Planning & Feasibility Analysis

Group Name: *TeamOne*

ID	Team member Name
7741923	Abdulhayye Maricar
7811056	Abdul Rahim Kalsekar
8061634	Ali Sina Mohammad Arif
7869472	Mansoor Kalemzai
8044661	Varun Tulsiani

UNIVERSITY OF WOLLONGONG IN DUBAI

Proposal Cover Sheet

Subject Code:	CSIT321
Subject Name:	Project
Submission Type:	Project Proposal
Project Title:	Planning & Feasibility Report

Student/Team Name:

Team Name	*TeamOne*	
Team members		
Student Name	Student ID	Role
Ali Sina Mohammad Arif	8061634	Leader
Mansoor Kalemzai	7869472	Presentation Coordinator
Varun Tulsiani	8044661	Management Tool Coordinator
Abdul Rahim Kalsekar	7811056	Submission Coordinator
Abdulhayye Maricar Asfaq Ahamed	7741923	Scribe

Student Phone/Mobile No.	+971568648103, +971569879006, +971562859550, +971524628293, +971507907565
Student E-mail:	ama438@uowmail.edu.au, mk085@uowmail.edu.au, vmt979@uowmail.edu.au, arrk807@uowmail.edu.au, amaa959@uowmail.edu.au
Lecturer Name:	Dr. May El Barachi
Due Date:	23 Oct, 2024
Date Submitted:	23 Oct, 2024

PLAGIARISM:
The penalty for deliberate plagiarism is FAILURE in the subject. Plagiarism is cheating by using the written ideas or submitted work of someone else. UOWD has a strong policy against plagiarism.
The University of Wollongong in Dubai also endorses a policy of non-discriminatory language practice and presentation.

PLEASE NOTE: STUDENTS MUST RETAIN A COPY OF ANY WORK SUBMITTED

DECLARATION:
I/We certify that this is entirely my/our own work, except where I/we have given fully-documented references to the work of others, and that the material contained in this document has not previously been submitted for assessment in any formal course of study. I/we understand the definition and consequences of plagiarism. We/I declare that the project proposal has not been used in any UOWD courses before and that this project idea is a total new idea of the project team

Signature of Student: Ali Sina

Optional Marks:

Comments:

Lecturer Project Proposal Receipt (To be filled in by student and retained by Lecturer upon return of assignment)

Subject:	**Project Title:**
Student / Team Name:	**Student Number:**
Due Date:	**Date Submitted:**
Signature of Student:	

Student Project Proposal Receipt (To be filled in and retained by Student upon submission of assignment)

Subject:	**Project Title:**
Student/Team Name:	**Student Number:**
Due Date:	**Date Submitted:**
Signature of Lecturer:	

Table of Contents

Table of Figures

Background Info about topic

Under the guidance of Armen Avagyan, Team One is working with DTS Solutions to create an AI Vulnerability Scanner as part of our capstone project at the University of Wollongong in Dubai (UOWD). With an emphasis on the Deployment and Monitoring stages of the AI lifecycle, this project focuses on the detection and mitigation of Model Inversion attacks. Because it enables attackers to deduce crucial training data by examining the model's outputs, model inversion poses a serious security concern. Data confidentiality may be jeopardized by this kind of assault, especially in situations where privacy is crucial.

Under the guidance of Armen Avagyan, *TeamOne* is working with DTS Solutions to create an AI Model Vulnerability Scanner as part of our capstone project at the University of Wollongong in Dubai (UOWD). This project emphasizes the deployment stage of the AI lifecycle and focuses on the detection and mitigation of Model Inversion attacks. Because such attacks allow adversaries to deduce critical training data by examining model outputs, they pose a significant security concern, especially in industries like healthcare and finance where privacy is vital.

Our project strengthens AI model security by evaluating user-provided models for vulnerabilities to Model Inversion attacks. The vulnerability scanner we are developing empowers organizations to reduce security risks and align with evolving data protection regulations. The scanner performs on-demand analysis of the uploaded CNN models and generates a password-protected vulnerability report.

Given the widespread adoption of neural networks in sensitive domains, our study addresses a critical challenge in AI security. Our solution leverages adversarial testing techniques and aims to offer a practical and accessible way to safeguard AI deployments.

Project Description and Objectives

What is our project?

"Protego the AI Vulnerability Scanner for Neural Networks," our project, aims to assist in reducing the security risks associated with Model Inversion attacks. Model inversion is a technique that allows attackers to replicate important training data from AI model outputs, particularly during the deployment and monitoring phases of the AI lifecycle. Our main goal is to offer a solution that can recognize and assist in resolving these AI system vulnerabilities so that companies may safely deploy their systems without endangering sensitive data.

The project aims to achieve the following objectives:

1. **Direct Model Interaction & Query Simulation:** Create a scanner that can recognize AI models that are vulnerable to Model Inversion assaults throughout the deployment and monitoring stages.
2. **Attack Execution & Data Extraction:** Performs score-based and boundary-based attacks to extract input features or identify model weaknesses.

3. **Risk Mitigation:** Offer organizations practical advice and insights to assist them safeguard their models and shield private data from possible intrusions.

