

**DECOY BASED PROTECTION FOR INTELLECTUAL
PROPERTY AGAINST AUTOMATED CYBERATTACKS**

by

GAYATHRI S

(Reg. No.: 963223622011)

A PROJECT REPORT

submitted to the

FACULTY OF

INFORMATION AND COMMUNICATION ENGINEERING

in partial fulfillment of the requirement

for the award of the degree

of

MASTER OF COMPUTER APPLICATIONS

in



PET ENGINEERING COLLEGE, VALLIOOR

ANNA UNIVERSITY

CHENNAI-600 025

AUGUST-2025

PET ENGINEERING COLLEGE, VALLIOOR**BONAFIDE CERTIFICATE**

Certified that this project report titled **“DECOY BASED PROTECTION FOR INTELLECTUAL PROPERTY AGAINST AUTOMATED CYBERATTACKS”** is the bonafide work of Ms. **Gayathri S (Reg. No.: 963223622011)** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the project work reported here in does not form part of any other project report on the basis of which a degree or award was confirmed on an earlier occasion on this or any other candidate.

PROJECT GUIDE

Mrs. P. Nazrin Beham, MCA., M.Phil.,
Assistant Professor,
Department of Computer Applications,
Vallioor.

HEAD OF THE DEPARTMENT

Dr. R. Kalaiselvi, B.E., M.E., Ph.D.,
Professor and Head,
Department of Computer Applications,
Vallioor.

PET ENGINEERING COLLEGE, VALLIOOR**Certificate of Viva-Voce Examination**

This is to certify that **Ms. Gayathri S (Reg. No.: 963223622011)**
has been subjected to viva-voce examination held on
..... in Department of Computer Applications, PET
Engineering College, Vallioor-627 117.

INTERNAL EXAMINER**EXTERNAL EXAMINER**

DECLARATION

I hereby declare that this project work titled **“DECOY BASED PROTECTION FOR INTELLECTUAL PROPERTY AGAINST AUTOMATED CYBERATTACKS”** submitted to **PET ENGINEERING COLLEGE, VALLIOOR** in partial fulfillment of the award of the degree of Master of Computer Applications, is a project work independently done by me under the guidance of **Mrs. P. Nazrin Beham, MCA., M.Phil., Assistant Professor**, Department of Computer Applications, PET Engineering College, Vallioor-627 117.

I hereby declare that this work was my original work and no part of it has been submitted for any other degree purpose or published in any other form till date.

Signature of the Candidate

(GAYATHRI S)

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I first thank the God Almighty who gave me this grace and strength to complete this project work successfully.

I express my sincere thanks to **Dr. K. Madhan Kumar**, M.E., Ph.D., Principal who encouraged and fulfilled all needs to complete this project.

I extend my profound thanks to **Dr. R. Kalaiselvi**, B.E., M.E., Ph.D., Professor and Head, Department of Computer Applications, who encouraged and gave me support to complete this project on time.

I sincerely thank my internal guide **Mrs. P. Nazrin Beham**, MCA., M.Phil., Assistant Professor, Department of Computer Applications, who has guided me to complete the project work in successful manner.

I also express my special thanks to all my teaching and non-teaching staff members of MCA Department for their assistance and to my parents, who have helped me both morally and financially to complete this project.



The Mind

building the future with mind...

Date: 26-06-2025



Project Completion Certificate

This is to certify that Ms. **GAYATHRI S (Reg.No:963223622011)**, a final year MCA student of **PET Engineering College, Tirunelveli**, has been successfully completed the internship with final semester project entitled "**Decoy Based Protection For Intellectual Property Against Automated Cyberattacks**" in our organization under the external guidance of Mr. **Tajudeen S**, during the period from **January 2025** to **May 2025**.

During this tenure, she displayed strong technical skills, eagerness to learn, and a professional attitude. She performed her tasks diligently, maintained regular attendance, and exhibited excellent conduct throughout the internship.

Congratulations on successfully completing this project! May this achievement be a stepping stone toward unlocking her full potential and paving the way for future accomplishments.

We wish her all the best in her future endeavors.

Sincerely



Project Coordinator

The Mind IT

TheMindIT

No.1, Ground Floor, ChristChurch Complex, Chatram BusStand, Trichy, Tamil Nadu-620002 www.themind.co.in hr@themind.co.in 0431-3559737

