradient Sign Method (FGSM): FGSM is one of the most popular and simplest approaches to constructing adversarial examples. It is called one-step gradient-based attacks. It is used to compute the gradient of the loss function with respect to the input, \mathbf{x} , and then the attacker creates the adversarial example by adding the sign of the gradient to the input data. It was first introduced by Goodfellow *et al.* [37]. The gradient sign is computed using the backpropagation algorithm. The steps are summarized as follows:

- Compute the gradient of loss function, $\nabla_{\mathbf{x}} \ell(\mathbf{x}, \mathbf{y})$
- Add the gradient to the input data, $\mathbf{x}_{adv} = \mathbf{x} + \epsilon \times \text{sign}(\nabla_{\mathbf{x}} \ell)$

where ϵ is the budget. FGSM attack has been used in [38] to attack channel estimation models.

2) BIM

Basic Iterative Method (BIM): BIM is one of the most popular attacks, which is called an iterative gradient-based attack. This attack is derived from the FGSM attack [39]. It is used to compute the gradient of the loss function with respect to the input, \mathbf{x} , and then the attacker creates the adversarial example by adding the sign of the gradient to the input data.

The gradient sign is computed using the backpropagation algorithm. The steps are summarized as follows:

- Initialize the adversarial example as $\mathbf{x}_{adv} = \mathbf{x}$
- Iterate i times, where $i = 0, 1, 2, 3, \dots, N$
- Compute the gradient of loss function, $\nabla_{\mathbf{x}} \ell(\mathbf{x}_{adv}, \mathbf{y})$
- Add the gradient to the input data, $\mathbf{x}_{adv} = \mathbf{x}_{adv} + \epsilon \times \text{sign}(\nabla_{\mathbf{x}} \ell)$

where ϵ is the budget, and N is the number of iterations. The BIM attack has been used in [38] to attack channel estimation models.

3) PGD

PGD is one of the most popular and powerful attacks, which is called gradient-based attacks [40], [41]. It is used to compute the gradient of the loss function with respect to the input, \mathbf{x} , and then the attacker creates the adversarial example by adding the sign of the gradient to the input data. The gradient sign is computed using the backpropagation algorithm. The steps are summarized as follows:

- Initialize the adversarial example as $\mathbf{x}_{adv} = \mathbf{x}$
- Iterate i times, where $i = 0, 1, 2, 3, \dots, N$
- Compute the gradient of loss function, $\nabla_{\mathbf{x}} \ell(\mathbf{x}_{adv}, \mathbf{y})$
- Add random noise to the gradient, $\hat{\nabla}_{\mathbf{x}} \ell(\mathbf{x}_{adv}, \mathbf{y}) = \nabla_{\mathbf{x}} \ell(\mathbf{x}_{adv}, \mathbf{y}) + \mathcal{U}(\epsilon)$
- Add the gradient to the input data, $\mathbf{x}_{adv} = \mathbf{x}_{adv} + \alpha \times \text{sign}(\hat{\nabla}_{\mathbf{x}} \ell)$

where ϵ is the budget, N is the number of iterations, and α is the step size. PGD can generate stronger attacks than FGSM and BIM.

4) MIM

Momentum Iterative Method (MIM): MIM is a variant of the BIM adversarial attack, introducing momentum term and integrating it into iterative attacks [42]. It is used to compute the gradient of the loss function with respect to the input, \mathbf{x} , and then the attacker creates the adversarial example by adding the sign of the gradient to the input data. The gradient sign is computed using the backpropagation algorithm. The steps are summarized as follows:

- Initialize the adversarial example $\mathbf{x}_{adv} = \mathbf{x}$ and the momentum, $\mu = 0$
- Iterate i times, where $i = 0, 1, 2, 3, \dots, N$
- Compute the gradient of loss function, $\nabla_{\mathbf{x}} \ell(\mathbf{x}_{adv}, \mathbf{y})$
- Update the momentum, $\mu = \mu + \frac{\eta}{\epsilon} \times \nabla_{\mathbf{x}} \ell(\mathbf{x}_{adv}, \mathbf{y})$
- Add random noise to the gradient, $\hat{\nabla}_{\mathbf{x}} \ell(\mathbf{x}_{adv}, \mathbf{y}) = \nabla_{\mathbf{x}} \ell(\mathbf{x}_{adv}, \mathbf{y}) + \mathcal{U}(\epsilon)$
- Add the gradient to the input data, $\mathbf{x}_{adv} = \mathbf{x}_{adv} + \alpha \times \text{sign}(\hat{\nabla}_{\mathbf{x}} \ell)$

where ϵ is the budget, N is the number of iterations, η is the momentum rate, and α is the step size.

5) C&W

The C&W attack was proposed as a targeted evasion attack by Carlini and Wagner [43]. It is based on the idea of a zero-sum game. In a zero-sum game, the total amount of value

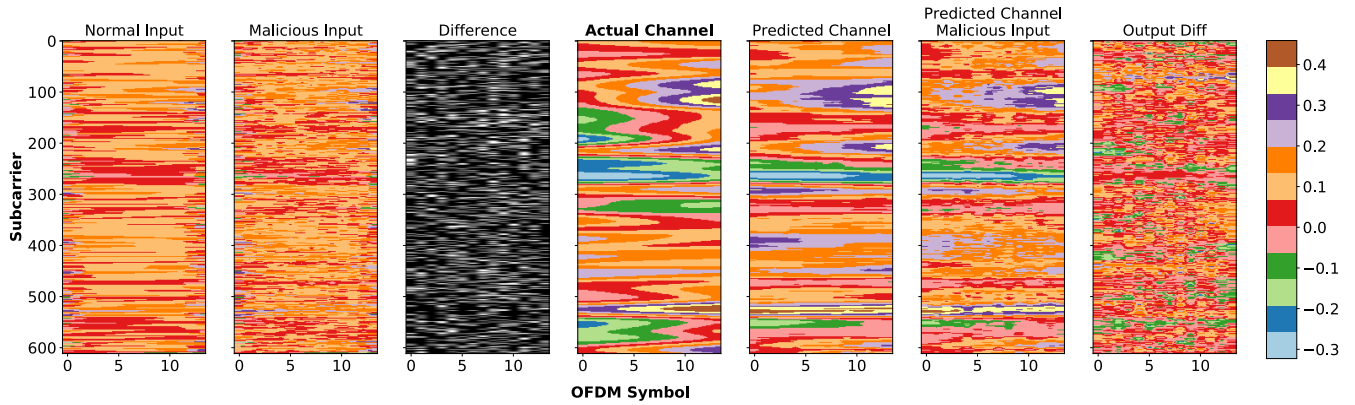


FIGURE 1. Typical adversarial ML-based adversarial sample generation.

