Causal Effects of Adversarial Attacks on AI Models in 6G Consumer Electronics

Da Guo, Zhengjie Feng, Zhen Zhang, Fazlullah Khan, *Senior Member, IEEE*, Chien-Ming Chen, *Senior Member, IEEE*, Ruibin Bai, *Senior Member, IEEE*, Marwan Omar, and Saru Kumar

***Abstract*—Adversarial examples are security risks in the implementation of artificial intelligence (AI) in 6G Consumer Electronics. Deep learning models are highly susceptible to adversarial attacks, and defense against such attacks is critical to the safety of 6G Consumer Electronics. However, there remains a lack of effective defensive mechanisms against adversarial attacks in the realm of deep learning. The primary issue lies in the fact that it is not yet understood how adversarial examples can deceive deep learning models. The potential operation mechanism of adversarial examples has not been fully explored, which constitutes a bottleneck in adversarial attack defense. This paper focuses on causality in adversarial examples such as combining the adversarial attack algorithms with the causal inference methods. Specifically, we will use a variety of adversarial attack algorithms to generate adversarial samples, and analyze the causal relationship between adversarial samples and original samples through causal inference. At the same time, we will compare and analyze the causal effect between them to reveal the mechanism and discover the reason of miscalculating. The expected contributions of this paper include: (1) Reveal the mechanism and influencing factors of counterattack, and provide theoretical support for the security of deep learning models; (2) Propose a defense strategy based on causal inference method to provide a practical method for the defense of deep learning models; (3) Provide new ideas and methods for adversarial attack defense in deep learning models.**

***Index Terms*—Adversarial example, adversarial attack, causal inference, causality, consumer electronics, 6G.**

I. Introduction

With the rise of 6G communication technology and the rapid development of the consumer electronics, the applications of artificial intelligence (AI) models in daily life have become popular. AI technology in smartphones, smart home devices, and other consumer electronics products provides users with a more intelligent and convenient experience [1]. For these 6G consumer products, developers are increasingly inclined to use advanced artificial intelligence to identify most problems [2]. Therefore, these devices become more intelligent but susceptible to adversarial attacks [3], [4], [5]. In this context, adversarial attacks pose new threats to AI models in 6G consumer electronics. Attackers may try to disrupt the normal operation of these AI models through cleverly designed adversarial examples, malicious injections, etc. In addition, unregulated technology integration can also make consumer electronics vulnerable, leading to security and privacy concerns [6]. Therefore, it is particularly important to conduct in-depth research on the causal effects of adversarial attacks on AI models in the 6G environment [7]. Although, deep learning models have achieved considerable success in areas such as image classification, object detection, and image segmentation [8], it also brings about issues of security and lack of transparency.

Deep learning models are susceptible to various adversarial attacks [9], [10], [11]. In deep learning, an adversarial attack refers to intentionally feeding the model slightly modified data, also known as adversarial examples, to manipulate the model's behavior. These changes are deliberate and aimed at misleading the deep learning model into producing incorrect or even harmful results. To human observers, these manipulated inputs are nearly indistinguishable from the original data. The concept of adversarial examples was introduced by Goodfellow et al. [12]. It is becoming an important issue to tackle because it can lead to serious security problems in the practical application of deep learning models. For instance, in image classification tasks, attackers can slightly perturb an image leading the deep learning model to misclassify it or make an incorrect decision [9]. This can have severe consequences, such as causing autonomous vehicles to misidentify vehicles, obstacles, or traffic signals, leading to traffic accidents [13]. At the same time, medical images should also be protected from cyberattacks in the health care system because they are important for diagnosing many diseases [14], [15]. The model's erratic behavior to abnormal inputs poses new challenges to the usability and safety of deep learning. Therefore, studying adversarial attack algorithms and their mechanisms is crucial for developing secure deep learning applications.

Received 25 November 2023; revised 27 January 2024, 31 March 2024, and 10 June 2024; accepted 1 July 2024. Date of publication 26 August 2024; date of current version 11 December 2024. This work was supported in part by the Ningbo Municipal Bureau of Science and Technology under Grant 2023J194, and in part by Guangdong Natural Science Foundation, China, under Grant 2024A1515011869. *(Corresponding author: Zhen Zhang.)*

Da Guo, Zhengjie Feng, and Zhen Zhang are with the College of Information and Science, Jinan University, Guangzhou 510632, China (e-mail: guodaa@stu2021.jnu.edu.cn; jerryfred@stu2021.jnu.edu.cn; zzhang@jnu.edu.cn).

Fazlullah Khan and Ruibin Bai are with the School of Computer Science, Faculty of Science and Engineering, University of Nottingham Ningbo China, Ningbo 315104, Zhejiang, China (e-mail: fazlullah@ieee.org).

Chien-Ming Chen is with the School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing 211544, China (e-mail: chienmingchen@ieee.org).

Marwan Omar is with the Information Technology and Management, Illinois Institute of Technology, Warrenville, IL 60555 USA.

Saru Kumar is with the Department of Mathematics, Chaudhary Charan Singh University, Meerut 250004, India.

Digital Object Identifier 10.1109/TCE.2024.3443328

Causal machine learning is an emerging method that aims to uncover the causal relationships behind data, not just correlations. In recent years, significant achievements have been made in the field, such as causal inference and causal discovery [16]. Compared to traditional machine learning methods, causal machine learning can better deal with potential confounding variables, describe the essential relationships between variables, and improve the model's predictive accuracy and interpretability, which cannot be achieved by data-driven statistical methods [17], [18]. A potentially confounding variable is one that is not observed in the study but may affect the relationship between the variables. The essential relationship is how a change in one variable leads to a change in another variable. These emphasize causal mechanisms. In the field of adversarial attacks, causal machine learning also provides new ideas for understanding and improving models, such as identifying and analyzing adversarial attacks by studying causal relationships [19]. It is still unclear how adversarial examples deceive deep learning models [10]. Using causal theory to study adversarial examples can help us research and understand semantic information within neural networks and discover decision boundaries for solving problems, which is of great research value for the reliability and security of neural network models.

The motivation of this study is to understand the security threats faced by AI models in consumer electronics products within the 6G communication environment. What are the potential impacts these attacks may have on user privacy and device functionality? By revealing the physical causes and evolutionary mechanisms of adversarial attacks, we aim to provide theoretical support for building more secure and robust AI systems. At the same time, an in-depth understanding of the challenges will help develop innovative vulnerability control strategies to ensure the stability and reliability of AI models in 6G consumer electronics in complex network environments. The main contributions of this paper are on the problem of adversarial examples in deep learning classification tasks.