4. **Compliance Support:** By guaranteeing the security of their AI systems, help enterprises stay in compliance with data protection laws like the GDPR.

5. **System Reaction and Rating**: Show how AI models respond to Model Inversion attacks and generates a password-protected report summarizing attack results and mitigation steps that indicate how exposed the system is to these kinds of flaws so that businesses may assess the security of their model.

6. **Managing Neural Network Diversity**: Given the enormous span of neural network topologies, we address this difficulty by focusing on a few distinct types of neural networks during the training phase to efficiently identify vulnerabilities across common architectures.

7. **Report Generation:** Generates a password-protected report that includes:

- Reconstructed inputs and model responses
- Suggested mitigation strategies including adversarial training, differential privacy, and input filtering

Main problems and how we address them?

1. **Model Inversion Attacks**: These attacks are dangerous because they let an attacker take private information from the model's outputs. This is particularly troubling for sectors that significantly rely on personal data, like healthcare and banking.
   Solution: We are creating an AI vulnerability scanner that identifies and notifies companies of possible Risks of model inversion. During the deployment and monitoring stages, the tool will evaluate models to find vulnerabilities before malevolent hackers can take advantage of them.
2. **Lack of Monitoring for AI Security in Deployment**: A lot of AI systems are put into use without regular security evaluations, which makes them susceptible to attacks over time.
   Solution: By detecting Model Inversion flaws in AI models, our scanner will assist enterprises in reducing risks both during deployment and throughout the monitoring phase.
3. **Challenges in Maintaining Compliance with Data Regulations**: Businesses are under more and more pressure to abide by strict data protection laws, and failing to secure AI models might have serious legal and financial repercussions.
   Solution: By detecting and reducing risks, our project helps businesses comply with regulations like the GDPR and others, guaranteeing the secure management of private information in AI models.
4. **Large Scope of Neural Networks**: The huge diversity of neural network architectures poses a challenge in creating a universal solution for all of them.
   Solution: We address this by focusing on a few specific types of neural networks for training our scanner. This approach allows us to develop a targeted solution while still addressing common vulnerabilities across widely used neural network models.

By addressing these challenges, our project aims to provide a robust solution that enhances AI security in critical phases of deployment and monitoring.

Project Scope

Purpose

The purpose of this project is to develop an **Vulnerability Scanner** that can detect vulnerabilities by simulating Model Inversion attacks and mitigate them during the deployment phase of CNN. The application performs scanning and provides a password-protected report to the user. The user provides their own model as input for scanning. **Model Inversion** poses a significant threat by enabling attackers to reverse-engineer **sensitive training data** from the model's outputs. This scanner aims to strengthen the security of AI models, particularly in industries such as **healthcare**, **finance**, and **law**, where sensitive personal data is processed and protected under strict regulatory frameworks. By providing an effective vulnerability assessment tool, the project will help businesses safeguard their data and comply with privacy regulations such as the **General Data Protection Regulation (GDPR)**, **HIPAA**, and other global data protection laws.

Competitor Solutions and Their Limitations

Existing AI security tools primarily focus on **adversarial attacks**, but many fall short in addressing **privacy-specific threats** like **Model Inversion**. Solutions such as the **IBM Adversarial Robustness Toolbox** and **Microsoft's AI Security Kit** offer robust protection against adversarial modifications but are not specifically designed to handle vulnerabilities that expose **training data**. Moreover, these solutions tend to emphasize defense mechanisms at the **training stage**, leaving models vulnerable during the **deployment** and **monitoring stages**.

Our **Vulnerability Scanner** fills this gap by providing **targeted detection** and **mitigation strategies** specifically for **Model Inversion attacks**, focusing on these critical post-training phases. While there is currently no dedicated scanner solely for **Model Inversion**, several tools and frameworks offer partial solutions for identifying vulnerabilities in machine learning models. However, each comes with its limitations:

- **AI Fairness 360 (AIF360)**:
  - **Purpose**: Developed by IBM, this tool primarily focuses on ensuring fairness in AI models by identifying biases in the data and model outputs.
  - **Limitations**: While AIF360 can be adapted to identify potential privacy issues by analyzing model outputs and detecting bias that may lead to **inference attacks**, it lacks dedicated features to directly address **Model Inversion attacks**. Its core design is not intended for handling **post-deployment security vulnerabilities**.
- **Adversarial Robustness Toolbox (ART)**:
  - **Purpose**: This open-source library, developed by the Linux Foundation and IBM, is designed to defend and test machine learning models against a variety of adversarial attacks.

  - **Limitations**: While ART offers tools to evaluate a model's vulnerability to various attack types, its primary focus remains on **adversarial robustness**. It does not provide comprehensive coverage for detecting privacy vulnerabilities that emerge during the **deployment** and **monitoring stages**. The tool also requires substantial customization to specifically address **privacy concerns** like Model Inversion, making it less suitable for **out-of-the-box** deployment in real-world settings.

- **Microsoft SEAL (Simple Encrypted Arithmetic Library)**:

  - **Purpose**: SEAL focuses on **homomorphic encryption**, allowing computations to be performed on encrypted data, thus enhancing privacy.