in the game is fixed. The winner of the game gets all of the value, and the loser gets nothing. The C&W method is an iterative attack that constructs adversarial examples by approximately solving the minimization problem $\min_d(x, x')$ such that $f(x') = t'$ for the attacker-chosen target t' , where $d(\cdot)$ is an appropriate distance metric. The optimization problem is shown in the following equation:

$$\min_{x \in \mathcal{X}} \mathbb{E}_{y \in \mathcal{Y}} [f(x) - y]^2$$

where $x \in \mathcal{X}$ is a training example, $y \in \mathcal{Y}$ is the target output, and $f(x)$ is the function to be estimated. The optimization is solved for a set of points x' that are close to the target t' , such that the function $f(x) - y$ is maximized for all y . This produces a set of adversarial examples x' that are likely to fool the defender model.

The most important difference between C&W and other adversarial ML attacks is that C&W does not require an ϵ value for the optimization. That is, C&W does not require that the attacker's goal be to find a set of points that are close to the target but instead find a set of points that are guaranteed to fool the defender. This makes C&W a more powerful attack.

D. DEFENSIVE DISTILLATION

Knowledge distillation was previously introduced by Hinton *et al.* [44] to compress the knowledge of a large, densely connected neural network (the teacher) into a smaller, sparsely connected neural network (the student). It was shown that the student was able to reach a similar performance as the teacher [44]. In the initial work, the knowledge distillation was used to solve a classification problem, which is also called the teacher-student framework. Papernot *et al.* [45] proposed this technique for the adversarial ML defense and demonstrated that it could make the models more robust against adversarial examples. The main contribution of this work was to introduce the knowledge distillation to the adversarial ML defense. Defensive distillation is an ML framework that can enhance the robustness of the model for classification problems. The first step is to train the

teacher model with a high temperature (T) parameter to soften the softmax probability outputs of the DL model. This can be done as follows:

$$p_{softmax}(z, T) = \frac{e^{z/T}}{\sum_{i=1}^n e^{z_{(i)}/T}} \quad (6)$$

where n is the number of labels and z is the output of the last layer of the DL model, i.e., $z = \mathbf{W}_n \cdot \mathbf{a}_{n-1} + b_n$. Here, \mathbf{W}_n is the weight matrix, and \mathbf{a}_{n-1} is the activation of the last layer. In the second step, the softmax probability outputs train the student model with a lower temperature parameter. The objective function is defined as

$$\begin{aligned} \mathcal{L}_{student}(T) &= \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^n y_{ij} \cdot \log p_{softmax}(z_{ij}, T) \\ &= \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^n y_{ij} \cdot \log \frac{e^{z_{ij}/T}}{\sum_{i=1}^n e^{z_{ij}/T}} \end{aligned} \quad (7)$$

where N is the number of training samples, y_{ij} is the training label, and z_{ij} is the logit. The objective function for the training the teacher model is defined as

$$\mathcal{L}_{teacher}(T) = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^n y_{ij} \cdot \log \frac{e^{z_{ij}/T}}{\sum_{i=1}^n e^{z_{ij}/T}} \quad (8)$$

Defensive distillation is a method that can enhance the robustness of the models, which are trained by the soft targets provided by the teacher model. By minimizing the objective functions, the model can be trained. This method helps build robust models against adversarial examples [45]. Figure 2 shows the overall steps for this technique. According to the figure, the teacher model is typically an extensive deep neural network, while the student model is usually a small and shallow neural network. The knowledge distillation process consists of two steps: (1) training the teacher model and (2) distilling the knowledge from the teacher to the student. The distillation can be performed using the teacher model's output probabilities, the teacher model's activations, or the

intermediate representations of the teacher model. The distillation can also be performed using a distillation loss, typically a combination of the cross-entropy loss and the distillation loss. The cross-entropy loss is used to minimize the difference between the output probabilities of the teacher and student models. In contrast, the distillation loss is used to minimize the difference between the intermediate representations of the teacher and student models.

Deep learning approaches have been shown to perform exceptionally well for a wide range of computer vision tasks (e.g., image classification, object, and action detection, scene segmentation, image generation, etc.). However, deep neural networks (DNNs) require large amounts of training data, which is not always available for new tasks or domains. Several knowledge distillation methods have been proposed to address this issue that can train a smaller student network to mimic the prediction of a more extensive and accurate teacher network.

Distillation has been applied in the field of intelligent systems, such as knowledge-based and rule-based systems, to reduce the system's size and improve the system's performance by improving the quality of the system's knowledge. The teacher and student models' differences can be considered a form of regularization, which is crucial to prevent overfitting. The algorithm 1 shows the pseudocode of distillation.

Algorithm 1 Pseudocode of Distillation

Input: Dataset D , teacher model T , student model S , loss function \mathcal{L} , learning rate η , number of epochs E
Output: Trained student model S
 Initialize the weights of the student model S
for $e = 1$ **to** E **do**
 Randomly shuffle the dataset D
 for $i = 1$ **to** $|D|$ **do**
 Extract the i^{th} sample (x_i, y_i) from D
 Forward propagate the sample x_i through the teacher model T to obtain the output probabilities \hat{y}_i
 Compute the loss \mathcal{L} using the output probabilities \hat{y}_i
 Backpropagate the loss \mathcal{L} through the student model S
 Update the weights of the student model S using the learning rate η
 end for
end for
return Trained student model S

In a typical wireless communication system, the channel estimation is done by the base station with the help of pilot signals sent by the user equipment (UE) during uplink. And the base station sends pilot signals toward the UE, which acknowledges the estimated channel information for the downlink transmission. Network operators and service providers are responsible for running their operations properly and meeting their obligations to the customers and the public related to privacy and data confidentiality. However, the network operations can be vulnerable to machine learning

adversarial attacks, especially 5G and beyond, due to using machine learning-based applications. In Figure 2, the training of the channel estimation prediction model (i.e., student model) is protected against adversarial ML attacks, and its use in base stations is shown in all its stages.

