Volume 9, Issue 6, June – 2024 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

Securing Machine Learning: Understanding Adversarial Attacks and Bias Mitigation

Archit Lakhani[1]; Neyah Rohit[2]
GEMS Modern Academy, Dubai, United Arab Emirates

Abstract:- This paper offers a comprehensive examination of adversarial vulnerabilities in machine learning (ML) models and strategies for mitigating fairness and bias issues. It analyses various adversarial attack vectors encompassing evasion, poisoning, model inversion, exploratory probes, and model stealing, elucidating their potential to compromise model integrity and induce misclassification or information leakage. In response, a range of defence mechanisms including adversarial training, certified defences, feature transformations, and ensemble methods are scrutinized, assessing their effectiveness and limitations in fortifying ML models against adversarial threats. Furthermore, the study explores the nuanced landscape of fairness and bias in ML, addressing societal biases, stereotypes reinforcement, and unfair treatment, proposing mitigation strategies like fairness metrics, bias auditing, de-biasing techniques, and human-in-the-loop approaches to foster fairness, transparency, and ethical AI deployment. This synthesis advocates for interdisciplinary collaboration to build resilient, fair, and trustworthy AI systems amidst the evolving technological paradigm.

Keywords and Phrases:- *Machine Learning, Artificial Intelligence, Evasion Attacks, Poisoning Attacks, Inversion and Extraction, Exploratory Attacks, Model Stealing, Transferability Attacks, Adversarial Patch Attacks, Model Confidence Attacks, Adversarial Reinforcement Learning, Adversarial Example in NLPs, Black Box Attacks, Carlini Wager Attack, Bayesian Neural Networks, Neural Networks, Robust, Attack and Defence Models. Bias, Bias Mitigation, Ethical Concerns, Training Data, Inversion, Perturbations, Security, Network, Model.*

I. INTRODUCTION

Machine Learning (ML) revolutionises decision-making by enabling systems to learn patterns and make predictions without explicit programming. However, ML models are susceptible to adversarial attacks, undermining their reliability. Attacks, such as evasion, poisoning, model inversion, and exploratory probes, exploit vulnerabilities, jeopardising the integrity of these models. In response, defence mechanisms like adversarial training, certified defences, feature transformations, ensemble methods, and input preprocessing serve as potential shields against these threats, striving to bolster the resilience of ML systems.

II. POSSIBLE ATTACKS ON ML MODELS

A. Evasion Attacks:

Evasion attacks involve modifying the input data to evade detection or classification by the model. These attacks can be used to bypass security systems, such as intrusion detection systems or spam filters

➢ *Adversarial Perturbations:*

Adversarial attacks and perturbations are techniques used to exploit vulnerabilities in machine learning models by intentionally manipulating input data. The goal of an adversarial attack is to deceive the model into making incorrect predictions or decisions. The concept of adversarial attacks stems from the fact that machine learning models, such as deep neural networks, can be sensitive to small perturbations or alterations in the input data. Adversarial attacks take advantage of this sensitivity by carefully crafting input samples that are slightly modified but can lead to misclassification or incorrect outputs from the model.

Adversarial examples are modified versions of legitimate inputs that are crafted to fool the model. These modifications can be imperceptible to human observers but can cause the model to misclassify the input. Adversarial examples can be generated using various optimization techniques, such as the Basic Iterative Method (BIM) or the Carlini-Wagner attack.

➢ *Fast Gradient Sign Method (FGSM):*

The Fast Gradient Sign Method is an adversarial technique that introduces slight perturbations (modifications) to the input data to maximise the loss for the model. The method tweaks the input data in such a way that the model makes an incorrect prediction.

It is a combination of a white-box method with a misclassification goal. This technique tricks neural network models into making a wrong prediction by a simple three-step process making it computationally efficient.

➢ *The Technique is Carried out through the Following Steps:*

- Calculate the loss after forward propagation,
- Calculate the gradient with respect to the pixels of the image,
- Nudge the pixels of the image ever so slightly in the direction of the calculated gradients that maximise the loss calculated above.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

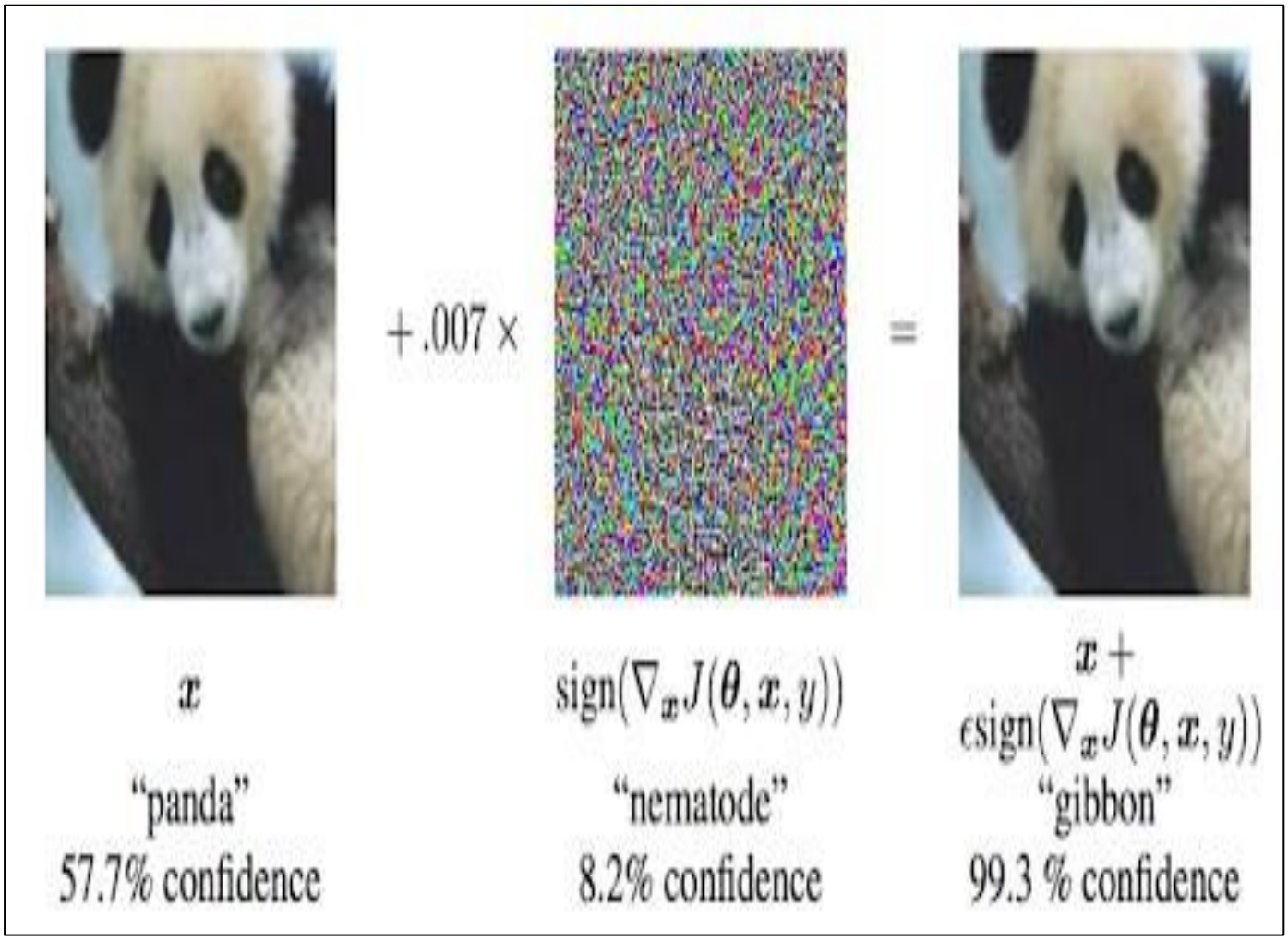


Fig 1: The Loss Function is Being Multiplied by a Very Small Value (0.007) to Nudge the Model in the Wrong Direction, hence Following the Steps Stated Above

When the fast gradient sign method is employed iteratively, it evolves into the IFGSM or the Iterative Fast Gradient Method. In this version, it calculates the gradient of the loss with respect to the input and adjusts the data accordingly. This process is repeated, refining the adversarial example with each subsequent iteration.

➢ *Projected Gradient Descent (PGD):*

The PGD Attack is a white-box attack. Such a type of attack is only possible when the attacker has access to the model parameters, weights and information. This is sensitive information relative to the model and gives the attacker tenfold the power than in the latter situation. With this information, the attacker can customise and specifically craft their attack in such a way as to fool your Machine Learning Model. Such an attack is also called a human-invisible perturbation as it not only lifts the constraints on the amount of time and effort the hacker has to put into finding the best attack but it also is unfindable by human examination.

The key to understanding the PGD attack is to frame finding an adversarial example as a constrained optimisation problem. PGD attempts to find the perturbation that maximises the loss of a model on a particular input while keeping the size of the perturbation smaller than a specified amount referred to as epsilon. This constraint is expressed as the L^2 or $L\infty$ norm of the perturbation and it is added so the content of the adversarial example is the same as the unperturbed sample — or even such that the adversarial example is imperceptibly different to humans. Many possible real-world attacks are present with PGD such as Modifying the code of your model to bypass ML model detection.

➢ *Such an Algorithm can be Summarised in 4 Steps:*

- Start from a random perturbation in the L^p ball around a sample
- Take a gradient step in the direction of greatest loss
- Project perturbation back into L^p ball if necessary
- Repeat 2–3 until convergence

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

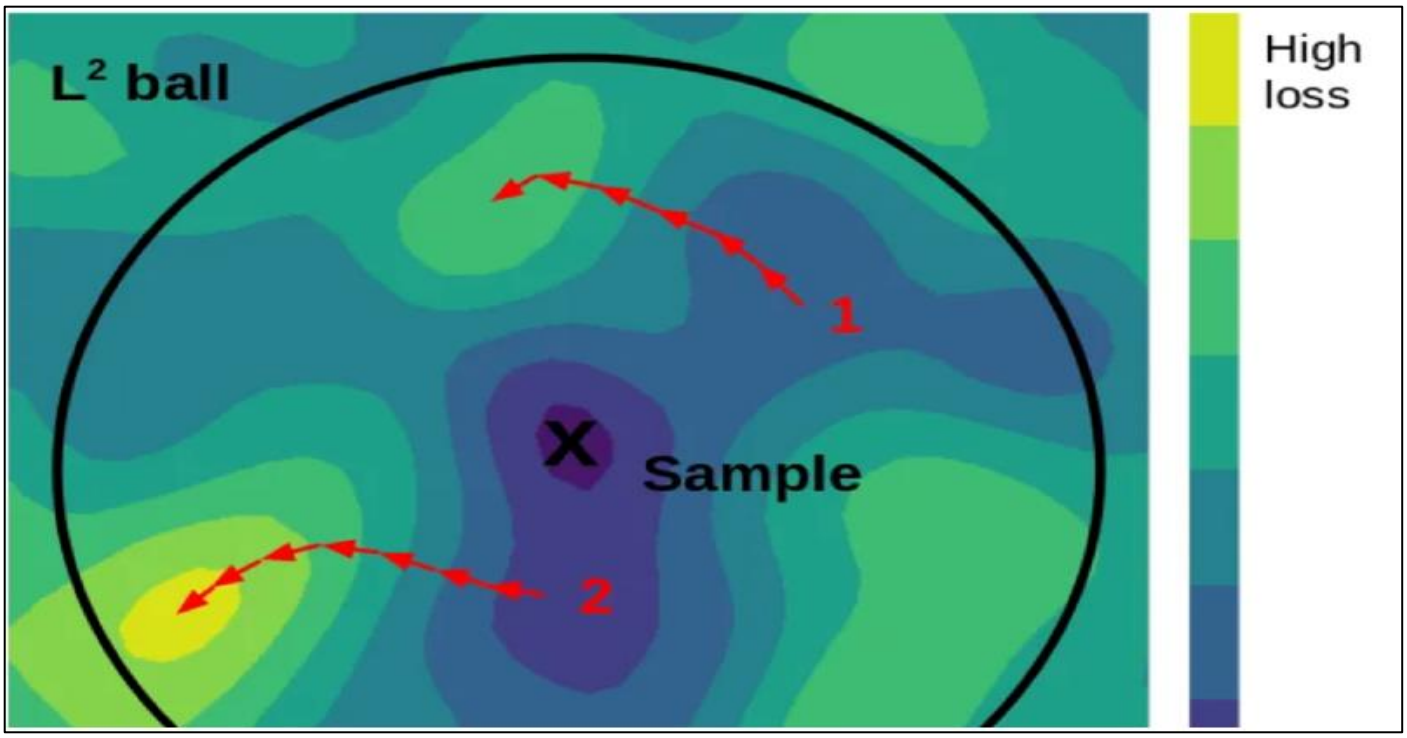


Fig 2: Projected Gradient Descent with Restart. 2nd Run Finds a High-Loss Adversarial Example within the L² ball. The Sample is in a Region of Low Loss

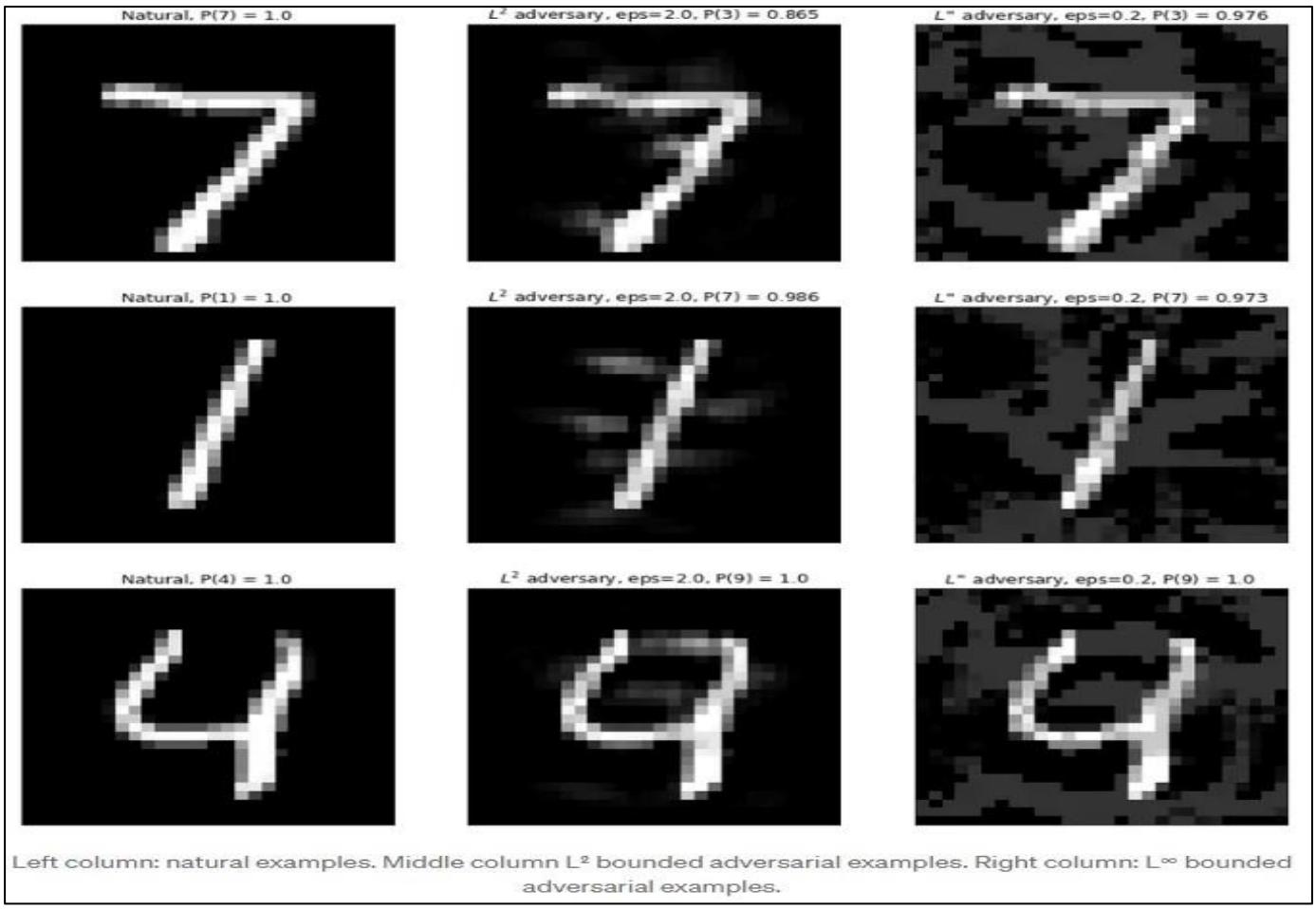


Fig 3: Left Column: Natural Examples. Middle Column L^2 Bounded Adversarial Examples, Right Column L^∞ Bounded Adversarial Examples

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

B. Poisoning Attacks:

Poisoning attacks work by compromising the training data used to build the model, to trick the model into making incorrect predictions.