- Our research is based on the conditions of a white-box deep neural network model and uses a variety of adversarial attack algorithms to generate adversarial examples based on different models in the context of 6G consumer electronics. The white-box deep neural network model we studied assumes a specific architecture, including key network hierarchies, activation functions, and connections. In addition, we treat its parameters as key conditions. This includes weights, biases, and other model parameters that determine how the model behaves when processing input data.
- We use causal inference methods, the average causal effects between image input neurons and output predictions in the neural network are calculated, and the obtained causal effects are visualized.
- The adversarial attacks are carried out on the neural network by calculating the causal effects between input neurons and output predictions in adversarial examples. The causal relationships between inputs and outputs are discovered, thereby capturing the main features of how adversarial examples influence model predictions.
- Comparing the causal effects between original samples and adversarial examples, we more effectively reveal the mechanism of action of adversarial examples within 6G consumer electronics, that is, determine which part of the adversarial example image caused the model to make a mistake.

The rest of the paper is organized as follows: Section II summarizes the exploratory work on adversarial attacks and causal discovery. Section III describes methods for computing causal effects in neural networks. Section IV details our experiments and evaluation. Section V concludes the paper and discusses potential future work.

II. Related Works

A. Adversarial Attacks

Adversarial attacks are a hot topic in the field of machine learning. Goodfellow et al. [12] first introduced the concept of adversarial example attacks and demonstrated their impact on existing deep learning models through experiments. In the field of white-box attacks on adversarial examples, Szegedy et al. [20] was the first to propose the L-BFGS method with box constraints for adversarial example attacks, the first to generate adversarial examples through optimization methods by traversing the manifold represented by the network and finding adversarial examples in the input space. Goodfellow et al. [12] proposed an efficient single-step attack method, the Fast Gradient Sign Method (FGSM), which generates adversarial examples by adding gradient-based disturbances to images. Compared to L-BFGS, this method is faster in generating adversarial examples, thus it can generate a large number of adversarial examples in a short time. Rozsa and Boult [21] improved upon FGSM using the Fast Gradient Method (FGM), which directly uses the real gradient direction to replace the sign of the gradient direction for the attack, allowing for higher pixel perturbations. Based on Saliency maps [22], Papernot et al. [23] proposed the Jacobian-based Saliency Map Attack (JSMA), which uses forward derivatives to create adversarial saliency maps and find the two features that have the greatest impact on the classifier for the attack. However, due to its high time complexity, this algorithm does not run well on high-resolution images.

Among the optimization-based adversarial attack algorithms, Moosavi-Dezfooli et al. [10] proposed the DeepFool algorithm, which tries to find the closest decision boundary to confuse the model. This method reliably quantifies the robustness of these classifiers by calculating the smallest disturbance to misclassify deep neural network models to generate good performance adversarial examples. The Projected Gradient Descent (PGD) [24] is an iterative attack that performs multiple iterations of attacks, unlike FGSM which only performs a single iteration. The basic idea of PGD is to project the generated adversarial examples at each step of gradient descent to ensure they still fall within the original data range. Also based on optimization, the C&W Attack algorithm proposed by Carlini and Wagner [25] is a solution to the joint optimization problem of the objective function and the

disturbance scale. This algorithm designs a new loss function and constraint conditions based on L-BFGS.

Under the formal setting of neural networks and fuzzy set theory, Khan [26] combined the learning ability of neural networks with the reasoning mechanism of fuzzy logic. The empirical results show that the proposed neural fuzzy model can not only provide better predictions (for both in-sample data and out-of-sample data) but also model the causal relationship between variables in more detail through the obtained knowledge base. Furthermore, it points to the significant causal role of deep inductive learning in research. However, image classification based on deep neural networks is susceptible to adversarial perturbation. It can be easily spoofed by adding artificially small and imperceptible perturbations to the input image. To address the vulnerability of classification models, Gu et al. [27] proposed an effective segmentation attack method called SegPGD, that is, create adversarial examples during training and inject them into training data. In addition, they also provide convergence analysis, which shows that under the same number of attack iterations, the proposed SegPGD can create more effective adduction examples than PGD, which greatly improves the robustness of the segmentation model. This development also provides a better idea for embedding causal effects in deep learning. Majidian et al. [28] used principal component analysis (PCA) to extract features, combined with error correction output code (ECOC) and adaptive neural fuzzy reasoning system (ANFIS) for classification, and then proposed a new method for detecting DoS attacks in computer networks. The intrusion detection process is divided into three stages: preprocessing, feature extraction, and classification. In this classification model, the particle swarm optimization (PSO) algorithm is used to optimize the structure of ANFIS. This method is significantly improved compared with the previous methods.

B. Causal Inference

Pearl [29] is known as a pioneer of causal inference. He systematically introduced the basic concepts, theoretical framework and methods of causality. He also demonstrated how causality developed from a vague concept into a mathematical theory. It laid the foundation for the development of causal reasoning. Greenland and Pearl [30] introduced applications of causal inference methods and proposed the concept of causal diagrams, from which we can see whether the causal effects of interest (target effects or causal estimands) can be estimated from available data, or what additional observations are needed to validly estimate those effects, which further promoted the development of causal inference methods. On this foundation, Chattopadhyay et al. [31] proposes a new attribution method for neural networks developed using the first principles of causality, which is also the first to treat neural network architecture as a structural causal model.

Traditional methods of causal inference require fulfilling certain strong assumptions, such as the rare event assumption, the unbiasedness assumption, etc. These assumptions are often difficult to verify and can be easily limited in practical application. Peter et al. [32] proposed a new method of causal inference under the model of causal reasoning, termed “Invariant Prediction Causal Inference”. This involves the use of an algorithm for identifying causal effects and calculating confidence intervals, based on the invariance of predictions. This method does not require satisfying certain strong assumptions, providing a new way to identify causal relationships based on observed data, and it can be flexibly applied to real problems. Kusner et al. [33] noted that traditional concepts of fairness are usually based on the equality or difference of group statistics, which cannot solve the issue of unfairness between individuals. Thus, they proposed the concept of “Counterfactual Fairness” based on causal inference and put forward an algorithm for evaluating and improving the counterfactual fairness of machine learning models, called the “Counterfactual Learning” method. The basic idea is to introduce some virtual samples during the model training, which are generated according to different protected characteristic values and have the same features as the real samples. By comparing the prediction results of real and virtual samples, the counterfactual fairness of the model can be calculated, and the model can be modified to meet the requirement of counterfactual fairness.