III. DATASET DESCRIPTION AND SCENARIO

MATLAB 5G Toolbox provides a wide range of reference examples for next-generation network communications systems, such as 5G [46]. It also allows to customize and generate several types of waveforms, antennas, and channel models to obtain datasets for DL-based models. In this study, the dataset used to train the DL-based channel estimation models is generated through a reference example in MATLAB 5G Toolbox, i.e., “Deep Learning Data Synthesis for 5G Channel Estimation”. In the example, a convolutional neural network (CNN) is used for channel estimation. Single-input single-output (SISO) antenna method is also used by utilizing the physical downlink shared channel (PDSCH) and demodulation reference signal (DM-RS) to create the channel estimation model.

The reference example in the toolbox generates 256 training datasets, i.e., transmit/receive the signal 256 times, for the DL-based channel estimation model. Each dataset consists of 8568 data points, i.e., 612 subcarriers, 14 OFDM symbols, 1 antenna. However, each data point of the training dataset is converted from a complex (real and imaginary) 612-14 matrix into a real-valued 612-14-2 matrix for providing inputs separately into the neural network during the training process. This is because the resource grids consist of complex data points with real and imaginary parts in the channel estimation scenario, but the CNN model manages the resource grids as 2-D images with real numbers. In this example, the training dataset is converted into 4-D arrays, i.e., 612-14-1-2N, where N presents the number of training examples, i.e., 256.

Complex numbers are used in wireless communication technologies. The complex number system modifies and demodulates wireless signals in digital wireless communication. The most significant distinction between the real and complex number systems is that the complex number system contains more than one dimension. Adversarial ML attacks, on the other hand, use real numbers to enter the decision boundaries of the victim DL models, and the final malicious inputs are in the real number domain. Complex numbers are split into real and imaginary elements to solve this challenge. Table 1 shows the example dataset.

For each set of the training dataset, a new channel characteristic is generated based on various channel parameters, such as delay profiles (TDL-A, TDL-B, TDL-C, TDL-D, TDL-E), delay spreads (1-300 nanosecond), doppler shifts (5-400 Hz), and Signal-to-noise ratio (SNR or S/N) changes between 0 and 10 dB. Each transmitted waveform with the DM-RS symbols is stored in the training dataset and the perfect channel values in train labels. The CNN-based channel estimation based is trained with the generated dataset.

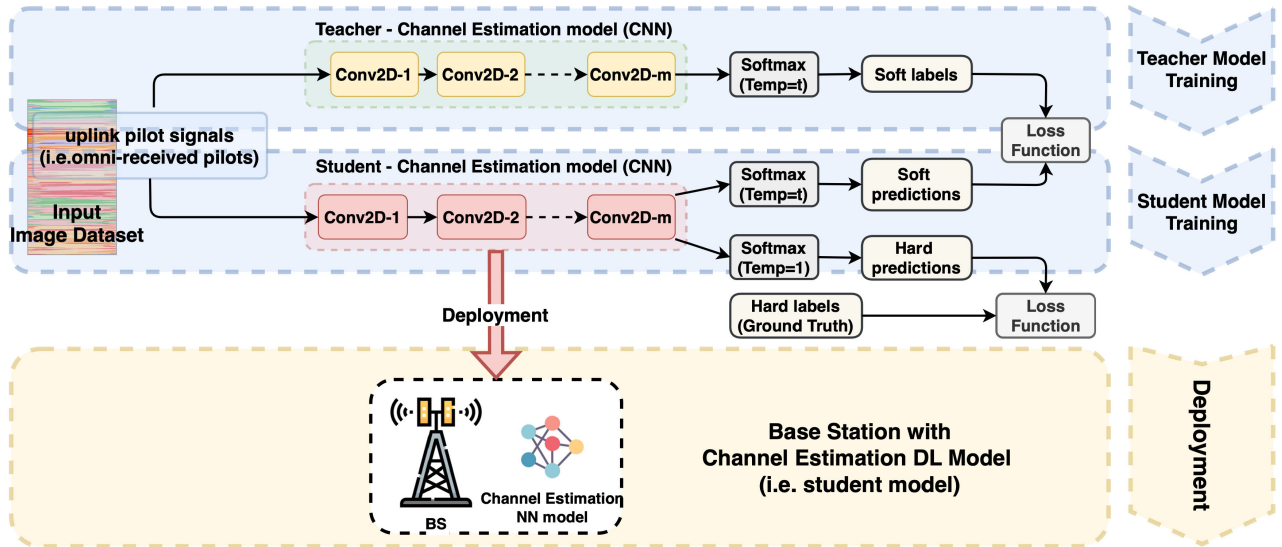


FIGURE 2. Overview of the system architecture with knowledge distillation.

TABLE 1. Example dataset. The original dataset is shown as complex numbers in the table at the top. The training dataset is represented in real numbers in the table below.

F1	F2	F3	F4
0.15+0.90j	0.26+0.90j	0.32+0.90j	0.41+0.88j
-0.39-0.84j	-0.46-0.83j	-0.55-0.79j	-0.61-0.72j
-0.26-0.89j	-0.38-0.87j	-0.44-0.84j	-0.50-0.80j
-0.56+0.78j	-0.45+0.82j	-0.37+0.89j	-0.28+0.89j
⋮	⋮	⋮	⋮
-0.86-0.43j	-0.88-0.35j	-0.87-0.23j	-0.89-0.12j



F1-1	F1-2	F2-1	F2-2	F3-1	F3-2	F4-1	F4-2
0.15	0.90	0.26	0.90	0.32	0.90	0.41	0.88
-0.39	0.84	-0.46	0.83	-0.55	0.79	-0.61	0.72
-0.26	0.89	-0.38	0.87	-0.44	0.84	-0.50	0.80
-0.56	0.78	-0.45	0.82	-0.37	0.89	-0.28	0.89
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
-0.86	0.43	-0.88	0.35	-0.87	0.23	-0.89	0.12

MATLAB 5G toolbox also allows tuning several communication channel parameters, such as the frequency, subcarrier spacing, number of subcarriers, cyclic prefix type, antennas, channel paths, bandwidth, code rate, modulation, etc. The channel estimation scenario parameters with values are given for each in Table 2.

