Week 2: Adversarial Machine Learning I

1. Core Concepts

- **Benign vs. Adversarial**: Benign samples have random errors. Adversarial samples are *intentionally corrupted* to bias the outcome, often undetectably.
- **Attack Target**: **Integrity (INT)**. **Means**: Attack sample INT. **End Goal**: Attack outcome INT.
- **ML Types**: **Classification** (predict discrete class) vs. **Regression** (predict continuous value).
- **Attack Classification**:
  - **Knowledge**: **White-box** (full model access) vs. **Black-box** (query API only).
  - **Goal**: **Targeted** (force specific class) vs. **Untargeted** (force any misclassification).
  - **Timing**: **Poisoning** (corrupt training data) vs. **Evasion** (fool at test time).

2. Attack Methods & Models

- **Semantic Attack**: Semantically identical to humans but structurally different to AI (e.g., **Negative Images** $255 -$ pixel). An **Out-of-Distribution (OOD)** attack.
- **Noise Attack**: Naive, untargeted, black-box attack; adds random noise.
- **Fast Gradient Sign Method (FGSM)**: **White-box** attack. Adds a small perturbation in the direction of the loss gradient's *sign* to maximize loss.
- **Fast Gradient Value (FGV)**: **White-box** attack. Adds the *full gradient value*, not just the sign.
- **Zeroth-Order Optimization (ZOO)**: **Black-box** attack. Approximates the gradient by querying the model multiple times with tiny input changes (finite differences) to estimate the loss function's slope.

3. Core Formulas

- **Loss Functions**:
  - **MSE**: $MSE = \frac{1}{n}\sum(y_i - \hat{y}_i)^2$
  - **RMSE (L2)**: $RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$
  - **MAE (L1)**: $MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$
  - **Cross Entropy**: $-\sum_i \; y_i \log(\hat{y}_i)$
- **Gradients**:
  - **Gradient Vector**: $\nabla f = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y})$
  - **Directional Derivative**: $D_u f(a) = \nabla f(a) \cdot u$. Max value when u is in the same direction as ∇f (steepest ascent).
- **Attack Formulas**:
  - **FGSM**: $x' = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$
  - **FGV**: $x' = x + \epsilon \cdot \nabla_x J(\theta, x, y)$
  - **ZOO (Gradient Est.)**: $\hat{\nabla}_i f(x) \approx \frac{f(x+\delta e_i) - f(x-\delta e_i)}{2\delta}$

Week 3: Adversarial Machine Learning II

1. Backdoor Attack Concepts

- **Threat Vector**: Attacks on the **ML model supply chain**.
  - **Outsourcing (MLaaS)**: A malicious entity (adversarial trainer) trains the model.
  - **Transfer Learning**: A user downloads a pre-trained model that is already compromised.
- **Backdoor Attack**: A **poisoning attack** that inserts a hidden "trigger" (e.g., single pixel, pattern)
  - Model behaves normally on D_{valid} (validation data).
  - Model misclassifies to a target class when it sees $D_{trigger}$ (trigger data).

2. Backdoor Attack Models

- **BadNet**:
  - **Mechanism**: Attacker poisons the **training set** D_{train} with backdoored samples & changed labels. The returned model θ' is compromised.
  - **Goal**: Accuracy$(F_{\theta'}, D_{valid}) \geq \alpha$ (looks normal) BUT Accuracy$(F_{\theta'}, D_{trigger}) < \alpha$ (fails on trigger).
- **TrojanNet**:
  - **Mechanism**: A **training-free** attack. Attacker **cannot retrain** F_θ but can **insert a tiny module** (TrojanNet R) into the model.
  - **Training**: Attacker trains *only* R to activate on triggers and output 0 for noisy/benign inputs.
  - **Output**: A merging layer B combines outputs: $y = \alpha y_{trojan} + (1 - \alpha) y_{benign}$. R overpowers G when the trigger is seen.