Pearl [34] later revisited the basic concept and properties of the do-operator in causal inference and proposed a new algorithm to handle the issues related to the do-operator. The do-operator is a common operation in causal inference, used to represent causal interventions on variables. However, the use of the do-operator requires meeting certain prerequisites, such as node independence, which are often difficult to verify, limiting the application of the do-operator. The algorithm he proposed is based on some simple graphical transformation rules that can simplify the do-operator through derivation, enabling it to handle more complex causal graphs and better deal with issues of conditional independence. This provides a more powerful tool for the field of causal inference, enabling researchers to infer more complex causal relationships. Traditional deep neural networks typically learn complex mappings between inputs and outputs by training on a large amount of data, but these mappings do not provide information about causal relationships between inputs and outputs. Harradon et al. [35] proposed a method based on autoencoders and Bayesian causal models. The method first uses a causal autoencoder to learn human-understandable concepts from data. Then, using these extracted concepts as variables, it builds a Bayesian causal model to perform causal analysis on each node in the network and can calculate the causal effect of each node on the output. This allows for an interpretation of the behavior of deep neural networks in image classification tasks. Cui and Athey [36], in order to bridge the gap between the tradition of precise modeling in causal reasoning and the black box approach in machine learning, clarified the sources of risk in machine learning models and discussed the benefit of introducing causality into learning as stable learning. If predictive stability, interpretability, and fairness are important, the idea of causality can also be used to improve predictive modeling, the stronghold of machine learning. After this, in order to solve the traditional point identification causation, Duarte et al. [37] proposed a general,

automated numerical method for causal reasoning in discrete Settings. They reduce the causal problem of discrete data to a polynomial programming problem and then propose an algorithm to automatically constrain causal effects using effective duality relaxation and spatial branch and bound techniques. The method can accommodate classical barriers, including confusion, selection, measurement errors, disobedience, and non-response. To further understand llm and its causal implications, taking into account the differences between different types of causal reasoning tasks, and the threat of entanglement in structure and measurement validity, Kıcıman et al. [38] envision llm being used alongside existing causal methods to open up new areas for advancing the study, practice, and adoption of causality.

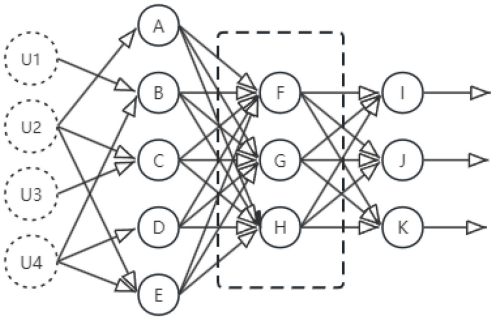

Fig. 1. The causal graph corresponds to the structure of a feedforward neural network, with a hidden layer in the middle.

III. Methodology

In this section, we give details of the methodologies used in this research.

A. Interpreting Neural Networks as Causal Structural Models

Neural networks are methods for approximating functions that can map inputs to outputs. A Structural Causal Model (SCM) also approximates functions but focuses more on describing causal relationships between variables. Therefore, we can simply interpret a neural network as a form of SCM. Specifically, we can view a neural network as a Directed Acyclic Graph (DAG), where each node represents a variable and each edge represents a dependency between variables. In this graph, input variables are nodes without incoming edges, output variables are nodes without outgoing edges, and intermediate nodes represent hidden variables. Each node has an associated function that calculates its output value. These functions are typically nonlinear, such as activation functions. To interpret a neural network as an SCM, we need to impose some constraints on the structure of the graph. Specifically, we need to satisfy two conditions: 1) No redundant paths: there are no redundant paths, i.e., there is only one path between any two nodes. 2) No feedback loops: there are no feedback loops, i.e., there is no path that starts from a node, goes through a series of nodes, and returns to the starting node. These constraints ensure that the causal structure of the graph is well-defined and can be inferred using causal inference methods.

Under these conditions, we can view a neural network as an SCM where each node represents the causal relationship between variables, and each function represents the strength of this causal relationship. It is important to note that neural networks are black-box models, and we typically cannot directly interpret their internal structure. Therefore, interpreting a neural network as an SCM is merely a form of abstraction and does not provide specific information about how the network works internally. Fig. 1 depicts a simple feedforward neural network structure, which includes one input layer, one hidden layer, and one output layer. The variables in the set U are exogenous variables. This neural network can be interpreted as a directed acyclic graph with directed edges, and the direction of the edges is the same as the forward propagation process from the input layer to the output layer in the neural network. Therefore, the final output can be attributed to the hierarchical interaction between lower-level nodes.

Specifically, the following explanation is given for this Structural Causal Model: For a l-layer feed-forward neural network $N(l_1, l_2, \ldots, l_n)$, where l_i is the set of neurons in the ith layer, there corresponds a Structural Causal Model $M([l_1, l_2, \ldots, l_n], U, [f_1, f_2, \ldots, f_n], P_U)$, where l_1 is the input layer, l_n is the output layer. For each l_i, there is a corresponding f_i, which refers to the set of causal relationship functions of the neurons in the ith layer. U refers to a set of exogenous random variables, which serve as causal factors for the input neurons l_1.

According to the mathematical derivation and proof by Chattopadhyay et al. [39], a new conclusion can be drawn: for each l-layer feed-forward neural network $N(l_1, l_2, \ldots, l_n)$, where l_i is the set of neurons in the ith layer, there corresponds a Structural Causal Model $M([l_1, l_2, \ldots, l_n], U, [f_1, f_2, \ldots, f_n], P_U)$, which can be simplified to $M'([l_1, l_n], U, f', P_U)$. Specifically, the following explanation is given for this Structural Causal Model: For a l-layer feed-forward neural network $N(l_1, l_2, \ldots, l_n)$, where l_i is the set of neurons in the ith layer, there corresponds a Structural Causal Model $M([l_1, l_2, \ldots, l_n], U, [f_1, f_2, \ldots, f_n], P_U)$, where l_1 is the input layer, l_n is the output layer. For each l_i, there is a corresponding f_i, which refers to the set of causal relationship functions of the neurons in the ith layer. U refers to a set of exogenous random variables, which serve as causal factors for the input neurons l_1.

According to the mathematical derivation and proof by Chattopadhyay et al. [31], a new conclusion can be drawn: for each l-layer feed-forward neural network $N(l_1, l_2, \ldots, l_n)$, where l_i is the set of neurons in the ith layer, there corresponds a Structural Causal Model $M([l_1, l_2, \ldots, l_n], U, [f_1, f_2, \ldots, f_n], P_U)$, which can be simplified to $M'([l_1, l_n], U, f', P_U)$.