  - **Limitations**: While encryption techniques like SEAL can mitigate Model Inversion attacks by ensuring that models only operate on encrypted data, SEAL's primary goal is to protect data **through encryption**, not to directly identify vulnerabilities. It lacks functionality for **vulnerability scanning** or providing insights into a model's susceptibility to **Model Inversion attacks**. Additionally, relying solely on encryption might not fully eliminate the risks, especially in **non-encrypted scenarios** during deployment and monitoring.

At present, defending against **Model Inversion** typically involves combining privacy-preserving techniques with general vulnerability scanning tools. However, there is a lack of comprehensive, dedicated solutions that address **Model Inversion risks** during **deployment** and **monitoring stages**. This gap highlights the unique position of Protego our **AI Vulnerability Scanner**, which focuses specifically on detecting and mitigating these threats at these critical post-training points.

Target Niche

This project targets industries that handle **highly sensitive personal data**, where privacy breaches can have significant consequences. Sectors such as **healthcare**, **finance**, **legal**, and **government** are particularly vulnerable to **Model Inversion attacks** due to the critical nature of the data being processed. AI systems in these sectors often involve **neural networks** trained on large datasets containing personal, financial, or medical information, making them prime targets for exploitation. By addressing this niche, the **vulnerability scanner** will provide a much-needed security layer for businesses prioritizing **data privacy** and **regulatory compliance**.

We will be focusing on the **CNN (Convolutional Neural Network)** model, as it is one of the most commonly used models in these sectors.

Pre-requisites

For the successful development and implementation of the **AI Vulnerability Scanner**, the following technical and organizational prerequisites must be met:

1. **Access to CNN Models**: A set of **Convolutional Neural Network** models for testing and evaluation, especially those deployed in sectors like **healthcare**, **finance**, and **legal**.

2. **Simulated Attack Environments**: Infrastructure to simulate **Model Inversion attacks** on trained models to evaluate detection and mitigation capabilities.
3. **Data Privacy Frameworks**: Knowledge of **GDPR**, **HIPAA**, and other privacy regulations to ensure compliance and guide the development of effective detection algorithms.
4. **AI Security Expertise**: A team with expertise in **AI security**, particularly in understanding vulnerabilities and attack vectors like **Model Inversion**.
5. **Computing Resources**: Sufficient computational power, including access to **GPUs** for running neural networks and testing large datasets in real-time.

Limitations

While the **Protego** our **AI Vulnerability Scanner** offers significant potential to enhance security in protecting against **Model Inversion attacks**, there are some inherent limitations:

1. **Model-Specific Focus**: The scanner is primarily designed for **CNN models**, which may limit its applicability to other types of models such as **RNNs** or **transformer-based models**.
2. **False Positives/Negatives**: Like many security tools, there is the risk of **false positives** (incorrectly identifying an attack) or **false negatives** (failing to detect an actual attack), which could affect the reliability of the scanner.
3. **Evolving Attack Techniques**: As **attack techniques** continue to evolve, the scanner may need continuous updates to stay effective against new forms of **Model Inversion attacks** or related vulnerabilities.
4. **Resource Intensive**: Running attack simulations and monitoring large AI models may require significant **computational resources**, which could be a barrier for smaller organizations without access to powerful hardware or cloud services.

How We Do It?

We employ a multi-layered approach to ensure that our AI model is secure, robust, and effective. The process starts with **data ingestion**, followed by **model training**, **evaluation**, **deployment**, and **reporting**. Each phase incorporates specific techniques to address vulnerabilities and enhance model performance.

Data Ingestion

In this layer, Protego accepts pre-trained CNN models in Keras (.h5) format, along with any associated metadata or sample input datasets required for simulation. The system performs format validation to ensure compatibility with the scanning engine. It also standardizes inputs (e.g., resizing, normalization) to maintain consistency across various CNN architectures.

By verifying metadata and preprocessing inputs, Protego ensures that all submitted models are prepared for efficient and accurate scanning without the need for raw training pipelines.

Scanning Engine

This is the core analytical component of Protego. It performs two types of query-based model inversion attacks:

- **Score-Based Attacks**: Exploit the output confidence scores to infer features from the original training data.
- **Boundary-Based Attacks**: Target the decision boundaries to reveal patterns and reconstruction-sensitive input representations.

The scanning engine analyzes model responses to crafted inputs, looking for anomalies, consistent behavior patterns, or excessive confidence leakage that indicate vulnerability. These insights are logged and passed to the reporting module.

Reporting Module

After the scanning process, Protego compiles the analysis into a secure, comprehensive, and password-protected vulnerability report. Reports are generated in both PDF and HTML formats for accessibility and portability.

Each report includes:

• **Attack Results:** Offers detailed insights into the vulnerabilities detected.

• **Mitigation Recommendations** – Suggested security enhancements tailored to the observed weaknesses. These include:

- Adversarial Training
- Differential Privacy
- Confidence Score Obfuscation
- Input Filtering

- Regularization Techniques

User Interface

Protego our **AI Vulnerability Scanner** features a modern, intuitive user interface built with **React**, designed to accommodate both technical and non-technical users. Through this dashboard, users can:

- Upload and manage multiple CNN models (.h5)
- Launch vulnerability scans and view progress in real-time
- Access detailed model summaries and scanning history
- Download finalized reports with embedded mitigation steps

The UI bridges usability and technical depth, making Protego a powerful tool for cybersecurity analysts, ML engineers, and compliance teams alike.

