International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

# Adversarial Attacks and Defences: Ensuring Robustness in Machine Learning Systems

[1]Gireesh Bhaulal Patil, [2]Uday Krishna Padyana, [3]Hitesh Premshankar Rai, [4]Pavan Ogeti, [5]Narendra Sharad Fadnavis, [6]Rajesh Munirathnam

**Submitted**: 12/05/2024 **Revised**: 25/06/2024 **Accepted**: 05/07/2024

**Abstract**: The current paper aims at presenting and discussing adversarial attacks and defence mechanisms in learning models, especially Deep Learning. First, the types of adversarial attacks, the working principle, and the effects on diversified architectures are discussed in this paper. We explicate the current best practices in defence mechanisms and measuring robustness, including various application areas. Real life examples from classification of images, text analysis, and uses of self-driving cars elaborate the real-life issues as well as approaches. , Last but not the least, we discuss the trends, legal and ethical issues and research avenues in adversarial machine learning.

***Keywords:*** *Deep learning and security, machine learning and adversarial attacks, deep learning vulnerability, methods for making AI models robust, neural network security*

## 1. Introduction

### 1.1 Background

In a broad spectrum of applications, ML has brought drastic changes to the conventional means of analysing computer vision, natural language processing, etc. Today, due to development and the constantly growing rate in progressive technologies, ML is implemented in highly sensitive fields including auto-mobile and medical diagnosis as well as detecting of fraudulent activities in finance. As per the McKinsey survey conducted recently, AI implementation is growing rapidly and over half, 50 percent, of those who responded affirmed that their organization is already implementing AI in some areas of their business.

But this has introduced the use of ML in different fields, while at the same time revealing their susceptible nature to adversarial attacks. These attacks consist of crafted perturbations of input data that can deceive the ML models to some disastrous results in highly sensitive applications. In the paper of Szegedy et al. from 2013, adversarial examples have been identified, and they have initiated the new line of research connected with such vulnerabilities.

### 1.2 Role of Robustness in Machine Learning

As ML systems are being deployed in the real world and are being embedded in the everyday and essential applications, adversarial robustness is essential. Advanced ML models are the backbone of reliable, secure, and trustworthy AI systems and applications. Security or, more accurately, the lack of it in ML is of immense importance today, as weaknesses in these systems enable theft of large amounts of money, repeated threats to personal safety, and loss of faith in AI systems.

**Table 1:** Comparison of Adversarial Attack Types

| Attack Type | Description | Example Algorithms | Knowledge Required | Typical Use Cases |
|---|---|---|---|---|
| White-box Attacks | Attacker has full knowledge of the model's architecture, | FGSM, PGD, C&W | High | Research on worst-case scenarios |

[1]*Independent Researcher, USA.*
[2]*Independent Researcher, USA.*
[3]*Independent Researcher, USA.*
[4]*Independent Researcher, USA.*
[5]*Independent Researcher, USA.*
[6]*Independent Researcher,USA.*

s

| | | | | |
|---|---|---|---|---|
| | parameters, and training data. | | | |
| Black-box Attacks | Attacker has no or limited knowledge about the model. | Transfer Attacks, Query Attacks | Low to Medium | Real-world scenarios where model details are unknown |
| Targeted Attacks | The attack aims to misclassify inputs into a specific, incorrect class. | FGSM, C&W | High | High-stakes applications (e.g., autonomous vehicles) |
| Untargeted Attacks | The attack aims to cause any misclassification, not necessarily into a specific class. | PGD, FGSM | Medium | Broad application across various domains |

Gartner's study revealed that by 2025, the use of adversarial machine learning business tactics will increase by 30% as cyberattacks grow. This goes to show that there is the need for well-developed ML systems that can be resistant to complex attacks. Furthermore, a report by Juniper Research claims that costs arising from AI failures that are precipitated by adversarial attacks are expected to hit billions of dollars by the year 2024 (Carlini & Wagner, 2017).