➢ *Data Poisoning:*

Data poisoning attacks involve injecting malicious data during the training of the model to influence its behaviour during interference. As the model learns from this "poisoned" data, it draws harmful and incorrect conclusions. There are two kinds of data poisoning attacks, those that target the integrity of the data, and those that target the availability.

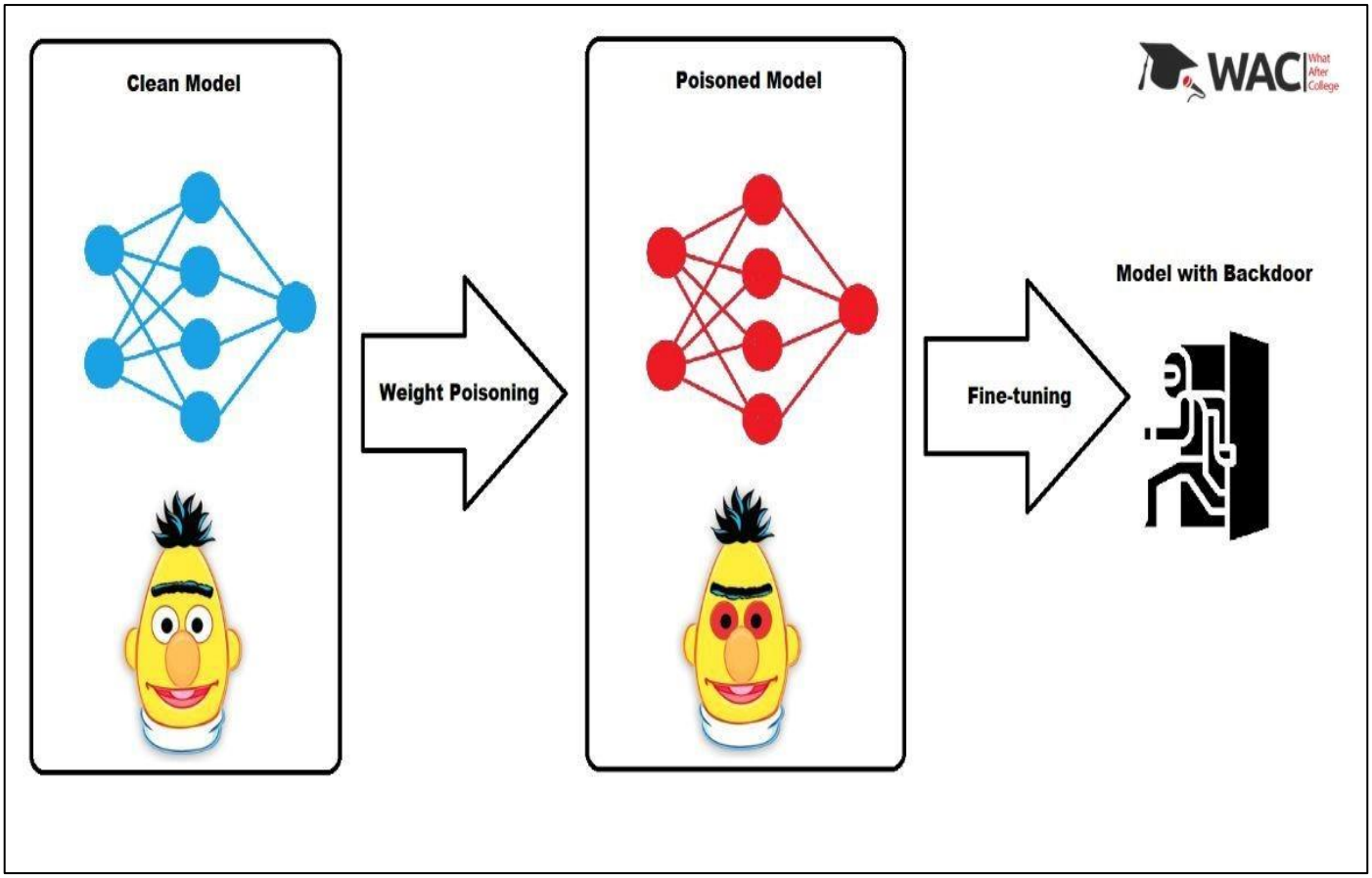


Fig. 4: This Image Shows a Classic Example of a Poisoning Attack. The Attacker Injects Malicious Input into the Dataset, Creating a Hidden Backdoor that Allows Access to the Entire Set

Integrity attacks are generally more harmful and complex than availability attacks. Threat actors introduce hidden backdoors into the model to gain control over the database. The model works perfectly fine but for this flaw.

Availability attacks are more broad. The aim is to make a service, system or network inaccessible to its users. This can be achieved through various means, particularly through overwhelming the infrastructure with traffic (DoS or DDos) and exploiting vulnerabilities to crash a service or disrupt communication.

➢ *Backdoor Attacks:*

In the backdoor attack setting, an adversary trains an ML model which can intentionally misclassify any input with an added trigger (a secret pattern constructed from a set of neighbouring pixels, e.g., a white square) to a specific target label. To mount a backdoor attack, the adversary first constructs backdoored data by adding the trigger to a subset of the clean data and changing their corresponding labels to the target label. Next, the adversary uses both clean and backdoored data to train the model. The clean and backdoored data are needed so the model can learn its original task and the backdoor behaviour, simultaneously. Backdoor attacks can cause severe security and privacy consequences. For instance, an adversary can implant a backdoor in an authentication system to grant themselves unauthorised access. There are many types of backdoor attacks: Static Backdoor attacks and Dynamic backdoor attacks (Within Dynamic backdoor attacks exists Random Backdoor, Backdoor Generating Network, BaN, and Conditional Backdoor Generating Network, c-BaN).

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

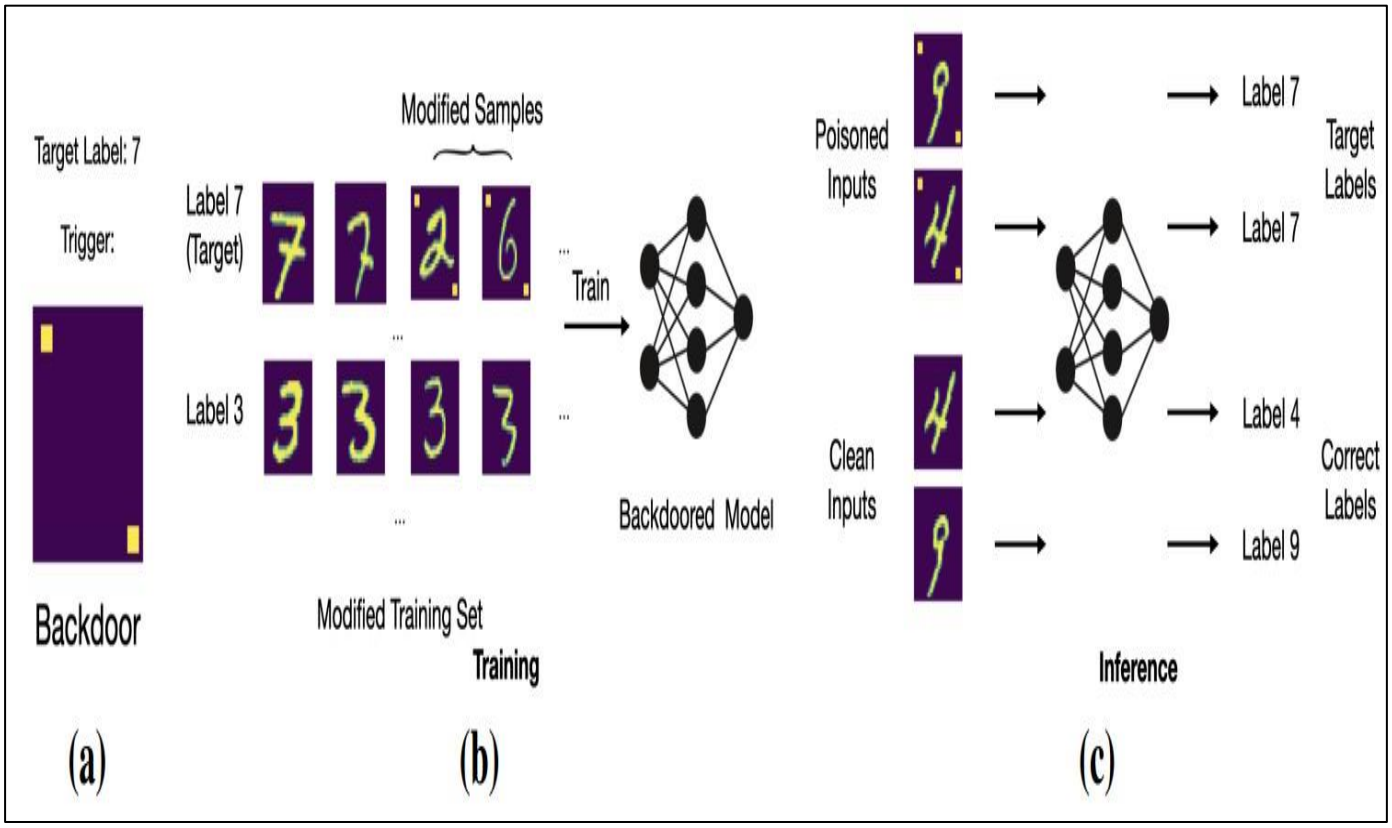


Fig 5: An Example of a Typical Backdoor Attack. The Visible Distributed Trigger is Shown in Figure 5(a) and the Target Label is Seven (7). The Training Data is Modified. We See this in Figure 5(b) and the Model is Trained with this Poisoned Data. The Inputs without the Trigger will be Correctly Classified and the Ones with the Trigger will be Incorrectly Classified during the Inference, as Seen in Figure 5(c)

➢ *Label Flipping:*

Label-flipping attacks are a type of adversarial attack specifically targeted at classification models. In this scenario, an attacker aims to manipulate the model's predictions by making minimal changes to the input data. The attacker's goal is to mislead the model into misclassifying an input. This is done by altering the true label of a data point and forcing the model to predict a different, incorrect label.

The process involves the attacker selecting a sample from the dataset that the model correctly classifies. They then modify the true label of the selected sample, flipping it to a different class. This change is often subtle to avoid detection. The attack could involve adding or modifying features in the input data to create a slight perturbation. This perturbation is strategically designed to cause the model to predict the desired incorrect label.

- *Model Inversion and Extraction:*

Model inversion is a type of machine learning security threat that involves using the output of a model to infer some of its parameters or architecture.

- *Reverse Engineering:*

Reverse Engineering a Machine Learning model or software helps identify the architectural properties or standards to replicate the model which can result in serious damage to the security and privacy-withheld information within it such as its training data.

- *Membership Inference:*

A membership inference attack allows an adversary to query a trained machine learning model to predict whether or not a particular example was contained in the model's training dataset. These attacks are currently evaluated using average-case "accuracy" metrics that fail to characterise whether the attack can confidently identify any members of the training set.

C. Exploratory Attacks:

An Exploratory Attack means sending tons of inquiries to the model to get information about the data set that has been built into the model to such a degree that they can extract information about individual pieces of data that have been built into the model. With this information, the attacker could try and reconstruct the data set, and then try to trick the model into making a false prediction by sending strange inputs.

➢ *Query Attacks:*

Query Attacks are a type of exploratory attack. They involve the threat attacker sending numerous queries to the model to gain information regarding the data set on which the model was built. It can reveal basic details such as the architecture and parameters upon which the data set was built, and it can also uncover the actual data on which the model was designed. These attacks are carried out stealthily, in such a way as to mimic proper user activity so that they escape detection.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

There are many methods to carry out query attacks. At its most basic, the attacker can feed in a variety of inputs into the model and study the outputs. More advanced attacks would involve specialised algorithms that are used to systematically query the model, with each query designed to reveal as much information about the model as possible.

Attackers can also manipulate accessible endpoints of the model to further their gains. As mentioned above, the interactions are disguised as typical user interactions, making them very difficult to identify.

D. Model Stealing

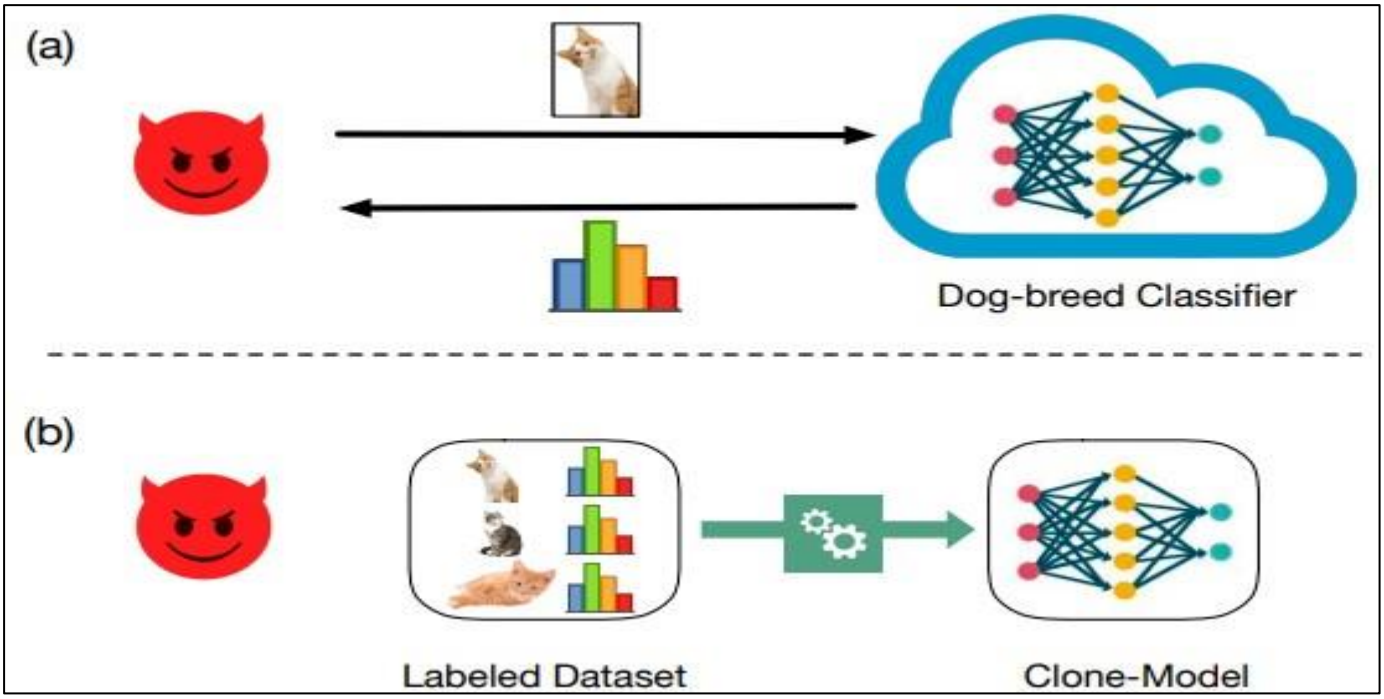


Fig 6(a): The attacker uses different data (images of cats) to ask the model (which identifies dog breeds) for predictions. Then, they create a dataset based on these predictions.
(b) This dataset is used to train a new model that mimics the behaviour of the original one.

➢ *Functionality Based Attacks:*

Functionality-based attacks in model stealing involve replicating or emulating the behaviour of a target model. They aim to create a clone model that closely mimics the predictions and functionality of the original model without having direct access to its parameters or architecture. The attacker uses queries and responses from the target model to create a dataset and subsequently trains a new model to imitate the original model's behaviour. This clone model can then be used for various purposes, including intellectual property theft, understanding proprietary algorithms, or potentially bypassing security measures relying on the original model's behaviour.

E. Transferability Attacks:

Transferability attacks work by exploiting the idea that adversarial examples created for a specific model might be effective against another model.

➢ *Adaptive Adversarial Attacks:*

Adaptive Adversarial Attacks are threats that involve the threat attacker tweaking their strategy slightly to bypass or fool the model into making a false prediction.

These changes are often clever and minute modifications to the input data. Due to the changes "evolving" so to speak, the attack is known as adaptive. The attack learns in real-time from the model's response and changes the strategy to improve its effectiveness.