Through recursive replacement, we can marginalize the hidden layer neurons in the neural network. This is equivalent in the corresponding causal Bayesian network to deleting the edges connecting these nodes, and then creating new directed edges from the parents of the deleted neurons (neurons in the

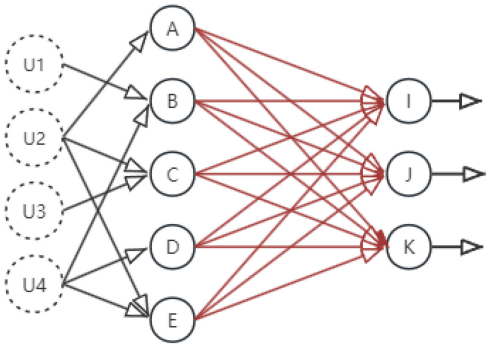

Fig. 2. The simplified causal graph.

input layer) to their respective children (neurons in the output layer). Based on this, we can establish the causal relationship between input and output from the graphical model. Fig. 2 shows the simplified causal graph obtained after marginalizing the hidden layer neurons based on the aforementioned 3-layer neural network example, that is, the simplified causal Bayesian network.

In practice, only the neurons in the l_1 layer and l_n layer are observable, which are input and output from the training dataset, respectively. Therefore, by marginalizing the neurons in the hidden layer, we can further simplify the original causal structure into the Structural Causal Model: $M'([l_1, l_n], U, f', P_U)$, thus establishing a causal relationship between input and output.

B. Computing Causal Relationships in Neural Networks

1) Causal Effect: For the neural network with structure $N(l_1, l_2, \ldots, l_n)$, a simplified Structural Causal Model: $M'([l_1, l_n], U, f', P_U)$ has been proposed in Section III-A. Based on this, this section studies how to quantify the causal relationship of a specific input neuron to a specific output neuron in the neural network.

To study the causal relationship of variables, we first need to intervene on them. Intervention refers to the deliberate change of a variable under certain conditions to study its impact on other variables. By intervening in a variable, we can infer the causal relationship within it. The difference between intervening on a variable and conditioning on it is quite clear. When we intervene on a variable in the model, we fix its value, meaning that the whole system changes and the values of other variables usually change as a result. When conditioning a variable, we don't make any changes to the system; we just focus on a subset of the problem where the values of the variables are of interest to us.

The concept of the *do* operator is a key aspect in causal inference, and it is used to represent an intervention on a variable. More specifically, a *do* expression can be understood as "setting a variable to a certain value", and this value is typically chosen by us rather than determined by the data itself. For example, $do(X = x)$ means "set the variable X to the value x". Therefore, $P(Y = v|X = x)$ represents the probability of $Y = y$ given $X = x$, while $P(Y = v|do(X = x))$ represents the probability of $Y = y$ when we intervene to make $X = x$. From a distributional standpoint, $P(Y = v|X = x)$ reflects the overall distribution of Y for individuals with $X = x$. On the other hand, $P(Y = v|do(X = x))$ reflects the distribution of Y when the value of X is set to x for every individual in the system.

To quantify the causal relationship between variables, we calculate the average causal effect (ACE) between two variables. The causal effect refers to the impact of a particular intervention on an outcome, i.e., the difference in the probability distribution of the outcome under intervention versus no intervention. The average causal effect (ACE) refers to the average impact of a particular intervention on the outcome across the entire domain, i.e., the average causal effect between the intervention and the outcome, controlling for other factors that may influence the outcome. By definition, the average causal effect of a binary random variable x on another random variable y is usually defined as:

$$E\big[y|do(x = 1)\big] - E\big[y|do(x = 0)\big] \tag{1}$$

This definition applies to binary random variables. However, the functions learned by neural networks are often continuous. Therefore, for a given neural network with input l_1 and output l_n, we measure the average causal effect of input feature $x_i \in l_1$ with value α on output feature $y \in l_n$ through the following formula:

$$ACE^{y}_{do(x_i=\alpha)} = E\big[y|do(x_i = \alpha)\big] - baseline_{x_i} \tag{2}$$

Therefore, the causal relationship of input neuron x_i to output neuron y is defined as $ACE^{y}_{do(x_i=\alpha)}$. Here, the baseline value of x_i is defined as the mean average causal effect of x_i on y, which is:

$$baseline_{x_i} = E_{x_i}\big[E_y\big[y|do(x_i = \alpha)\big]\big] \tag{3}$$

It has been proven that this definition of the baseline for x_i is reasonable. The rationale behind this definition can be found in another basic principle: $E[y|do(x_i = \alpha)]$ represents the expected value of the random variable y when the random variable x_i is set to α. If the expected value of y is constant for all possible intervention values of x_i, it can be considered that the causal impact of x_i on y is 0 for any value of x_i. In this case, the baseline value $baseline_{x_i}$ will be the same constant, leading to $ACE^{y}_{do(x_i=\alpha)} = 0$.

2) Computing the Mathematical Expectation of Interventions: Based on the above, to calculate the causal effect between two variables, we need to calculate the expected value of the intervention. We previously defined the intervention expectation value of y as: $E[y|do(x_i = \alpha)]$. Specifically, to calculate this intervention expectation value, we can define:

$$E\big[y|do(x_i = \alpha)\big] = \int_y yp(y|do(x_i = \alpha))dy \tag{4}$$

This calculation involves sampling all other input features from the empirical distribution while ensuring feature $x_i = \alpha$, then taking the output values' average. It's important to note that this is based on the assumption that the input features do not influence each other. However, in problems involving vector calculations, the computation exponentially increases with dimensionality. Due to the curse of dimensionality,

this unbiased estimation of $E[y|do(x_i = \alpha)]$ will have a high variance. In addition, running the entire training data query for prediction results for each intervention will consume a large amount of computation. Therefore, to reduce time complexity, the following procedure optimizes the calculation of the intervention expectation value.

Based on the previous discussion, we can represent the causal mechanism as follows:

$$f_{y|do(x_i=\alpha)'(x_1,x_2,\ldots,x_k)} \tag{5}$$

After conducting the intervention operation $do(x_i = \alpha)$ on the network, the causal mechanism can be expressed as:

$$f_{y|do(x_i=\alpha)'(x_1,\ldots,x_{i-1},\alpha,x_{i+1},\ldots,x_k)} \tag{6}$$

Let $\mu j = E[x_j|do(x_i = \alpha)], \forall x_j \in l_1$. Since f'_y is a neural network, it is smooth. The second-order Taylor expansion of the causal mechanism $fy|do(x_i = \alpha)'$ around the vector $\mu = [\mu_1, \mu_2, \ldots, \mu_K,]^T$ is then expressed as follows:

$$\begin{aligned} f'_y(l_1) \approx{}& f'_y(\mu) + \nabla^T f'_y(\mu)(l_1 - \mu) \\ &+ \frac{1}{2}(l_1 - \mu)^T \nabla^2 f'_y(\mu)(l_1 - \mu) \end{aligned} \tag{7}$$

Calculating the expectations on both sides, we get:

$$\begin{aligned} &\mathrm{E}\Big[f'_y(l_1)|do(x_i = \alpha)\Big] \\ &\approx f'_y(\mu) + \frac{1}{2} Tr\Big(\nabla^2 f'_y(\mu)\mathrm{E}\big[(l_1 - \mu)(l_1 - \mu)^T|do(x_i = \alpha)\big]\Big) \end{aligned} \tag{8}$$

It can be observed that the first-order term in the expansion has disappeared. This is because $E(l_1|x_i = \alpha) = \mu$. Now we only need to calculate the intervention covariance between a single intervention mean μ and the input features to compute $E[y|do(x_i = \alpha)]$.