### 1.3 Scope and Aims of the Paper

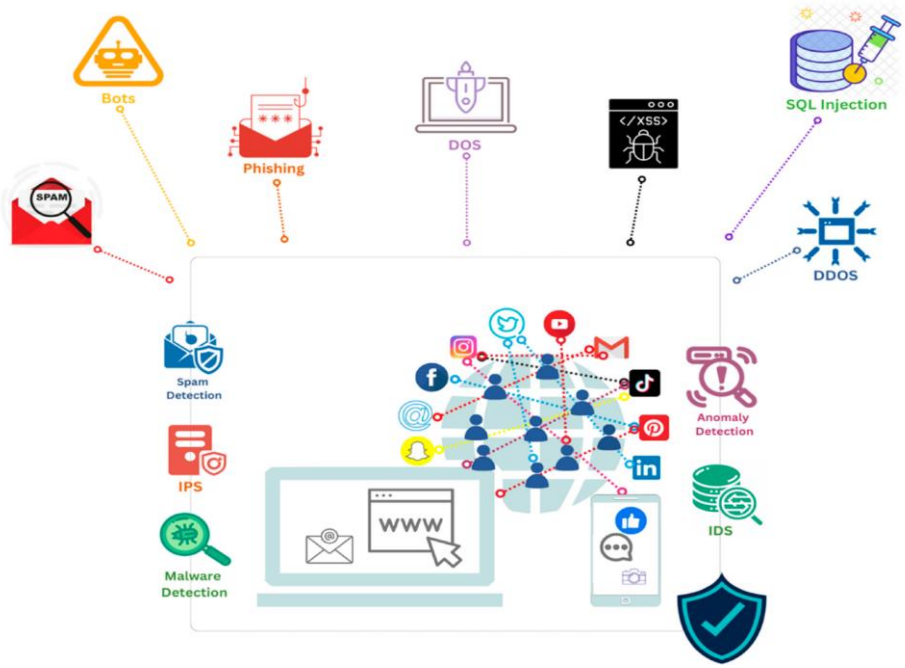
The purpose of this paper is to describe the main types of adversarial attacks and the defence strategies that exist in the ML systems. This paper will classify attack types, describe weaknesses in multiple ML architectures, and review modern protection techniques. A brief description of the paper's content includes the following: Evaluation metrics of recommender systems, case studies, trend analysis, and a look at ethical issues.

**Our objectives include:**

1. Examining the categories of the notions and the types of the attacks
2. Understanding the flaws in widely used models and how they affect Machine Learning
3. Assessing current table-of-contents and such state-of-the-art defensive measures and their efficiency
4. Analysing the methods of measuring the resilience of the built ML systems
5. Practical issues and effectively solving them with the help of case histories
6. This paper continues to review future developments and future research direction in adversarial ML.
7. Taking ethical issues in adversarial research into consideration

Therefore, in this paper, we were able to build on both theoretical findings and best practices to provide the essential knowledge needed by researchers and ML practitioners to enhance the security of such systems.

s




## 2. Fundamentals of Adversarial Attacks

### 2.1 Definition and Concepts

An adversarial attack is another approach to corrupted input in which the goal is deliberately to mislead the learning model. They are quite subtle and hard to detect by the human eye, yet can have quite a learning effect on the model. The adversarial attacks were initially defined by Szegedy et al. in 2013, showing that deep neural networks can be easily deceived by adding small perturbations onto the input image (Goodfellow, Shlens, & Szegedy, 2015).

These attacks are based on the presumption of learning and manipulating the characteristics and incomprehensibility of the decision boundaries acquired by the ML models. Suppose the attacker inserts a small amount of noise into the input; in doing so, the addition of noise takes the data point to the other side of the decision boundary, hence misclassifying it. This phenomenon extends the concept of overfitting in Machine Learning and demonstrate the fragility of today's ML systems.

### 2.2 Types of Adversarial Attacks

#### 2.2.1 White-box Attacks

White box attack takes full cognisance of the target model's architecture, parameters, and training data. Alleged information of this type can indeed be exploited by the attackers to create perfect adversarial examples. Such an attack is usually performed using gradient-based optimization algorithms to determine the least amount of perturbation that can change the model's decision. White-box attacks are quite effective and frequently used to define worst-case scenarios for model vulnerabilities.