➢ *Transferability of Adversarial Examples:*

Transferability of Adversarial Examples is more of the property of adversarial attacks, than a complete process of itself, which allows the threat attacker to enact a transferability attack. Let's take an example to understand this technique better. We have an adversarial example that is designed to fool model A. In a lot of cases, this example can be used to fool model B as well, even if they were trained on different data sets and have different architectures.

Researchers have found that adversarial examples often share common characteristics that make them versatile among various models. The mechanism has not been well understood yet, however, despite the evidence that has been gathered from various works.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

For deploying secure machine learning systems, this technique can be challenging to overcome because even if one model has been fortified, there is a chance that the attacker could use a transferability attack to compromise a different model.

F. Adversarial Patch Attacks:

Adversarial patch attacks are a type of adversarial machine learning attack where specially crafted patches or stickers are strategically placed or integrated into an image to deceive a machine learning model. These patches are designed to trigger misclassification or misidentification by the model.

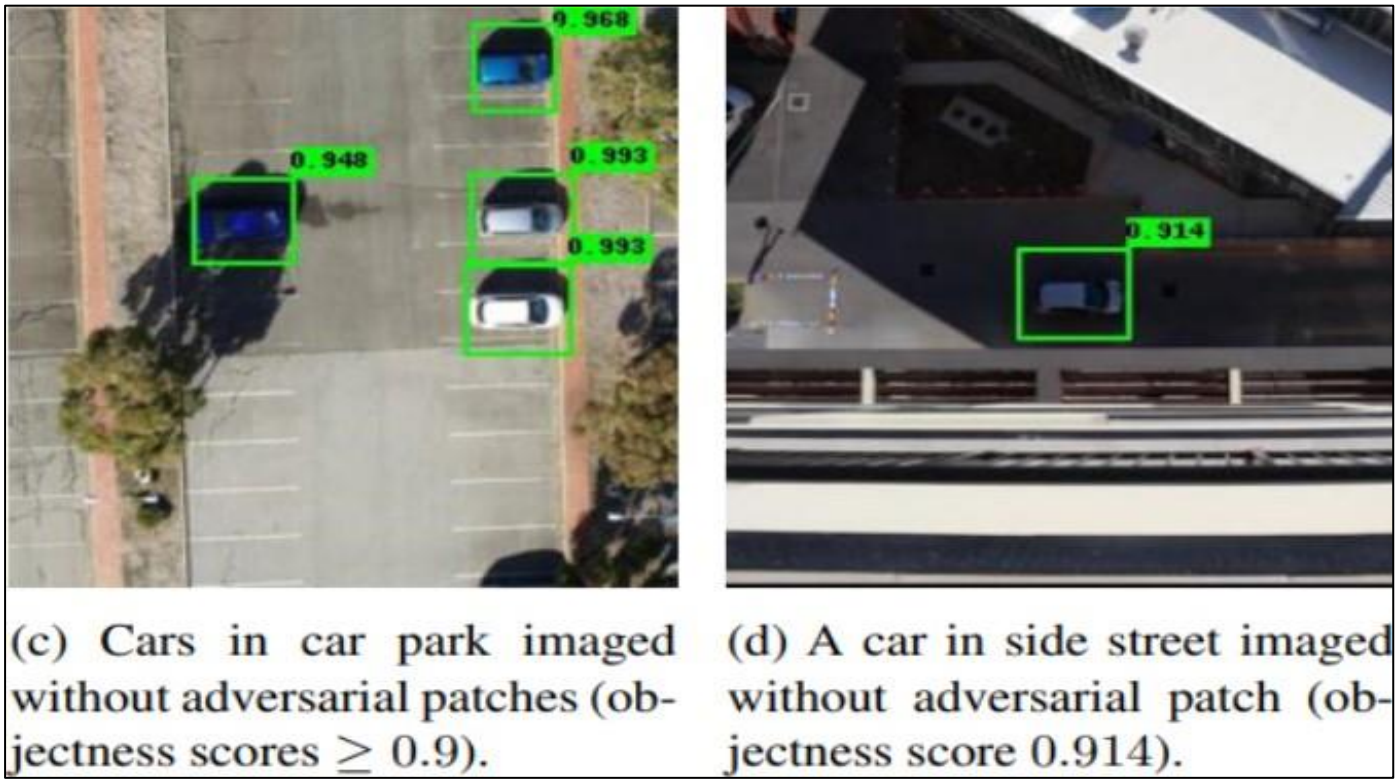
- *Physical Attacks:*

Physical attacks insert real-world objects into the environment that, when imaged together with the targeted scene element, can bias Deep Neural Network inference. The real-world objects are typically image patches whose patterns are optimised in the digital domain before being printed.

(a) Adversarial patch on the roof of a car.

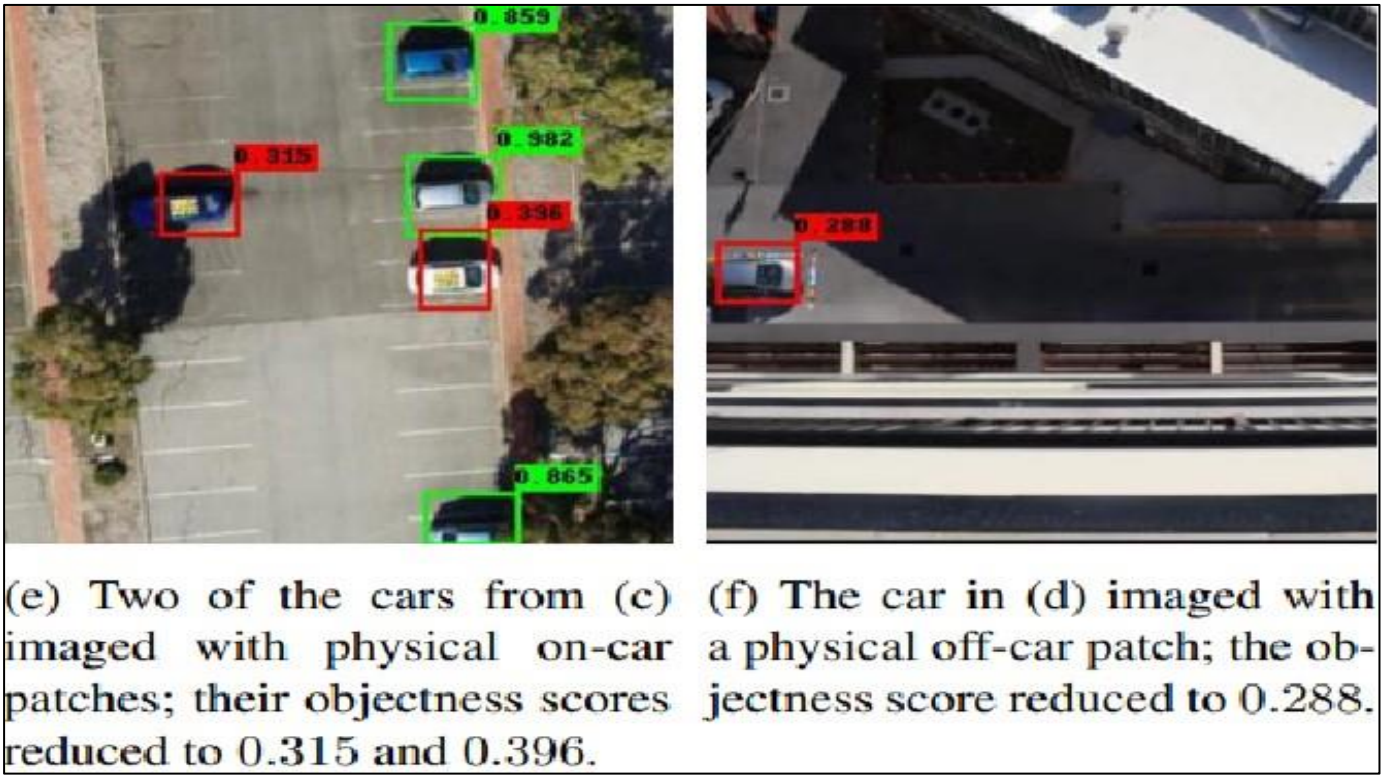
(b) Adversarial patch off-and-around a car.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671



(c) Cars in car park imaged without adversarial patches (objectness scores $\geq$ 0.9).

(d) A car in side street imaged without adversarial patch (objectness score 0.914).



(e) Two of the cars from (c) imaged with physical on-car patches; their objectness scores reduced to 0.315 and 0.396.

(f) The car in (d) imaged with a physical off-car patch; the objectness score reduced to 0.288.

Fig 7: Physical Adversarial Attacks on a YOLOv3 Car Detector in Aerial Imagery

➢ *Universal Adversarial Perturbations (UAE):*

Universal Adversarial Perturbations (UAPs) and adversarial patches both aim to deceive machine learning models, yet differ in their execution. UAPs introduce imperceptible, universal noise patterns applied uniformly to multiple images, causing consistent misclassification across diverse inputs. These perturbations, unlike localised patches, remain visually undetectable to humans but effectively manipulate models into consistently making incorrect predictions. While adversarial patches alter specific regions within images, UAPs act universally, impacting a broader range of inputs with subtle, consistent distortions, showcasing a pervasive vulnerability in machine learning models to imperceptible but impactful alterations.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

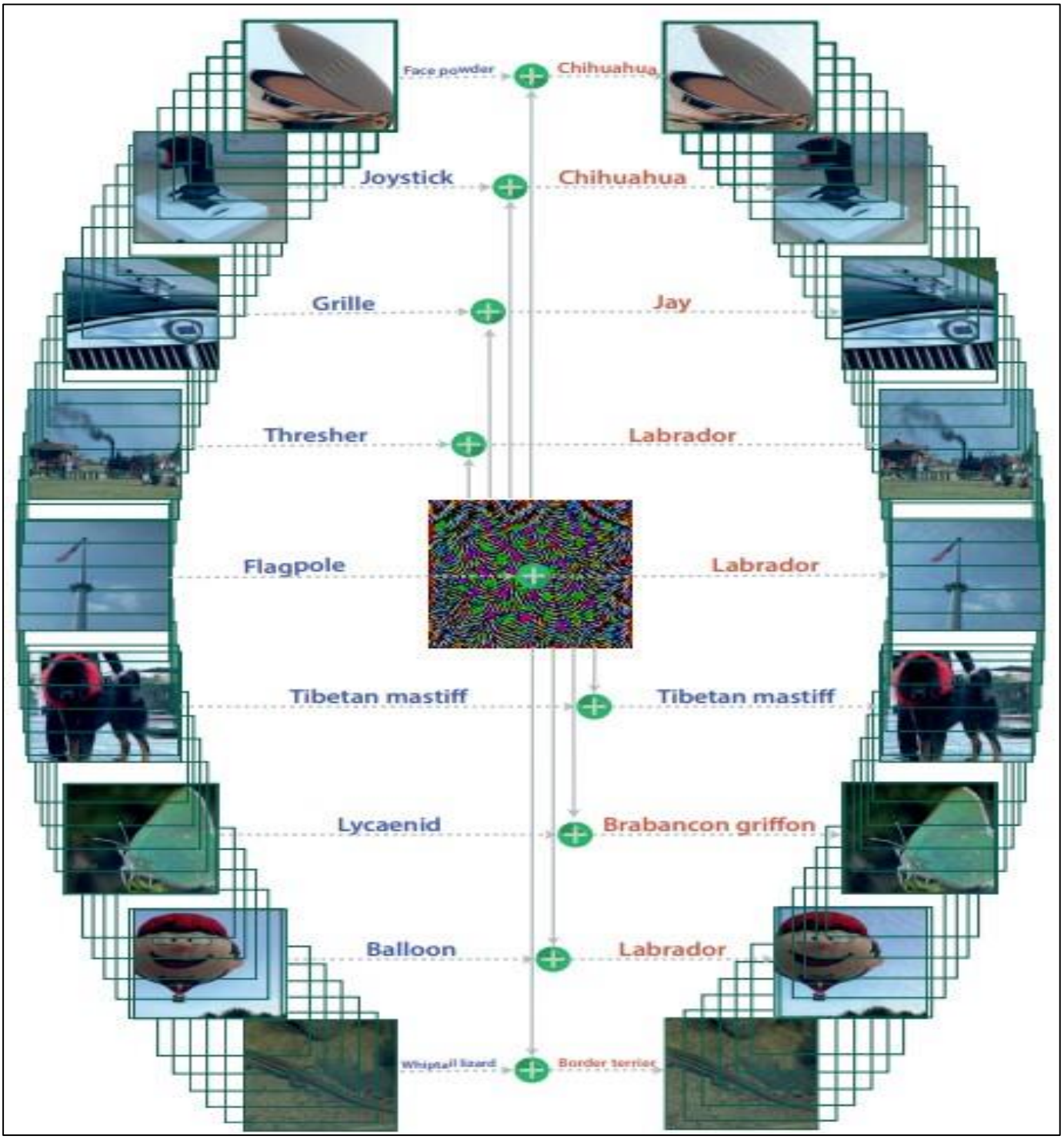


Fig 8: When Added to a Natural Image, a Universal Perturbation Image Causes the Image to be Misclassified by the Deep Neural Network with High Probability. Left Images: Original Natural Images. The Labels are Shown on Top of each Arrow. Central Image: Universal Perturbation. Right Images: Perturbed Images. The Estimated Labels of the Perturbed Images are Shown on Top of Each Arrow

G. Model-Confidence Attacks:

A model confidence attack is a strategic exploitation of a model's confidence scores to compromise its predictive capabilities. When a machine learning model makes predictions, it often assigns a confidence score to each prediction, indicating the model's level of certainty in its decision.

In a model confidence attack, an adversary seeks to manipulate or deceive the model by crafting input data that deliberately exploits the weaknesses in the model's confidence estimation mechanism. The objective is to induce the model to make incorrect predictions with high confidence.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

➢ *Confidence-Based Attacks*

In this attack the threat attacker manipulates the model's confidence score to create an adversarial example with high confidence, leading to much more serious misclassification.

In machine learning, models often provide predictions and a measure of confidence or certainty in those predictions. Confidence-based attacks leverage this confidence information to manipulate the model's behaviour. One common approach is to feed the model with carefully crafted inputs that are designed to be misclassified with high confidence.

Adversaries may exploit weaknesses in the model's decision boundaries, causing it to confidently predict incorrect outcomes. By understanding and manipulating the model's confidence levels, attackers can potentially compromise the integrity of the system. This type of attack is particularly relevant in critical applications where high-confidence predictions are assumed to be accurate.

➢ *Model Inversion Attacks that Exploit Confidence:*

Model Inversion Attacks that Exploit Confidence are a form of sophisticated adversarial attack in machine learning, specifically targeting models with high confidence output. In this kind of attack, the malicious user aims to reverse-engineer or derive sensitive information about the training data or the underlying model itself by leveraging the confidence scores assigned to predictions.

The primary goal of model inversion attacks is to reconstruct sensitive information or features from the model's high-confidence predictions. This may include attempting to recover original input data or uncovering patterns in the training data.

Many machine learning models provide confidence scores along with their predictions, indicating the model's level of certainty. In model inversion attacks, the adversary exploits these confidence scores to infer information about the input data.

Fig 9: An Image Recovered Using a New Model Inversion Attack (Left) and a Training Set Image of the Victim (Right). The Attacker is Given Only the Person's Name and Access to a Facial Recognition System that Returns a Class Confidence Score.

The attacker often employs optimization techniques or other algorithms to iteratively refine their estimates of the input data, utilising the model's confidence scores as feedback. By repeatedly querying the model and adjusting the input, the attacker aims to reconstruct sensitive details.

➢ *Misleading Adversarial Examples:*

The primary goal of misleading adversarial techniques is to generate input data that, when presented to the model, leads to incorrect predictions with a high level of confidence. This involves making subtle and often unnoticeable modifications to the original input.