One thing to note on this basis is that the intervened input neuron is mutually independent of all other input neurons. This implies that if an intervention is made on neuron x_i, the probability distribution of all other input neurons does not change. In other words, the calculation method above has a hidden condition, that is, there is no causal dependency between different input neurons, an assumption often found in machine learning models (apply methods such as Principal Component Analysis if any correlation between input dimensions needs to be eliminated). If there is a dependency between input neurons, it is considered due to the existence of underlying confounding factors, not the influence of one input on another input.

In the previous text, we first define the baseline value of each input neuron as: $baseline_{x_i} = E_{x_i}[E_y[y|do(x_i = \alpha)]]$. Specifically, within the domain of input neuron x_i, we uniformly intervene on the value of input neuron x_i at fixed intervals from low to high, thus calculating $baseline_{x_i}$. Combined with the previously calculated intervention expectation $E[y|do(x_i = \alpha)]$, we can finally calculate the average causal effect value of input neuron x_i on output neuron y.

3) Causal Effect Estimation Based on the β-VAE Model: In Section III-B, the basic calculation method and principle of the causal effect between input neurons and output neurons were introduced. To further investigate the correctness of this causal attribution method, we evaluated it on data where the causal relationship is clearly known. To do so, we trained a β-VAE [40] on the MNIST dataset to obtain disentangled representations that can represent unique generative factors of numerical characters. VAE is a generative model and belongs to a form of autoencoder. β-VAE is an extension of the variational autoencoder. In β-VAE, unique generative factors refer to a set of variables in the latent space, which control different aspects of the generative model. For example, in the MNIST dataset, unique generative factors generating numerical characters might include stroke thickness, rotation angle, font style, etc. Disentanglement is the process of mapping raw data (like images) to a latent high-dimensional vector space, presuming independence between dimensions of the latent encoding in this process.

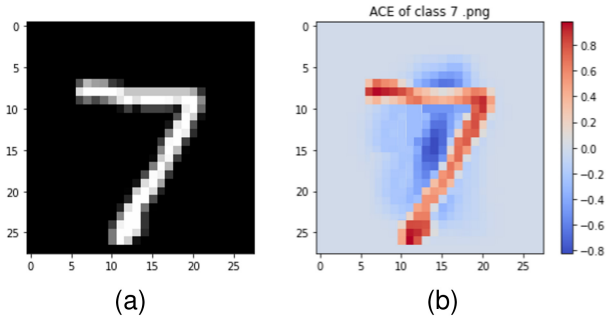

Fig. 3. The left image (a) is the original picture of sample 7 in MNIST. The right image (b) shows the average causal effect of the sample 7.

Specifically, this experiment defines a β-VAE_mnist_mod class as the β-VAE model. The model includes an encoder and a decoder. In the training loop, we use the trainloader to train the model and output the training loss at the end of each epoch. After the training is complete, we use the testloader to evaluate the model. When defining the loss function, the β parameter is used to control the weight of the KL divergence. KL divergence is a measure of the difference between two probability distributions. The smaller the KL divergence value, the more similar the two probability distributions; the larger the KL divergence value, the less similar the two probability distributions. During training, we set the β parameter to 10.

Then, a latent vector z is sampled from the model, and a random vector can be generated using a normal distribution sampler. Z is then concatenated with the numerical label, where the label is the numerical label we want to generate and can be any integer between 0-9, represented by a one-hot encoding. The combined z and label are then passed as inputs to the decoder. We can ultimately use the decoder of the β-VAE to produce reconstructed images. Since we can obtain a probability model through VAE, the intervention expectation can be calculated by Eq. (4). Then, we use $ACE^y_{do(x_i=\alpha)} = E[y|do(x_i = \alpha)] - baseline_{x_i}$ to calculate the causal effect between input neurons (pixels of each input sample) and output neurons (prediction results), and finally visualize the obtained causal effect.

Fig. 3 is a sample example of the MNIST dataset, and Fig. 4 is a graph of the causal effect between the input and output of the sample, where positive causal effects are