Reporting

After processing the uploaded model, the system conducts an in-depth scan to identify vulnerabilities related to query-based model inversion attacks. It then generates a password-protected report summarizing the detected vulnerabilities and provides recommended mitigation steps based on **OWASP** guidelines and the **CVE** database. The report is securely shared with developers for review and further action.

High-Level System Architecture

The architecture of our AI vulnerability scanning system consists of **four primary layers**, each dedicated to a key part of the **query-based model evaluation lifecycle**. These layers are designed to work in tandem for secure, efficient, and accurate vulnerability analysis.

1. Data Ingestion Layer

- Imports **pretrained CNN models (.h5)** from users (e.g., MobileNet, ResNet, EfficientNet)
- Validates model structure, input/output compatibility, and associated metadata
- Standardizes input format (e.g., dimensions, normalization) to ensure compatibility with Protego's scanning engine

Note: Unlike traditional pipelines, this layer does not ingest raw training data.

2. Scanning Engine Layer

- Executes **Score-Based** and **Boundary-Based** model inversion attacks
- Simulates queries and analyzes model responses for **sensitive feature reconstruction risk**
- Identifies **decision boundary leakage**, output confidence anomalies, and reconstruction patterns
- Logs findings for reporting and assigns initial vulnerability severity levels

3. Reporting Module Layer

- Compiles results into a **password-protected PDF or HTML report**
- Includes:
    - Attack Type Breakdown
    - Suggested Mitigation Strategies (e.g., adversarial training, differential privacy, confidence score suppression)
    - Designed for **technical review**

4. User Interface Layer

The **Protego dashboard**, developed using **React**, offers a clean and professional interface that supports secure and structured model evaluations. It is designed for ease of use by both **cybersecurity analysts** and **machine learning engineers**, regardless of their background in adversarial machine learning.

Key functionalities include:

• Model Upload Interface

Users can securely upload **pretrained CNN models (.h5 format)** for evaluation.

• Attack-Based Scanning Initiation

Users can trigger Protego's **score-based** and **boundary-based** model inversion attack simulations.

• Scan History Access

The interface maintains a structured **scan history**, allowing users to revisit and manage previous evaluations.

• Comprehensive Report Downloads

After the scan, users can download a **password-protected, detailed report** summarizing:

- Attack simulation results
- Tailored mitigation strategies

Usability Across Teams

Designed with simplicity and security in mind, the UI supports collaboration between **AI engineers, security teams, and compliance departments**.

The High-Level System Operation

Protego follows a structured operational flow that mirrors its modular architecture. From model ingestion to report generation, each step aligns with one of the four core layers of the system to ensure secure, standardized, and thorough vulnerability evaluation

1. Model Ingestion

- The user uploads a pretrained **CNN model (.h5)** through the dashboard.

- Protego verifies the model's structure, input shape, and metadata for compatibility.
- If the model is malformed or unsupported, the system flags it and halts the scan.

2. Attack Simualtion

- Protego launches **score-based** and **boundary-based** inversion attacks.
- These are executed in a **sandboxed environment**, mimicking real-world query scenarios.
- The model's output behavior is recorded and analyzed for **reconstruction risks**, **sensitive feature leakage**, and **decision boundary exposure**.

3. Vulnerability Scoring & Mitigation Steps

- The system evaluates the model's responses and determines its **vulnerabilities**.
- Vulnerabilities are scored by severity and classified by architecture-specific risks.
- Based on the findings, Protego identifies and recommends appropriate **mitigation steps.**

4. Report Generation and Delivery

- Protego generates a **password-protected vulnerability report** in PDF and HTML format.
- The report contains:
- Attack type breakdown
- Specific, actionable mitigation guidance
- Once completed, the report is made available for download via the dashboard.

5. Scanning and Reporting

The user uploads their CNN model. The application runs a vulnerability scan and generates a downloadable, password-protected comprehensive report summarizing vulnerabilities and mitigation steps.

Software and Hardware Requirements

Software Requirements:

Platforms

- The software is compatible with both Linux and Windows operating systems; however, Windows is recommended for deployment due to its stability, security, and strong support for machine learning tools.

Tools

- Python is the primary programming language, providing access to essential libraries which are crucial for handling and manipulating neural network models.

- TensorFlow and PyTorch will be used for building, training, and testing neural network models, and Keras will provide a high-level neural network API to simplify model creation. Scikit-learn will be useful for implementing standard machine learning models and metrics, while the Adversarial Robustness Toolbox (ART) will be crucial for assessing vulnerabilities in AI models, particularly in relation to model inversion attacks. Libraries like OpenSSL are optional but important for ensuring secure communication between system components in case they are distributed across different networks.

- Scripting languages like Bash or PowerShell may be necessary for automating environment setups, deployments, and running scans on various models.