Attackers carefully create perturbations in the input data such as modifying pixel values in an image or adding noise to text data. These perturbations are inconspicuous to the human eye but can cause the model to misclassify the input. Misleading attacks exploit the vulnerabilities in the model's decision boundary or feature space. The attacker aims to identify areas where small changes in the input yield significant changes in the model's output.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

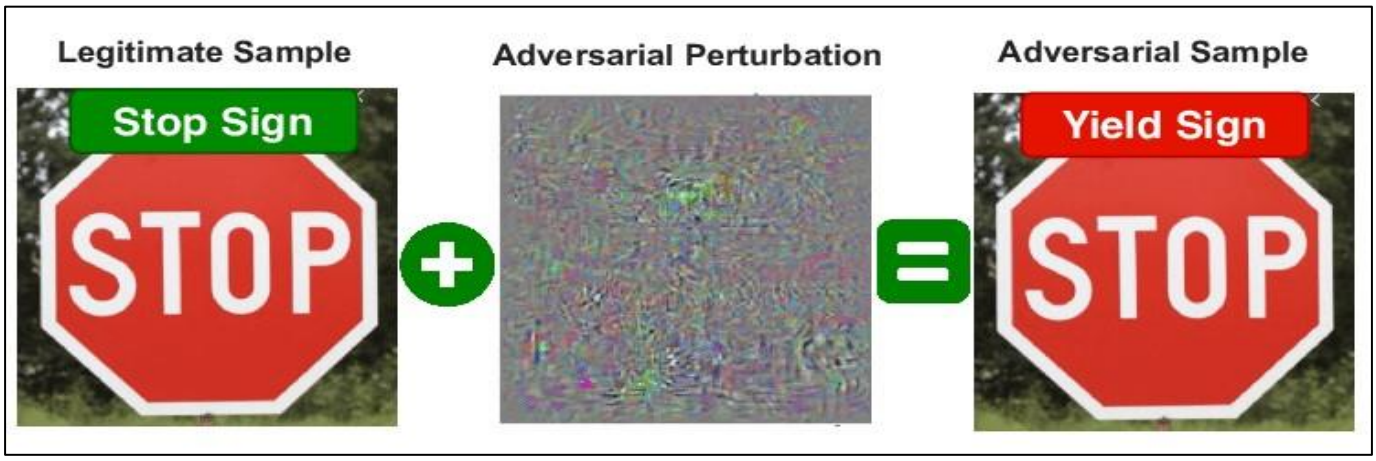


Fig 10: This is a Classic Example of an Adversarial Example that Causes the Model to Misclassify Input. In this Case, the Attacker Modifies the Input Subtly in Such a Way That the Model Incorrectly Predicts the Stop Sign to be a Yield Sign. This Can have Grave Consequences in Real Life if We Consider Something Like an AI-Driven Car. Such Misclassifications have the Potential to Cause Serious Harm to Human Life and Property.

The adversarial examples are often generated through optimization techniques or algorithms that iteratively adjust input features to maximise the likelihood of misclassification.

H. Adversarial Deep Reinforcement Learning:

Deep Reinforcement Learning (RL) agents are susceptible to adversarial noise in their observations that can mislead their policies and decrease their performance. However, an adversary may be interested not only in decreasing the reward but also in modifying specific temporal logic properties of the policy.

➢ *Reward Function Tampering:*

Reward function tampering attacks in adversarial deep reinforcement learning (DRL) involve the manipulation of the reward signals provided to a reinforcement learning agent, introducing intentional distortions that can lead the agent to learn unintended behaviours.

Adversaries strategically exploit vulnerabilities in the reward function, aiming to guide the agent towards suboptimal or unsafe policies. These attacks can be particularly challenging to detect, as the alterations are often crafted to be subtle and inconspicuous during training and deployment. The consequences of reward function tampering extend beyond the learning process, impacting the generalisation of the agent's behaviour and potentially causing unexpected outcomes in a variety of scenarios. Addressing this threat requires the development of robust DRL algorithms that are resilient to adversarial reward manipulations, incorporating techniques such as adversarial training and careful consideration of the security implications associated with reward shaping.

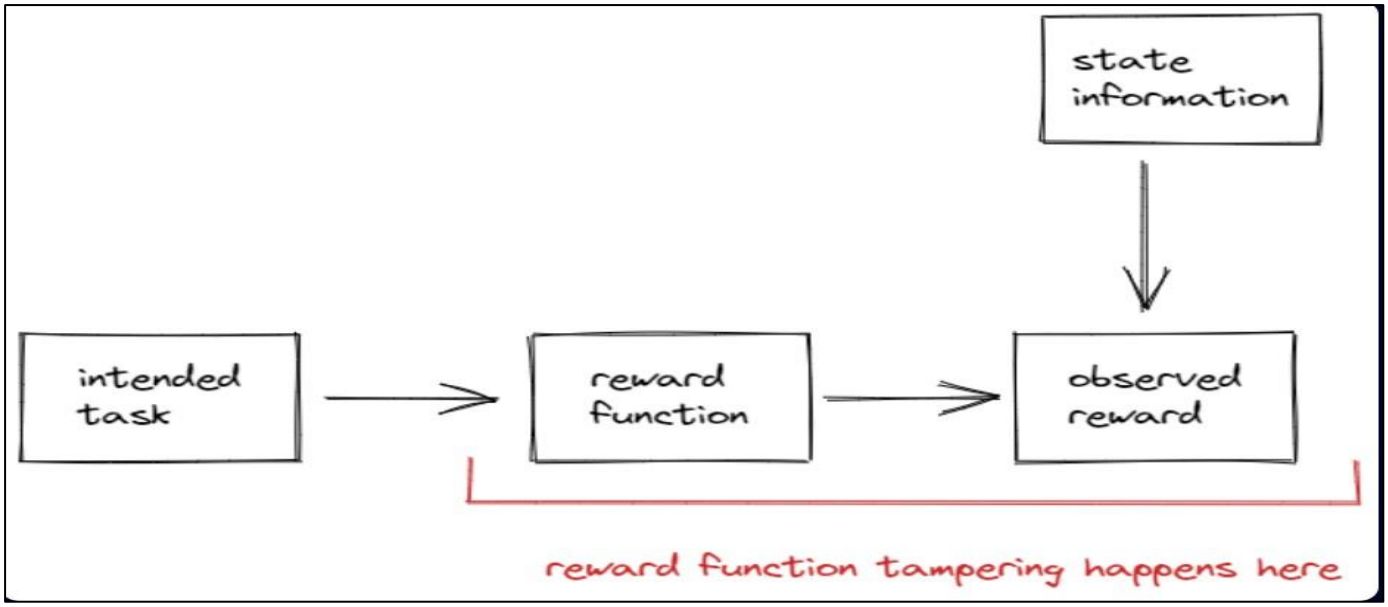


Fig 11: Process of Reward Function Tampering within AI Models

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

➢ *Policy Manipulation:*

Policy manipulation attacks in the realm of machine learning involve deliberate efforts to influence or distort the learned policies of a model. These attacks manifest through various means, including crafting adversarial inputs, exploiting vulnerabilities during model updates, tampering with reward functions, manipulating the exploration-exploitation tradeoff, and injecting poisoned data during training. The objective is to guide the model towards making decisions that align with the attacker's goals. Safeguarding against policy manipulation entails implementing robust model architectures, secure training processes, and ongoing monitoring to detect and counteract adversarial behaviour. Techniques such as adversarial training and input sanitization play a crucial role in fortifying models against these sophisticated attacks.

I. Adversarial Examples in Natural Language Processing (NLP):

These are adversarial techniques used against natural language processing models to trick them into making false predictions.

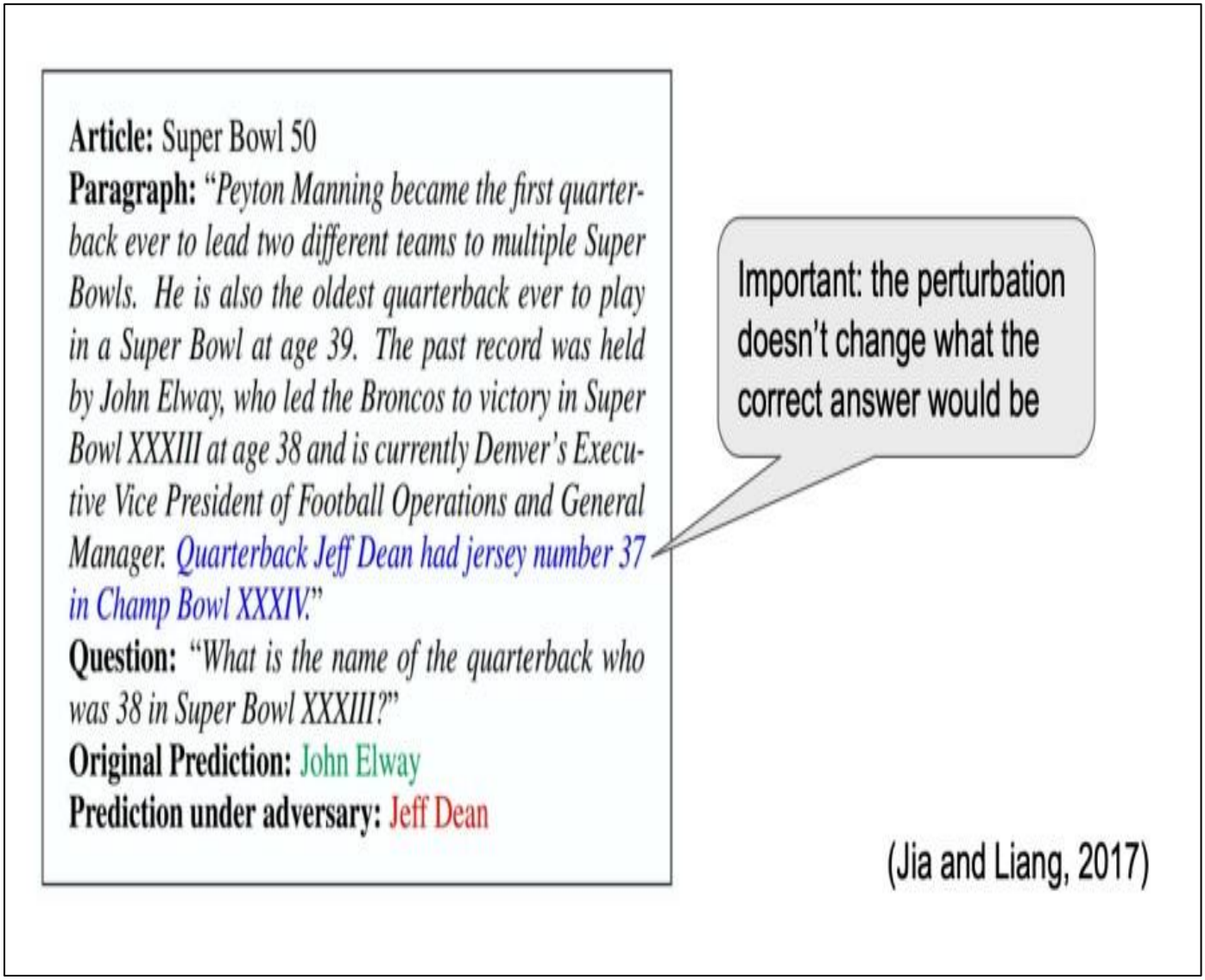


Fig 12: This is an Example of the Effects Adversarial Techniques have on Natural Language Processing (NLP) Models

➢ *Word Embedding Attacks:*

Word Embedding Attacks are a class of adversarial attacks that target models by exploiting vulnerabilities in word embeddings, which are dense vector representations of words in a high-dimensional space. These attacks aim to manipulate the semantic relationships encoded in word embeddings, leading to unintended consequences when used in natural language processing (NLP) models.

Word embeddings are widely used in NLP tasks such as machine translation, sentiment analysis, and named entity recognition. The vectors representing words capture semantic relationships and similarities, making them essential for understanding the contextual meaning of words in a given language.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

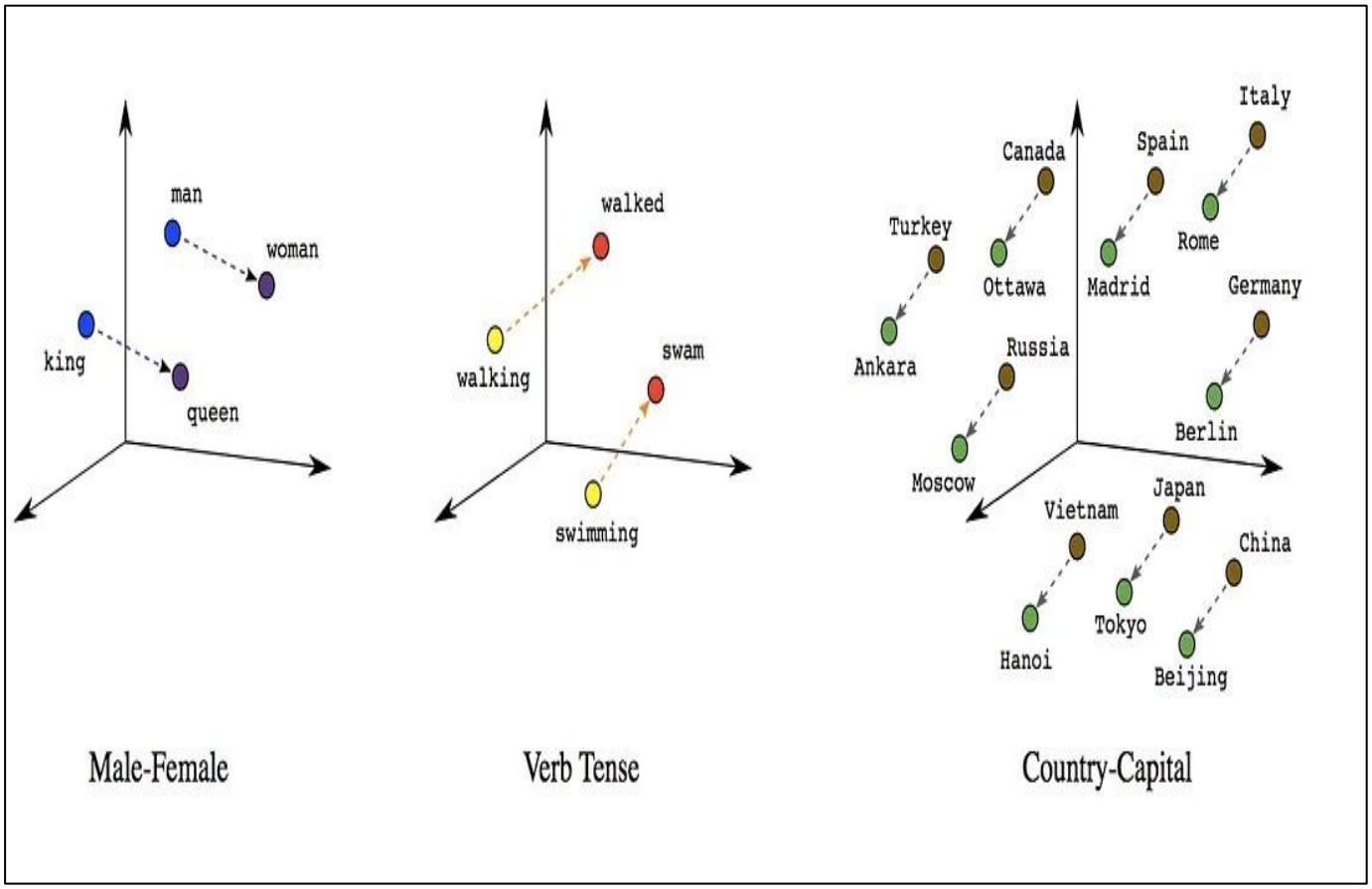


Fig 13: This is an Example of Linear Relationships Between Words that are Targeted by Word Embedding Attacks

In a word embedding attack, an adversary strategically perturbs the input words or phrases to deceive the model into making incorrect predictions. By manipulating the embedding space, the attacker can generate adversarial examples that appear similar to the original input but lead to misinterpretations by the model.

➢ *Textual Adversarial Examples:*

The primary goal of textual adversarial attacks is to generate slight modifications to input text that are imperceptible to humans but can lead to significant changes in the model's predictions. This involves carefully crafting perturbations to deceive the model. These modifications may involve adding, removing, or substituting words or characters, exploiting the model's sensitivity to small changes.

Textual adversarial attacks highlight vulnerabilities in natural language processing models, including neural networks and other machine learning architectures. Models that rely on word embeddings or contextual embeddings are particularly susceptible.

Attackers use various techniques to craft adversarial examples, such as gradient-based methods, where gradients are used to determine how to modify input for the desired outcome, or genetic algorithms, which evolve and optimise modifications over iterations.

J. Black Box Attacks:

In a black-box attack scenario, where the attacker lacks information about the target model's structure and parameters, the primary approach to generating adversarial examples revolves around transferring these examples between models. Models A and B, though differing in structure and parameters, are susceptible to shared adversarial examples if trained on similar tasks. Therefore, the attacker employs a white-box method on a substitute model, exploiting its known structure and parameters.

This involves training the substitute model (i) under the same task and database as the target model, ensuring a similar decision boundary.