TABLE OF CONTENTS

| CHAPTER NO. | TITLE | PAGE NO. |
|-------------|------------------------------|----------|
| | LIST OF TABLES | x |
| | LIST OF FIGURES | xi |
| | LIST OF ABBREVIATIONS | xiii |
| | ABSTRACT | xiv |
| I | INTRODUCTION | 1 |
| | 1.1 ORGANIZATION PROFILE | 2 |
| | 1.2 PROJECT OVERVIEW | 3 |
| | 1.2.1 Need of Study | 4 |
| | 1.2.2 Objective | 5 |
| | 1.2.3 Future Scope | 6 |
| | 1.3. SYSTEM ANALYSIS | 7 |
| | 1.3.1 Existing System | 7 |
| | 1.3.2 Proposed System | 9 |
| | 1.3.3 Feasibility study | 12 |
| | 1.4 SYSTEM DESIGN | 13 |
| | 1.4.1 Input Design | 14 |
| | 1.4.2 Output Design | 16 |
| | 1.4.3 Architecture Design | 17 |
| | 1.4.4 Database Design | 19 |
| | 1.4.5 UML Desing | 23 |
| | 1.4.6 ER Diagram | 33 |

| | | |
|------------|---|-----------|
| | 1.5 SYSTEM SPECIFICATION | 35 |
| | 1.5.1 Hardware Specification | 35 |
| | 1.5.2 Software Specification | 35 |
| II | PROJECT DESCRIPTION | 37 |
| | 2.1 SYSTEM DESCRIPTION | 37 |
| | 2.2 MODULE DESCRIPTION | 38 |
| | 2.2.1 End User Dashboard | 39 |
| | 2.2.2 Adversary Model | 40 |
| | 2.2.3 Adversary Detection | 42 |
| | 2.2.4 Document Manipulation | 45 |
| | 2.2.5 Document Repository Modification | 47 |
| | 2.2.6 Alert Generator | 48 |
| | 2.3 SOFTWARE DESCRIPTION | 49 |
| | 2.4 FEATURES AND ADVANTAGES | 53 |
| III | TESTING METHODOLOGIES | 55 |
| | 3.1 TYPES OF TESTING | 55 |
| | 3.1.1 Unit Testing | 55 |
| | 3.1.2 Integration Testing | 56 |
| | 3.1.3 Performance Testing | 56 |
| | 3.1.4 Security Testing | 57 |
| | 3.1.5 Validation Testing | 57 |

| | | |
|-----------|------------------------------|-----------|
| IV | SYSTEM IMPLEMENTATION | 60 |
| | 4.1 INTRODUCTION | 60 |
| | 4.2 PRE IMPLEMENTATION | 61 |
| | 4.3 POST IMPLEMENTATION | 62 |
| V | RESULT AND CONCLUSION | 64 |
| | 5.1 CONCLUSION | 64 |
| | 5.2 FUTURE ENHANCEMENT | 65 |
| | APPENDIX | 66 |
| | REFERENCES | 76 |

LIST OF TABLES

| TABLE NO. | TABLE NAME | PAGE NO. |
|------------------|--------------------|-----------------|
| 1.1 | Users Table | 19 |
| 1.2 | Admin Table | 20 |
| 1.3 | Documents Table | 20 |
| 1.4 | Adversary Table | 21 |
| 1.5 | Detection Table | 21 |
| 1.6 | Decoy Table | 22 |
| 1.7 | Alert Table | 23 |
| 1.8 | Notification Table | 23 |

LIST OF FIGURES

| FIGURE NO. | FIGURE NAME | PAGE NO. |
|-------------------|------------------------|-----------------|
| 1.1 | Architecture Diagram | 18 |
| 1.2 | Use Case Diagram | 25 |
| 1.3 | Class Diagram | 26 |
| 1.4 | Collaboration Diagram | 28 |
| 1.5 | Sequence Diagram | 29 |
| 1.6 | Activity Diagram | 31 |
| 1.7 | Deployment Diagram | 32 |
| 1.8 | ER Diagram | 34 |
| App 1.1 | Home Page | 66 |
| App 1.2 | Admin Login | 66 |
| App 1.3 | Document Management | 67 |
| App 1.4 | User Management | 67 |
| App 1.5 | Adversary Monitoring | 68 |
| App 1.6 | Attacker Query | 68 |
| App 1.7 | Adversary File Request | 69 |
| App 1.8 | Adversary Alert | 69 |
| App 1.9 | Basic Shuffle | 70 |
| App 1.10 | Shuffle Increment | 70 |
| App 1.11 | Shuffle Reduction | 71 |
| App 1.12 | Change Topic | 71 |
| App 1.13 | Decoy Serving | 72 |

| | | |
|----------|---------------------|----|
| App 1.14 | Decoy Files Preview | 72 |
| App 1.15 | User Login | 73 |
| App 1.16 | User Home Page | 73 |
| App 1.17 | User File Access | 74 |
| App 1.18 | SMS Notification | 74 |
| App 1.19 | Email Notification | 75 |

LIST OF ABBREVIATIONS

| | | |
|---------------|---|---|
| DARD | - | Decoy Approaches for Robust Defense |
| LDA | - | Latent Dirichlet Allocation |
| TFIDF | - | Time Frequency Inverse Document Frequency |
| VAE | - | Variational Auto Encoder |
| IP | - | Intellectual Property |
| NLP | - | Natural Language Processing |
| NLTK | - | Natural Language Tool Kit |
| SQL | - | Structured Query Language |
| AWS | - | Amazon Web Services |
| KL Divergence | - | Kullback Leiber Divergence |
| PHP | - | Hypertext Preprocessor |
| HTML | - | Hypertext Markup Language |
| CSS | - | Cascading Style Sheets |
| CSV | - | Comma Seperated Values |
| ER | - | Enitivity Relationship |