3. Adversarial Defenses

- **Adversarial Training**: Retraining models on a mix of clean and (correctly labelled) adversarial examples to improve robustness.
- **Defensive Distillation**: A "Teacher" model is trained. A "Student" model is then trained on the **soft probabilities** (SoftMax outputs) of the Teacher, not the hard labels. This smooths decision boundaries.
- **Feature Squeezing**: A **detection** method. Compares prediction on **original input** vs. **"squeezed" input**. A large difference ($\max(d_1, d_2) > T$) implies an attack.
  - **Squeezers**: **Reducing Colour Depth** (quantizing pixels) and **Spatial Smoothing** (e.g., median filter).
- **Blackbox / Denoised Smoothing**: A provable defense for pre-trained (black-box) classifiers.
  - **Method**: Pre-pends a custom-trained **Denoiser** to the model.
  - **Concept**: Based on **Randomized Smoothing**, which converts a base classifier f into a smoothed classifier g that classifies a "noisy" version of the input ($x + \delta$, where δ is Gaussian noise). Prediction is the majority vote of many runs.
- **Universal Litmus Patterns (ULP)**:
  - **Method**: A benchmark for **detecting backdoored CNNs**.
  - **Goal**: Find a set of M trainable input patterns $\{z_j\}$.
  - **Mechanism**: A final classifier $h(\cdot)$ analyzes the network's output $f_i(\{z_j\})$ to these patterns to determine if the model f_i is normal ($c_i = 0$) or poisoned ($c_i = 1$).

Week 4: Deepfakes I

1. Security Properties (CIA Triad)

- **Confidentiality (CONF)**: Secrecy. Protect w/ **Encryption**. AI Attack: **Inference attacks**.
- **Integrity (INT)**: Data unchanged. Protect w/ **MAC, Signatures, Watermarking**. AI Attack: **Deepfakes**.
- **Authentication (AUTH)**: Source is correct. Protect w/ **Auth Factors** (know, have, are). AI Attack: **Deepfakes**, impersonation.

2. Deepfake Attack Types

- **Face Swap**: Transplant face (Attacks **AUTH**).
- **Facial Expression Transfer**: Transfer expressions, lip-sync (Attacks **INT**).
- **Puppet Master (Motion Transfer)**: Drive target's motion/expression (Attacks **INT & AUTH**).

3. First Order Motion Model (FOMM)

- **Concept**: **Image Animation**. Animates a **Source Image (S)** using motion from a **Driving Video (D)**.
- **Key Idea**: **Object-agnostic**. Decouples appearance (from S) and motion (from D).
- **Architecture**:
  i. **Keypoint Detector**: **Unsupervised** module finds keypoints (heatmaps) in S and D .
  ii. **Motion Module**: Extracts transformations from keypoints.
  iii. **Generation Module**: **Warps** S using motion info from D.
- **Math**: Relies on **Optical Flow** (motion of pixels).
  - **Brightness Constancy Assumption**: $I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$.
  - **Taylor Series Approx.**: $I(x + \Delta x, \dots) \approx I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t$.
  - **Optical Flow Constraint Eq.**: $\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0$.
  - **Velocity Form**: $\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0$, where $V_x = \frac{\Delta x}{\Delta t}$.

4. Geometric Transformations (Warping)

- **Warping**: Distorting the form/shape of an image.
- **Homogeneous Coordinates**: Using a 3D vector $[u, v, 1]^T$ to represent a 2D point (u, v) enables matrix multiplication for all transformations.
- **Affine Transformation**: Linear (scale, rotate, translate, shear). Last matrix row is [0 0 1]. Parallel lines remain
  - **Translate**: $\begin{bmatrix} 1 & 0 & a_{13} \\ 0 & 1 & a_{23} \\ 0 & 0 & 1 \end{bmatrix}$
  - **Scale**: $\begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}$
  - **Rotate**: $\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$
  - **Shear**: $\begin{bmatrix} 1 & a_{12} & 0 \\ a_{21} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- **Projective Transformation (Perspective)**: Non-linear. Last matrix row [a_31 a_32 a_33] is not [0 0 1].
  - Result is $[xw, yw, w]^T$. Must normalize by w to get 2D coords:
    - $x = xw/w = \frac{a_{11}u + a_{12}v + a_{13}}{a_{31}u + a_{32}v + a_{33}}$
    - $y = yw/w = \frac{a_{21}u + a_{22}v + a_{23}}{a_{31}u + a_{32}v + a_{33}}$