The training dataset is split into validation and training sets to avoid overfitting the training data. The training set is used to train and fit the model, while the validation data is used for monitoring the performance of the trained neural network at certain intervals, i.e., 5 per epoch. The training is expected to stop when the validation loss stops decreasing and improving the model. In this study, most part of the dataset is used for training, i.e., 80% for training, and 20% for testing.

TABLE 2. The channel estimation parameters with values.

Channel Parameter	Value
Delay Profile	TDL-A, TDL-B, TDL-C, TDL-D, TDL-E
Delay Spread	1-300 ns
Maximum Doppler Shift	5-400 Hz
NFFT	1024
Sample Rate	30720000
Symbols Per Slot	14
Windowing	36
Slots Per Subframe	2
Slots Per Frame	20
Polarization	Co-Polar
TransmissionDirection	Downlink
NumTransmitAntennas	1
NumReceiveAntennas	1
FadingDistribution	Rayleigh
Modulation	16QAM

IV. SIMULATION MODEL, SETTINGS AND PERFORMANCE METRIC

A. SIMULATION MODEL

Figure 3 shows the CNN-based DL model used in this paper for the channel estimation. The input to the model is the pilot signals with different subcarriers and OFDM symbols. The input is first passed through a convolutional layer, followed by a max-pooling layer. The output of the max-pooling layer is then passed through a fully connected layer, followed by a softmax layer. The final output of the model is the channel estimation.

We use the channel estimation dataset described in Section III to train the model. We use five different attacks (i.e., FGSM, BIM, MIM, PGD, and C&W) to evaluate the proposed mitigation methods. The deep learning-based channel estimation model is trained in the TensorFlow environment. The proposed mitigation methods are implemented in the Keras environment. The MSE performance metric is used to evaluate the accuracy of the channel estimation model.

B. SIMULATION SETTINGS

The teacher and student models are DNNs with 3 convolutional layers. They are trained using stochastic gradient descent with a momentum of 0.9 and a learning rate of 0.001 for 100 epochs. The batch size is set to 256. Table 3 shows the DL model parameters.

TABLE 3. CNN Model architecture parameters for the teacher and the student models.

	Name	Type	Filters	Kernel Size	Padding	Output
Teacher	Conv-1	Conv2D	48	(9, 9)	same	(1, 612, 48)
	Conv-2	Conv2D	16	(5, 5)	same	(1, 612, 16)
	Conv-3	Conv2D	1	(5, 5)	same	(1, 612, 1)
Student	Conv-1	Conv2D	24	(9, 9)	same	(1, 612, 48)
	Conv-2	Conv2D	8	(5, 5)	same	(1, 612, 16)
	Conv-3	Conv2D	1	(5, 5)	same	(1, 612, 1)

Figure 3 shows the architecture of the teacher and student models.

The models are generative and supervised models trained to predict channel parameters defined at the receiver. The input and output size is 612×14 (e.g. *Subcarriers* \times *OFDM symbols*). Table 4 shows the CNN model’s hyper-parameters.

TABLE 4. CNN Model architecture parameters for the teacher and the student models.

Hyper parameter	Value
Framing Problem	Supervised Regression
Initialization method	GlorotUniform
Activation functions	<ul style="list-style-type: none"> • Conv-1: Selu • Conv-2: Softplus • Conv-3: Selu
Number of parameters	<ul style="list-style-type: none"> • Undefended: 6977 • Teacher: 23533 • Student: 6977

Figure 4 shows the training history of all three models.

C. PERFORMANCE METRIC

The performance metric, MSE (Mean Squared Error), is used to evaluate and compare CNN-based models. The MSE scores are utilized for further analyses of the model. MSE equation is given below. It measures the average squared difference between the actual and predicted values. The MSE equals zero when a model has no error. The model error increases along with the MSE value.

$$MSE = \frac{\sum (Y_t - \hat{Y}_t)^2}{n} \tag{9}$$

where : Y_t :The actual t^{th} instance, \hat{Y}_t : The forecasted t^{th} instance, n: The total number of instance

V. EVALUATION AND PERFORMANCE RESULTS

This section provides the experimental results to evaluate the proposed defensive distillation-based mitigation method for DL-based channel estimation models in next-generation networks. We applied the attack success ratio (ASR) as the performance metric. ASR is the ratio of test samples that an attacker can mispredict to the total number of test samples. The highest ASR indicates that the attack is more effective. The following equation is used to calculate ASR:

$$ASR = \frac{1}{m} \sum_{i=0}^m \frac{MSE(\mathbf{x}_{(i)}^{adv}, \mathbf{y}_{(i)}) - MSE(\mathbf{x}_{(i)}, \mathbf{y}_{(i)})}{MSE(\mathbf{x}_{(i)}^{adv}, \mathbf{y}_{(i)})} \tag{10}$$

Table 5 shows the initial prediction performance results of all models with the test dataset.

TABLE 5. Initial MSE values with test (i.e., benign) dataset.

Model	MSE
Undefended	0.02766
Teacher	0.02484
Student	0.02558

The first experiment is to perform attacks on the undefended model, as shown in Table 6.

TABLE 6. Experimental results for the undefended DL model. The results show that the initial DL model is vulnerable to adversarial ML attacks.