- Data management will involve using lightweight databases such as SQLite for small-scale testing, and Amazon Aurora for larger storage needs.

- For deployment, Docker will be used to containerize the application, ensuring consistency across different environments, and Jenkins or GitLab CI will automate testing and deployment pipelines to ensure code functionality and security.

- For hosting models in production and large-scale testing of vulnerabilities, where GPU or TPU instances are required, cloud environments like AWS will be utilized.

Hardware Requirements:

System Setup Hardware

- On the hardware side, a system with a multi-core processor is ideal for running simulations and parallel processing involving handling large neural networks and model inversion attacks. High RAM is recommended for handling large models or batch processing. High capacity SSDs are required to handle datasets, models, and results, with optional external/cloud storage for backups. Firewalls and VPNs are needed to maintain the security of environments, preventing breaches during testing and deployment phases.

Specialized Hardware

- For specialized hardware, Graphics Processing Units (GPUs) will significantly accelerate neural network training and scanning, and Tensor Processing Units (TPUs) may be used in cases requiring high scalability. Custom hardware like Field-Programmable Gate Arrays (FPGAs) or Application-Specific Integrated Circuits (ASICs) could also be considered for accelerating specific aspects of neural network processing.

Challenges

While our system aims to be robust and secure, several challenges arise:

1. Limited Ground Truth for Evaluation

- **Challenge:** There is no universal ground truth to measure the accuracy of model inversion, especially in score-based attacks, making it hard to benchmark attack success.
- **Impact:** Difficulties in validating Protego's effectiveness and measuring improvement after applying mitigations.

2. Model Diversity and Compatibility

- **Challenge:** CNN architectures differ widely in structure and behavior—even across variants of ResNet or EfficientNet—making consistent scanning and interpretation challenging.

- **Impact:** Could lead to false positives or incomplete vulnerability detection in lesser-tested architectures.

3. Performance Bottlenecks on Large Models

- **Challenge:** Query-based inversion simulations can be computationally intensive, especially on large models like EfficientNet-B7.
- **Impact:** Higher latency and memory usage, limiting real-time scanning or parallel scans.

4. Simulating Realistic Adversarial Behavior

- **Challenge:** Adversarial models in the real world may use advanced tactics beyond typical score or boundary-based attacks.
- **Impact:** Protego might fail to simulate certain real-world attack strategies, reducing overall robustness.

5. Risk of Misinterpretation by End-Users

- **Challenge:** Security ratings and vulnerability metrics can be misunderstood by non-technical users.
- **Impact:** Organizations may take inappropriate mitigation actions or develop a false sense of security.

6. Generalization to Non-Image Models

- **Challenge:** Protego is designed for CNNs used in image analysis. It may struggle to scale for NLP models or multi-modal networks.
- **Impact:** Limits applicability in organizations using diverse AI models beyond computer vision.

7. Dataset Sensitivity and Legal Constraints

- **Challenge:** Some scans may require datasets to validate attack outputs or mitigation strategies, which may not be available due to privacy regulations (e.g., HIPAA).
- **Impact:** Hampers verification of reconstructed data and the accuracy of inversion attacks.

8. Keeping Pace with Evolving Attack Techniques

- **Challenge:** Model inversion attacks continue to evolve rapidly in academia and black-hat communities.
- **Impact:** Protego must be constantly updated to remain relevant, which can be resource-intensive.

User Flow Diagram:

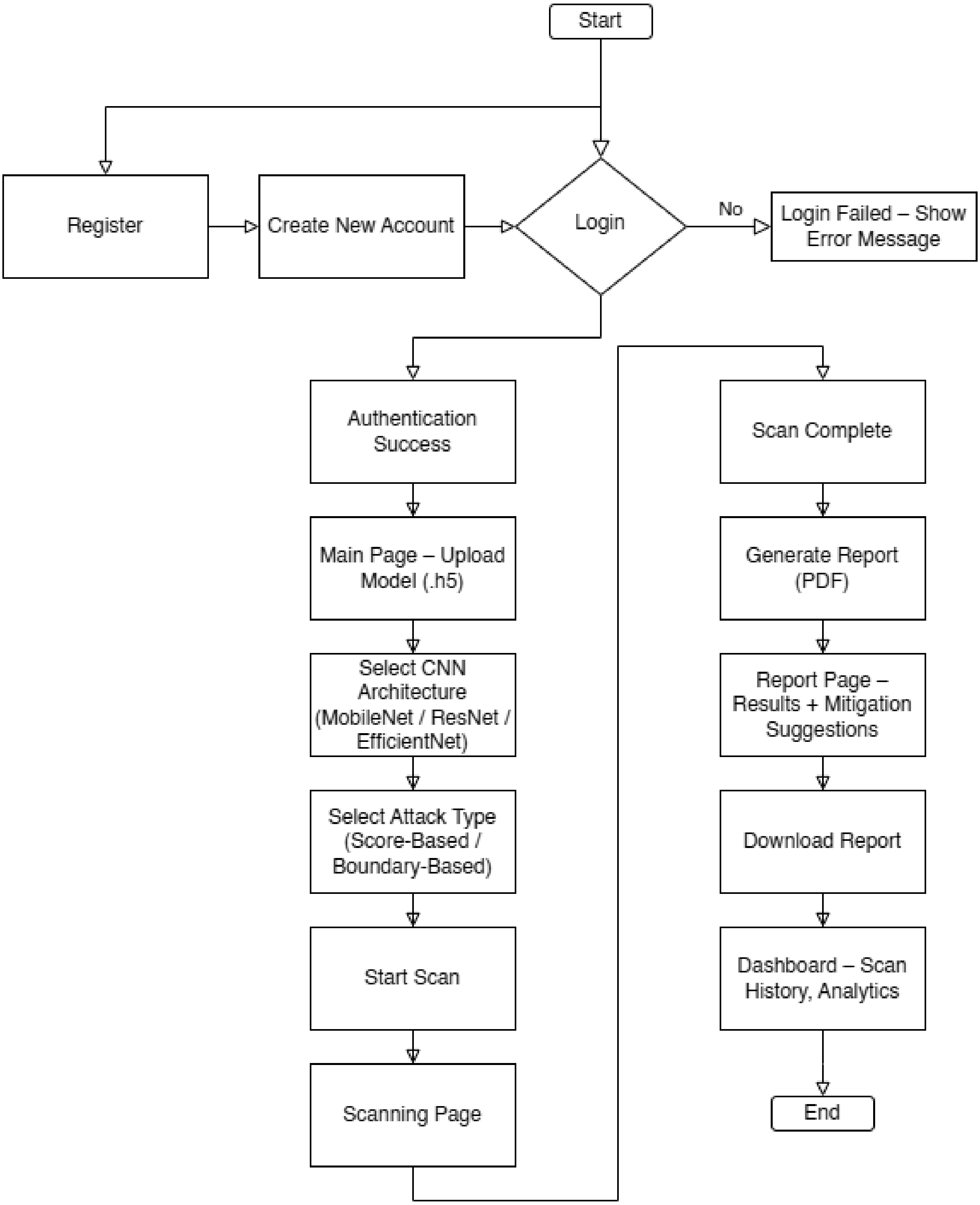

Start
Register
Create New Account
Login
No
Login Failed – Show Error Message
Authentication Success
Main Page – Upload Model (.h5)
Select CNN Architecture (MobileNet / ResNet / EfficientNet)
Select Attack Type (Score-Based / Boundary-Based)
Start Scan
Scanning Page
Scan Complete
Generate Report (PDF)
Report Page – Results + Mitigation Suggestions
Download Report
Dashboard – Scan History, Analytics
End

Figure 1: User-Flow Diagram

Future improvements

1. Support for More AI Architectures

- **Enhancement:** Extend compatibility beyond CNNs to include NLP models (e.g., BERT, GPT) and transformer-based vision models (e.g., ViT, Swin Transformer).
- **Impact:** Broaden Protego's applicability across industries using text, audio, and multi-modal data.

2. Cloud-Native Deployment & API Access

- **Enhancement:** Offer Protego as a **SaaS platform** with secure RESTful APIs for CI/CD integration and remote scans.
- **Impact:** Enables enterprise-wide adoption, allowing teams to integrate vulnerability scans into ML pipelines.

3. Federated & Differentially Private Analysis Modes

- **Enhancement:** Add scanning capabilities under **federated learning** or **differential privacy** settings to test robustness in decentralized data scenarios.
- **Impact:** Enhances use in privacy-critical environments like healthcare or finance, ensuring Protego aligns with future AI compliance models.

4. Custom Attack Profile Builder

- **Enhancement:** Allow security experts to define their own attack parameters, confidence score manipulation strategies, or gradient approximations for testing.
- **Impact:** Makes Protego flexible for advanced users performing customized penetration testing.

5. Plugin Ecosystem for Community Extensions

- **Enhancement:** Enable third-party developers to build and share attack plugins, custom scanners, and visualization tools via a plug-and-play architecture.
- **Impact:** Encourages open-source contributions, making Protego a community-driven security platform.

Timeline

In this report, we have included a Gantt chart to effectively manage and visualize our project timeline, ensuring we stay on track throughout its development. The chart spans from Autumn 2024 to Winter 2025, outlining the key phases of our final year project. These phases include **Planning**, where we set objectives and milestones; **Requirement Analysis**, where we gather and define project specifications; **Design**, focusing on the architecture and layout; **Implementation**, where the actual development takes place; and finally, **Deployment and Testing**, where we ensure the system functions as intended and is ready for delivery. This Gantt chart serves as a structured guide, helping us to allocate time efficiently and track progress across the project's lifecycle.