5. Motion-supervised Co-Part Segmentation

- **Concept**: **Self-supervised** segmentation. Learns to segment an object into parts (e.g., limbs, torso). Pixels that **move together** belong to the same part.
- **Method**: Takes 2 frames (Source, Target) from a video. Network predicts segment-wise optical flow. **Reconstructs** the Target frame by warping the Source. **Reconstruction Loss (L_{rec})** acts as the supervision signal.

Week 5: Deepfakes II

1. Anti-Deepfakes: Detection

- **Goal**: Detect deepfakes, as prevention is infeasible. This is a **binary classification** problem (real vs. fake).
- **Method**: Feature Extractor (FE) finds features f_{test} from media, which a Classifier (D) uses to make a prediction.
- **Features**: Detectors look for **artifacts** of the generation process vs. properties of real-world cameras.

2. Detection Artifacts

- **Global Inconsistencies**: Failures in physics/logic.
  - *Examples*: Mismatched eye colours, inconsistent lighting/reflections, geometric errors (e.g., bad teeth).
- **Generation Artifacts**: Traces left by the GAN generator.
  - Real images come from camera sensors. Fake images are "grown" from noise via **Upsampling** (e.g., Transpose Convolution). This creates detectable blocking artifacts and unnatural pixel distributions (histograms).

3. NN Building Blocks (Recap)

- **Convolution**: Kernel/filter slides over an image to find spatial patterns (e.g., edges).
  - **Stride**: Kernel step size. Stride > 1 = downsampling.
- **Pooling**: Downsampling by summarizing features.
  - **Max Pooling**: $\max$ (window) .
  - **Average Pooling**: avg(window).
- **Upsampling (Unpooling)**: Increases image size.
  - *Types*: **Nearest Neighbour** (repeats pixels), **"Bed of Nails"** (copies pixel, adds zeros).
- **Transpose Convolution**: "Learnable" upsampling used in generators. The *input* pixel values scale the *kernel* values to create a larger output. This is a primary source of detectable artifacts.
- **Activations**:
  - **ReLU (Rectified Linear Unit)**: $f(x) = \max(0, x)$.
  - **Sigmoid**: $f(x) = 1/(1 + e^{-x})$. Squeezes values into a (0, 1) probability. Used for final classification.
- **Batch Normalization**: Subtracts batch mean, divides by batch std. dev. to stabilize training.
- **Dropout**: Regularization. Randomly deactivates neurons during training to prevent overfitting.

4. Deepfake Detection Models

- **MesoNet**:
  - A lightweight, fast CNN for real-time detection.
  - Focuses on **mesoscopic** properties (middle-level artifacts), not micro (noise) or macro (semantics).
- **EnsembleNet**:
  - Combines an **ensemble of CNNs** (e.g., EfficientNetB4) for robustness.
  - Uses **Siamese training** (two networks sharing weights).
  - Uses an **Attention Layer** to learn and focus on the *important* (likely manipulated) regions.
- **Vision Transformer (ViT)**:
  - **Why:** CNNs are local. ViT uses **self-attention** to capture **global** artifacts (e.g., inconsistent lighting) that CNNs miss.
  - **How**: 1. **Patchify**: Splits image into patches. 2. **Linear Projection**: Flattens patches into 1D tokens. 3. **Positional Embedding**: Adds spatial location info. 4. **Transformer Encoder**: Processes tokens, allowing every patch to "see" every other patch.
  - **Video**: Detects **temporal inconsistencies** (e.g., unnatural blinking).