Attack	ϵ	MSE		ASR
		Benign Input	Malicious Input	
BIM	0.1	0.028126	0.028485	0.018932
	0.5	0.028128	0.036766	0.289385
	1.0	0.028106	0.073039	0.613742
	2.0	0.028222	0.192034	0.832142
	3.0	0.027837	0.306523	0.904284
FGSM	0.1	0.028213	0.028477	0.013840
	0.5	0.028223	0.034714	0.215770
	1.0	0.028106	0.052979	0.433404
	2.0	0.028121	0.121161	0.617207
	3.0	0.028126	0.234474	0.689940
MIM	0.1	0.028126	0.028493	0.019298
	0.5	0.028229	0.037301	0.297863
	1.0	0.027990	0.069112	0.599503
	2.0	0.028228	0.162054	0.825845
	3.0	0.028228	0.323735	0.908491
PGD	0.1	0.028000	0.028363	0.019231
	0.5	0.028205	0.036839	0.288993
	1.0	0.028106	0.073028	0.613850
	2.0	0.028141	0.192549	0.833080
	3.0	0.027913	0.317048	0.905127
C&W	-	0.028314	0.029803	0.066435

The results of the first experiment show that the initial DL model is vulnerable to adversarial ML attacks. As expected, the ASR value has a positive correlation with ϵ value. The results also show that the BIM, MIM, and PGD attacks are more effective than the FGSM and C&W (without ϵ) attacks model under the same ϵ . The success rate of the C&W attack model is lower than that of the BIM, MIM, and PGD attack models.

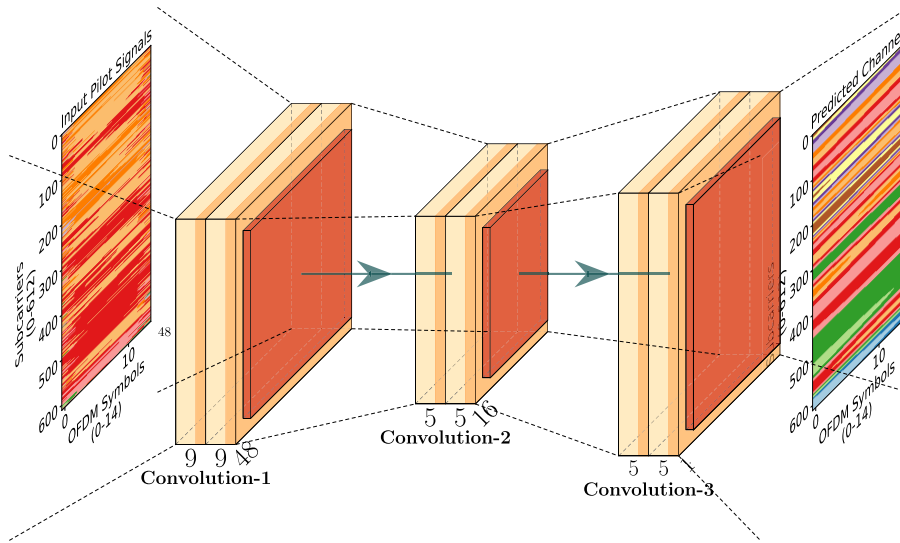


FIGURE 3. The vulnerable CNN model-based channel estimation overview.

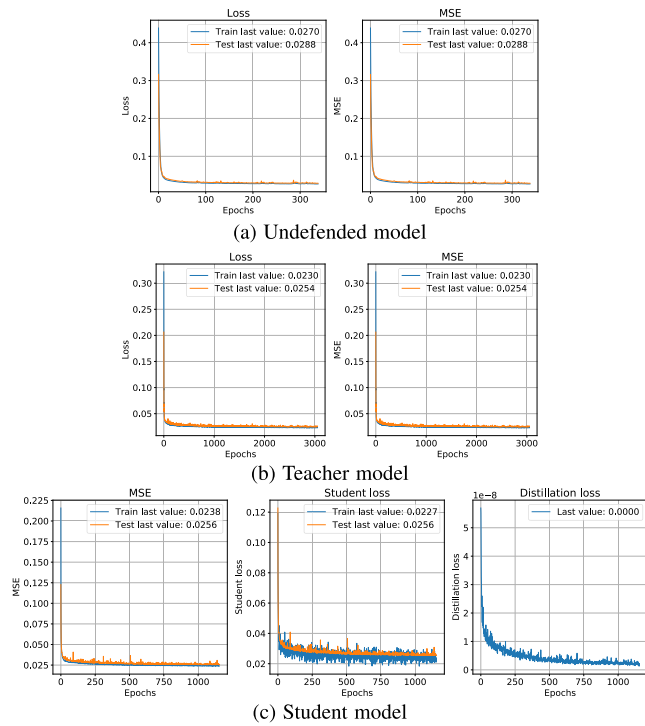


FIGURE 4. Training history of all three models.

Experimental results for the proposed defensive distillation-based mitigation method are shown in Table 7.

The experimental results show that the proposed method can improve the accuracy of the channel estimation model. The results also show that the proposed method can provide better results for the attacks (i.e., FGSM, BIM, MIM, PGD, and C&W).

Figure 5 shows the MSE results with 6 different ϵ values (i.e., 0.0, 0.1, 0.5, 1.0, 2.0, 3.0) for the undefended and

TABLE 7. Experimental results for the proposed defensive distillation-based mitigation method. The results show that the proposed method can improve the accuracy of the channel estimation model. The results indicate that the proposed method can provide better results for the attacks (i.e., FGSM, BIM, MIM, PGD, and C&W).

Attack	ϵ	MSE		ASR
		Benign Input	Malicious Input	
BIM	0.1	0.027861	0.028192	0.018048
	0.5	0.027859	0.029179	0.066479
	1.0	0.027857	0.029177	0.066474
	2.0	0.027860	0.029179	0.066478
	3.0	0.027865	0.029185	0.066460
FGSM	0.1	0.027851	0.027854	0.000118
	0.5	0.027853	0.027921	0.003289
	1.0	0.027845	0.028108	0.012865
	2.0	0.027851	0.028870	0.047475
	3.0	0.027851	0.030105	0.095989
MIM	0.1	0.027864	0.028198	0.018295
	0.5	0.027863	0.029232	0.068893
	1.0	0.027863	0.029232	0.068896
	2.0	0.027860	0.029229	0.068908
	3.0	0.027860	0.029229	0.068914
PGD	0.1	0.027859	0.028190	0.018059
	0.5	0.027866	0.029183	0.066392
	1.0	0.027865	0.029183	0.066400
	2.0	0.027857	0.029175	0.066423
	3.0	0.027862	0.029180	0.066412
C&W	-	0.027263	0.027408	0.00793

defensive distillation-based defended DL model for the all attacks. There is only one bar chart for the C&W attack because there is no ϵ value for the C&W attack. The results show that the proposed method can improve the accuracy of the channel estimation model.