Gantt chart

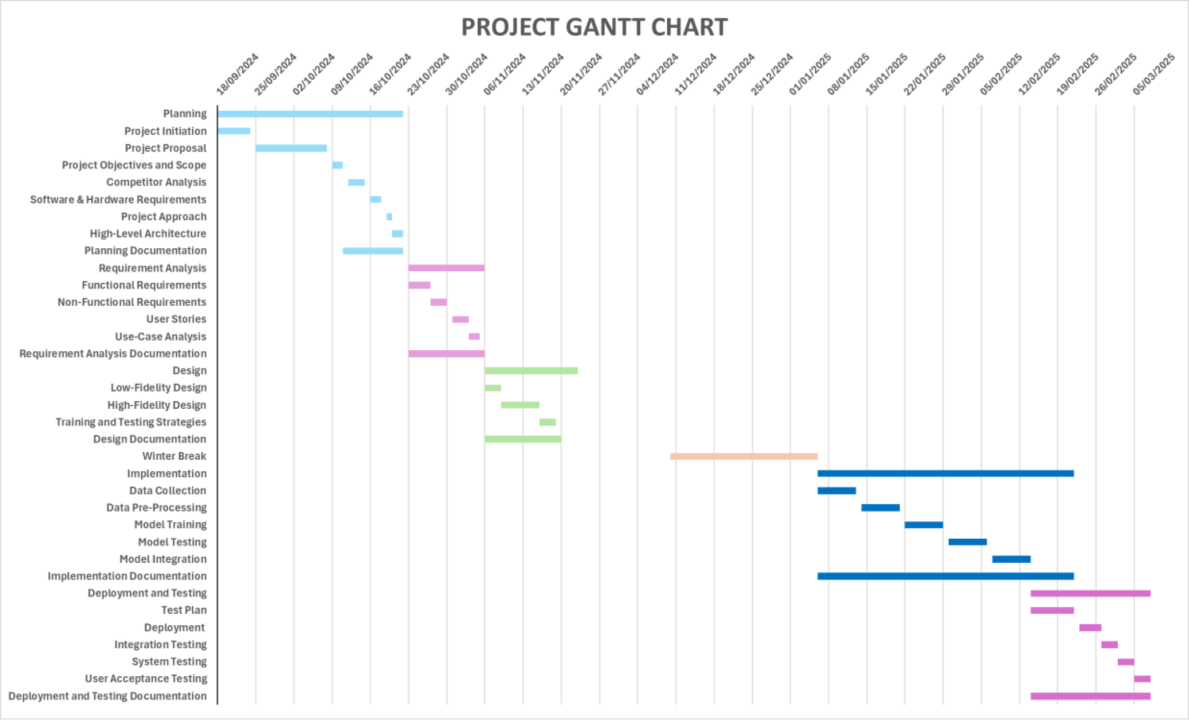
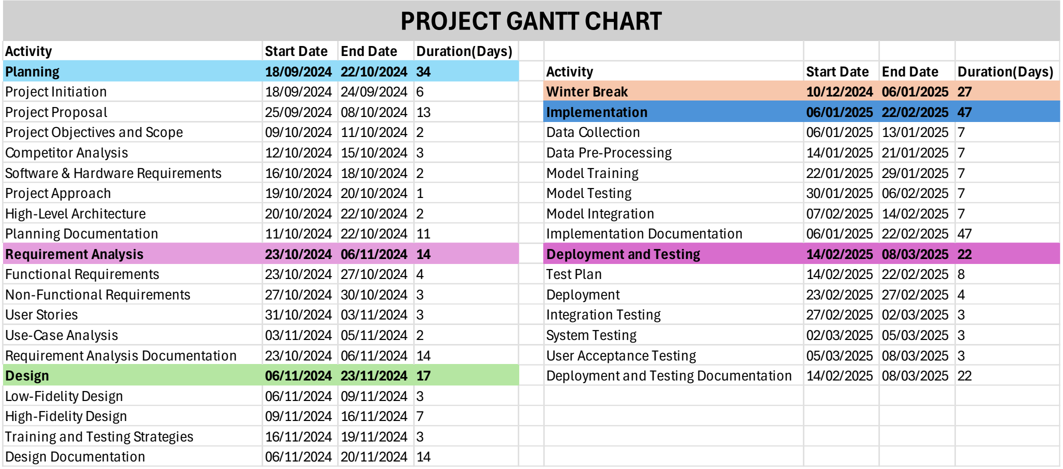


Figure 2: Gantt Chart



PROJECT GANTT CHART

Activity	Start Date	End Date	Duration(Days)
Planning	**18/09/2024**	**22/10/2024**	**34**
Project Initiation	18/09/2024	24/09/2024	6
Project Proposal	25/09/2024	08/10/2024	13
Project Objectives and Scope	09/10/2024	11/10/2024	2
Competitor Analysis	12/10/2024	15/10/2024	3
Software & Hardware Requirements	16/10/2024	18/10/2024	2
Project Approach	19/10/2024	20/10/2024	1
High-Level Architecture	20/10/2024	22/10/2024	2
Planning Documentation	11/10/2024	22/10/2024	11
Requirement Analysis	**23/10/2024**	**06/11/2024**	**14**
Functional Requirements	23/10/2024	27/10/2024	4
Non-Functional Requirements	27/10/2024	30/10/2024	3
User Stories	31/10/2024	03/11/2024	3
Use-Case Analysis	03/11/2024	05/11/2024	2
Requirement Analysis Documentation	23/10/2024	06/11/2024	14
Design	**06/11/2024**	**23/11/2024**	**17**
Low-Fidelity Design	06/11/2024	09/11/2024	3
High-Fidelity Design	09/11/2024	16/11/2024	7
Training and Testing Strategies	16/11/2024	19/11/2024	3
Design Documentation	06/11/2024	20/11/2024	14