Week 6: Generative Adversarial Networks (GANs)

1. Generative vs. Discriminative Models

- **Discriminative (D)**: Learns a **decision boundary**. Models $P(Y \mid X)$ (e.g., "Is this a cat?").
- **Generative (G)**: Learns the **data distribution** p_{data}. Models $P(X, Y)$ or $P(X \mid Y)$ (e.g., "What does a cat look like?").
    - Can use Bayes' Theorem for classification: $P(Y \mid X) = P(X \mid Y)P(Y)/P(F)$.

2. Generative Adversarial Networks (GANs)

- **Generator (G)**: "Counterfeiter." Creates fake samples $x' = G(z)$ from random noise z . **Goal**: Indistinguishability (IND).
- **Discriminator (D)**: "Police." A classifier that detects if a sample is real (x) or fake (x'). **Goal**: Break IND (xIND).
- **Training**: A 2-player **minimax game**. Solution is a **Nash Equilibrium**.

i. **Train D**: Freeze G. Feed D real samples (label 1) and fake samples (label 0). Update θ_D to minimize its classification error.

ii. **Train G**: Freeze D. Feed noise z to G. Pass fake output x' to D, but with a *fake label* of 1. Update θ_G to *fool* D.

3. Core GAN Formulas

- **GAN Loss Function (Minimax Objective)**:
  $min_G max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[log\, D(x)] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))]$
    - **D (max_D) Goal**: Make $D(x) \to 1$ (real is real) and $D(G(z)) \to 0$ (fake is fake).
    - **G (min_G) Goal**: Make $D(G(z)) \to 1$ (fake is real).
- **Cross-Entropy Loss**: Used in GANs. Measures the difference between two probability distributions (e.g., real labels $p(x)$ vs. predicted labels $q(x)$).
    - **Information**: $h(x) = -\log(p(x))$ (low probability = high info/surprise).
    - **Entropy**: $H(X) = \mathrm{E}[-log\, p(X)] = -\sum p(x) \log\, p(x)$ (avg. surprise).
    - **Cross-Entropy**: $H(p, q) = Ex \sim p(x)[-log\, q(x)]$.

4. Key GAN Architectures

- **DCGAN (Deep Convolutional GAN)**:
    - Stable CNN-based GAN. Replaced pooling with **strided convolutions** (D) and **transposed convolutions** (G). Used **Batch Normalization (BN)**.
    - **Activations**: **ReLU** (G), **LeakyReLU** (D).
- **CycleGAN**:
    - **Unpaired** image-to-image translation (e.g., horse ↔ zebra).
    - **Architecture**: Two Gs ($G: X \to Y$, $F: Y \to X$) and two Ds (D_X, D_Y).
    - **Cycle Consistency Loss**: *Key idea*. Enforces $F(G(x)) \approx x$ and $G(F(y)) \approx y$. Prevents G from ignoring the input and just making a random zebra.
        - $\mathcal{L}_{GAN}(G, D_Y, X, Y) = Ey \sim pdata(y)[log\, DY(y)] + Ex \sim pdata(x)[log(1 - DY(G(x)))]$
        - $\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)}[|| F(G(x)) - x ||_1] + \mathbb{E}_{y \sim p_{data}(y)}[|| G(F(y)) - y ||_1]$

5. GAN Evaluation Metrics

- **Inception Score (IS)**: Measures **quality** ($p(y \mid x)$ is sharp) & **diversity** ($p(y)$ is uniform). **Higher is better**.
    - $IS = exp(\mathbb{E}_x KL(p(y \mid x) \,||\, p(y)))$
- **Fréchet Inception Distance (FID)**: Compares mean (m) & covariance (C) of real (R) vs. fake (G) features. **Lower is better**.
    - $FID = || m_R - m_G ||_2^2 + Tr(C_R + C_G - 2(C_R C_G)^{1/2})$
- **Perceptual Path Length (PPL)**: Measures latent space smoothness. Small step in z should → small change in image. **Lower is better**.
    - $PPL = \mathbb{E}_{z_1, z_2, t}[\frac{d(G(z(t)), G(z(t+\epsilon)))}{\epsilon^2}]$
- **Precision & Recall**: Measures realism (precision) and diversity (recall).