Figure 6 shows the MSE change with different ϵ values for each attack for the undefended and defensive-distillation based defended DL model. The defended model's MSE values (i.e., the right figure) are almost similar to each attack and ϵ values. We can see that the defensive distillation-based

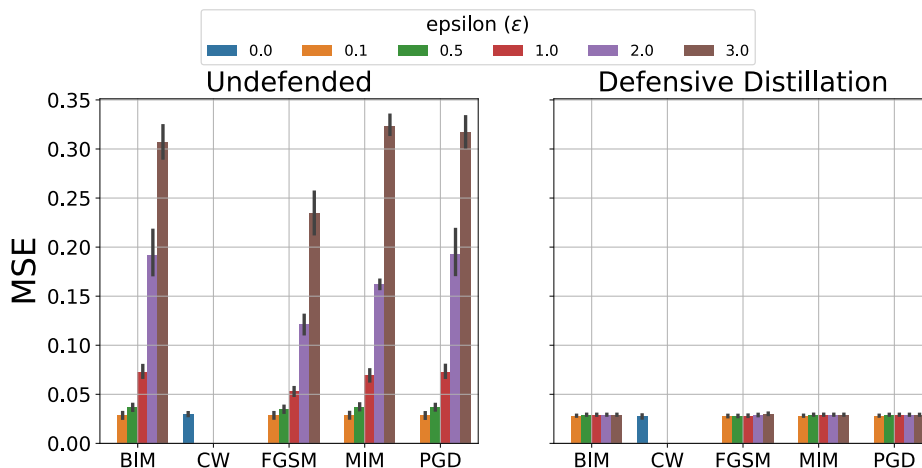


FIGURE 5. Experimental results for the proposed defensive distillation-based mitigation method. The results show that the proposed method can improve the accuracy of the channel estimation model.

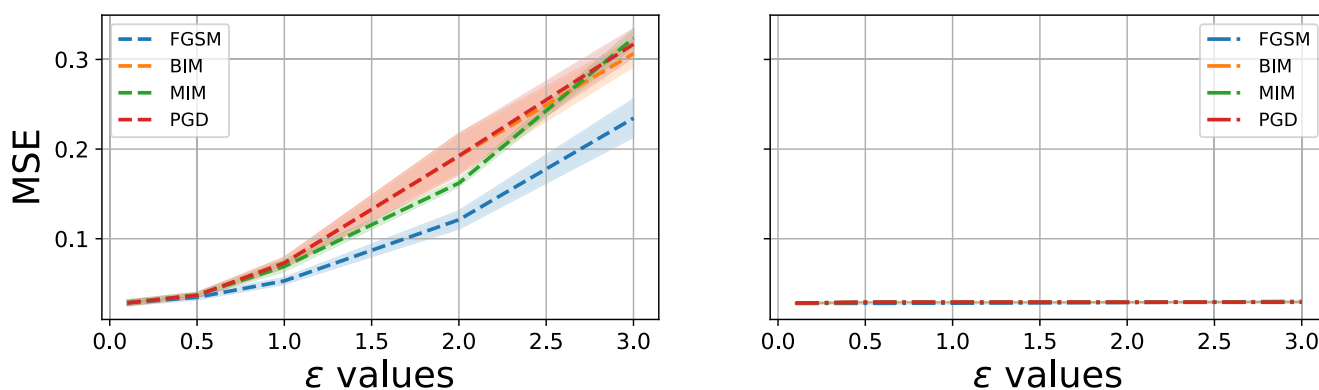


FIGURE 6. MSE trend line for the undefended and defended DL models.

mitigation method works pretty well against all types of adversarial attacks.

VI. DISCUSSION

This study provides a comprehensive analysis of the DL-based channel estimation model in terms of vulnerabilities. The model’s vulnerabilities are studied for various adversarial attacks, including FGSM, BIM, PGD, MIM, and C&W, as well as the mitigation method, i.e., defensive distillation. The results show that CNN-based channel estimation models are vulnerable to adversarial attacks, i.e., FGSM, BIM, MIM, PGD, and C&W. The attack success ratio is also pretty much high, i.e., 0.9, under a higher power attack (ϵ equals 3.0) for BIM, MIM, and PGD attacks. On the other hand, the rate is very low for C&W attacks, i.e., 0.06, compared with the others. Fortunately, the proposed defensive distillation-based mitigation method performs better against higher-order adversarial attacks, and the attack success ratio goes down to 0.06 for BIM, MIM, and PGD attacks. The impact of the mitigation method on FGSM is lower than others, i.e., the attack success rate is 0.09. For C&W, the attack success rate goes from 0.06 to 0.007 after

applying the proposed defensive distillation-based mitigation method. According to the results, adversarial attacks on DL-based channel estimation models and the use of the proposed defensive distillation-based mitigation method can be summarized as:

Observation 1: The DL-based channel estimation models are vulnerable to adversarial attacks, especially BIM, MIM, and PGD.

Observation 2: BIM, MIM, and PGD attacks are the most successful attack success rate.

Observation 3: The DL-based channel estimation models are more robust against C&W attacks.

Observation 4: A strong negative correlation exists between attack power ϵ and the performance of channel estimation models.

Observation 5: The proposed mitigation method, i.e., defensive distillation, offers a better performance against adversarial attacks.

