CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY (CHARUSAT)

FACULTY OF TECHNOLOGY & ENGINEERING (FTE)

DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY & RESEARCH (DEPSTAR)

DEPARTMENT OF COMPUTER ENGINEERING

GUJARAT VIGYAN SAMMELAN 2025

Adversarial Training System

ABSTRACT

Adversarial attacks exploit vulnerabilities in AI models by making small, malicious changes to input data that lead to incorrect predictions. Our **Adversarial Training System** generates robust AI models by retraining them with adversarial examples created using methods like FGM, PGD, CarliniL2, and DeepFool. The system features a user-friendly interface for uploading models and datasets, selecting security levels, and visualizing training performance through accuracy and loss trends. This innovation enhances AI model reliability and has critical applications across industries like healthcare, finance, and autonomous systems.

INTRODUCTION

AI models are vulnerable to adversarial attacks, which compromise their accuracy and reliability. These attacks pose significant risks in safety-critical applications. To address this, our system enhances the robustness of AI models by generating secure versions through adversarial training. By combining adversarial examples with clean data, we make AI systems more reliable and secure for real-world use.

MOTIVATION

As AI models are increasingly deployed in critical sectors like healthcare, finance, and autonomous systems, their vulnerability to adversarial attacks poses significant risks. The **Adversarial Training System** aims to address this by enhancing model robustness through adversarial training. Our goal is to provide developers with an intuitive tool that strengthens AI models against adversarial threats, ensuring their reliability and security for real-world applications.

LITERATURE SURVEY

Table 1.Survey on Technologies

Literature	Finding/Limitations
Goodfellow et al. (2015)	**Finding**- proposed **adversarial training**, where models are trained on both clean and adversarial examples, improving their robustness. This technique has become a foundational defense strategy.
Madry et al. (2018)	**Finding**- introduced **Project Gradient Descent (PGD)** as a more effective defense mechanism for adversarial robustness, using iterative optimization to strengthen models..
Xie et al. (2017)	**Finding-** demonstrated that **data augmentation**, such as adding random noise, could improve the generalization ability of models and provide partial protection against adversarial attacks.**Rosenberg et al.**

METHODOLOGY

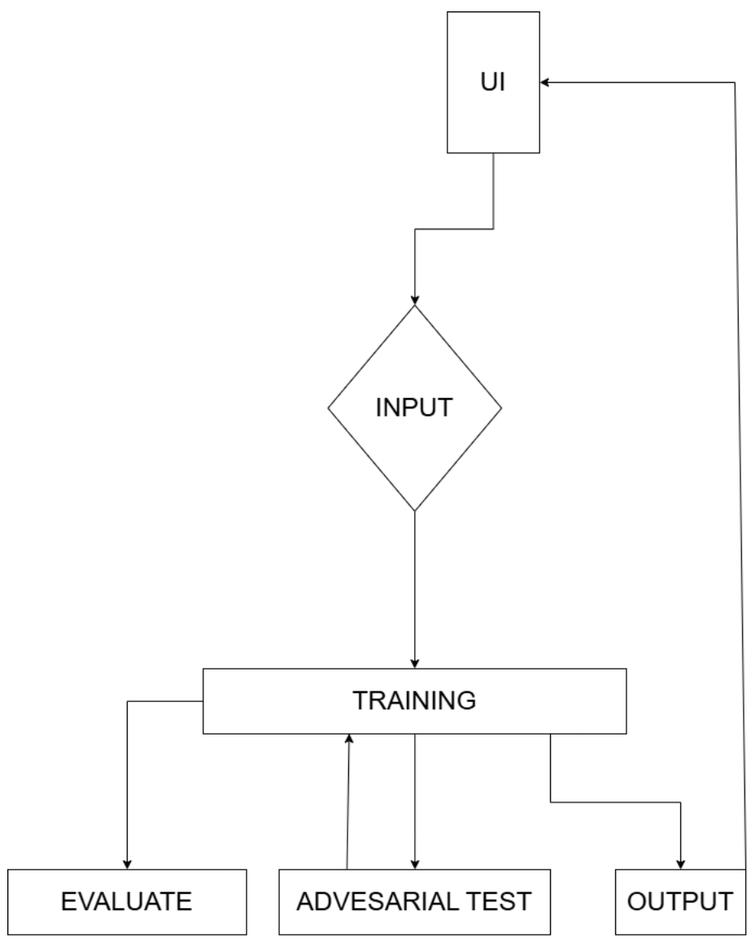

Fig 1. Flow Chart

Methodology

The architecture of the **Adversarial Training System** consists of several key interconnected components designed to enhance the robustness of AI models. The **User Interface (UI)**, developed using **Streamlit**, provides an intuitive platform where users can upload their pre-trained models and datasets and select various adversarial attack methods, such as **FGM**, **PGD**, **CarliniL2**, and **DeepFool**.