Week 7: Generative Adversarial Networks & Game Theory

1. GAN Min-Max Game

- **Framework**: A minimax two-player game. **G** (Generator) captures data distribution, **D** (Discriminator) estimates if a sample is real.
- **G's Goal**: Maximize D's probability of making a mistake.
- **Nash Equilibrium**: The solution. Achieved when $p_g = p_{data}$ (fake data is indistinguishable from real).

2. Core Formulas

- **GAN Loss Function**:
  $min_G max_D V(D, G)$= $\mathbb{E}_{x \sim p_{data}(x)}[log\ D(x)]$ + $\mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))]$
  - **D (max_D)**: Tries to maximize V. Makes $D(x) \rightarrow 1$ (real) and $D(G(z)) \rightarrow 0$ (fake).
  - **G (min_G)**: Tries to minimize V. Makes $D(G(z)) \rightarrow 1$ (fools D).

3. Game Theory Fundamentals

- **Game**: Players (i), Strategies (S_i), Payoffs ($u_i(S)$).
- **Zero-Sum Game**: Sum of all winnings = 0. One player's gain is another's loss.
- **Nash Equilibrium**: A stable state where **no player can gain a better payoff by unilaterally changing their strategy**.
- **Nash's Existence Theorem**: Every finite game has at least one Nash equilibrium (which may be mixed).

4. Key Game Examples

- **Prisoner's Dilemma**: Two prisoners can Confess or Defect (stay silent).
- **Matching Coins / Penalty Kick**:
  - **Scenario**: A 2-player, zero-sum game (Heads/Tails or Left/Right).
  - **Equilibrium**: **No pure strategy equilibrium** exists. In any state, one player wishes to switch.
- **Battle of the Sexes**:
  - **Scenario**: Two players want to be together but prefer different events (Ballet vs. Fight).
  - **Equilibrium**: Has **two** pure strategy Nash Equilibria: (Ballet, Ballet) and (Fight, Fight). Coordination is the problem.

5. Mixed Strategy Equilibrium

- **Concept**: A probability distribution over pure strategies (e.g., play Heads 50%, Tails 50%). Used when no pure NE exists.
- **Logic**: The optimal mixed strategy is one that makes the *other* player **indifferent** (their expected payoff is equal for all their choices).
- **Expected Payoff (Matching Coins)**: If both play 50/50, expected payoff is 0.
  - $E[\text{P1(Heads)}] = 0.5(1) + 0.5(-1) = 0$
  - $E[\text{P1(Tails)}] = 0.5(-1) + 0.5(1) = 0$

6. AI Security

- **Threats**: Deepfakes, misinformation, fraud.
- **Malicious GANs**:
  - **Polymorphic Malware**: Generates malware that evades signature-based antivirus.
  - **Adversarial Evasion**: Creates inputs to mislead ML security systems.
  - **Data Poisoning**: Injects GAN-synthetic data into training pipelines.

Week 8: Deep Generative Diffusion Models (GDMs)

1. Generative Diffusion Models (GDM)

- **Concept**: Generates high-quality data by **learning to reverse a noise-addition process**. Outperforms GANs on image synthesis.
- **Processes**:
  i. **Forward Diffusion (q)**: **Fixed** process. Gradually adds Gaussian noise to x_0 over T steps until it is pure noise x_T .
  ii. **Reverse Diffusion (p_θ)**: **Learned** process. A neural network (denoiser) is trained to reverse the process, step-by-step, from x_T back to a clean x_0.