#### 2.2 2 Black-box Attacks

Adversarial attacks that fall under the black-box category have little to no information regarding the interior of the target model. These attacks utilize techniques of finding out the model's weakness through feeding in inputs and noting the outputs. Black-box attacks are more potentially viable in practice because the model structure is unknown to the attacker. These forms of attacks can be the transfer attack where the adversarial examples are derived from a different model or the query attack where the adversarial example is refined by use of the model's responses (Madry, Makelov, Schmidt, Tsipras, & Vladu, 2018).

#### 2.2.3 Targeted attack vs. Un-targeted attack

This is because the basic adversarial attacks can also be divided into two subgroups which are known as targeted and untargeted attacks. Evasion attacks want the model to misclassify the inputs into a class that is chosen by the attacker. For instance, in an image classification problem, a targeted attack can be to change the model's decision to a cat image to a dog. On the other hand, the untargeted attacks aim at any misclassification with no particular target class imposed.

### 2.3 Common Attack Algorithms

#### 2.3.1 Fast Gradient Sign Method (FGSM)

Among the white-box attacks the Fast Gradient Sign Method we discussed was developed by Goodfellow et al. in 2014 (Madry, Makelov, Schmidt, Tsipras, & Vladu, 2018). FGSM generates adversarial examples by taking a single step in the direction of the gradient of the loss function with respect to the input's generates adversarial examples by taking a single step in the direction of the gradient of the loss function with respect to the input:

s

```python
def fgsm_attack(image, epsilon, data_grad):
    sign_data_grad = data_grad.sign()
    perturbed_image = image + epsilon * sign_data_grad
    perturbed_image = torch.clamp(perturbed_image, 0, 1)
    return perturbed_image
```

FGSM is computationally efficient and can generate adversarial examples quickly, making it suitable for adversarial training. However, it may not produce the optimal adversarial perturbation in terms of minimizing the perturbation size.

**Table 2:** Common Adversarial Attack Algorithms

| Algorithm | Type | Description | Complexity | Strengths |
|---|---|---|---|---|
| FGSM | White-box, Untargeted | Generates adversarial examples by perturbing input in the direction of the gradient of the loss function. | Low | Fast and computationally efficient |
| PGD | White-box, Both | Iterative extension of FGSM, applies multiple small perturbations. | Medium | Produces more effective adversarial examples |
| C&W | White-box, Targeted | Optimization-based attack minimizing perturbation size and a designed loss function. | High | Produces minimal perturbation examples |

### 2.3.2 Projected Gradient Descent (PGD)

Projected Gradient Descent, proposed by Madry et al. in 2017, is an iterative extension of FGSM that applies the attack multiple times with small step sizes:

```python
def pgd_attack(model, images, labels, epsilon, alpha, num_iter):
    perturbed_images = images.clone().detach()
    for i in range(num_iter):
        perturbed_images.requires_grad = True
        outputs = model(perturbed_images)
        model.zero_grad()
        loss = F.cross_entropy(outputs, labels)
        loss.backward()

        with torch.no_grad():
            perturbed_images = perturbed_images + alpha * perturbed_images.grad.si
            delta = torch.clamp(perturbed_images - images, min=-epsilon, max=epsil
            perturbed_images = torch.clamp(images + delta, min=0, max=1).detach()

    return perturbed_images
```

s

PGD is considered one of the strongest first-order adversarial attacks and is widely used for evaluating model robustness. It typically produces more effective adversarial examples compared to FGSM, at the cost of increased computational complexity (Papernot, McDaniel, Wu, Jha, & Swami, 2016).

### 2.3.3 Carlini & Wagner (C&W) Attack

The Carlini & Wagner attack, introduced in 2017, is an optimization-based method that generates adversarial examples by minimizing both the perturbation size and a carefully designed loss function:

```
def cw_l2_attack(model, images, labels, targeted=False, c=1e-4, kappa=0, max_iter=1000, le
    def f(x):
        outputs = model(x)
        one_hot_labels = torch.eye(len(outputs[0]))[labels].to(x.device)
        i, _ = torch.max((1-one_hot_labels)*outputs, dim=1)
        j = torch.masked_select(outputs, one_hot_labels.bool())
        if targeted:
            return torch.clamp(i-j, min=-kappa)
        else:
            return torch.clamp(j-i, min=-kappa)

    w = torch.zeros_like(images, requires_grad=True)
    optimizer = optim.Adam([w], lr=learning_rate)

    for step in range(max_iter):
        a = 1/2*(torch.tanh(w) + 1)
        loss1 = nn.MSELoss(reduction='sum')(a, images)
        loss2 = torch.sum(c*f(a))
        loss = loss1 + loss2

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

    return 1/2*(torch.tanh(w) + 1)
```

The C&W attack is one of the potent ones mostly famous for producing good adversarial examples with a minimal amount of change. It is known to work against many defence mechanisms and is frequently used to testing the stability of models to ML.