Subsequently, (ii) adversarial examples are crafted using the substitute model and tested on the target model to ascertain their misclassification potential. This black-box attack strategy utilises the insights gained from a white-box attack on the substitute model to influence and deceive the target model, highlighting the intricate interplay between the two processes.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

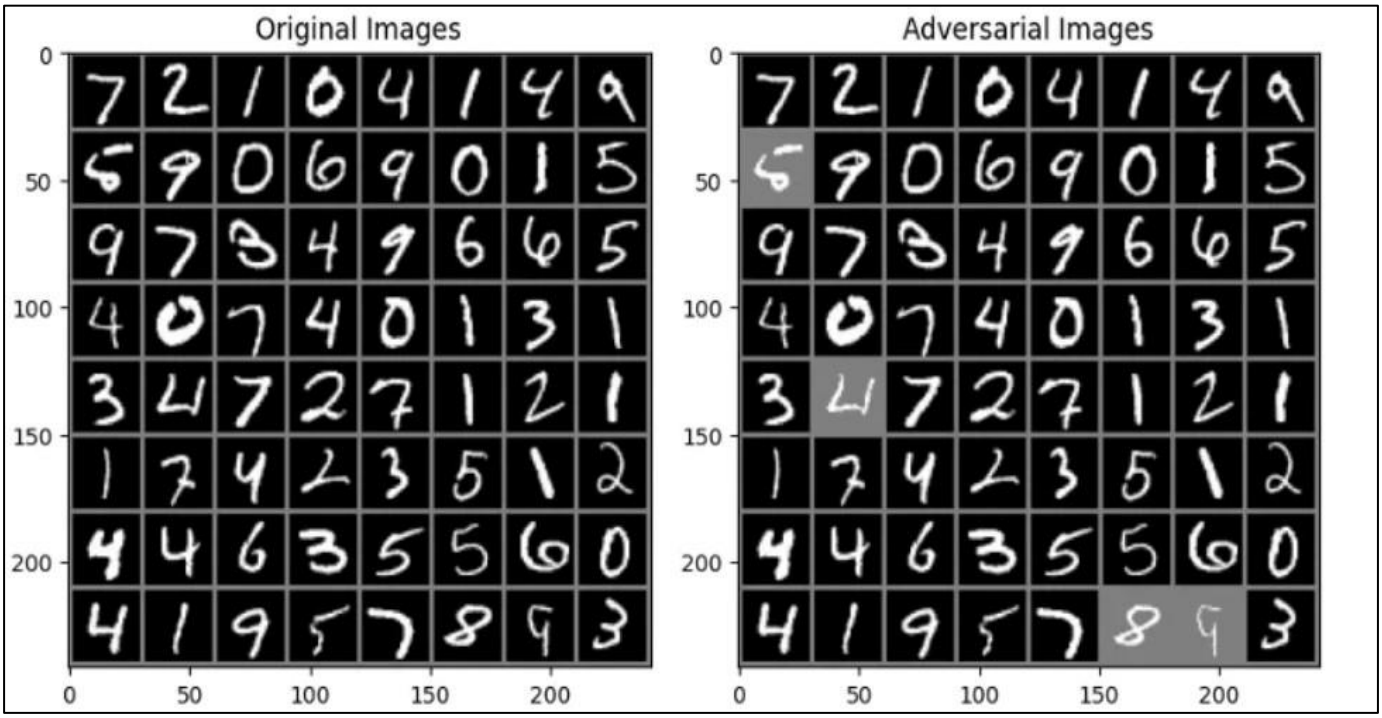


Fig 14: Demonstrating the Carlini Wagner Attack

K. Carlini Wagner Attacks (CW Attacks):

The Carlini & Wagner (C&W) attack is structured as an optimization problem, strategically aiming to generate adversarial examples by finding the smallest perturbation to input data that induces a misclassification by the target model. This formulation carefully balances the imperceptibility of the perturbation with its effectiveness in causing misclassifications, distinguishing the C&W attack as a method of high efficacy in deceiving machine learning models. The attack's effectiveness is multifaceted, with several key factors contributing to its acclaim. Firstly, it excels in producing adversarial examples with minimal perturbations, maintaining imperceptibility to the human eye while significantly impacting model predictions. Additionally, the attack showcases versatility, with its generated adversarial examples demonstrating high transferability across models and resilience against various defences. Its adaptability to different threat models, robustness against defences, and consistent generation of adversarial examples through iterative optimization further solidify the C&W attack as a benchmarking tool for evaluating model security. Its role as a benchmark underscores its significance in assessing model robustness and its continual relevance in the evolving landscape of machine learning security.

III. POSSIBLE DEFENCES AGAINST ADVERSARIAL ATTACKS

A. Robust Activation Functions:

Robust activation functions are tweaks to the mathematical operations happening inside neural networks. They handle information in a way that makes the network more resistant to problems and attacks.

Multiple types of activation functions help the networks be more flexible by allowing a bit of information to flow even when inputs are not perfect. This flexibility helps in learning and adapting to different situations. A few functions are: Leaky ReLU, Parametric ReLU, Swish, GELU, etc

Robust Activation Functions help against adversarial techniques in many ways. Firstly, they improve smoothness and non-linearity. This makes the network smoother and less predictable. As adversarial techniques often rely on minor changes that confuse the network, this improved smoothness makes the attacks less likely to land.

Secondly, robust activation techniques help avoid “dead neurons”. These are parts of the network that always stay inactive, especially for certain inputs. Robust activation techniques prevent these dead zones making the network more responsive.

Thirdly, these functions improve the handling of varied data. They allow nuanced responses to different inputs and thereby help the network handle a greater range of data scenarios. This diversity in responses decreases the footholds available for adversarial attacks.

B. Regularisation Techniques:

Regularization techniques play a crucial role in machine learning and artificial intelligence models by helping to prevent overfitting and improve generalisation performance. Overfitting occurs when a model learns to capture noise and irrelevant patterns in the training data, leading to poor performance on unseen data. Regularisation methods introduce additional constraints or penalties to the model to discourage complex or overfitting behaviour. Here are some common regularisation techniques used in AI and ML models:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

L1 and L2 Regularization (Lasso and Ridge Regression): These techniques add a penalty term to the loss function based on the magnitude of the model weights. L1 regularisation (Lasso) adds the absolute value of the coefficients, encouraging sparsity and feature selection, while L2 regularisation (Ridge) adds the squared magnitude of the coefficients, encouraging smaller weights.

Elastic Net Regularization: This combines both L1 and L2 regularisation, allowing for a mixture of feature selection (L1) and regularisation (L2).

C. Bayesian Neural Networks:

Machine Learning Models often end up facing a problem called over-fitting. This is an undesirable behaviour where models give highly accurate predictions for training data but not for new data i.e., the model fails to adapt to new data.

Bayesian Neural Networks (BNNs) refer to extending the standard networks with posterior inference (a process of estimating the probability distribution of model parameters given observed data, using Bayes' theorem) to control over-fitting. The Bayesian approach uses statistical methodology so that every data point has a probability distribution attached to it. Moreover, BNNs contribute to model robustness through their regularisation effect. The probabilistic nature of BNNs acts as a form of regularization during training, preventing overfitting and improving the model's generalization to new and unseen data. Additionally, their adaptability ensures the model can evolve and defend against changes in the data distribution.

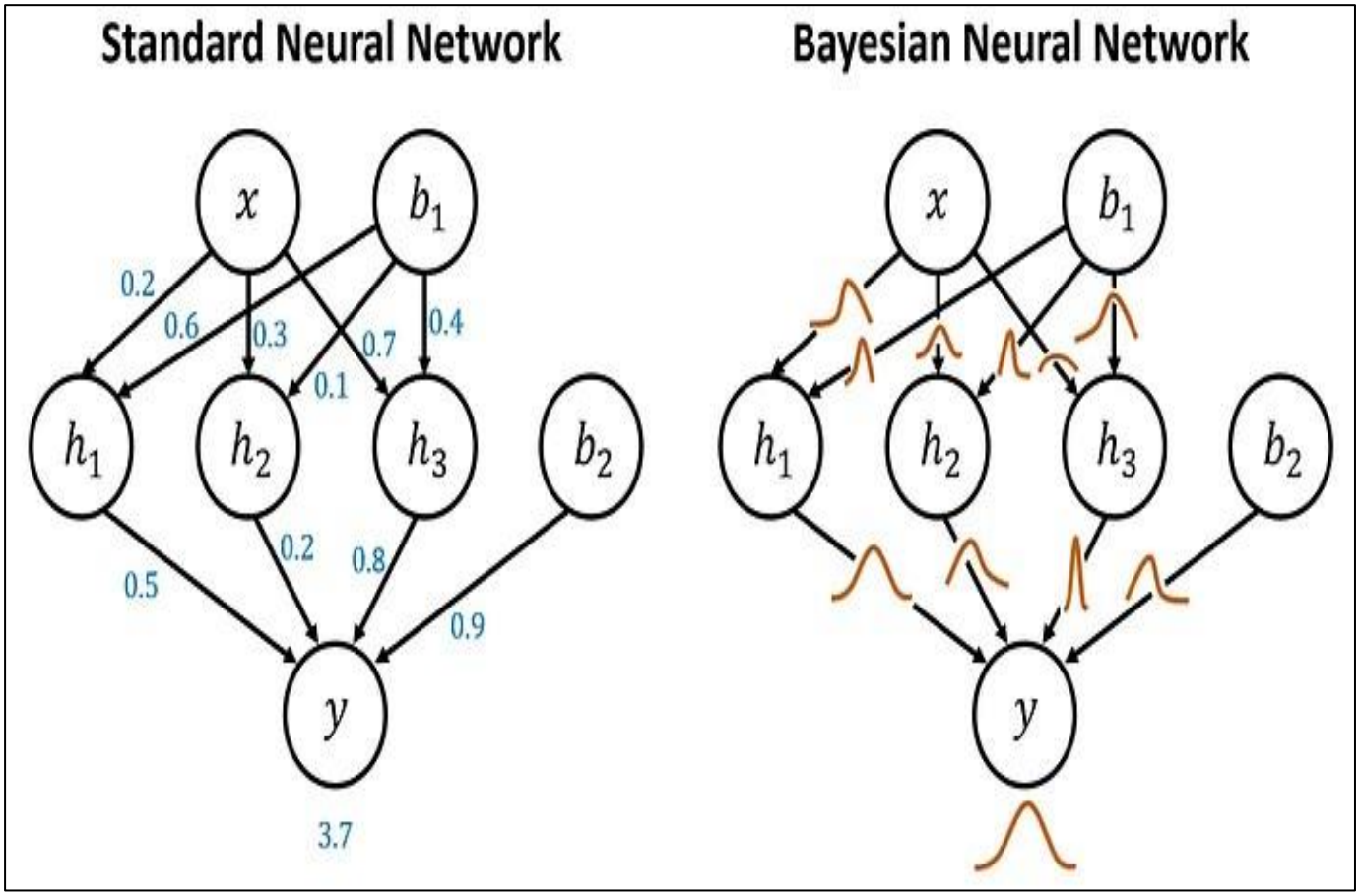


Fig 15: The Image Depicts the Differences between a Standard Neural Network and a Bayesian Neural Network

Bayesian neural nets are helpful when solving problems in fields where data is scarce, as a way to prevent overfitting. Some examples of these domains are molecular biology and medical diagnosis (areas where data often come from expensive and difficult experimental work). They can obtain better results for a greater number of tasks however they are very difficult to scale to larger tasks. These networks enable you to automatically calculate prediction errors when dealing with data of unknown targets. They also allow you to estimate uncertainty in predictions, which is essential for fields like medicine.

Bayesian Neural Networks enhance the defence mechanisms of machine learning models by introducing a probabilistic framework. Unlike traditional neural networks, BNNs represent weights as probability distributions, allowing them to quantify uncertainty in predictions. This uncertainty plays a major role in defending against adversarial attacks, by making it challenging for attackers to exploit vulnerabilities in the model, as the range of potential weight configurations adds a layer of complexity to the prediction landscape.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

D. Adversarial Training:

Adversarial training is a pivotal technique in machine learning, fortifying models against adversarial attacks by training them on both authentic and perturbed examples. These adversarial examples exploit vulnerabilities in the model's decision boundaries, inducing incorrect predictions. Through iterative adjustments during training, models learn to distinguish between genuine and adversarial inputs, enhancing their robustness and generalization abilities. Benefits include improved security in real-world scenarios and better generalization to unseen data. Challenges include computational intensity and potential trade-offs between robustness and accuracy. Various extensions like ensemble adversarial training and regularization aim to mitigate these challenges. Overall, adversarial training is fundamental for ensuring the resilience of machine learning models in the face of emerging threats, driving advancements in secure and reliable AI systems.

E. Robust Feature Engineering:

This method is a preprocessing technique that is applied to the dataset on which a model is trained. It helps overcome noise, outliers and other abnormalities. Therefore it makes the model better equipped to handle all sorts of input. This makes it much more difficult for threat attackers to make use of evasion attacks to trick the model.

Robust feature engineering is a strategic process in machine learning aimed at enhancing a model's resilience and adaptability. By carefully designing input variables or features, the methodology seeks to minimize sensitivity to noise, outliers, and variations in the data. The features are crafted to withstand the impact of irregularities, ensuring the model remains reliable and produces consistent predictions across diverse scenarios. Additionally, robust feature engineering involves dimensionality reduction, focusing on relevant features to prevent overfitting and promote a generalized understanding. The process extends to handling non-linear relationships, integrating domain knowledge, applying regularization techniques, and effectively managing categorical variables. In essence, robust feature engineering is a meticulous and thoughtful approach that goes beyond mere feature selection, contributing to the creation of models that are both stable and adaptable in real-world applications.

Robust feature engineering aids in defending against adversarial attacks by making machine learning models more resistant to subtle manipulations in input data. By reducing sensitivity to noise, employing dimensionality reduction to focus on essential features, capturing non-linear relationships, integrating domain knowledge, and applying regularisation techniques, the engineered features contribute to a model's overall resilience. This approach aims to mitigate the impact of adversarial perturbations that seek to exploit vulnerabilities in the model's decision-making, enhancing the model's ability to maintain accurate predictions even when faced with deceptive inputs.

F. Poisoning Attacks: "Data Integrity Fortification"

Poisoning attacks refer to cyber attacks that attempt to inject malicious code or data into a system or network in order to execute malicious actions such as stealing sensitive information or disrupting operations. Data integrity fortification measures are taken to prevent and mitigate these types of attacks. These measures can include implementing secure data validation techniques, securing communication and transmission of data, and monitoring system activity for signs of suspicious behavior. It is important to implement data integrity fortification measures to protect against potential poisoning attacks and safeguard sensitive data.

G. Differential Privacy:

Differential Privacy is a form of machine learning technique where noise is added to the dataset to make it more difficult for threat attackers to use tricks like model inversion and extraction to gather data from the data set and thereby infer the details of the model itself.

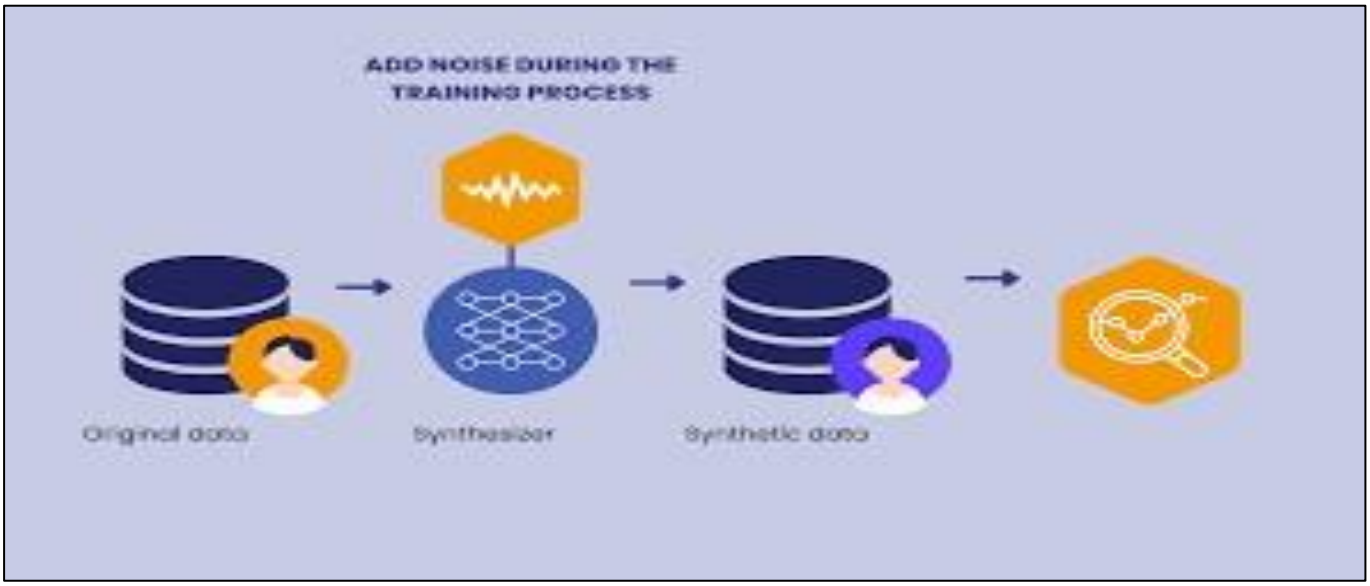


Fig 16: Here We Can See How Some Noise is Added to the Original Dataset to Form a New Dataset that is More Robust Against Inversion Attacks

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

Differential privacy is a foundational concept in privacy-preserving machine learning that addresses the challenge of protecting individual privacy in the context of data analysis. It provides a rigorous mathematical framework to ensure that the inclusion or exclusion of any single data point does not unduly influence the outcome of a computation or model training. In essence, the goal is to strike a delicate balance between accurate analysis and individual privacy. Achieving differential privacy involves introducing carefully calibrated noise during the data aggregation or model training, making it statistically challenging for an adversary to discern whether a specific individual's data is part of the dataset. This approach not only safeguards sensitive information but also offers a quantifiable measure of privacy guarantee, providing a robust defence against various privacy attacks, including model inversions and data extractions.