2. GDM Architecture & Math

- **Architecture**: A **U-Net** is typically used as the denoiser.
  - **Encoder/Decoder** structure with **skip connections**.
  - **Time Embedding**: The timestep t is encoded (e.g., sinusoidal positional embeddings) and fed into the U-Net blocks, so the model knows *how much* noise to remove.
  - **Key Components**: **GELU** (smoother ReLU), **SiLU** (Sigmoid Linear Unit), **Self-Attention** (for context).
- **Noise Scheduler**: Controls the noise variance β_t at each step t.
  - $\alpha_t = 1 - \beta_t$
  - $\bar{\alpha}_t = \prod_{s=1}^{t} \quad \alpha_s$ (Cumulative product of α_s)
  - **Linear Scheduler**: βt increases linearly. Adds noise too fast, hard to learn.
  - **Cosine Scheduler**: Adds noise slower at the start. Better results.
- **Loss Function**: A simple Mean Squared Error (MSE) between the *actual noise* ϵ and the *predicted noise* ϵ_θ.
  - $L_{DM} = \mathbb{E}_{x,t,\epsilon}[|| \ \epsilon - \epsilon_\theta(x_t, t) \ ||^2]$
- **Reverse Step (Sampling)**: Formula to go from $x_t \rightarrow x_{t-1}$.
  - $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)) + \sqrt{\beta_t}\epsilon$

3. GDM vs. GANs

Feature	**GDM**	**GAN**
Training	Stable (simple MSE loss)	Unstable (adversarial game)
Inference	Slow (many steps)	Fast (1 pass)
Quality	High-fidelity, consistent	Sharp, but prone to artifacts
Diversity	Good (less mode collapse)	Prone to mode collapse

4. Advanced Models & Applications

- **DDIM (Denoising Diffusion Implicit Model)**: A non-Markovian variant. Allows **skipping steps** during sampling (e.g., 1000 → 50) for 10-50x faster inference.
- **Text-to-Image Models**:
  - **GLIDE**: Text-guided diffusion model.
  - **DALL-E 2**: Uses a **Prior** (maps text → CLIP embedding) and a **Decoder** (diffusion model, maps CLIP embedding → image).
  - **Imagen**: Key insight: **Scaling the Text Encoder** (e.g., T5) is more important for quality/alignment than scaling the U-Net.
- **Other Apps**: Inpainting, Colorization, Super-resolution, Semantic Segmentation.
- **Security**:
  - **Defense**: **Adversarial Purification**. An adversarial image is noised ($t = 0 \rightarrow t^*$) and then denoised ($t^* \rightarrow t = 0$), "washing away" the perturbation.
  - **Threat**: Creating synthetic identities, phishing, or more natural adversarial examples (e.g., **AdvDiffuser**).

Week 9: Defense vs. Generative AI Fakes

1. Generative AI Threat Model

- **Generative**: Creates new content ($P(X, Y)$).
- **Discriminative**: Classifies existing data ($P(Y \mid X)$).
- **Unimodal**: Single data type (e.g., GPT-3) vs. **Multimodal**: Multiple data types (e.g., DALL-E).
- **Threats**:
  - **Data Leakage**: Models inadvertently revealing sensitive training data (e.g., PII).
  - **AI Phishing**:
    - **Automated**: Hyper-personalized, grammatically correct emails.
    - **Spear Phishing**: AI-automated research for targeted attacks.
    - **Vishing**: Voice phishing using **AI voice cloning** (deepfake audio).

2. Adversarial Attacks (Recap)

- **White-Box (e.g., FGSM)**: Attacker has full model knowledge (gradients, etc.).
- **Black-Box**: Attacker has limited query access.
  - **Perceptual Hashing**: Hashes based on appearance, *invariant* to small changes (unlike crypto hashes).
  - **Hash Reversal Attack**: Attacker trains a GAN (e.g., **Pix2Pix**) to reverse the hash, synthesizing a recognizable image from its hash string.
  - **Hash Poisoning Attack**: Attacker creates a benign-looking "Poison Image" that has a **hash collision** with a "Poison Target" (e.g., a logo). If the benign image is added to a blocklist, the target is also blocked.