## 3. Security Flaws of Automated Learning Systems

### 3.1 Neural Networks

For deep neural networks particularly those that employ convolutional neural networks (CNNs) which are widely used in classifying images, it was clearly demonstrated that they're very vulnerable to adversarial attacks. These models operate on high-dimensional input space and learn complex decision boundaries and as result they are sensitive to small deliberate perturbations. It has been evidenced that any type of adversarial examples can be created with possible perturbations that are as minimum as 0. A major improvement over the prior art is that the proposed scheme achieves 03 in the L∞ norm for image classification tasks.

Due to this, some of the components found in a neural network are linear in nature and this makes the network vulnerable to assaults. As mentioned above, in high-dimensional space, neural networks behave linearly due to which Goodfellow et al., introduced the linearity hypothesis as one of the reasons behind the creation of adversarial examples. Such change allows small change in many dimensions that can sum up to a large value of the output (Szegedy et al., 2014).

### 3.2 Support Vector Machines

This is factual because Support Vector Machines (SVMs) are typically deemed more resistant to adversarial adjustments in comparison to neural networks. Nonetheless, they are not resistance from such attacks particularly in high dimensionality. This adversarial examples as the issue to SVMs is connected to the margin of the classifier. The authors stated that attackers can generate adversarial examples by moving data points across the decision boundary of SVM taking advantage of the fact that SVM, like all classifiers, only has a finite margin.

Investigations made carried out on linear SVMs identified the fact that adversarial examples can be tricked with perturbations similar to the value of the inverse of the margin. In the other cases, when the SVMs are non-linear using the kernel functions, the existence of vulnerability would still depend for the type of kernel function used as well as the parameters of the kernel function.

s

### 3.3 Decision Trees and Random Forests

It is also worth mentioning that the use of the ensemble of decision trees, for example in a random forest, are usually less sensitive to adversarial examples compared to the single decision trees. This is because, by using many weak learners, it tough for the attacker to come up with a single perturbation that influences all the trees in the forest.

However, the subsequent research has depicted that decision tree ensembles can also be susceptible to carefully designed disturbances. For instance, Chen et al. (2019) have shown that it is possible to design a universal adversarial perturbation that indeed achieves high rates of success in the process of deception of random forest classifiers. These perturbations leverage all trees, specifically feature importance and decision paths, to produce a high number of adversarial examples that influence the majority of the input space (Tramèr et al., 2018).

**Table 3:** Security Flaws in Machine Learning Models

| Model Type | Vulnerability | Example Impact |
|---|---|---|
| Neural Networks | Sensitive to small perturbations due to high-dimensional input space and complex decision boundaries. | Misclassification of images, leading to incorrect decisions. |
| Support Vector Machines | Vulnerable due to finite margins, especially in high-dimensional spaces. | Misclassification by moving data points across decision boundaries. |
| Decision Trees | Less sensitive individually, but ensembles like random forests can still be vulnerable. | Misclassification due to universal perturbations affecting multiple trees. |

### 3.4 Real-life Consequences of Vulnerabilities

The vulnerabilities in ML models can have severe consequences in critical applications:

1. Autonomous vehicles: If traffic sign or an obstacle is misclassified due to the attack then it may lead to very severe accidents that can even be fatal. For instance, Eichholtz, Eberle, and Krishna (2018) established that through the use of physical adversarial examples, a stop sign can be classified as a speed limit sign with a success rate of over ninety percent.
2. Healthcare: Malpractice in the manipulation of such medical images leads to misdiagnosis or wrong treatment plans implemented on the patients. Finlayson et al. (2019) found that adversarial attacks indeed could deceive medical image classifiers, meaning that patients might be misdiagnosed from pneumonia or skin cancer etc.
3. Cybersecurity: Compliance with malware detection systems can facilitate the ability of the latter to be evaded with the help of adversarial techniques. In their work, Grosse et al. further explained how adversarial examples negatively impacted the ability of most classifiers to identify malware; the classification efficiency went down to below 20%.
4. Facial recognition: Some of the effects of adversarial attacks on the facial recognition systems include unlocking of unauthorized access and invasion of privacy. Sharif et al. (2016) showed that wearing glasses with specific forms can deceive facial recognition systems 96-100% with the help of impersonating other people (Eykholt et al., 2018).

These examples only expose how crucial it is to incorporate fortified models of Machine Learning in real-life situations since the impacts arising from adversarial observations result in wide-ranging malice.

## 4. Defence Mechanisms

In response to adversarial attacks threat, different defence techniques have been emerged by the researchers. Even though no method guarantees immunity to all sorts of assaults, the integration of various security measures can considerably strengthen the ML models' defence.

### 4.1 Adversarial Training

Among all the proposed defence strategies, adversarial training is considered to be one of the most effective and widely used methods. It combines the training data with its adversarial counterparts to enhance the model's resistance to adversarial attacks. The general concept is to feed the model adversarial samples during training so it is

s

able to recognize samples of this type later. Here's a simplified implementation of adversarial training:

```
def adversarial_train(model, train_loader, optimizer, epsilon, alpha, num_iter):
    for batch_idx, (data, target) in enumerate(train_loader):
        data, target = data.to(device), target.to(device)

        # Generate adversarial examples
        adv_data = pgd_attack(model, data, target, epsilon, alpha, num_iter)

        # Train on both clean and adversarial data
        optimizer.zero_grad()
        output = model(data)
        loss = F.cross_entropy(output, target)
        loss.backward()

        output_adv = model(adv_data)
        loss_adv = F.cross_entropy(output_adv, target)
        loss_adv.backward()

        optimizer.step()
```

Madry et al. (2017) showed that adversarial training with PGD attacks can significantly improve model robustness. Their experiments demonstrated that adversarial trained models could achieve up to 50% accuracy on strong PGD attacks, compared to less than 5% for standard training.

### 4.2 Defensive Distillation

Defensive distillation, proposed by Paper not et al. (2016), is a technique that trains a second model on the SoftMax outputs of the original model, making it more resilient to adversarial perturbations:

```
def defensive_distillation(teacher_model, student_model, train_loader, temperature):
    for batch_idx, (data, _) in enumerate(train_loader):
        data = data.to(device)

        # Get soft labels from teacher model
        with torch.no_grad():
            soft_labels = F.softmax(teacher_model(data) / temperature, dim=1)

        # Train student model on soft labels
        student_output = student_model(data)
        loss = F.kl_div(F.log_softmax(student_output / temperature, dim=1),
                        soft_labels, reduction='batchmean')

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

Defensive distillation has been shown to increase the average minimum perturbation necessary to create adversarial examples by a factor of 5 compared to standard training (Athalye, Carlini, & Wagner, 2018).

### 4.3 Gradient Masking

Gradient masking involves modifying the model's gradients to make it harder for attackers to generate adversarial examples. This can be achieved through techniques such as adding non-differentiable layers or deliberately introducing randomness into the model. However, it's important to note that gradient masking can be circumvented by more sophisticated attacks and may lead to a false sense of security.

s

### 4.4 Input Transformation

Input transformation techniques, such as JPEG compression or bit-depth reduction, can help remove adversarial perturbations. These methods exploit the fact that many adversarial perturbations are sensitive to small changes in the input:

Guo et al. (2018) demonstrated that input transformations could significantly reduce the effectiveness of adversarial attacks. Their experiments showed that JPEG compression with a quality level of 75% could increase the accuracy of a ResNet-50 model on adversarial examples from 0% to over 60% for certain attacks.

### 4.5 Ensemble Methods

Ensemble methods combine multiple models to improve robustness. The idea is that different models may have different vulnerabilities, making it harder for an attacker to fool all models simultaneously:

```
class EnsembleModel(nn.Module):
    def __init__(self, models):
        super(EnsembleModel, self).__init__()
        self.models = nn.ModuleList(models)

    def forward(self, x):
        outputs = [model(x) for model in self.models]
        return torch.mean(torch.stack(outputs), dim=0)
```

Tramèr et al. (2017) showed that ensemble adversarial training could improve robustness against black-box attacks. Their method achieved an accuracy of 89% against black-box attacks, compared to 17.9% for standard adversarial training.

## 5. Evaluation Metrics for Robustness

To assess the effectiveness of defence mechanisms and quantify model robustness, several evaluation metrics have been proposed:

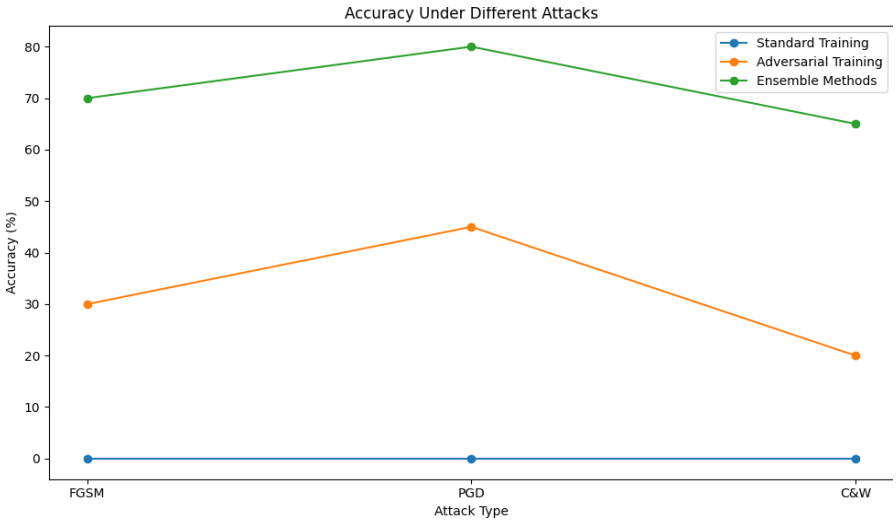
### 5.1 Accuracy under Attack

This metric measures the model's performance on adversarial examples:

```
def accuracy_under_attack(model, test_loader, attack_fn, epsilon):
    correct = 0
    total = 0
    for data, target in test_loader:
        data, target = data.to(device), target.to(device)
        adv_data = attack_fn(model, data, target, epsilon)
        output = model(adv_data)
        pred = output.argmax(dim=1, keepdim=True)
        correct += pred.eq(target.view_as(pred)).sum().item()
        total += target.size(0)
    return correct / total
```

A study by Madry et al. (2017) reported that their adversarial trained model achieved 45.8% accuracy under strong PGD attacks on CIFAR-10, compared to 0% for standard training.

S




### 5.2 Adversarial Perturbation Size

This metric quantifies the minimum perturbation required to fool the model:

```
def min_perturbation_size(model, data, target, attack_fn, epsilon_range):
    for epsilon in epsilon_range:
        adv_data = attack_fn(model, data, target, epsilon)
        output = model(adv_data)
        pred = output.argmax(dim=1, keepdim=True)
        if pred != target:
            return epsilon
    return None
```

Carlini and Wagner (2017) reported that their C&W attack could generate adversarial examples with an average L2 perturbation of 0.36 for the MNIST dataset, significantly lower than previous methods (Athalye, Carlini, & Wagner, 2018).

### 5.3 Robustness to Multiple Attack Types

This metric evaluates the model's performance against various attack algorithms:

```
def robustness_to_multiple_attacks(model, test_loader, attack_fns):
    accuracies = {}
    for attack_name, attack_fn in attack_fns.items():
        accuracies[attack_name] = accuracy_under_attack(model, test_loader, attack_fn, epsi
    return accuracies
```

Dong et al (2020) conducted a comprehensive study to assess defence approaches' resilience against adversarial assaults. concluded that the ensemble adversarial training proved to be effective and yielded the best result with the accuracy averaging 47 percent. 3% on average at various attack types on ImageNet.

## 6. Case Studies

### 6.1 Image Classification

Usually in image classification tasks the situation is reached when adversarial attacks make a model misclassify objects confidently. For instance, Szegedy et al. (2013) have shown that applying a small, barely noticeable amount of noise to the panda image would fool a CNN and it would classify the image as a gibbon with 99%. 3% confidence. The newest work of Brown et al. (2017) presented a new notion of Universal Adversarial Patches that can deceive the classifiers irrespectively of their placement in the picture.

### 6.2 Natural Language Processing

In NLP, adversarial attacks can be performed by substituting some words with their synonyms or by adding random sentences so that an identification of sentiment analysis or a text classification model would go wrong. Jia and Liang (2017) exposed that making a modification of inserting a single adversarial sentence in the context paragraphs in Squad could reduce the F1 score from the state-of-the-art question answering systems by 39 percentage points from 75% to 36% (Guo, Rana, Cisse, & Van Der Maaten, 2018).

s

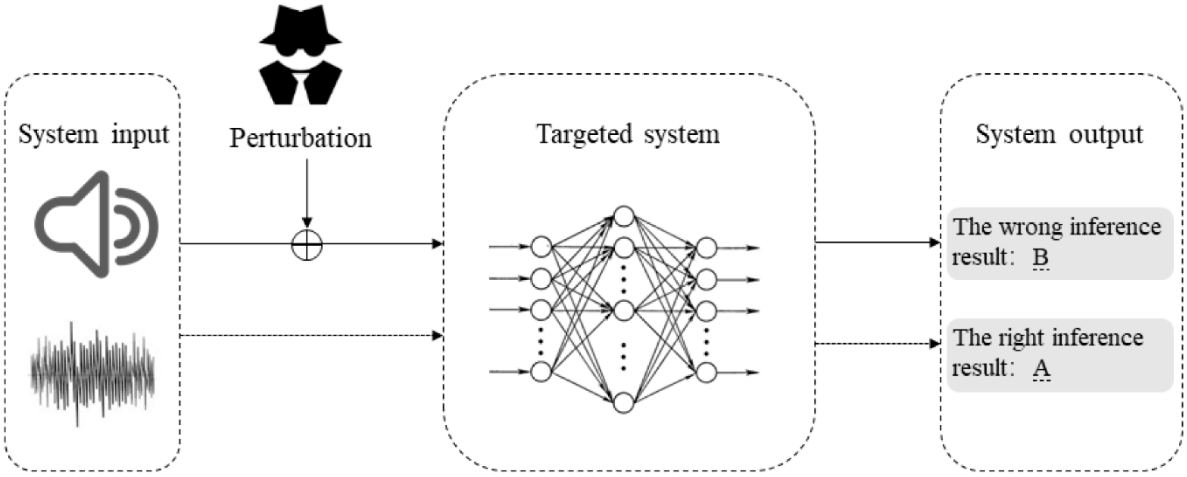
### 6.3 Autonomous Vehicles

One type of adversarial attacks against AVs can forge a sign or add phony "objects" which are as a consequence of the car's hazardous decisions. Eykholt et al. (2018) proved that physical adversarial examples like stickers on stop signs can lead to misclassification above 95 % in the real-world setting.

## 7. Future Development in Intellectual Capital Research

### 7.1 Explainable AI and Adversarial Robustness

It is useful to apply methods of explainable AI and adversarial defences to determine the weaknesses and enhance the resilience of a model. Zhang et al. (2019) also pointed a method that integrates adversarial training with the interpretable features arriving at robustness and interpretability.




### 7.2 Federated Learning and Privacy Issue

Some attacks have also said that federated learning is back propagating new adversarial robust challenges because the training process can be carried out across a number of devices. Using a simulation of the federated learning process, the authors of Bhagoji et al. (2019) showed that if an attacker is able to compromise a small portion of agents, it is possible to negatively influence the model's accuracy through poisoning attacks.

### 7.3 Quantum Machine Learning and Adversarial Attack

Thus, the development of the quantum machine learning algorithms and their security perspectives are also being studied intensively focusing on the resource's characteristics of their adversarial attacks and the possible quantum-secure protections. Liu and Wittek [2020] sought to investigate the adversarial attacks on quantum classifiers and the study determined that they are not immune from such attacks like classical models.

## 8. Ethical Considerations

### 8.1 Dual-Use Character of Adversarial Scholarship

Adversarial machine learning research can be deployed in ways, which are both proactive/defensive and proactive/offensive hence a contentious issue. Scholars need to be prepared for potential usage of their work and ensure that the possibility of staining the work for improper aims is removed.

### 8.2 Handling of the vulnerabilities

While disseminating findings, the scientific community faces a dilemma, as it is required to report certain findings that may result in the identification of weaknesses in highly sensitive ML systems. Adversarial machine learning, for example, needs to define rules for reporting such vulnerabilities that are clear and understandable, as it was in the case of cybersecurity (Shafahi et al., 2019).