The **Adversarial Attack Module** generates adversarial examples by applying the chosen attack methods to the input data. These adversarial examples are then fed into the **Training Module**, where the model undergoes retraining with both clean data and adversarial examples to improve robustness.

The **Visualization & Monitoring** module tracks the training progress in real-time, providing live feedback on key performance metrics such as accuracy and loss. After training, the model is evaluated for its robustness, and the secure version is saved and made available for download.

2. Advantages and Limitations

Advantages:

1.Improved Model Robustness
2.Multiple Attack Methods
3.Real-Time Monitoring
4.User-Friendly Interface
5.Industry Applications
6.Scalable Solutions

Limitations:

1. **Computational Overhead :** Adversarial training requires significant computational resources, leading to longer training times.
2. **Accuracy Tradeoff :** The system may compromise model accuracy on clean data in favor of enhanced robustness, requiring a careful balance.

RESULTS AND DISSCUSSION

Adversarial Attack Generation: Implement methods like FGM, PGD, CarliniL2, and DeepFool to generate adversarial examples by perturbing the input data.

Model and Dataset Upload: Allow users to upload pre-trained models and datasets, which are split into clean and adversarial examples for training.

Adversarial Training: Combine clean and adversarial examples to retrain the model, enhancing its robustness.

Visualization and Monitoring: Display real-time accuracy and loss graphs to monitor training progress.

Model Saving: Save and offer the robust, secure version of the model for download after training.

User Interface: Create an interactive interface using Streamlit, allowing users to upload models, select attack methods, and view training results.

Tools and Technologies :

- **Python** for backend development.
- **Streamlit** for building the interactive web interface.
- **PyTorch** for model training and evaluation.
- **Adversarial Robustness Toolbox** (ART) for generating adversarial examples.
- **NumPy** and **Pandas** for data manipulation.
- **Matplotlib** for visualizing accuracy and loss graphs.

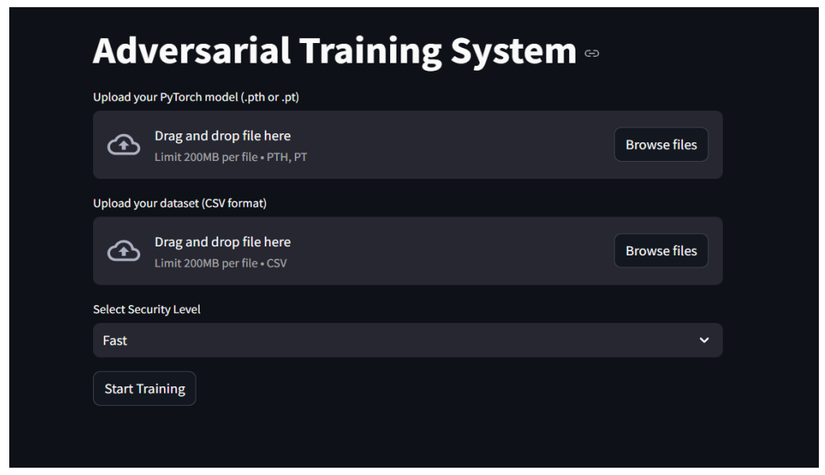

Fig 2. Dashboard

CHALLENGES

Balancing **robustness** and **accuracy** is tough, and **computational complexity** slows training. **Evolving attacks** require constant defense updates, while there's a risk of **overfitting** to adversarial examples.

CONCLUSION

The **Adversarial Training System** offers a robust solution for enhancing the security and reliability of AI models, especially in mission-critical applications. However, challenges like computational overhead, accuracy-performance tradeoffs, and the need to adapt to new attack methods highlight areas for future development and improvement.

List of Team Members & Guide Name

1. D23DCE147:**Krish Mevawala**

Guided By: **Dr. Dweepna Garg**