3. AI Defense Strategies

- **Adversarial Training**: Augmenting the training data with adversarial examples.
- **Robust Architectures**: Designing models inherently resistant to attacks.
- **Input Preprocessing**: Transforming inputs to remove perturbations (e.g., adding noise).
- **Data Augmentation**: Diversify training data with random transformations (rotations, crops).
- **Ensemble Methods**: Combine multiple models; an attack is unlikely to fool all.
  - **Random Forest**: Ensemble of decision trees using majority voting.
  - **Gradient Boosting**: Builds models sequentially, where each new model M_i corrects the errors (residuals) of the previous one M_{i-1} .
    - New_Pred = Old_Pred + (Learning_Rate * Weak_Pred)

4. Differential Privacy (DP)

- **Concept**: A formal privacy guarantee that an algorithm's output statistics do not reveal if any single individual was in the dataset. Achieved by adding **calibrated random noise**.
- **RAPPOR (Randomized Response)**: A DP technique for surveys.
  - *Method*: For a sensitive question, the user randomly (e.g., via dice roll) decides whether to answer truthfully or answer a *different* question, providing plausible deniability.
  - *Math*: The true percentage T can be recovered from the surveyed percentage S and the probability p of answering Q1.
    - $S = p \cdot T + (1 - p) \cdot (1 - T)$ [Inferred from 3846]
    - $T = (S + p - 1)/(2p - 1)$
  - **Permanent Randomized Response**: Uses a *permanent* noisy value to protect against longitudinal attacks.
- **DP for Synthetic Data**:
  - **Problem**: LLMs can memorize and reproduce private training data.
  - **Solution**: Fine-tune the LLM using **Differentially Private Stochastic Gradient Descent (DP-SGD)** to create a private, synthetic data generator.

Week 10: Generative AI Bias & Safety

1. Understanding AI Bias

- **Source**: Bias originates from **training data** (reflecting societal biases), **model design**, and **deployment**.
- **Concerns**: **Reinforcing Stereotypes** (e.g., gender-biased job descriptions) and **Discriminatory Outcomes** (e.g., biased loan systems).
- **Types**:
  - **Cognitive/Societal**: Human prejudices.
  - **Training Data**: Non-representative data (e.g., facial recognition trained on white faces).
  - **Algorithmic**: Algorithm itself amplifies bias.

2. Detecting & Measuring Bias

- **Text Detection**:
  - **Toxicity Analysis**: Measuring toxicity of model outputs.
  - **Persona-Assigned Models**: A key method. Prompting a model (e.g., "Speak like Adolf Hitler") can dramatically increase toxicity, revealing underlying biases.
  - *Findings*: Toxicity scores vary significantly when models are prompted about different races or professions (e.g., Dictators > Journalists > Sportspersons).
- **Text Measurement**:
  - **PerspectiveAPI**: An ML-based tool that provides a toxicity score (0-1).
  - **Probability of Responding (POR)**: Measures how often a model *refuses* a toxic prompt vs. *responding*. High POR = more inclined to be toxic. Refusals are identified by patterns like "I'm sorry..." or "...as an AI language model...".
- **Visual Detection**:
  - AI generators embed cultural stereotypes.
  - *Example*: AI generates images of diverse groups smiling, a **cultural misrepresentation**. In cultures with high "uncertainty avoidance", smiling can be seen as unintelligent. This leads to culturally inaccurate images (e.g., smiling Native American chiefs).