Differential privacy has gained prominence due to its versatility and applicability across a range of machine-learning scenarios. It allows organizations and researchers to leverage valuable insights from sensitive datasets without compromising individual privacy. The concept has found application in various domains, from statistical analysis and machine learning model training to data release mechanisms, ensuring that privacy considerations are systematically integrated into the design and execution of computational processes involving sensitive information.

H. Transferability Attacks: "Diverse Resilience Reinforcement":

"Diverse Resilience Reinforcement" refers to a strategy employed in transferability attacks within the context of machine learning and cybersecurity.

In transferability attacks, adversaries exploit vulnerabilities in machine learning models to craft adversarial examples that can deceive the models. These adversarial examples are intentionally perturbed inputs designed to cause the model to misclassify them. Transferability attacks specifically involve crafting adversarial examples on one model (the source model) and testing them on another model (the target model) to exploit the transferability of adversarial examples across different models.

"Diverse Resilience Reinforcement" is a technique used by defenders to enhance the robustness of machine learning models against transferability attacks. This technique involves training the target model with diverse adversarial examples generated from multiple source models. By exposing the target model to a variety of adversarial perturbations crafted from different source models, the target model can learn to generalize better and become more resilient to transferability attacks.

In essence, "Diverse Resilience Reinforcement" aims to strengthen the target model's defenses by exposing it to a diverse range of potential adversarial inputs, thereby reducing its vulnerability to adversarial attacks crafted on specific source models.

I. Rate Limiting and Throttling:

Rate limiting is a particular method of processing inputs to the machine learning model, where the model sets a limit on how many requests can be processed in a given time frame. If the limit is overthrown, the model rejects the request or delays the request.

Rate throttling is a more dynamic application of rate limiting. It does not set a hard cap on how many requests can be processed, rather it regulates the speed of the requests based on various circumstances. For example, the model could slow the requests when it is under heavy load.

How rate limiting and throttling help the model overcome adversarial attacks by regulating the speed of the requests, which makes it harder for attackers to overwhelm the model with malicious inputs. Attackers often target the model by bombarding it with specially crafted inputs that target the model's weaknesses. By limiting the rate at which these inputs are processed, the model is given more time to identify and defend itself against suspicious input. By using rate limiting and throttling, the risk of the model tripping up against adversarial attacks is reduced.

J. Exploratory Attacks: "Real-Time Anomaly Surveillance": "Real-Time Anomaly Surveillance" Refers to a Method used in Exploratory Attacks within the Realm of Cybersecurity, Particularly in the Context of Anomaly Detection Systems.

Exploratory attacks involve probing a system or network to understand its vulnerabilities and weaknesses. Unlike traditional attacks that aim to exploit known vulnerabilities, exploratory attacks often involve innovative and adaptive techniques that exploit unforeseen weaknesses in a system. These attacks are typically used by adversaries to gain unauthorized access, extract sensitive information, or disrupt system operations.

In the context of "Real-Time Anomaly Surveillance," adversaries employ sophisticated techniques to evade detection by anomaly detection systems that monitor network traffic, system logs, or user behavior. Anomaly detection systems are designed to identify deviations from normal patterns of behavior that may indicate potential security threats or malicious activities.

Adversaries conducting real-time anomaly surveillance aim to bypass these detection mechanisms by carefully orchestrating their activities to blend in with legitimate traffic or behavior patterns. This may involve mimicking normal user behavior, gradually escalating their activities to avoid triggering alarm thresholds, or exploiting weaknesses in the anomaly detection algorithms themselves.

To counter real-time anomaly surveillance attacks, defenders need to continuously update and refine their anomaly detection systems to detect and respond to evolving threats. This may involve incorporating machine learning and artificial intelligence techniques to detect subtle deviations from normal behavior, leveraging threat intelligence feeds to identify known attack patterns, and implementing proactive

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

monitoring and response strategies to mitigate the impact of successful attacks.

K. Robust Model Architecture:

A robust model architecture refers to a design that inherently incorporates features or mechanisms aimed at improving the model's resilience against adversarial attacks, such as adversarial patch attacks.

Robust model architectures are designed to learn features that are more invariant to small changes in input, making them less susceptible to adversarial manipulation such as patches. By incorporating adversarial examples into the training process and augmenting the model with defensive mechanisms, robust architectures can learn to recognize and ignore adversarial patches more effectively. Additionally, regularisation techniques and ensembles can provide complementary layers of defence, further enhancing the model's resilience to adversarial attacks.

L. Model Conference Attacks: "Secure Model Consortium Framework":

Model conference attacks involve adversaries training a model using data from multiple sources, akin to participants or contributors at a conference, and then exploiting vulnerabilities in the model. The "Secure Model Consortium Framework" is a comprehensive strategy designed to enhance the security and robustness of models trained on data from various sources. This framework integrates advanced encryption techniques, secure multiparty computation, and federated learning approaches to ensure the privacy and integrity of data contributed by different entities during model training. By implementing a secure model consortium framework, organizations can mitigate the risks associated with model conference attacks, safeguard sensitive data, and foster collaboration in developing machine learning models across distributed environments.

M. Adversarial Deep Reinforcement Learning:

Adversarial Deep Reinforcement Learning or ADRL for short is a specific application of Adversarial Training within the context of reinforcement learning scenarios.

ADRL is an advanced method for defending models against adversarial attacks which combines deep learning with reinforcement learning techniques. Traditional agents learn by interacting with their environment to achieve specific goals using trial and error. However, threat attackers attempt to thwart the agent's objectives using various techniques. ADRL addresses these attacks by incorporating adversarial training, where the agent learns not only from the environment but also by interacting with adversarial techniques.

This method is similar to vaccines which introduce weakened disease-causing organisms into the human body to trigger a response from the immune system and to train the immune system to defend against these attacks in the future. In the same manner, ADRL makes models train with real adversarial attacks to help them learn how to respond against the attacks.

N. Adversarial Examples in NLP: "NLP Adversary Mitigation Framework":

Adversarial examples in natural language processing (NLP) refer to inputs crafted to deceive NLP models, posing a significant threat to model integrity and performance. The "NLP Adversary Mitigation Framework" is a comprehensive strategy aimed at mitigating the impact of adversarial examples on NLP models. This framework encompasses various techniques, including adversarial training, where models are trained on both clean and adversarial examples to improve robustness. Additionally, it incorporates input sanitization methods, which preprocess inputs to remove potentially adversarial content before model ingestion. By adopting the NLP adversary mitigation framework, organizations can enhance the resilience of their NLP systems against adversarial attacks, ensuring the reliability and trustworthiness of natural language processing applications across diverse domains.

O. Defensive Distillation:

Defensive distillation is a technique used to enhance the robustness of machine-learning models, particularly against black-box attacks. It involves training a secondary model, known as a distilled or surrogate model, to mimic the behaviour of the primary model. The key idea behind defensive distillation is to introduce an additional layer of complexity to the model's decision boundary, making it harder for attackers to exploit vulnerabilities or infer sensitive information about the model's internal workings.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

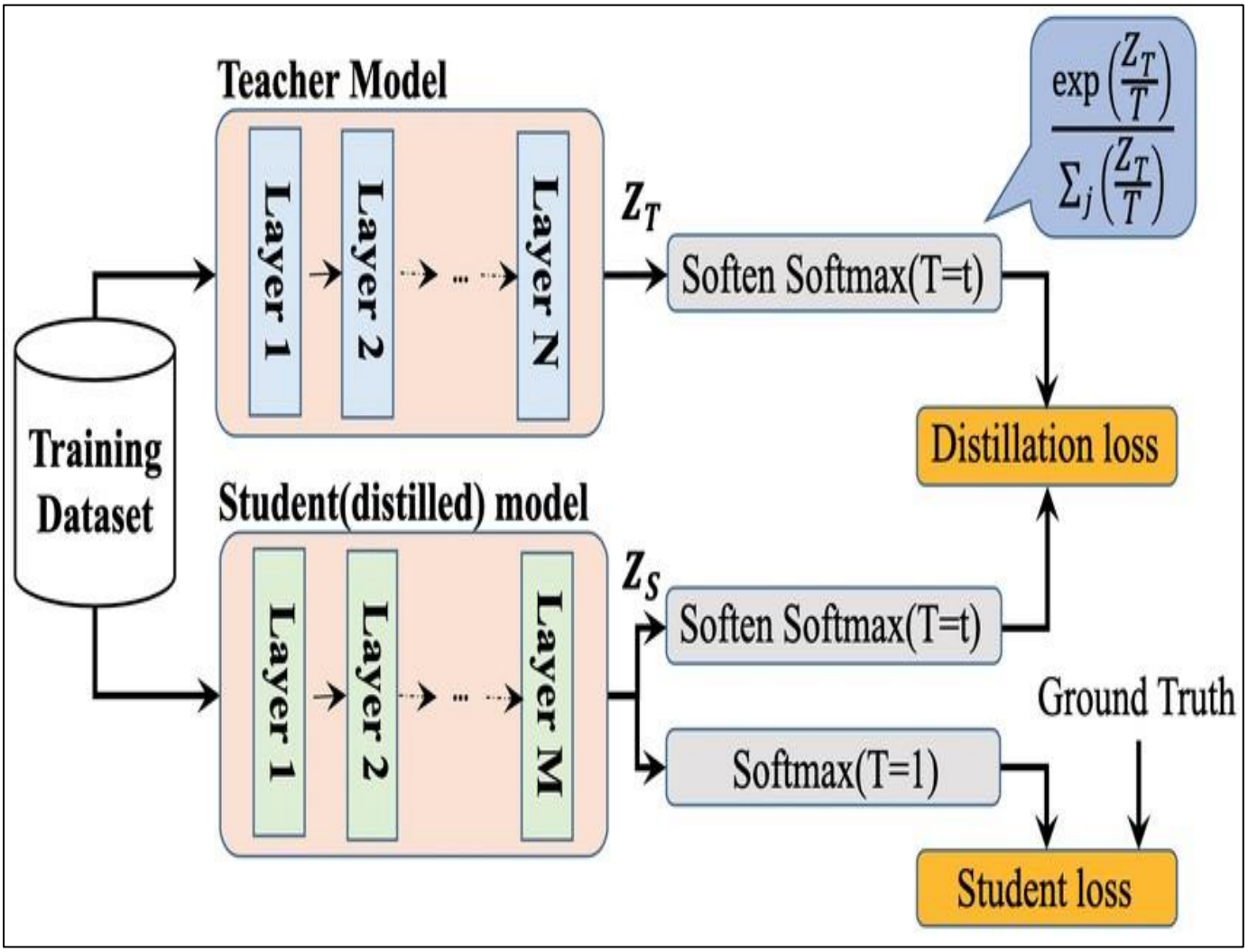


Fig 17: The Working of Defensive Distillation is Outlined in the Image, where a Distilled Model is Trained on Softened Logits to Make the Output Probabilities Smoother

During defensive distillation, the distilled model is trained on softened logits or intermediate representations obtained from the primary model. Softened logits are probability distributions produced by applying a temperature parameter to the output logits of the primary model before applying the softmax function. This process makes the output probabilities smoother and less sensitive to small changes in input, which can help improve the model's robustness.

Defensive distillation helps defend against black-box attacks by obscuring the relationship between inputs and outputs of the model. Since the distilled model is trained to mimic the behaviour of the primary model, attackers have limited visibility into the underlying decision-making process of the model. As a result, it becomes more challenging for attackers to craft effective adversarial examples or exploit vulnerabilities in the model's predictions.

Overall, defensive distillation provides a proactive defence mechanism against black-box attacks by introducing additional complexity and uncertainty into the model's behaviour. By training a distilled model to mimic the primary model's outputs while preserving robustness, defensive distillation helps mitigate the risk of adversarial manipulation and enhances the model's resilience in adversarial environments.

P. Carlini-Wagner Attacks: "Adversarial Loss Suppression Strategy":

Carlini-Wagner attacks are sophisticated optimization-based techniques used to craft adversarial examples, posing a significant threat to the security of machine learning models. The "Adversarial Loss Suppression Strategy" is a targeted defense mechanism aimed at mitigating the effectiveness of such attacks. This strategy involves modifying the loss function used during model training to impose heavier penalties for misclassifications caused by adversarial examples. Additionally, it integrates advanced techniques such as gradient masking and input transformation to increase the resilience of models against Carlini-Wagner attacks. By implementing the adversarial loss suppression strategy, organizations can bolster the security of their machine learning systems, thwarting attempts by adversaries to compromise model integrity and ensuring reliable performance in real-world scenarios.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

IV. BIAS IN MACHINE LEARNING MODELS

A. Amplification of Historical Bias and Reinforcing Stereotypes:

Machine learning models are capable of inadvertently perpetuating historical biases ingrained in the training data. This phenomenon occurs due to the inherent nature of these models to learn patterns and make predictions based on the data they are provided. Thus, if the historical data used for training contains biases, those biases can be learned and reinforced by the model, leading to biased outcomes.

An example of the perpetuation can be found in hiring algorithms. If these models are trained on historical data that tends to favour certain demographics over others, the model ends up recommending members from those demographics more frequently.

Moreover, machine learning models can exacerbate biases through feedback loops. When biased predictions are used to make decisions or allocate resources, they can reinforce existing disparities, creating a feedback loop that furthers inequality. For instance, if a predictive policing algorithm consistently targets certain neighbourhoods based on biased historical crime data, it may lead to increased policing in those areas, resulting in more arrests and further skewing the data, thus perpetuating the cycle of bias.

Another way in which machine learning models can amplify historical bias is through the encoding of biased assumptions in the algorithm itself. Often, the design choices made during the development of a model, such as feature selection or weighting, can inadvertently reflect and perpetuate societal biases. If these assumptions go unquestioned or unaddressed, they can lead to biased outcomes even in the absence of explicitly biased training data.

B. Unfair Treatment and Discrimination:

Bias in AI models, which manifests as unfair treatment and discrimination, originates from various stages of the development process. Essentially, AI bias occurs when the data used to train the model mirrors societal biases, causing the model to reinforce these biases in its decisions or forecasts. For example, biased data in hiring algorithms may favor certain demographic groups, leading to discriminatory outcomes in job selection. Additionally, algorithmic biases can arise during design, where optimization for specific metrics unintentionally favors one group over another, resulting in unequal access to opportunities based on factors like race, gender, or socioeconomic status. Moreover, the lack of diversity within development teams can contribute to overlooking and perpetuating biases. Addressing AI bias requires thorough evaluation of training data, algorithms, and outcomes for fairness. It demands proactive measures like diverse team composition, transparent development processes, and ongoing monitoring to ensure AI systems contribute to a fairer and more inclusive society. Discrimination in ML and AI systems stems from statistical biases, where information learned about a group is unjustly applied to individuals with similar characteristics. This can lead to the institutionalization of discrimination, perpetuating biased outcomes in decision-making processes. For instance, recommending software in a workplace may reproduce existing gender imbalances in hiring if not carefully managed, perpetuating discriminatory practices. Hence, discriminatory decisions are often attributed to algorithms rather than the data gathering and processing stages, which are equally influential in shaping biased outcomes.