VII. CONCLUSION AND FUTURE WORK

Mobile wireless communication networks are rapidly developing with the high demand and advanced communication

and computing technologies. The last few years have experienced remarkable growth in the wireless industry, especially for NextG networks. This paper provides a comprehensive vulnerability analysis of deep learning (DL) based channel estimation models for adversarial attacks (i.e., FGSM, BIM, PGD, MIM, and C&W) and defensive distillation-based mitigation methods in NextG networks. The results confirm that the original DL-based channel estimation model is significantly vulnerable to adversarial attacks, especially BIM, MIM, and PGD. The attack success rate increases under a heavy adversarial attack ($\epsilon = 3.0$) up to 0.9 for those attacks. There is a high positive correlation between attack power ϵ and the attack success rate as expected, i.e., a high ϵ increases as the attack success rate. On the other hand, the proposed defensive distillation-based mitigation method can improve the accuracy of the channel estimation model and provide better results against higher-order adversarial attacks, e.g., the attack success rate goes from 0.9 to 0.06 after applying the proposed mitigation method. The overall results prove that the proposed method can provide better results for the attacks (i.e., FGSM, BIM, MIM, PGD, and C&W) in terms of the model accuracy and the attack success rate. The scope of this study is restricted to one of the 5G physical layer applications, its vulnerability analysis under selected adversarial machine learning attacks, and the defensive distillation mitigation method. As future work, we plan to focus on other standard defenses, e.g., adversarial training, for the deep learning-based channel estimation models against adversarial attacks and parameter-free attack methods like the C&W attack. As another future work, the authors will focus on the Intelligent Reflecting Surface (IRS) and spectrum sensing using AI-based models and their cybersecurity risks.

REFERENCES

- [1] E. Bertino, D. Bliss, D. Lopresti, L. Peterson, and H. Schulzrinne, "Computing research challenges in next generation wireless networking," 2021, *arXiv:2101.01279*.
- [2] N. A. Johansson, Y.-P.-E. Wang, E. Eriksson, and M. Hessler, "Radio access for ultra-reliable and low-latency 5G communications," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, Jun. 2015, pp. 1184–1189.
- [3] M. E. M. Cayamcela and W. Lim, "Artificial intelligence in 5G technology: A survey," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2018, pp. 860–865.
- [4] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.
- [5] V. Ziegler and S. Yrjola, "6G indicators of value and performance," in *Proc. 2nd 6G Wireless Summit (6G SUMMIT)*, Mar. 2020, pp. 1–5.
- [6] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84–90, Aug. 2019.
- [7] J. Kaur, M. A. Khan, M. Iftikhar, M. Imran, and Q. E. Ul Haq, "Machine learning techniques for 5G and beyond," *IEEE Access*, vol. 9, pp. 23472–23488, 2021.
- [8] F. Wilhelmi, M. Carrascosa, C. Cano, A. Jonsson, V. Ram, and B. Bellalta, "Usage of network simulators in machine-learning-assisted 5G/6G networks," *IEEE Wireless Commun.*, vol. 28, no. 1, pp. 160–166, Feb. 2021.
- [9] A. Yazar and H. Arslan, "A waveform parameter assignment framework for 6G with the role of machine learning," *IEEE Open J. Veh. Technol.*, vol. 1, pp. 156–172, 2020.
- [10] S. Khan, A. Hussain, S. Nazir, F. Khan, A. Oad, and M. D. Alshehri, "Efficient and reliable hybrid deep learning-enabled model for congestion control in 5G/6G networks," *Comput. Commun.*, vol. 182, pp. 31–40, Jan. 2022.
- [11] M. Jalil Piran and D. Young Suh, "Learning-driven wireless communications, towards 6G," in *Proc. Int. Conf. Comput., Electron. Commun. Eng. (iCCECE)*, Aug. 2019, pp. 219–224.
- [12] B. Ozpoyraz, A. T. Dogukan, Y. Gevez, U. Altun, and E. Basar, "Deep learning-aided 6G wireless networks: A comprehensive survey of revolutionary PHY architectures," *Tech. Rep.*, 2022.
- [13] Y. Jin, J. Zhang, S. Jin, and B. Ai, "Channel estimation for cell-free mmWave massive MIMO through deep learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10325–10329, Nov. 2019.
- [14] Y. Jin, J. Zhang, B. Ai, and X. Zhang, "Channel estimation for mmWave massive MIMO with convolutional blind denoising network," *IEEE Commun. Lett.*, vol. 24, no. 1, pp. 95–98, Jan. 2020.
- [15] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "What should 6G be?" *Nature Electron.*, vol. 3, no. 1, pp. 20–29, Jan. 2020.
- [16] M. Kuzlu, C. Fair, and O. Guler, "Role of artificial intelligence in the Internet of Things (IoT) cybersecurity," *Discover Internet Things*, vol. 1, no. 1, pp. 1–14, Dec. 2021.
- [17] P. Porambage, G. Gur, D. P. Moya Osorio, M. Livanage, and M. Ylianttila, "6G security challenges and potential solutions," in *Proc. Joint Eur. Conf. Neww. Commun. 6G Summit (EuCNC/6G Summit)*, Jun. 2021, pp. 1–6.
- [18] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "AI and 6G security: Opportunities and challenges," in *Proc. Joint Eur. Conf. Neww. Commun. 6G Summit (EuCNC/6G Summit)*, Jun. 2021, pp. 1–6.
- [19] G. Li, K. Ota, M. Dong, J. Wu, and J. Li, "DeSVig: Decentralized swift vigilance against adversarial attacks in industrial artificial intelligence systems," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3267–3277, May 2020.
- [20] C.-X. Wang, M. D. Renzo, S. Stanczak, S. Wang, and E. G. Larsson, "Artificial intelligence enabled wireless networking for 5G and beyond: Recent advances and future challenges," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 16–23, Feb. 2020.
- [21] F. O. Catak, M. Kuzlu, E. Catak, U. Cali, and D. Unal, "Security concerns on machine learning solutions for 6G networks in mmWave beam prediction," *Phys. Commun.*, vol. 52, Jun. 2022, Art. no. 101626.
- [22] E. Catak, F. O. Catak, and A. Moldsvor, "Adversarial machine learning security problems for 6G: MmWave beam prediction use-case," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, May 2021, pp. 1–6.
- [23] O. O. Oyerinde and S. H. Mneney, "Review of channel estimation for wireless communication systems," *IETE Tech. Rev.*, vol. 29, no. 4, pp. 282–298, Jul. 2012.
- [24] H. Saad, "Enhanced channel estimation for MIMO-OFDM in 5G NR: Or an analysis of the channel estimation performance," *Tech. Rep.*, 2021.
- [25] Y. Yang, F. Gao, X. Ma, and S. Zhang, "Deep learning-based channel estimation for doubly selective fading channels," *IEEE Access*, vol. 7, pp. 36579–36589, 2019.
- [26] O. Simeone, "A very brief introduction to machine learning with applications to communication systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 648–664, Dec. 2018.
- [27] H. A. Le, T. Van Chien, T. H. Nguyen, H. Choo, and V. D. Nguyen, "Machine learning-based 5G-and-beyond channel estimation for MIMO-OFDM communication systems," *Sensors*, vol. 21, no. 14, p. 4861, Jul. 2021.
- [28] M. Koller, C. Hellings, M. Knodlseder, T. Wiese, D. Neumann, and W. Utschick, "Machine learning for channel estimation from compressed measurements," in *Proc. 15th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2018, pp. 1–5.
- [29] *Study on Channel Model for Frequency Spectrum Above 6 GHz*, Standard 3GPP ETSI TR 138 900 v14.2.0, Jun. 2017.
- [30] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and Cooperation in Neural Nets*. Cham, Switzerland: Springer, 1982, pp. 267–285.
- [31] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.
- [32] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 253–256.