ABSTRACT

ABSTRACT

Cyberattacks on Intellectual Property are increasingly common, with adversaries leveraging automated tools to exfiltrate and classify large document volumes to uncover sensitive data like trade secrets [1][6]. Traditional IP protection methods such as encryption and firewalls often fail against such advanced threats [2][7]. This system proposes the Decoy Approaches for Robust Defense against IP Theft, which uses deceptive techniques to mislead automated classification systems [6]. To detect adversarial behavior, DARD system employs a Variational Autoencoder to identify anomalies in user access and activity logs. Upon detection, the system alters the document repository to manipulate clustering and topic modeling, thus confusing adversaries [9]. This approach uses four manipulation techniques, Basic Shuffle, Shuffle Increment, Shuffle Reduction and Change Topic which replace original keywords with decoys to form misleading clusters. The pipeline includes text preprocessing using NLP, feature extraction with TFIDF, KMeans clustering, and topic modeling via Latent Dirichlet Allocation [8]. These techniques jointly ensure that adversaries receive deceptive outputs, while legitimate users access accurate content through a secure enclave architecture [4][5].

INTRODUCTION

CHAPTER I

1. INTRODUCTION

In the modern digital era, organizations face an increasing threat to the security of their Intellectual Property, particularly in environments where sensitive documents are stored and accessed electronically [1][7]. With the rise of automated tools and advanced data mining techniques, adversaries are now able to exploit digital document repositories using machine learning algorithms such as document classification, clustering, and topic modeling to extract valuable insights with minimal effort [6][9].

Traditional IP protection methods, such as encryption, authentication, and access control, are primarily passive and often fail to detect or mitigate the actions of sophisticated adversaries who mimic legitimate user behavior [2][4]. These techniques do not disrupt the automated analysis methods that attackers rely on to infer the structure, purpose, and importance of documents [6].

As a result, once access is gained either through credential theft, insider threats, or social engineering adversaries can rapidly classify and extract sensitive knowledge without raising immediate suspicion [3][5]. Additionally, most current systems lack the ability to detect anomalies in access patterns that may signify malicious behavior [9]. The absence of proactive and intelligent interference mechanisms allows attackers to exploit organizational data using unsupervised learning methods like KMeans or Latent Dirichlet Allocation which can automatically discover hidden topics and document relationships[6][8]. To address these gaps, the proposed DARD system introduces a novel and proactive defense mechanism by

integrating anomaly detection through Variational Autoencoders and deploying decoy based manipulation strategies[6]. These techniques mislead adversarial systems by shuffling document contents, altering topic structures, and generating plausible decoys, thereby disrupting automated analysis without hindering legitimate users [9].

By incorporating deception, anomaly detection, and intelligent manipulation, the DARD system redefines IP protection strategies to meet the demands of today's threat landscape ensuring secure, intelligent, and robust defense against automated IP theft [5][10].

1.1. Organization Profile

The Mind IT is a technology driven software company with a development center located in Trichy, Tamil Nadu. The company focuses on delivering tailor made software solutions that empower clients across various sectors such as healthcare, finance, logistics, and education. The Trichy branch plays a critical role in project development, testing, and remote client support. Known for its commitment to innovation and quality, The Mind IT follows a client centric approach and adopts the latest technologies like Python, React, Angular, and cloud computing.

Mission

The mission of The Mind IT is to deliver innovative, reliable, and client centric IT solutions that empower businesses to achieve their goals. The company is committed to building long term relationships with clients by maintaining transparency, quality, and integrity in every project. By fostering a collaborative work environment and embracing emerging technologies, The Mind IT aims to create value through continuous

improvement and technical excellence. The organization focuses on solving real world problems with scalable software solutions tailored to each client's unique needs.

Vision

The vision of The Mind IT is to become a globally recognized technology partner known for delivering impactful and sustainable digital solutions. The company aspires to lead the IT industry through its dedication to quality, innovation, and customer satisfaction.

Services

The Mind IT provides a wide range of IT services designed to support clients at every stage of their digital journey. Its offerings include custom software development, web and mobile application development, and enterprise solution implementation. The company also delivers cloud integration, DevOps, and software testing services to ensure scalable and high performance system.

1.2 Project Overview

The proposed system, DARD is a robust, security solution designed to protect intellectual property documents within the insurance domain[1][4]. This system addresses the growing threat of cyberattacks and insider threats by providing a proactive defense mechanism that not only detects adversarial activity but also deploys deceptive tactics to mislead and neutralize potential attackers[2][6]. The system centralizes the storage of sensitive documents including insurance policies, claim records, and financial reports within a secure IP repository accessible only to authenticated users under strict access control protocols[1][3].

DARD integrates advanced behavioral analysis using Variational Autoencoders to monitor document access patterns and identify anomalies such as unusual login behavior or high volume data retrieval[5][6]. Upon detecting adversarial behavior, the system activates its Document Manipulation Module, which intelligently modifies the document content through shuffling, keyword injection, content reduction, and topic alteration to prevent meaningful data extraction[6][8].