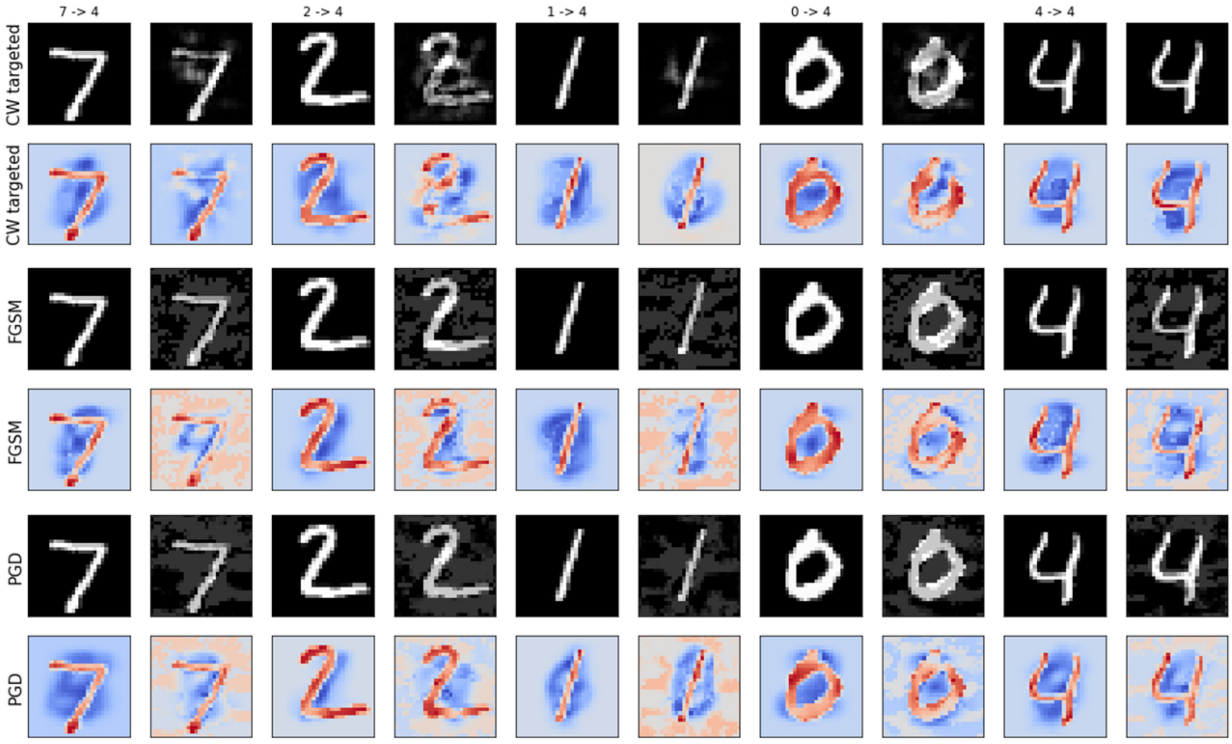

Fig. 4. Comparison of causal effects between MNIST's original samples and adversarial examples under C&W, FGSM, PGD algorithm.

represented in red, and negative causal effects are represented in blue. The results show that the input neurons of the digit image have a positive causal relationship (ACE $>$ 0) with the model's prediction results at their corresponding spatial positions, which is consistent with the causal structure. This indicates that the causal perspective method captures the main features of the data sample. In addition, we can also know which parts of the data are beneficial to the prediction of the neural network model and which are not.

IV. Experiments

In order to explore the mechanism of adversarial examples generated by different adversarial attack algorithms and the differences between different algorithms, this section first generates adversarial examples from several adversarial attack algorithms described in Section II-A, including the C&W adversarial attack algorithm, the FGSM adversarial attack algorithm, and the PGD adversarial attack algorithm. Then, calculate the adversarial examples generated by each algorithm and their corresponding causal effects, and make comparisons and analyses.

A. Generating Adversarial Examples Based on the LeNet

In this section, we have chosen the MNIST and Fashion-MNIST datasets for training the neural network model, and carry out adversarial attacks based on the trained models. For the neural network model, LeNet [41] was selected for training. In the training parameters, the epochs were set to 10 and 20, the batch size was set to 10, the learning rate was set to 0.001, and the optimizer was chosen as SGD. After the training was completed, the accuracy of the model trained with the MNIST dataset on the validation set was 95.23%, and the accuracy of the model trained with Fashion-MNIST was 89.32%. Finally, the test sets of MNIST and Fashion-MNIST were chosen as the datasets for generating adversarial examples.

As for the specific attack design, this paper adopted the open-source adversarial example library Foolbox. Foolbox is a Python-based library for generating and evaluating adversarial examples against machine-learning models. It provides an easy-to-use interface that can help to quickly generate adversarial examples against different neural network learning models. We first selected part of the dataset's images for the attack to verify the correctness of the algorithm, setting the attack parameter ε=0.3 for this experiment.

B. Causal Effects of Adversarial Examples

Based on the completion of model training in Section IV-C1 of this paper, adversarial attacks were carried out on the model. After obtaining the adversarial examples, according to the theoretical basis of the previous text, the average causal effect between the input image and the prediction result of the neural network was calculated for each generated adversarial sample. On the MNIST dataset, the images of the original samples and adversarial examples generated by several attack algorithms, and their corresponding average causal effects, are shown in Fig. 4. Positive causal effects are represented in red, and negative causal effects are represented in blue.

The results show that the method of causal perspective has captured the main features of the adversarial examples. After the adversarial attack, in the main feature areas of each

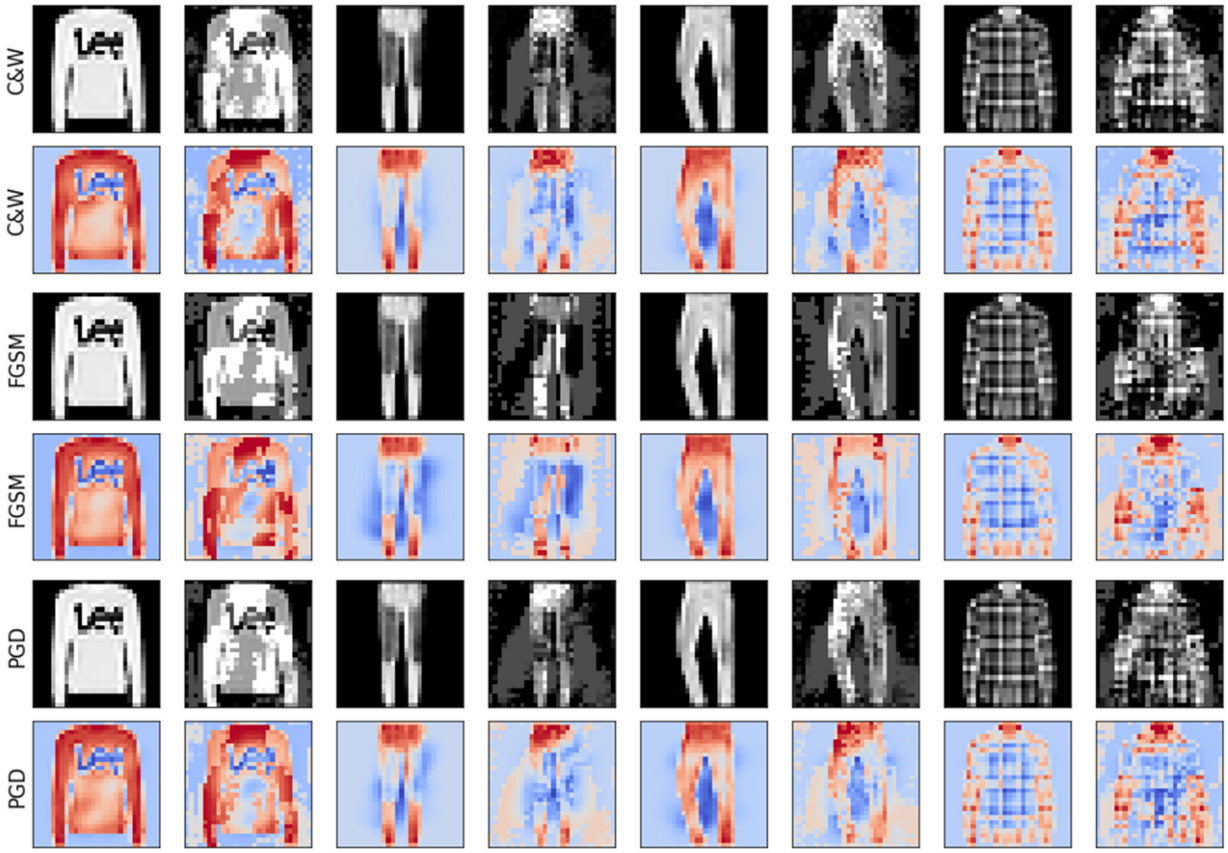

Fig. 5. Comparison of causal effects between FashionMNIST's original samples and adversarial examples under C&W, FGSM, PGD algorithm.

digit of the adversarial examples, the causal effect intensity of the positive causal effects has noticeably weakened, and its spatial continuity has also reduced. In the non-main feature areas of the data, the negative causal effects have increased, and some areas have turned from negative causal effects into positive causal effects. This explains the mechanism of adversarial attacks: on the one hand, the main features of the samples after the attack are weakened; on the other hand, new influencing factors appear in other areas of the image to disturb the prediction of the model. To avoid randomness, the mechanism of adversarial attacks can be observed through more samples. Fig. 5 shows the causal effect maps before and after adversarial attacks on the Fashion-MNIST dataset.