Activity	Start Date	End Date	Duration(Days)
Winter Break	**10/12/2024**	**06/01/2025**	**27**
Implementation	**06/01/2025**	**22/02/2025**	**47**
Data Collection	06/01/2025	13/01/2025	7
Data Pre-Processing	14/01/2025	21/01/2025	7
Model Training	22/01/2025	29/01/2025	7
Model Testing	30/01/2025	06/02/2025	7
Model Integration	07/02/2025	14/02/2025	7
Implementation Documentation	06/01/2025	22/02/2025	47
Deployment and Testing	**14/02/2025**	**08/03/2025**	**22**
Test Plan	14/02/2025	22/02/2025	8
Deployment	23/02/2025	27/02/2025	4
Integration Testing	27/02/2025	02/03/2025	3
System Testing	02/03/2025	05/03/2025	3
User Acceptance Testing	05/03/2025	08/03/2025	3
Deployment and Testing Documentation	14/02/2025	08/03/2025	22

Figure 3: Timeline Details

Project Management Tools to be Used

- Project management will play a critical role in ensuring the project's success. Tools like Notion will track project progress, manage sprints, organize tasks, and for documentation to centralize project specifications, development reports, and design details, while Trello can serve as an alternative for teams preferring visual task management.

- Version Control Systems like Git, alongside GitHub, will manage source code, team collaboration, and track changes throughout the development process.

- Google Docs will be used for collaborative document creation and editing, allowing the team to work together on shared files.
- Lucidchart will be used to create flowcharts, system architecture diagrams, and project process maps, offering clear visual representations of project workflows.
- Communication among team members will take place on platforms like WhatsApp and Google Meets for real-time collaboration.
- Google Drive will serve as the primary file storage and sharing platform, ensuring that all documents, reports, and design files are easily accessible by team members.
- Lastly, for prototyping machine learning models and sharing code, cloud-based notebooks like Google Colab will be used to facilitate collaboration and experimentation.

Team Work Distribution

S no.	Name	Profile	Expertise	Work Distribution
1	Ali Sina Mohammad Arif	Leader	Application Dev, AI/ML, DevOps, CyberSecurity	Developing Front-end and of The AI Model (Reporting and Evaluation Layer)
2	Mansoor Kalemzai	Presentation Coordinator	AI/ML, Big Data, DevOps	Developing Front-end, AI Model (Model Training and Adversarial Testing Layer)
3	Varun Tulsiyani	Management Tool Coordinator	AI/ML, Big Data, DevOps	Developing, Front-end, AI Model (Model Training and Adversarial Testing Layer)
4	Abdul Rahim Kalsekar	Submission Coordinator	Cyber Security, DevOps, WebDev	Research in Cyber Security Perspective and assisting in Model Development (Data Ingestion and Validation Layer)
5	Abdulhayye Maricar Asfaq Ahamed	Scribe	CyberSecurity, DevOps, WebDev.	Research in Cyber Security Perspective and assisting in Model Development (Deployment and Monitoring Layer)

References

OWASP (2023) *ML03*

Model Inversion Attack, OWASP. Available at: https://owasp.org/www-project-machine-learning-security-top-10/docs/ML03_2023-Model_Inversion_Attack (Accessed: 1 October 2024).

IBM AI Fairness 360 (AIF360)

IBM (n.d.) *AI Fairness 360*. Available at: https://aif360.mybluemix.net/ (Accessed: 19 October 2024).

IBM Adversarial Robustness Toolbox (ART)

The Linux Foundation and IBM (n.d.) *Adversarial Robustness Toolbox*. Available at: https://adversarial-robustness-toolbox.readthedocs.io/ (Accessed: 19 October 2024).

Microsoft SEAL

Microsoft (n.d.) *Microsoft SEAL (Simple Encrypted Arithmetic Library)*. Available at: https://www.microsoft.com/en-us/research/project/microsoft-seal/ (Accessed: 19 October 2024).

General Data Protection Regulation (GDPR)

European Union (2016) *General Data Protection Regulation (GDPR)*. Available at: https://eur-lex.europa.eu/eli/reg/2016/679/oj (Accessed: 19 October 2024).

Health Insurance Portability and Accountability Act (HIPAA)

U.S. Department of Health & Human Services (1996) *Health Insurance Portability and Accountability Act (HIPAA)*. Available at: https://www.hhs.gov/hipaa/index.html (Accessed: 19 October 2024).

Rauber, J., Brendel, W., & Bethge, M. (2020). **Foolbox: A Python toolbox to benchmark the robustness of machine learning models.** *Journal of Machine Learning Research*, 21(197), 1-6. Available at: https://arxiv.org/abs/1707.04131 [Accessed 19 Oct. 2024].

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., & Swami, A. (2018). **CleverHans v2.1.0: An adversarial machine learning library.** *arXiv preprint*. Available at: https://arxiv.org/abs/1610.00768 [Accessed 19 Oct. 2024].