[33] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, no. 6433, pp. 1287–1289, 2019, doi: [10.1126/science.aaw4399](https://doi.org/10.1126/science.aaw4399).

[34] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, "Adversarial attacks on deep-learning models in natural language processing: A survey," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 3, pp. 1–41, Jun. 2020, doi: [10.1145/3374217](https://doi.org/10.1145/3374217).

[35] P. Żelasko, S. Joshi, Y. Shao, J. Villalba, J. Trmal, N. Dehak, and S. Khudanpur, "Adversarial attacks and defenses for speech recognition systems," 2021, *arXiv:2103.17122*.

[36] J. Lin, L. Dang, M. Rahouti, and K. Xiong, "ML attack models: Adversarial attacks and data poisoning attacks," 2021, *arXiv:2112.02797*.

[37] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[38] O. F. Tuna, F. O. Catak, and M. T. Eskil, "Exploiting epistemic uncertainty of the deep learning models to generate adversarial samples," 2021, *arXiv:2102.04150*.

[39] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial Intelligence Safety and Security*. Boca Raton, FL, USA: CRC Press, 2018, pp. 99–112.

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

[41] Y. Jiang, G. Yin, Y. Yuan, and Q. Da, "Project gradient descent adversarial attack against multisource remote sensing image scene classification," *Secur. Commun. Netw.*, vol. 2021, pp. 1–13, Jun. 2021.

[42] I. Fostirooulos, B. Shbita, and M. Marmarelis, "Robust defense against L_p -norm-based attacks by learning robust representations," Tech. Rep.

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

[44] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," Tech. Rep., 2015.

[45] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," Tech. Rep., 2016.

[46] MathWorks. *MATLAB 5G Toolbox*. Accessed: Sep. 30, 2021. [Online]. Available: <https://www.mathworks.com/products/5g.html>



EVREN CATAK (Member, IEEE) received the B.Sc. degree in electrical and electronics engineering from Eskisehir Osmangazi University, Turkey, in 2002, the M.Sc. degree in electronics engineering from Kadir Has University, Istanbul, Turkey, in 2012, and the Ph.D. degree in communication engineering from Yildiz Technical University, Istanbul, in 2017. She was a Postdoctoral Fellow at the Norwegian University of Science and Technology. Her research interests include the physical layer design of emerging communication systems, communication theory, signal processing, and wireless communications.



UMIT CALI (Member, IEEE) received the B.E. degree in electrical engineering from Yildiz Technical University, Istanbul, Turkey, in 2000, and the M.Sc. degree in electrical communication engineering and the Ph.D. degree in electrical engineering and computer science from the University of Kassel, Germany, in 2005 and 2010, respectively. In 2020, he joined as an Associate Professor with the Department of Electric Power Engineering, Norwegian University of Science and Technology, Norway. He worked at the University of Wisconsin—Platteville and the University of North Carolina at Charlotte as an Assistant Professor, from 2013 to 2020, respectively. His current research interests include energy informatics, artificial intelligence, blockchain technology, renewable energy systems, and energy economics. He is serving as an Active Vice-Chair for the IEEE Blockchain in Energy Standards WG (P2418.5).



FERHAT OZGUR CATAK (Member, IEEE) received the B.Sc. degree in electrical/electronic engineering, in 2002, and the Ph.D. degree in informatics, in 2014. He is currently an Associate Professor at the University of Stavanger, Norway. Previously, he worked at TUBITAK, Turkey, NTNU, and Simula Research Laboratory, Norway. His research interests include cyber security, malware analysis, secure multi-party computation, and privacy methods.



MURAT KUZLU (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electronics and telecommunications engineering from Kocaeli University, Turkey, in 2001, 2004, and 2010, respectively. In 2018, he joined as an Assistant Professor with the Department of Engineering Technology, Old Dominion University (ODU). From 2006 to 2011, he worked as a Senior Researcher at TUBITAK-MAM (Scientific and Technological Research Council of Turkey—The Marmara Research Center). Before joining ODU, he was a Research Assistant Professor at the Advanced Research Institute, Virginia Tech, from 2011 to 2018. His research interests include cyber-physical systems, smart cities, smart grids, artificial intelligence, and next-generation networks.



OZGUR GULER received the B.S. degree in computer science and the M.S. degree in computer science with a focus on image-guided surgery from the University of Innsbruck, Innsbruck, Austria, and the Ph.D. degree from the Medical University of Innsbruck, Austria, with a focus on image-guided diagnosis and therapy. He is currently an Imaging Scientist and an AI Researcher specialized in 3-D chronic wound imaging and computer vision. Prior to joining eKare Inc., he was a Researcher with the Sheikh Zayed Institute (SZI) for Pediatric Surgical Innovation Center, Washington, DC, USA, where he developed the segmentation and classification algorithms that laid the groundwork of the eKare inSight system.

...