These alterations are designed to confuse clustering algorithms and topic modeling techniques commonly used by attackers. To reinforce document integrity, the system applies reclustering and retopic modeling using KMeans and Latent Dirichlet Allocation, generating misleading clusters and topics that degrade the attacker's ability to decipher sensitive content[6][9].

Real time alerts and notifications are automatically triggered to inform system administrators and security personnel of suspicious activities, ensuring swift incident response and containment. By combining deception based defense, machine learning, and secure repository management, the DARD system offers a novel and intelligent approach to safeguarding critical IP assets[2][6].

1.2.1 Need of Study

In today's digital ecosystem, safeguarding intellectual property and sensitive organizational data has become a growing challenge, especially with the rise in sophisticated and automated cyberattacks. Organizations often rely on centralized document repositories to store confidential information, which makes them attractive targets for adversaries.

Conventional security solutions such as firewalls and access controls, although essential, are no longer sufficient to deter advanced persistent threats and insider attacks. Therefore, there is a critical need to develop proactive, intelligent mechanisms that not only detect malicious access but also respond dynamically to potential threats.

This system addresses that gap by introducing a decoy based approach that intelligently detects unauthorized or abnormal access attempts using adversary modeling and machine learning. By simulating adversarial behavior and monitoring document access patterns, the system can distinguish between legitimate users and potential attackers.

Once suspicious behavior is detected, the system manipulates the documents in real time such as replacing content with decoy or redacted versions to mislead and delay the attacker, preventing real data leakage. This approach adds a deceptive and adaptive defense layer, which is essential in securing IP repositories. The need for this study arises from the increasing frequency of cyberattacks targeting sensitive data and the lack of existing systems that combine detection, deception, and response in a unified manner. Therefore, this study is vital for improving document security strategies in enterprise and institutional environments.

1.2.2 Objective

The primary objective of this system is to develop a secure and intelligent document access system that can proactively detect, mislead, and respond to unauthorized or suspicious access attempts. Traditional systems rely solely on authentication and authorization mechanisms, which may not be sufficient to protect intellectual property from evolving cyber threats,

especially those originating from insider threats or automated adversarial bots. To address this limitation, the system aims to integrate a decoy based defense strategy that introduces deceptive elements into the system to actively mislead potential attackers.

A key goal of the system is to implement an adversary model that simulates potential attack scenarios and detects abnormal access behaviors using pattern recognition and anomaly detection techniques. Upon detecting a threat, the system triggers document manipulation mechanisms such as injecting decoy content, redacting sensitive sections, or delivering fake documents thereby protecting the real data from being exposed or stolen.

By combining proactive detection, deceptive response, and administrative control, the system aims to build a comprehensive solution that enhances document security, mitigates the risk of data breaches, and supports real time decision making in sensitive environments.

1.2.3 Future Scope

One of the promising directions for future development is the integration of advanced machine learning and deep learning techniques to improve adversary detection accuracy. Moreover, the use of behavioral biometrics, such as typing patterns and access timing, can further enhance the precision of anomaly detection mechanisms.

The future version of this system can incorporate blockchain technology to ensure data integrity and maintain an immutable log of access and manipulation events. Integration with cloud storage services can also expand the usability of the system, allowing secure document management across distributed platforms without compromising on performance or

security. Moreover, mobile application support can empower administrators to receive and respond to security alerts on the go, improving reaction time in case of critical incidents.

The system can be extended to support multilingual documents and dynamic decoy generation based on document context, making it applicable in a wider range of industries such as finance, healthcare, and defense. Role based access controls can also be enhanced to provide finer grained permission structures. These advancements would not only strengthen the technical capabilities of the system but also expand its relevance and adoption across various sectors.

1.3 System Analysis

The increasing frequency and sophistication of cyberattacks on sensitive digital assets have necessitated the development of intelligent and adaptive security mechanisms. Traditional access control systems, while effective in managing user permissions, often fail to detect or prevent internal threats and automated intrusions that exploit vulnerabilities in the document storage layer. This system addresses these limitations through a detailed system analysis that identifies potential security gaps in conventional document repositories and proposes a robust defense framework using decoy based adversary detection and document manipulation techniques.

1.3.1 Existing System

Traditional systems primarily rely on user authentication and access control mechanisms, where login credentials, roles, and permission levels are used to restrict access to critical files and documents. These systems often include secure file storage platforms, firewalls, and intrusion detection

systems, which help monitor access activities and raise alerts when suspicious behavior is detected. However, such mechanisms are usually passive in nature, providing limited protection once an attacker bypasses authentication or when credentials are compromised.

Encryption transforms Intellectual Property into a secure format using algorithms, making it unreadable without the corresponding decryption key. This ensures that even if the data is intercepted, unauthorized individuals cannot access its contents. Common encryption standards include Advanced Encryption Standard and Rivest Shamir Adleman encryption, which provide high levels of security for protecting IP.

Firewalls act as a barrier between a trusted internal network and untrusted external sources. They control incoming and outgoing network traffic based on predefined security rules. Firewalls help prevent unauthorized access to systems and block potential malicious activity targeting sensitive IP.