On the whole dataset, we have conducted a quantitative analysis of the causal effect. For the causal effect of the samples, this paper uses two quantitative indicators to measure the differences between adversarial examples and original samples. The first indicator is the sparsity of the positive causal effect:

$$S = 1 - \frac{N_{positive}}{N_{total}} \tag{9}$$

Here, S represents the sparsity of the positive causal effect in the input image, $N_{positive}$ represents the number of pixel blocks with positive causal effects, and N_{total} represents the total number of pixels in the image sample. The higher the S value, the sparser the pixel blocks with positive causal effects are; conversely, it means that the pixel blocks with positive causal effects are denser.

The second indicator is the Total Variation of the causal effect, which measures the discontinuity of the causal effect:

$$TV = \sum_{m,n} \sqrt{\left(\varphi_{m+1,n} - \varphi_{m,n}\right)^2 + \left(\varphi_{m,n+1} - \varphi_{m,n}\right)^2} \tag{10}$$

Here TV is the total change of causal effect, (m, n) is the two-dimensional coordinate index of the pixel block in the input image, and $\varphi_{m,n}$ is the causal effect of the pixel block at the two-dimensional index (m, n). The higher the TV value, the greater the degree of change in causal effect and the greater the discontinuity; conversely, it means greater continuity.

By using the sparsity of positive causal effects and the total variation degree as two indicators to measure the causal effects of the overall dataset samples, the quantitative results of the causal effects of images before and after the adversarial attack on the MNIST and Fashion-MNIST datasets are shown in Table I. Where Smean is the mean of the sparsity of positive causal effects, Sstd is the standard deviation of sparsity; similarly, TVmean and TVstd are the mean and standard deviation of the total variation of causal effects, respectively. Clean refers to the original samples of the dataset. C&W, FGSM, and PGD are adversarial examples generated by three different attack algorithms.

The visualization results and quantitative analysis results show:

TABLE I
QUANTITATIVE ANALYSIS RESULTS OF CAUSAL EFFECTS ON MNIST AND FASHIONMNIST

	MNIST				FashionMNIST			
	Smean	Sstd	Tvmean	TVstd	Smean	Sstd	Tvmean	TVstd
Clean	0.751	0.164	60.32	30.65	0.535	0.067	65.20	24.68
C&W	0.874	0.073	73.41	28.71	0.789	0.139	76.83	33.93
FGSM	0.867	0.096	68.58	33.28	0.816	0.092	71.93	35.76
PGD	0.905	0.062	72.90	33.32	0.823	0.089	79.81	28.82

(1) The pixel areas at the spatial locations corresponding to each digit feature have a positive causal relationship, which is consistent with the known causal relationship in the handwritten digit dataset and the Fashion-MNIST dataset.

(2) Moreover, compared with the original samples, the areas with positive causal relationships in the adversarial examples are sparser, and the size of the causal effects is relatively weaker.

(3) The causal effects of adversarial examples are highly discontinuous in space, that is, their continuity is lower than that of original samples. This may be one of the important reasons why models misrecognize adversarial examples.

(4) The causal effect image shows that only a small portion of the area helps to deceive the recognition model, while the causal effects of most areas are negative or negligible. This phenomenon shows that in adversarial attacks, although the whole sample is disturbed by adversarial noise, only a few key areas of adversarial examples play a decisive role.

In the horizontal comparison of different attack algorithms, it can be seen that the more effective C&W attack algorithm causes the important features in the original image to be disturbed more, which is consistent with the conclusion drawn from the comparison of the accuracy of adversarial examples; for the FGSM and PGD algorithms, which have similar effects on adversarial sample perturbations, some areas in their causal effect maps of adversarial examples turn from negative causal effects into positive causal effects, showing similar patterns, which also coincides with the conclusion from the comparison of the accuracy of adversarial examples. These phenomena all demonstrate the effectiveness of the causal perspective, indicating that this method can capture the main features of adversarial examples and effectively reveal the mechanism of attack algorithms.

V. CONCLUSION

This study combines adversarial attack algorithms with causal reasoning methods to propose an innovative approach for adversarial sample problems in deep learning classification tasks in 6G consumer electronics. By calculating the average causal effect between the input neuron and the output prediction, we successfully capture the main features that affect the model prediction, reveal the mechanism and influencing factors of counterattack, and thus identify the reasons for the model's misjudgments in the adversarial sample.

This approach may have limited generalization across different data sets or tasks, and may not perform well against other types of adversarial attacks. Future work needs to validate our approach on a broader set of data and tasks. We will continue to investigate the application of causal inference methods to adversarial attacks, exploring more adversarial attack algorithms and more general deep learning models. At the same time, we plan to expand the application of causal inference methods to improve the interpretability and transparency of the model. In terms of defense against counterattacks, we will work to come up with more effective defense mechanisms to ensure that deep learning models are more robust in practical applications. These efforts will promote the in-depth solution of the adversarial sample problem and improve the security and reliability of the model [42].

REFERENCES

[1] M. A. Jan, W. Zhang, F. Khan, S. Abbas, and R. Khan, "Lightweight and smart data fusion approaches for wearable devices of the Internet of Medical Things," *Inf. Fusion*, vol. 103, Mar. 2024, Art. no. 102076.

[2] Z. Teimoori, A. Yassine, and M. S. Hossain, "Smart vehicles recommendation system for artificial intelligence-enabled communication," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3914–3925, Feb. 2024.

[3] A. A. Ahmed et al., "Secure AI for 6G mobile devices: Deep learning optimization against side-channel attacks," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3951–3959, Feb. 2024.

[4] M. A. Lopez, G. N. N. Barbosa, and D. M. F. Mattos, "New barriers on 6G networking: An exploratory study on the security, privacy and opportunities for aerial networks," in *Proc. 1st Int. Conf. 6G Netw. (6GNet)*, 2022, pp. 1–6.