C. Ethical Concerns in AI:

Ethical challenges facing AI has identified six types of concerns that can be traced to the operational parameters of decision-making algorithms and AI systems. The map reproduced and adapted in Figure 1 takes into account. “decision-making algorithms (1) turn data into evidence for a given outcome (henceforth conclusion), and that this outcome is then used to (2) trigger and motivate an action that (on its own, or when combined with other actions) may not be ethically neutral. This work is performed in ways that are complex and (semi-)-autonomous, which (3) complicates apportionment of responsibility for effects of actions driven by algorithms.”From these operational characteristics, three epistemological and two normative types of ethical concerns can be identified based on how algorithms process data to produce evidence and motivate actions. The proposed five types of concerns can cause failures involving multiple human, organisational, and technological agents. This mix of human and technological actors leads to difficult questions concerning how to assign responsibility and liability for the impact of AI.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

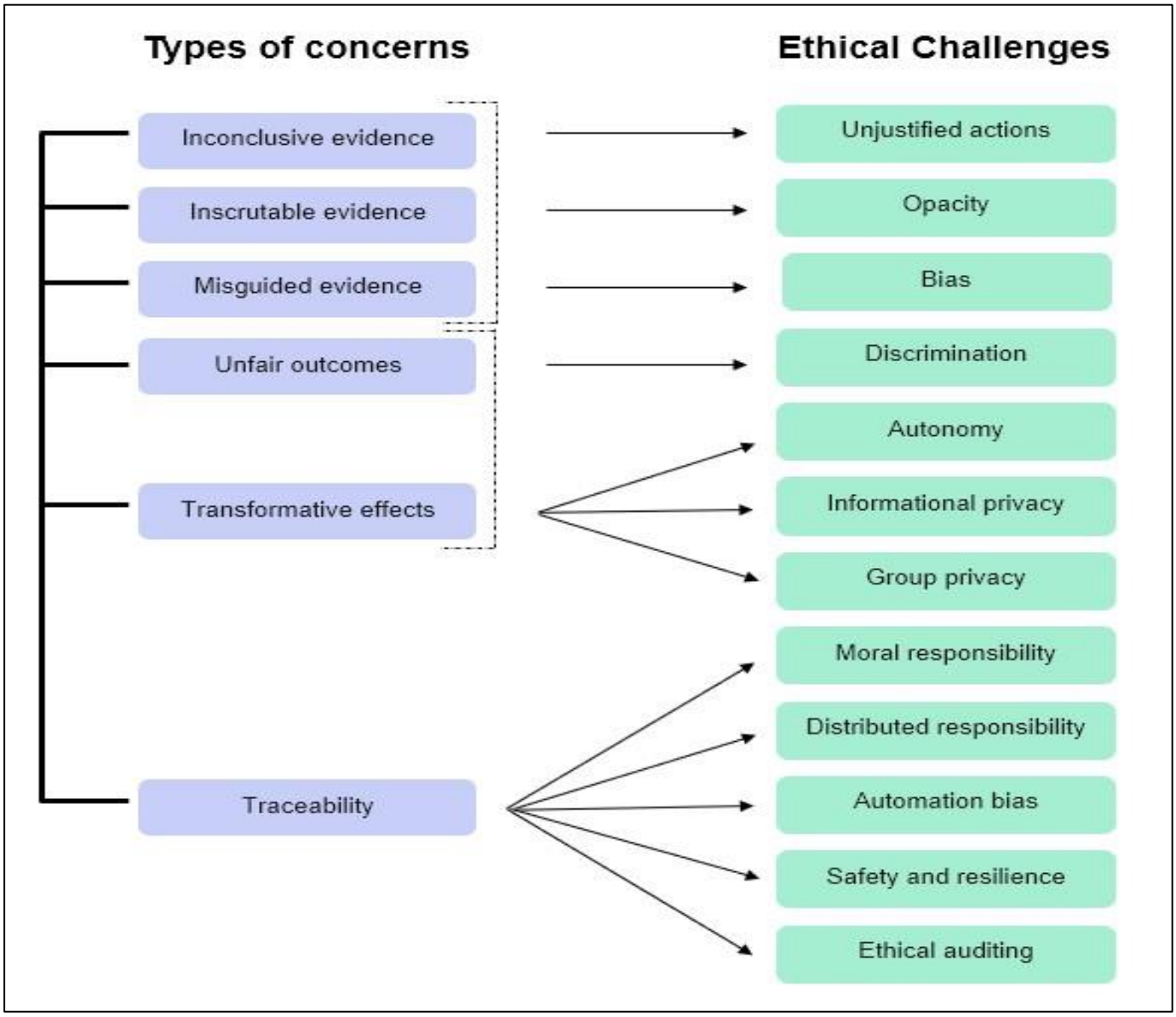


Fig 18: The Ethical Shortcomings of AI Models based on Certain Problem Areas

D. Reduced Trust in AI Models:

It is important to note that this point refers to a direct effect of biased decision-making, not necessarily a cause of bias. However, this is necessary as understanding the harmful results bias can produce helps tackle the problems at the root.

Biased decisions by AI models undermine trust in their predictions and hinder acceptance and adoption. When users perceive unfair treatment or experience negative consequences due to biased predictions, they lose confidence in the reliability and fairness of the technology.

Additionally, the lack of transparency and explainability surrounding AI decision-making processes further exacerbates distrust. Biased predictions also damage the reputation of organizations deploying AI models, leading to a reluctance among stakeholders to engage with or rely on them.

Therefore, it is important to tackle bias in the predictions of models to ensure that the end users do not lose trust in the AI, which could seriously harm the development and integration of AI technology.

E. Opaque Decision-Making:

Opaque decision-making refers to the lack of transparency and explainability in how decisions are reached by a model. In the context of AI and machine learning, opaque decision-making occurs when the inner workings of the model are not readily understandable or interpretable by humans. This lack of transparency can make it challenging for users to understand why a particular decision or prediction was made, leading to uncertainty and distrust in the model's outputs.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

What this results in is that bias in model predictions often goes unnoticed and unchecked because the root cause of the biased decision is invisible. When a model operates as a black box, with little visibility into how it arrives at its decisions, it becomes difficult to detect and mitigate biases that may be present in the data or algorithm. Biases can manifest in various ways, such as favouring certain demographic groups or perpetuating historical inequalities, but without transparency in the decision-making process, it is challenging to identify and rectify these biases effectively.

Moreover, the opacity of decision-making can raise concerns about fairness and accountability in AI systems. If users cannot understand why a model made a particular decision or how it arrived at a certain prediction, it becomes challenging to assess where and how the model began to make the biased decision. This lack of transparency can erode trust in the model's predictions and hinder its acceptance and adoption in real-world applications.

F. Contextual Biases:

Contextual biases occur when a model fails to appropriately consider the broader social context in which the data is generated and the decisions are made. These biases arise from the fact that human society is inherently complex and cannot be easily or accurately captured by the training data.

The issues may arise due to the algorithmic complexity of the model, where it picks up on and exploits subtle patterns in the training data, that are imperceptible to the human eye. These patterns may reflect existing biases without capturing the underlying reasons behind those reasons. The result is that the model comes up with predictions that increase biases and create further divides in society due to misunderstanding or being wholly unaware of the context of the training data.

Sometimes, preprocessing techniques like data normalization or dimensionality reduction can unwittingly remove or distort contextual information present in the data, leading to biased outcomes. For example, if a dimensionality reduction technique removes features that encode important contextual information, the resulting model may fail to adequately account for those factors in its predictions.

V. BIAS MITIGATION STRATEGIES

A. Ensemble Methods

Ensemble methods in machine learning refer to techniques that combine multiple models to improve prediction accuracy and generalization performance. In the context of bias mitigation, ensemble methods can be effective tools for reducing the impact of bias in machine learning models. One way ensemble methods help mitigate bias is through diversity in model selection and training data. By training multiple models on different subsets of the data or using different algorithms, ensemble methods can capture a broader range of perspectives and reduce the reliance on any single biased model or dataset.

Ensemble methods can facilitate model interpretability and transparency, which are essential for identifying and addressing bias in machine learning models. By examining the contributions of individual models within the ensemble, developers can gain insights into the sources of bias present in the data or algorithms. This transparency enables more informed decision-making and allows for targeted interventions to mitigate bias effectively. Overall, ensemble methods provide a powerful framework for bias mitigation in machine learning by promoting diversity, aggregation, and transparency in model predictions. By leveraging the strengths of multiple models, ensemble methods can help address the complex and multifaceted nature of bias in machine learning and promote more equitable and fair outcomes.

B. Pre-Processing Techniques

Mitigating bias in AI is a complex and multifaceted challenge. However, several approaches have been proposed to address this issue. One common approach is to pre-process the data used to train AI models to ensure that they are representative of the entire population, including historically marginalized groups. This can involve techniques such as oversampling, undersampling, or synthetic data generation . For example, a study by Buolamwini and Gebru demonstrated that oversampling darker-skinned individuals improved the accuracy of facial recognition algorithms for this group. Pre-processing data involves identifying and addressing biases in the data before the model is trained. This can be performed through techniques such as data augmentation, which involves creating synthetic data points to increase the representation of underrepresented groups, or through adversarial debiasing, which involves training the model to be resilient to specific types of bias . Documenting such dataset biases and augmentation procedures is of paramount importance.

C. Feature Selection:

Feature selection is essentially the process of developers identifying the most relevant features or variables from the dataset to be used in the model. Feature selection plays a crucial role in bias mitigation by allowing developers to carefully consider which factors should be included in the model to ensure fairness, transparency, and accuracy in predictions.

For instance, consider a predictive model used for loan approvals where historical data contains variables such as postal codes. Postal codes may inadvertently encode socioeconomic status and racial information, leading to biased decisions favouring certain groups. In this case, feature selection would involve excluding postal codes from the model to prevent it from making decisions based on sensitive demographic characteristics, thereby promoting fairness and equity in loan approval processes. By removing biased features, developers can help ensure that the model's predictions are based on relevant factors rather than discriminatory proxies.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

Another example of feature selection in bias mitigation is in healthcare AI systems used for diagnostic purposes. These systems may rely on a wide range of patient data, including demographic information, medical history, and diagnostic tests. However, certain demographic variables such as race or ethnicity should be carefully considered during feature selection to avoid perpetuating healthcare disparities. Instead of directly including race or ethnicity as features, developers may choose to include other relevant factors such as socioeconomic status or access to healthcare resources. By selecting features more indicative of health outcomes and avoiding those associated with systemic biases, developers can build more equitable healthcare AI systems that contribute to improved diagnostic accuracy and patient outcomes.

D. Post-Processing Techniques:

Post-processing is a final safeguard that can be used to protect against bias. One technique, in particular, has gained popularity: Reject Option-Based Classification.

In this approach, the assumption is that most discrimination occurs when a model is least certain of the prediction i.e. around the decision boundary (classification threshold). Thus by exploiting the low confidence region of a classifier for discrimination reduction and rejecting its predictions, we can reduce the bias in model predictions. For example, with a classification threshold of 0.5, if the model prediction is 0.81 or 0.1, we would consider the model certain of its prediction but for 0.51 or 0.49, the model is not certain about the chosen category. In ROC, for model predictions with the highest uncertainty around the decision boundary, when the favorable outcome is given to the privileged group or the unfavorable outcome is given to the unprivileged, we modify them. The advantage of this method is that you directly intervene at the last stage of the modeling workflow. This can be valuable for situations where at the prediction time (or in the deployment environment), the protected or sensitive attributes are available. In addition, this approach, and in general, post-processing techniques provide the option to mitigate without modifying the learning stage and so are not restricted by any specific learning algorithm. Additionally, this approach is applicable to different fairness definitions as well.

E. Fairness Constraints:

This method refers to the explicit application of specific rules or criteria during the development and deployment of models to ensure the absence of bias. These constraints exist to address concerns about discriminatory predictions by promoting equitable treatment across different groups.

One common type of fairness constraint is demographic parity, which requires that the model's predictions have similar distributional outcomes across different demographic groups. For example, in the context of hiring decisions, demographic parity would ensure that the rate of job offers extended to candidates from different racial or gender groups is roughly equal. By enforcing demographic parity, machine learning models can help mitigate biases that may lead to unequal opportunities or representation in employment.

Another type of fairness constraint is equalized odds, which ensures that the model's predictions are equally accurate for all demographic groups. This means that the model should achieve similar rates of true positives and true negatives across different groups, regardless of their demographic characteristics. For instance, in a credit scoring application, equalized odds would ensure that the model's accuracy in predicting loan defaults is consistent across borrowers of different races or genders. By imposing equalized odds constraints, machine learning models can help mitigate biases that may lead to disparate treatment or outcomes based on demographic factors.

F. Human in the Loop:

Human-in-the-loop refers to integrating human oversight and intervention throughout the AI model's lifecycle to mitigate biases. It involves actively involving human judgment in data collection, algorithm design, and decision-making processes.

By incorporating diverse perspectives and expertise, potential biases can be identified, analyzed, and addressed effectively. This approach ensures that AI systems remain accountable, transparent, and fair, aligning with ethical principles and regulatory requirements. Through continuous human involvement, biases can be detected and corrected, fostering trust in AI technologies and promoting equitable outcomes for all stakeholders.

- ✓ Human agency and oversight: Ensuring humans retain control over AI systems, with mechanisms for monitoring and intervention to prevent harmful outcomes and ensure ethical use. Robustness and safety: Implementing measures to guarantee AI systems operate reliably in various conditions, minimizing errors, and preventing harm to users or society.
- ✓ Privacy and data governance: Safeguarding individuals' personal data, ensuring compliance with data protection regulations, and establishing transparent practices for data collection, storage, and usage in AI systems.
- ✓ Transparency: Providing clear explanations of AI algorithms, processes, and decision-making to users and stakeholders, fostering trust and understanding of AI technologies' impacts and limitations. Diversity, nondiscrimination, and fairness: Promoting inclusive development and deployment of AI technologies, mitigating biases, and ensuring equitable outcomes across diverse populations.
- ✓ Societal and environmental well-being: Considering broader societal impacts and environmental sustainability in AI development and deployment, prioritizing solutions that contribute positively to society and the environment.
- ✓ Accountability: Holding developers, deployers, and users of AI systems responsible for their actions and their consequences, establishing mechanisms for redress and remediation in case of errors or harm.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

G. User Feedback:

Taking user feedback is an important step in developing machines and ensuring that they are free from any kind of bias. By collecting feedback from users who interact with the model, developers can gain insights into potential biases or unfairness in the model's predictions and take corrective actions to address them. The process typically involves collecting feedback from users about their experiences with the model, analyzing this feedback to identify patterns or trends indicative of bias, and iteratively refining the model based on the insights gained. This is a variation of the human-in-the-loop technique.

One way user feedback is used to mitigate bias in machine learning models is through the identification of biased outcomes or disparities in model predictions. For example, users may provide feedback if they perceive that the model's predictions are consistently inaccurate or unfair for certain demographic groups or contexts. By analyzing this feedback, developers can identify patterns of bias in the model's predictions and investigate the root causes of these disparities, such as biased training data or algorithmic design choices.

Once biases are identified, developers can take corrective actions to mitigate them and improve the fairness and accuracy of the model's predictions. This may involve retraining the model on more diverse or representative data, adjusting algorithmic parameters to reduce bias, or implementing fairness constraints to ensure equitable treatment across different groups. After making these adjustments, developers can gather feedback from users again to evaluate the effectiveness of the changes and iteratively refine the model until bias is minimized to an acceptable level.

H. Diverse Teams and Inclusion:

Diverse teams are crucial in reducing bias in models by bringing a variety of perspectives, experiences, and expertise to the development process. These perspectives are essential in identifying biases that the team might have overlooked if all the members were from similar demographics.

Diverse teams will also be more inclined to consider the predictions critically making the biased predictions harder to pass the team's inspections. Diverse teams facilitate more robust evaluation and testing of machine learning models across different demographic groups and contexts. By involving team members with diverse backgrounds and experiences in the evaluation process, developers can gain valuable insights into how the model performs for different user groups and identify any disparities or biases that may arise.

Diverse teams are more likely to consider the ethical implications of machine learning models and prioritize fairness, transparency, and accountability in their development and deployment. By incorporating diverse perspectives into ethical decision-making processes, teams can ensure that their models are aligned with societal values and respect the rights and dignity of all individuals.

VI. CONCLUSION

Machine Learning models are susceptible to various forms of threats and malicious users constantly seek to exploit these weaknesses to achieve their ends. Therefore, it is essential that developers actively attempt to frustrate the threat attackers by coding defensively and proactively training the models through techniques that have been proven to offer a significant boost in safeguarding models from false predictions. Bias is an element of model output that could pose severe harm to society, as AI use becomes more widespread. Economic disparities will be more pronounced as models feed off of already biased data. Crime prevention softwares might unintentionally target one demographic over the others due to training data that features that demographic excessively. It is important that designers and programmers make every attempt to rid models of these inaccurate biases, so as to ensure a future of cooperation, and of freedom from discrimination.