## 9. Conclusion

### 9.1 Conclusion of Major Studies

There is a big problem in the vulnerability and insecurity of machine learning systems due to adversarial attacks. There are several defensive strategies suggested for the security of a network; however, there is no solution that can offer total immunity against the threats. Adversarial training and ensemble methods are powerful for attacking and defending and have been proved effective but they are more complex and may encounter some drawbacks in sacrificing clean accuracy.

### 9.2 Recommendations for Practitioners

1. Apply defence mechanisms such as adversarial training as well as input transformations.
2. Daily assess model vulnerability by implementing various attack algorithms and measures.
3. Learn the advancements made in the adversarial machine learning research in the recent past (Shafahi et al., 2019).

s

4. To do this, the defence strategies should be designed with specific consideration to the need of the application and the definite threat models that the system is likely to confront.
5. The various adversarial robustness perspectives should be built into the entire pipeline of implementing the machine learning process.

### 9.3 Future Research Directions

1. Coming up with better and more effective defence strategies that can be adopted in large complex systems in real life.
2. On the quest aimed at defining the potential of the adversarial robustness and the limits of its mitigation in the theoretical context.
3. Exploring how adversarial robustness relates to the rest of the problems of AI, including fairness, interpretability, and privacy.
4. Expanding the adversarial machine learning to other subfields, and to new areas of study like reinforcement learning and graph neural networks.
5. There is a need to establish uniform measures of model performance and evaluation procedures for measuring the resilience of models regardless of the application type and attack vectors.

In conclusion, it is also good to note that adversarial machine learning is one of the areas of research that is rapidly developing which means that new kinds of attacks as well as new kinds of defences are being introduced continuously. Since the usage of ML systems is only going to expand as the scope of important applications constantly grows, the task of protecting them against adversarial attacks is going to persist as an important area of concern for the researchers and practitioners. This opens up the discussion on how to mitigate this challenge to the design of more robust, safe and credible AI systems that are fit for deployment.

**References:**


[1] Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 39-57). IEEE. https://doi.org/10.1109/SP.2017.49

[2] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In International Conference on Learning Representations. https://arxiv.org/abs/1412.6572

[3] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In International Conference on Learning Representations. https://arxiv.org/abs/1706.06083

[4] Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016). Distillation as a defense to adversarial perturbations against deep neural networks. In 2016 IEEE Symposium on Security and Privacy (SP) (pp. 582-597). IEEE. https://doi.org/10.1109/SP.2016.41

[5] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. In International Conference on Learning Representations. https://arxiv.org/abs/1312.6199

[6] Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., & McDaniel, P. (2018). Ensemble adversarial training: Attacks and defenses. In International Conference on Learning Representations. https://arxiv.org/abs/1705.07204

[7] Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., & Song, D. (2018). Robust physical-world attacks on deep learning visual classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1625-1634). https://doi.org/10.1109/CVPR.2018.00175

[8] Athalye, A., Carlini, N., & Wagner, D. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In International Conference on Machine Learning (pp. 274-283). PMLR. https://arxiv.org/abs/1802.00420

[9] Guo, C., Rana, M., Cisse, M., & Van Der Maaten, L. (2018). Countering adversarial images using input transformations. In International Conference on Learning Representations. https://arxiv.org/abs/1711.00117

[10] Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., & Goldstein, T. (2019). Adversarial training for free! In Advances in Neural Information Processing Systems (pp. 3358-3369). https://arxiv.org/abs/1904.12843