[5] X. Wang, A. Shankar, K. Li, B. D. Parameshachari, and J. Lv, "Blockchain-enabled decentralized edge intelligence for trustworthy 6G consumer electronics," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 1214–1225, Feb. 2024.

[6] C. K. Wu, C.-T. Cheng, Y. Uwate, G. Chen, S. Mumtaz, and K. F. Tsang, "State-of-the-art and research opportunities for next-generation consumer electronics," *IEEE Trans. Consum. Electron.*, vol. 69, no. 4, pp. 937–948, Nov. 2023.

[7] L. Xu, X. Zhou, Y. Tao, X. Yu, M. Yu, and F. Khan, "Af relaying secrecy performance prediction for 6G mobile communication networks in industry 5.0," *IEEE Trans. Ind. Informat.*, vol. 18, no. 8, pp. 5485–5493, Aug. 2022.

[8] R. Cong et al., "Boundary guided semantic learning for real-time COVID-19 lung infection segmentation system," *IEEE Trans. Consum. Electron.*, vol. 68, no. 4, pp. 376–386, Nov. 2022.

[9] C. Zhang, X. Costa-Perez, and P. Patras, "Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms," *IEEE/ACM Trans. Netw.*, vol. 30, no. 3, pp. 1294–1311, Jun. 2022.

[10] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2574–2582.

[11] M. L. Das, A. Saxena, and V. P. Gulati, "A dynamic ID-based remote user authentication scheme," *IEEE Trans. Consum. Electron.*, vol. 50, no. 2, pp. 629–631, May 2004.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015, *arXiv:1412.6572*.

[13] S. A. Haider et al., "Secure artificial intelligence for precise vehicle behavior prediction in 6G consumer electronics," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3898–3905, Feb. 2024.

[14] P. Kumar, M. Rahman, S. Namasudra, and N. R. Moparthi, "Enhancing security of medical images using deep learning, chaotic map, and hash table," *Mobile Netw. Appl.*, vol. 23, pp. 1–15, Sep. 2023.

[15] K. Wang, Z. Chen, M. Zhu, Z. Li, J. Weng, and T. Gu, "Score-based counterfactual generation for interpretable medical image classification and lesion localization," *IEEE Trans. Med. Imaging*, early access, Mar. 14, 2024, doi: 10.1109/TMI.2024.3375357.
[16] P. Ma, Z. Ji, Q. Pang, and S. Wang, "NoLeaks: Differentially private causal discovery under functional causal model," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 2324–2338, 2022.
[17] J. G. Richens, C. M. Lee, and S. Johri, "Improving the accuracy of medical diagnosis with causal machine learning," *Nat. Commun.*, vol. 11, no. 1, p. 3923, 2020.
[18] B. Schölkopf, "Causality for machine learning," in *Probabilistic and Causal Inference: The Works of Judea Pearl*. New York, NY, USA: ACM, 2022, pp. 765–804.
[19] R. Huang and Y. Li, "Adversarial attack mitigation strategy for machine learning-based network attack detection model in power system," *IEEE Trans. Smart Grid*, vol. 14, no. 3, pp. 2367–2376, May 2023.
[20] C. Szegedy et al., "Intriguing properties of neural networks," 2014, *arXiv:1312.6199*.
[21] A. Rozsa and T. E. Boult, "Improved adversarial robustness by reducing open space risk via tent activations," 2019, *arXiv:1908.02435*.
[22] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," 2014, *arXiv:1312.6034*.
[23] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Security Privacy (EuroS P)*, 2016, pp. 372–387.
[24] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2019, *arXiv:1706.06083*.
[25] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Security Privacy (SP)*, 2017, pp. 39–57.
[26] H. A. Khan, "AI, deep machine learning via neuro-fuzzy models: Complexities of international financial economics of crises," *Int. J. Comput. Neural Eng.*, vol. 7, pp. 122–34, Nov. 2021.
[27] J. Gu, H. Zhao, V. Tresp, and P. H. S. Torr, "SegPGD: An effective and efficient adversarial attack for evaluating and boosting segmentation robustness," in *Proc. 17th Eur. Conf. Comput. Vis.*, 2022, pp. 308–325.
[28] Z. Majidian, S. TaghipourEivazi, B. Arasteh, and S. Babai, "An intrusion detection method to detect denial of service attacks using error-correcting output codes and adaptive neuro-fuzzy inference," *Comput. Elect. Eng.*, vol. 106, Mar. 2023, Art. no. 108600.
[29] J. Pearl, *Causality*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[30] S. Greenland and J. Pearl, "Causal diagrams," Dept. Stat. Papers, Univ. California, Oakland, CA, USA, Rep. R-332, 2007.
[31] A. Chattopadhyay, P. Manupriya, A. Sarkar, and V. N. Balasubramanian, "Neural network attributions: A causal perspective," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 981–990.
[32] J. Peters, P. Bühlmann, and N. Meinshausen, "Causal inference by using invariant prediction: Identification and confidence intervals," *J. Royal Stat. Soc. Ser. B, Stat. Methodol.*, vol. 78, no. 5, pp. 947–1012, 2016.
[33] M. J. Kusner, J. Loftus, C. Russell, and R. Silva, "Counterfactual fairness," in *Proc. 31st Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.
[34] J. Pearl, "The do-calculus revisited," 2012, *arXiv:1210.4852*.
[35] M. Harradon, J. Druce, and B. Ruttenberg, "Causal learning and explanation of deep neural networks via autoencoded activations," 2018, *arXiv:1802.00541*.
[36] P. Cui and S. Athey, "Stable learning establishes some common ground between causal inference and machine learning," *Nat. Mach. Intell.*, vol. 4, no. 2, pp. 110–115, 2022.
[37] G. Duarte, N. Finkelstein, D. Knox, J. Mummolo, and I. Shpitser, "An automated approach to causal inference in discrete settings," *J. Am. Stat. Assoc.*, vol. 119, no. 547, pp. 1778–1793, 2024.
[38] E. Kıcıman, R. Ness, A. Sharma, and C. Tan, "Causal reasoning and large language models: Opening a new frontier for causality," 2024, *arXiv:2305.00050*.
[39] P. K. Chattopadhyay, *Mathematical Physics*. New Delhi, India: New Age Int., 1990.
[40] I. Higgins et al., "Early visual concept learning with unsupervised deep learning," 2016, *arXiv:1606.05579*.
[41] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[42] J. Deacon. "Model-view-controller (MVC) architecture." Accessed: Mar. 10, 2006, 2009. [Online]. Available: http://www.jdl.co.uk/briefings/MVC.pdf