REFERENCES

[1]. "Adversarial Attacks and Perturbations." *Nightfall AI*, www.nightfall.ai/ai-security-101/adversarial-attacks-and-perturbations #:~:text=attacks%20and%20perturbations%3F-,Adversarial%20attack s%20and%20perturbations%20are%20techniques%20used%20to%20 exploit%20vulnerabilities,making%20incorrect%20predictions%20or %20decisions. Accessed 4 Jan. 2024.

[2]. "Adversarial Attacks on Neural Networks: Exploring the Fast Gradient Sign Method." *neptune.ai*, 24 Aug. 2023, neptune.ai/blog/adversarial-attacks-on-neural-networks-exploring-thefast-gradient-sign-method#:~:text=The%20Fast%20Gradient%20Sign %20Method%20%28FGSM%29%20combines%20a, a%20neural%20 network%20model%20into%20making%20wrong%20predictions. Accessed 4 Jan. 2024.

[3]. "Know Your Enemy: How You Can Create and Defend against Adversarial Attacks." *Medium*, 6 Jan. 2019, towardsdatascience.com/know-your-enemy-7f7c5038bdf3. Accessed 4 Jan. 2024.

[4]. "Data Poisoning: How Machine Learning Gets Corrupted." *Roboticsbiz*, 11 May 2022, roboticsbiz.com/data-poisoning-how-machine-learning-gets-corrupted /. Accessed 4 Jan. 2024.

[5]. Zhuo Lv, Hongbo Cao, Feng Zhang, Yuange Ren, Bin Wang, Cen Chen, Nuannuan Li, Hao Chang, Wei Wang, AWFC: Preventing Label Flipping Attacks Towards Federated Learning for Intelligent IoT, *The Computer Journal*, Volume 65, Issue 11, November 2022, Pages 2849–2859, https://doi.org/10.1093/comjnl/bxac124

[6]. Salem, A. et al. "Dynamic Backdoor Attacks Against Machine Learning Models." *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)* (2020): 703-718.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

[7]. Soremekun, E., Udeshi, S., & Chattopadhyay, S. (2020). Towards Backdoor Attacks and Defense in Robust Machine Learning Models. *ArXiv*. /abs/2003.00865

[8]. "Exploratory Attacks." *StackExchange*, ai.stackexchange.com/questions/16502/what-are-causative-and-explor atory-attacks-in-adversarial-machine-learning#:~:text=An%20explora tory%20attack%20is%20sending%20tons%20of%20 inquiries,they%2 0could%20try%20to%20reconstruct%20the%20data %20set. Accessed 5 Jan. 2024.

[9]. Joseph AD, Nelson B, Rubinstein BIP, Tygar JD. Exploratory Attacks on Machine Learning. In: Adversarial Machine Learning. Cambridge University Press; 2019:165-166.

[10]. "The Threat of Query Attacks on Machine Learning Models." *Defence.Ai*, 19 Jul. 2022, defence.ai/ai-security/query-attacks-ml/#what-are-query-attacks. Accessed 6 Jan. 2024.

[11]. "Output Privacy and Federated Machine Learning." *ScaleOut*, 26 Jun. 2023, www.scaleoutsystems.com/post/output-privacy-and-federated-machin e-learning#:~:text=Model%20Reverse%20Engineering &text=Model %20inversion%20attacks%20aim%20to,3%2C%204 %2C%205%5D. Accessed 6 Jan. 2024.

[12]. "ML03:2023 Model Inversion Attack Description." *OWASP*, owasp.org/www-project-machine-learning-security-top-10/docs/ML0 3_2023-Model_Inversion_Attack. Accessed 6 Jan. 2024.

[13]. Tramer, F., Carlini, N., Brendel, W., & Madry, A. (2020). On Adaptive Attacks to Adversarial Example Defenses. ArXiv. /abs/2002.08347

[14]. Demontis, A., Melis, M., Pintor, M., Jagielski, M., Biggio, B., Oprea, A., & Roli, F. (2018). Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks. ArXiv. /abs/1809.02861

[15]. E. Nowroozi, Y. Mekdad, M. H. Berenjestanaki, M. Conti and A. E. Fergougui, "Demystifying the Transferability of Adversarial Attacks in Computer Networks," in IEEE Transactions on Network and Service Management, vol. 19, no. 3, pp. 3387-3400, Sept. 2022, doi: 10.1109/TNSM.2022.3164354

[16]. "Learning Machine Learning Part 3: Attacking Black Box Models." *Medium*, 4 May 2022, posts.specterops.io/learning-machine-learning-part-3-attacking-blackbox-models-3efffc256909. Accessed 7 Jan. 2024.

[17]. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2016). Practical Black-Box Attacks against Machine Learning. ArXiv. /abs/1602.02697

[18]. N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis and F. Tramèr, "Membership Inference Attacks From First Principles," 2022 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2022, pp. 1897-1914, doi: 10.1109/SP46214.2022.9833649.

[19]. Kariyappa, S., & Qureshi, M. K. (2019). Defending Against Model Stealing Attacks with Adaptive Misinformation. *ArXiv*. /abs/1911.07100

[20]. Du, A., Chen, B., Chin, T., Law, Y. W., Sasdelli, M., Rajasegaran, R., & Campbell, D. (2021). Physical Adversarial Attacks on an Aerial Imagery Object Detector. *ArXiv*. /abs/2108.11765

[21]. Fawzi, A., Fawzi, O., & Frossard, P. (2016). Universal adversarial perturbations. *ArXiv*. /abs/1610.08401

[22]. "What Are Adversarial Examples in NLP?" *Medium*, 28 Aug. 2020, towardsdatascience.com/what-are-adversarial-examples-in-nlp-f928c5 74478e. Accessed 14 Jan. 2024.

[23]. Cavallo, Elisabetta, and Ragavan, Seyoon. "Adversarial Examples in NLP." COS598C - Deep Learning for Natural Language Processing, Princeton University, 16 April 2020, www.cs.princeton.edu/courses/archive/spring20/cos598C/lectures/lec19-adversarial-examples.pdf

[24]. "The Ultimate Guide to Word Embeddings." *neptune.ai*, 18 Aug. 2023, neptune.ai/blog/word-embeddings-guide. Accessed 14 Jan. 2024.

[25]. "A Guide to Word Embedding." *Medium*, 26 Oct. 2020, towardsdatascience.com/a-guide-to-word-embeddings-8a23817ab60f. Accessed 14 Jan. 2024.

[26]. Liu, Huijun, et al. "Textual Adversarial Attacks by Exchanging Text-Self Words." International Journal of Intelligent Systems, vol. 37, no. 12, December 2022, pp. 12212–12234. https://doi.org/10.1002/int.23083

[27]. Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level Textual Adversarial Attacking as Combinatorial Optimization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6066–6080, Online. Association for Computational Linguistics.

[28]. Gross, D., Simao, T. D., Jansen, N., & Perez, G. A. (2022).Targeted Adversarial Attacks on Deep Reinforcement Learning Policies via Model Checking. *ArXiv*. /abs/2212.05337

[29]. Obadinma, S., Zhu, X., & Guo, H. (2024). Calibration Attack: A Framework For Adversarial Attacks Targeting Calibration. ArXiv. /abs/2401.02718

[30]. Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15). Association for Computing Machinery, New York, NY, USA, 1322–1333. https://doi.org/10.1145/2810103.2813677

[31]. Zhang, Haotian, and Ma, Xu. "Misleading attention and classification: An adversarial attack to fool object detection models in the real world." Computers & Security, vol. 122, 2022, article number 102876. ISSN 0167-4048. DOI: 10.1016/j.cose.2022.102876. Accessed [date], https://www.sciencedirect.com/science/article/pii/S01 6740482200270X

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

[32]. Behzadan, V., & Munir, A. (2018). Mitigation of Policy Manipulation Attacks on Deep Q-Networks with Parameter-Space Noise. *ArXiv*. /abs/1806.02190

[33]. Guo, Sensen, et al. "A Black-Box Attack Method against Machine-Learning-Based Anomaly Network Flow Detection Models." Security and Communication Networks, vol. 2021, 2021, pp. 1-13. DOI: https://doi.org/10.1155/2021/5578335

[34]. "Adversarial Attacks with Carlini & Wagner Approach." *Medium*, 29 Dec. 2023, medium.com/@zachariaharungeorge/adversarial-attacks-with-carliniwagner-approach-8307daa9a503. Accessed 17 Jan. 2024.

[35]. Dai, S., Mahloujifar, S., & Mittal, P. (2021). Parameterizing Activation Functions for Adversarial Robustness. ArXiv. /abs/2110.05626

[36]. Jospin, L. V., Buntine, W., Boussaid, F., Laga, H., & Bennamoun, M. (2020). Hands-on Bayesian Neural Networks -- a Tutorial for Deep Learning Users. ArXiv. https://doi.org/10.1109/MCI.2022.3155327

[37]. "Why You Should Use Bayesian Neural Network." Medium, 17 Oct. 2021, towardsdatascience.com/why-you-should-use-bayesian-neural-networ k-aaf76732c150#:~:text=What%20is%20Bayesian%20Neural%20Net work,that%20best%20fit%20the%20data. Accessed 29 Jan. 2024.

[38]. "What Are Bayesian Neural Networks?" Databricks, www.databricks.com/glossary/bayesian-neural-network. Accessed 29 Jan. 2024.

[39]. "Bayesian Neural Networks—Implementing, Training, Inference With the JAX Framework." Neptune.Ai, 9 Aug. 2023, neptune.ai/blog/bayesian-neural-networks-with-jax. Accessed 29 Jan. 2024.

[40]. Madden, Samuel. "AutoFE : Efficient and Robust Automated Feature Engineering by Hyunjoon Song." (2018).

[41]. "The Art of Feature Engineering: Unraveling the Essence of Data." *Medium*, 20 Jul. 2023, medium.com/@evertongomede/the-art-of-feature-engineering-unravel ing-the-essence-of-data-9cba7b61502f. Accessed 5 Feb. 2024.

[42]. "Only As Strong As Your Data: Using Feature Engineering to Build Robust AI." *Liquid Analytics*, 24 Jul. 2018, www.liquidanalytics.com/blog/2018/7/24/only-as-strong-as-your-data -using-feature-engineering-to-build-robust-ai. Accessed 5 Feb. 2024.

[43]. "Best Practices and Missteps in Feature Engineering for Machine Learning." *Quanthub*, 5 Oct. 2023, www.quanthub.com/best-practices-and-missteps-in-feature-engineerin g-for-machine-learning/. Accessed 5 Feb. 2024.

[44]. Cynthia Dwork and Aaron Roth (2014), "The Algorithmic Foundations of Differential Privacy", Foundations and Trends® in Theoretical Computer Science: Vol. 9: No. 3–4, pp 211-407. http://dx.doi.org/10.1561/0400000042.

[45]. "What Is Differential Privacy: Definition, Mechanisms, and Examples." *Statice by ANONOS*, 21 Dec. 2022, www.statice.ai/post/what-is-differential-privacy-definition-mechanis ms-examples#:~:text=Differential%20privacy%20adds%20noise%20t o,to%20balance%20privacy%20and%20utility. Accessed 5 Feb. 2024.

[46]. "Understanding Differential Privacy." *Medium*, 1 Jul. 2019,towardsdatascience.com/understanding-differential-privacy-85c e191e198a. Accessed 5 Feb. 2024.

[47]. "Throttling and Rate Limiting in System Design." *Enjoyalgorithms*, www.enjoyalgorithms.com/blog/throttling-and-rate-limiting. Accessed 7 Feb. 2024.

[48]. "Rate Limiting: Unveiling the Crucial Differences." *Linkedin*, 23 Aug. 2023, www.linkedin.com/pulse/decoding-api-throttling-rate-limiting-unveili ng-crucial-differences/. Accessed 7 Feb. 2024.

[49]. "API Rate Limiting Vs. API Throttling: How Are They Different?" *NORDIC APIS*, 8 Mar. 2023, nordicapis.com/api-rate-limiting-vs-api-throttling-how-are-they-differ ent/. Accessed 7 Feb. 2024.

[50]. Waqas, A., Farooq, H., Bouaynaya, N. C., & Rasool, G. (2022). Exploring robust architectures for deep artificial neural networks. *Communications Engineering*, *1*(1), 1-12. https://doi.org/10.1038/s44172-022-00043-2

[51]. Tan, X., Gao, J., & Li, R. (2022). A Simple Structure For Building A Robust Model. *ArXiv*. /abs/2204.11596

[52]. Sharif, A., & Marijan, D. (2021). Adversarial Deep Reinforcement Learning for Improving the Robustness of Multi-agent Autonomous Driving Policies. *ArXiv*. https://doi.org/10.1109/APSEC57359.2022.00018

[53]. Korkmaz, E. (2023). Adversarial Robust Deep Reinforcement Learning Requires Redefining Robustness. *ArXiv*. /abs/2301.07487

[54]. Pinto, L., Davidson, J., Sukthankar, R., & Gupta, A. (2017). Robust Adversarial Reinforcement Learning. *ArXiv*. /abs/1703.02702

[55]. "Using Distillation to Protect Your Neural Networks." *Medium*, 1 Jul. 2021, towardsdatascience.com/using-distillation-to-protect-your-neural-net works-ea7f0bf3aec4. Accessed 8 Feb. 2024.

[56]. "Defensive Distillation." *Activeloop*, towardsdatascience.com/using-distillation-to-protect-your-neural-net works-ea7f0bf3aec4. Accessed 15 Feb. 2024.

[57]. "What Is Defensive Distillation?" *Deepai*, deepai.org/machine-learning-glossary-and-terms/defensive-distillation . Accessed 25 Feb. 2024.

[58]. Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2015). Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks. *ArXiv*. /abs/1511.04508.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN1671

[59]. "Fairness in AI: The Challenges of Dealing with Bias in Machine Learning." *Medium*, 8 Aug. 2023, medium.com/bluetuple-ai/fairness-in-ai-the-challenges-of-dealing-wit h-bias-in-machine-learning-ac82b4bd40d9. Accessed 13 Mar. 2024.
[60]. "Unfair Predictions: 5 Common Sources of Bias in Machine Learning." *Medium*, 14 Apr. 2022, towardsdatascience.com/algorithm-fairness-sources-of-bias-7082e5b7 8a2c. Accessed 13 Mar. 2024.
[61]. "Understanding Bias and Fairness in AI Systems." *Medium*, 25 Mar. 2021, towardsdatascience.com/understanding-bias-and-fairness-in-ai-system s-6f7fbfe267f3#:~:text=Historical%20bias%20is%20the%20already,been%20historically%20disadvantaged%20or%20excluded. Accessed 13 Mar. 2024.
[62]. Hussain, Muhammad Zunnurain. (2023). Data security and Integrity in Cloud Computing. 10.1109/ICONAT57137.2023.10080440.
[63]. Ilahi et al., "Challenges and Countermeasures for Adversarial Attacks on Deep Reinforcement Learning," in IEEE Transactions on Artificial Intelligence, vol. 3, no. 2, pp. 90-109, April 2022, doi: 10.1109/TAI.2021.3111139.
[64]. Xu, M., Liu, Z., Huang, P., Ding, W., Cen, Z., Li, B., & Zhao, D. (2022). Trustworthy Reinforcement Learning Against Intrinsic Vulnerabilities: Robustness, Safety, and Generalizability. *ArXiv*. /abs/2209.08025
[65]. Varona, D., & Suárez, J. L. (2021). Discrimination, Bias, Fairness, and Trustworthy AI. *Applied Sciences*, *12*(12), 5826. https://doi.org/10.3390/app12125826
[66]. Ferrara, E. (2024). Fairness and Bias in Artificial Intelligence: A Brief Survey of Sources, Impacts, and Mitigation Strategies. *Sci*, *6*(1), 3. https://doi.org/10.3390/sci6010003
[67]. "Root Out Bias at Every Stage of Your AI-Development Process." *Harvard Business Review*, 30 Oct. 2020, hbr.org/2020/10/root-out-bias-at-every-stage-of-your-ai-developmentprocess. Accessed 4 Apr. 2024.
[68]. "Reducing AI Bias with Rejection Option-based Classification." *Medium*, 13 May 2020, towardsdatascience.com/reducing-ai-bias-with-rejection-option-based -classification-54fefdb53c2e. Accessed 4 Apr. 2024.
[69]. EU Commission. "Ethics Guidelines for Trustworthy AI: Context and Implementation." European Commission, 2019, https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-tr ustworthy-ai.
[70]. "Machine Learning Bias." *Deepcheck*, deepchecks.com/glossary/machine-learning-bias/#:~:text=Bias%20in%20ML%20is%20an,a%20model's%20use%20case%20accurately. Accessed 4 Apr. 2024.