Access control systems are used to restrict access to sensitive intellectual property based on predefined user roles and permissions. This ensures that only authorized personnel can access specific documents or resources. Common techniques include Role Based Access Control and Mandatory Access Control.

Intrusion Detection Systems are designed to monitor network traffic and system behavior to detect signs of unauthorized access or malicious activity. These systems alert administrators about potential threats, such as attempts to breach security measures or exploit vulnerabilities within the network before compromising the valuable IP.

Disadvantages

- **Ineffective Against Automated Attacks:** Most traditional systems are not equipped to detect or respond to sophisticated automated attacks that rapidly scan and extract data.
- **Vulnerable to Insider Threats:** Existing systems rely heavily on trust-based role access, making them easy to exploit by insiders who already have legitimate credentials.
- **Lack of Real Time Deception:** These systems focus mainly on access control and alerting, but do not employ dynamic deception techniques like document redirection or content manipulation to mislead attackers.
- **Delayed Response Mechanism:** Intrusion detection and log analysis often require manual review, causing delays in identifying and mitigating threats.
- **No Intelligent Adversary Profiling:** Current systems rarely include intelligent modeling of adversarial behavior, which limits their ability to predict or adapt to evolving threats.
- **Does Not Prevent Data Exfiltration:** These systems are mainly reactive and unable to prevent attackers from accessing or exporting real data once inside the system.

1.3.2 Proposed System

The Proposed System, the DARD system is designed to enhance the security of sensitive intellectual property by using advanced decoy and detection mechanisms. It specifically targets and disrupts automated techniques like clustering and topic modeling used by cyber adversaries to steal or analyze sensitive documents.

Decoy Based Protection

This technique revolves around deliberately altering or manipulating document content in a way that misguides adversaries. When attackers attempt to use data mining or natural language processing methods to analyze the repository, they are presented with misleading or irrelevant information. This confuses automated tools such as clustering algorithms and topic models making it difficult to extract meaningful insights or detect patterns. The actual sensitive data remains safe and hidden from these automated threats.

Variational Autoencoder for Adversary Detection

A Variational Autoencoder is a deep learning technique used for unsupervised anomaly detection. In the DARD system, VAE is trained on normal user behavior like access frequency, document types viewed, and navigation patterns. When a user's behavior significantly deviates from the learned norm such as bulk document access, unusual login times, or access to diverse unrelated categories it is flagged as suspicious or adversarial. This enables the system to detect and respond to potential IP theft attempts early.

Document Repository Modification

To proactively combat automated attacks, the system modifies the document repository once adversarial behavior is detected. These modifications involve changing the structure and content of documents in ways that prevent successful execution of text mining techniques. Even if an attacker bypasses access restrictions, the manipulated content prevents meaningful data extraction. Meanwhile, legitimate users still access the original documents securely via the Secure Access Module.

Four Manipulation Techniques

The DARD system applies four strategic document manipulation techniques to effectively misguide the adversaries.

- **Basic Shuffle:** Randomly rearranges keywords within the document. This disrupts the original semantic flow, making it harder for adversaries to comprehend or classify the document accurately.
- **Shuffle Increment:** Adds decoy or irrelevant keywords to the document. These additional terms increase noise and obscure the document's true context, confusing topic modeling algorithms like LDA.
- **Shuffle Reduction:** Removes selected original keywords from the document. This leads to partial information loss from an attacker's perspective, weakening the clustering results and making topic identification imprecise.
- **Change Topic:** Replaces original keywords with unrelated or misleading terms, effectively altering the perceived topic of the document. This misguides automated models trying to determine the document's true content or purpose.

1.3.2.1 Advantages

- Detects adversarial behavior using intelligent pattern analysis.
- Protects sensitive documents through real time manipulation techniques.
- Misleads attackers by delivering decoy or redacted documents.
- Sends instant notifications to admin via SMS and email alerts.
- Reduces the risk of data exfiltration through proactive response.

- Ensures only admin approved users can access the document repository.
- Supports dynamic and adaptive security based on user behavior.
- Increases document access security without affecting usability.
- Helps in early identification of both internal and external threats.
- Minimizes manual intervention through automation and alerting.

1.3.3 Feasibility Study

The proposed system is highly feasible as it leverages widely supported technologies such as Python for adversary detection, secure document handling, and integration with web based front ends. The system architecture supports modular design, enabling smooth integration of document manipulation, user authentication, and alert generation components. Modern encryption libraries, secure file storage practices, and notification APIs like SMS and email services are readily available and compatible with the technology stack.

1.3.3.1 Technical Feasibility

The system is technically feasible as it leverages proven technologies such as Python for backend logic, MySQL for database management, and machine learning frameworks like TensorFlow, Keras, and Scikit learn for implementing document classification disruption and adversary detection. NLP libraries such as NLTK and SpaCy are used for document processing and topic modeling interference. The system employs Variational Autoencoders for anomaly detection, and integrates manipulation techniques such as Basic Shuffle, Shuffle Increment, Shuffle Reduction, and Change Topic. All required components are supported by open source ecosystems and are compatible with modern development environments.