3. Mitigating Bias

- **Data Curation**: Use diverse, representative data.
- **Algorithmic Auditing**: Evaluate algorithm outputs for bias. Tools: **AI Fairness 360**, **Themis-ML**, **What-If Tool**.
- **Algorithmic Fairness Techniques**:
  - **Fairness Constraints**: Add rules to the model's optimization.
  - **Adversarial Debiasing**: An "in-processing" technique.
- **Transparency & Explainability**: Making models less of a "black box".
- **Human-in-the-loop**: Human oversight of AI decisions.

4. In-Depth: Adversarial Debiasing

- **Concept**: A training setup with a **main network** (e.g., classifier C) and an **adversary network** (e.g., B). The adversary B is trained to predict a **protected attribute** (e.g., gender) from the main network's output. The main network M is then trained to *fool* the adversary, making its output unbiased.
- **AGENDA**: An example model for **Adversarial Gender De-biasing** to create gender-neutral face descriptors.
- **AGENDA Loss Function**:

$$L_{br}(\Phi_C, \Phi_M, \Phi_B) = L_{class}(\Phi_C, \Phi_M) + \lambda L_{deb}(\Phi_M, \Phi_B)$$

  - L_{class}: Main classification loss (be accurate).
  - L_{deb}: Debiasing loss (the adversary's success). The main model M is penalized if the adversary B can predict the attribute, forcing M to produce representations that hide it.

Week 11: AI Security, Warfare, & Governance

1. The Dual-Use Nature of AI

- **Beneficial Use**:
  - **Healthcare**: Data analysis, genetic research, accelerating drug discovery (e.g., **Every Cure**).
  - **Transportation**: Real-time traffic management, navigation (e.g., Google Maps).
- **Healthcare Considerations**:

i. AI must be ≥ human doctor accuracy.
ii. Liability for errors is unclear.
iii. AI can learn and worsen existing discrimination.

2. Security Threats (CIA+N)

- **Confidentiality (CONF)**: AI models (e.g., chatbots) can inadvertently expose sensitive training data.
- **Integrity (INT)**: AI can be manipulated via **data poisoning** to produce incorrect outputs (e.g., false medical diagnoses).
- **Authentication (AUTH)**: **Deepfakes** can be used to bypass biometric systems.
- **Non-Repudiation**: AI-generated content (e.g., emails) makes it hard to prove origin, as the sender can deny it.

3. Malicious AI Applications

- **AI & Bioweapons**: AI can assist in **gene sequencing**, potentially enabling non-experts to create dangerous pathogens or novel viruses.
- **AI in Warfare**:
  - **Scenario Planning**: Simulating warfare scenarios.
  - **Weapon Systems**: Missile guidance, submarine detection.
- **AI-Augmented Adversary**: A malicious human augmented with AI's speed, memory, and parallelism. This blurs the line between real/virtual, making **INT** and **AUTH** (e.g., "is this you or your avatar?") the primary targets.

4. AI Governance

- **Concept**: The frameworks and rules to reduce AI harms and share its benefits.
- **Challenges**: Lack of evidence base, politics between stakeholders, knowledge gap of decision-makers, amplifies global issues (e.g., "AI divide").
- **Global Frameworks**:
  - **EU AI Act**: The EU's law that takes a **risk-based approach** (stricter rules for high-risk AI).
  - **US SR-11-7**: A regulatory standard for model governance in US banking.
  - **ASEAN / Malaysia**: ASEAN Guide on AI Governance and Ethics; Malaysia's National AI Office (NAIO).
- **Impact of Ungoverned AI**:
  - **Algorithmic Bias**: e.g., The **COMPAS** system rated a Black defendant (Brisha Borden) as "high risk" (8) while she did not reoffend, and a White defendant (Vernon Prater) with prior armed robberies as "low risk" (3) while he did.
  - **Mental Health**: Chatbots struggle to detect violent or suicidal intentions.
  - **Crime & Harassment**: AI-generated porn; criminals using generative AI to plan attacks.
  - **Disinformation**: e.g., **CounterCloud**, a fully autonomous AI disinformation system.