1.3.3.2 Economical Feasibility

The economic feasibility of the system is strong due to the use of open source technologies such as Python, TensorFlow, and MySQL, minimizing initial development costs. Infrastructure needs are modest, with deployment possible on local servers or cloud platforms like AWS or Google Cloud. Investment may be required for scaling the system to large document repositories or for advanced threat detection capabilities. In return, the system offers significant value by preventing intellectual property theft, reducing legal risks, and protecting proprietary data, making it a cost effective solution for organizations.

1.3.3.3 Operational Feasibility

The system is designed to function seamlessly within an organization's existing document storage and access workflows. It introduces minimal disruption to legitimate users while strategically confusing unauthorized entities attempting automated analysis. The integration of decoy mechanisms and secure user access controls ensures a balanced approach to usability and security. The user friendly interfaces for administrators and users support efficient management of document repositories and security events. Overall, the system enhances operational resilience against cyber threats targeting intellectual property.

1.4 System Design

The system design of this system focuses on building a secure, intelligent, and modular architecture that ensures controlled document access, dynamic threat detection, and real time response. The architecture is based on a client server model where the administrator plays a central role

in managing users, documents shown in App 1.3 and App 1.4 respectively, and monitoring alerts. At the core of the system lies the adversary detection engine outlined in App 1.5, which uses access behavior patterns and predefined anomaly indicators to detect suspicious activity.

Once an anomaly is flagged the system initiates document manipulation techniques that replace the original content with decoy or redacted versions shown in App 1.10 to App 1.13 mislead potential attackers.

The front end design offers intuitive user interactions while the back end handles data processing and response automation. Through this well structured system design, the system ensures both security and efficiency in protecting sensitive documents from theft or unauthorized access.

1.4.1 Input Design

The input design focuses on collecting, validating, and processing data securely across different modules.

User Authentication

Users provide login credentials such as email/username and password to access the system. One of the critical input areas is user authentication, shown in App 1.16, App 1.15 and App 1.2 where both admins and authorized users enter their login credentials, such as email or username and passwords.

These inputs are securely validated to prevent unauthorized access and ensure that only legitimate users interact with the system. The login process also logs timestamps, which become vital in adversary behavior tracking illustrated in Table 1.1.

Document Upload

Admins upload documents by entering titles, selecting categories, adding descriptions, and attaching files described in App 1.3. Keywords and timestamps are also captured. During this process, the admin provides metadata such as the document title, category, descriptive tags, and optionally keywords, along with the actual file stored in Table 1.3.

These inputs are not only used for organizing the repository but also serve as parameters in search and manipulation operations. User access requests are also a major input type, where users enter search queries, filter options, or document criteria as seen in App 1.17.

Behavior Monitoring

The system automatically records user activities like login times and document access frequency, shown in App 1.5 to detect adversarial behavior patterns also illustrated in Table 1.5.

Manipulation Trigger

When suspicious behavior is detected, the manipulation engine is triggered using internal rule based inputs, which determines what kind of manipulation illustrated in App 1.9 to App 1.12 should be applied to the requested documents that are managed in the Table 1.6.

Alert Configuration

Alert configuration inputs allow the admin to define thresholds for abnormal activity, specify notification preferences SMS as seen in App.1.19, email as shown in App 1.18 and provide contact details to receive alerts in real time. These inputs collectively ensure that the system operates securely and responsively that is described in the Table 1.7 and Table 1.8.

1.4.2 Output Design

The output design ensures users receive relevant and timely information clearly and securely.

Admin Dashboard

Admins view summaries of user management, document stats, and alerts shown in App 1.3 to App 1.5. This dashboard becomes the central hub for tracking potential threats and overall repository health. This leads to an effective user and document management as seen in Table 1.1 to 1.3 and 1.7.

User Document Access

Users see results with document previews, and related files as illustrated in Table 1.1 and 1.3 if authorized, described in App 1.17.

Adversary Detection

When suspicious activity is detected, the system flags it as shown in Table 1.5 and App 1.8 for review and triggers document manipulation. Output indicators include access time anomalies, spike patterns, or unusual topic interest, visualized in App 1.6 and App 1.7.

Manipulated Document

In cases where manipulation is triggered as illustrated in App 1.9 to App 1.12, manipulated document outputs are sent to adversaries. These outputs may include decoy content, altered keywords or redacted sections as shown in Table 1.6, making sensitive data inaccessible and protecting the integrity of the real information as visualized in App 1.13 and App 1.14.

Notification

Notifications are sent via dashboard, SMS, and email to notify admins about critical events instantly which is managed in Table 1.8 as shown in App 1.18 and App 1.19.

1.4.3 Architectural Design

The architectural design of the DARD system lays the foundation for a secure, modular, and scalable framework for protecting sensitive documents against unauthorized access and adversarial threats. This layered architecture integrates traditional user access workflows with advanced cybersecurity measures, including adversary behavior detection, intelligent document manipulation, and real time alerting.

Figure 1.1 illustrates the architectural design of the DARD system, which integrates traditional document management workflows with intelligent adversary detection and decoy response mechanisms. The architecture is segmented into several key functional components: User, Admin, Adversary, Document Repository, and the DARD System.

The Admin module allows authorized users to authenticate and securely access documents stored in the Document Repository as shown in App 1.15 to App 1.17, also manages system operations, including authentication, user management, document uploading and deletion, and alert monitoring as described in App 1.2 to App 1.5. Only the admin can add valid users and upload or manage sensitive documents.

The Adversary section models an external entity attempting unauthorized data access as portrayed in App 1.6 and App 1.7. The adversary bypasses the login process and sends direct file requests to the repository using techniques such as TFIDF for querying, Latent Dirichlet Allocation for topic modeling, and KMeans for document clustering. These actions aim to extract intellectual content or patterns from the documents. The DARD System responds to such behavior with multiple defensive layers.

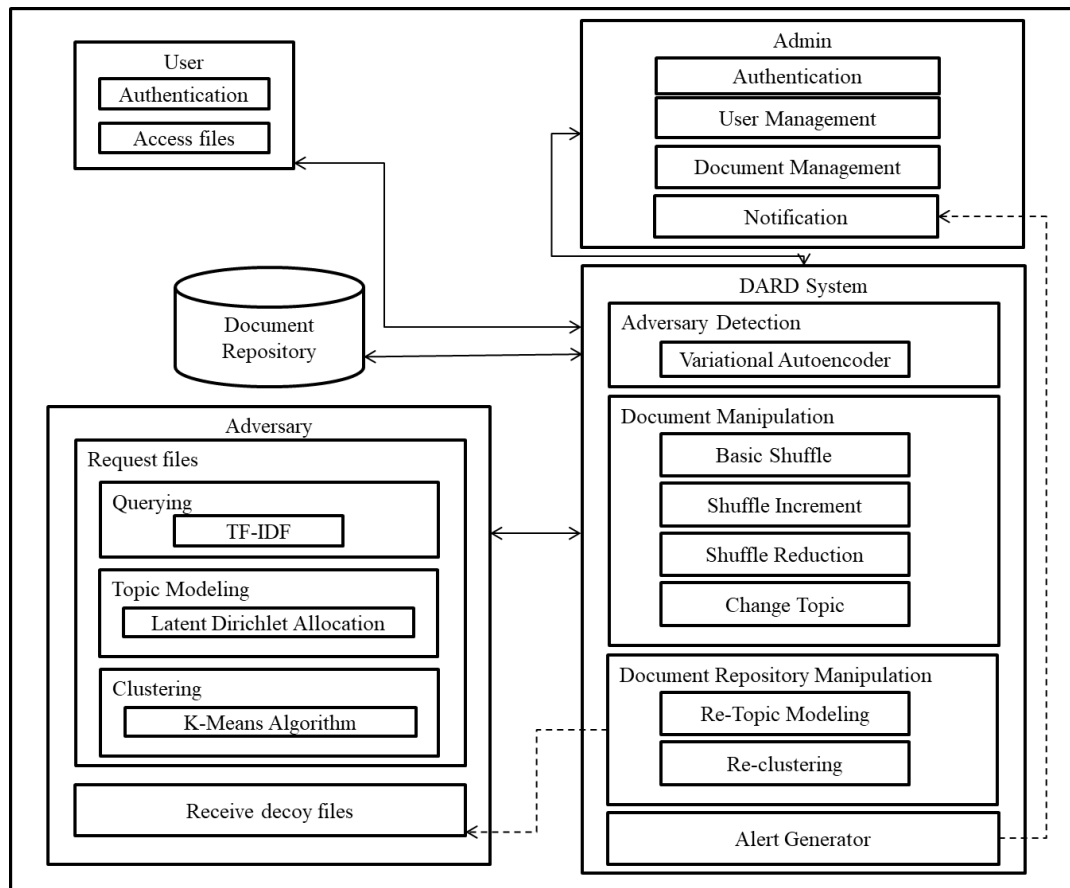


Figure 1.1 Architecture Diagram

Adversary Detection can be seen in App 1.8, powered by a Variational Autoencoder, monitors access patterns and identifies anomalous behavior indicative of adversarial intent. Adversary Detection module, which employs a Variational Autoencoder model trained on a synthetic dataset. This dataset is auto generated with 100 samples and 5 distinct behavioral features, allowing the model to learn patterns. These patterns are about normal and adversarial behavior. Once trained, the VAE detects deviations from typical user behavior, signaling potential attacks. When such behavior is identified, the Document Manipulation module is triggered as visualized in App 1.9 to App 1.13. The Document Repository Manipulation module performs Re Topic Modeling and Re Clustering to serve decoy files to the adversaries as shown in App 1.13 and App 1.14.

Upon detection, the Alert Generator immediately notifies the admin through SMS and email notifications as described in App 1.18 and App 1.19. Figure 1.1 represents a robust and adaptive architecture designed to protect organizational intellectual property against evolving cyber threats.

1.4.4 DATABASE DESIGN

Database design is a foundational aspect of the DARD system, ensuring the secure storage, efficient retrieval, and relational integrity of all system data. The database schema supports the entire functionality of the system including user authentication, document management, adversary detection, manipulation of documents and generating alerts and notifications. This section details the various tables used in the system along with their structure. Each table is uniquely identified by a Primary Key and is connected to related tables using Foreign Keys to maintain referential integrity.

1.4.4.1 Users Table

| Column Name | Data Type | Constraints |
|---------------|-------------------|---------------------------|
| user_id | Int (Primary Key) | Auto_Increment |
| Username | Varchar(25) | Unique, Not Null |
| Email | Varchar(40) | Unique, Not Null |
| password_hash | Varchar(255) | Not Null |
| created_at | Timestamp | Default Current_Timestamp |
| mobile_no | Bigint(10) | Unique, Not Null |

Table 1.1 Users Table

The table 1.1 stores data related to the end users who interact with the document system. This includes login credentials and profile information.

Key attributes include user_id, username, email, password, created_at and mobile_no. This table is crucial for authenticating legitimate users and maintaining a secure login mechanism as described in App 1.15.

1.4.4.2 Admin Table

| Column Name | Data Type | Constraints |
|-------------|-------------------|---------------------------|
| admin_id | Int (Primary Key) | Auto_Increment |
| Username | Varchar(20) | Unique, Not Null |
| Email | Varchar(30) | Unique, Not Null |
| Password | Varchar(255) | Not Null |
| created_at | Timestamp | Default Current_Timestamp |
| mobile_no | Bigint(10) | Unique, Not Null |

Table 1.2 Admin Table

The table 1.2 contains details about administrative users who manage document uploads, notifications, and user behavior monitoring. Attributes include admin_id, username, email, password, and created_at. Similar to the user table, it supports secure login for admin roles and keeps audit trails for administrative operations as visualized in App 1.2.

1.4.4.3 Documents Table

| Column Name | Data Type | Constraints |
|----------------|-------------------|---------------------------|
| doc_id | Int (Primary Key) | Auto_Increment |
| doc_name | Varchar(25) | Not Null |
| doc_type | Varchar(10) | Null |
| upload_date | Timestamp | Default Current_Timestamp |
| uploaded_by | Int | Foreign Key (ins_admin) |
| encrypted_data | Blob | Not Null |
| access_control | JSON | Not Null |

Table 1.3 Documents Table

The Table 1.3 manages metadata for both original and manipulated documents. The primary key is `doc_id`, and other attributes include `doc_name`, `doc_type`, `upload_by`, `upload_date`. The `uploaded_by` field is a foreign key referencing `admin_id`, denoting which admin uploaded the document as shown in App 1.3.

1.4.4.4 Adversary Table

| Column Name | Data Type | Constraints |
|-------------------------------|-------------------|---------------------------|
| <code>attack_id</code> | Int (Primary Key) | Auto_Increment |
| <code>doc_id</code> | Int | Foreign Key (ins_docs) |
| <code>attack_type</code> | Varchar(25) | Not Null |
| <code>attack_timestamp</code> | Timestamp | Default Current_Timestamp |

Table 1.4 Adversary Table

The Table 1.4 captures details of simulated attack behaviors that are detected by the adversary detection module. Its primary key is `attack_id`, and the foreign key `doc_id` links it to the table 1.3. It also includes fields such as `attack_type`, indicating how the simulated attack was generated and what kind it was that is keyword based and clustering shown in App 1.6.

1.4.4.5 Detection Table

| Column Name | Data Type | Constraints |
|----------------------------------|-------------------|---------------------------|
| <code>detection_id</code> | Int (Primary Key) | Auto_Increment |
| <code>doc_id</code> | Int | Foreign Key (ins_docs) |
| <code>anomaly_type</code> | Varchar(25) | Not Null |
| <code>detection_timestamp</code> | Timestamp | Default Current_Timestamp |

Table 1.4 Detection Table

The Table 1.5 records instances of potential adversarial behavior detected by the system as visualized in App 1.8. The primary key is

detection_id, and foreign keys and document_id connects it to the Table 1.3. Additional fields include anomaly_type and which define the nature and likelihood of the detected adversarial activity. This table feeds into the alert generation and document manipulation components.

1.4.4.6 Decoy Table

| Column Name | Data Type | Constraints |
|-----------------|--|------------------------------|
| manip_id | Int (Primary Key) | Auto Increment |
| doc_id | Int | Foreign Key (ins_docs) |
| manip_type | Enum('Basic Shuffle', 'Shuffle Increment', 'Shuffle Reduction', 'Change Topic') | Not Null |
| manip_data | Blob | Not Null |
| manip_timestamp | Timestamp | Default Current Timestamp |

Table 1.6 Decoy Table

The Table 1.6 keeps a record of all manipulated or decoy documents. The primary key is manip_id, and the foreign key doc_id points to the original entry in the table 1.3. It includes manip_type such as shuffle or topic change and modified_data, which holds the actual manipulated content shown to adversaries as described in App 1.9 to App 1.13.

1.4.4.7 Alert Table

The Table 1.7 documents security alerts generated during system operation. The primary key is alert_id, and the foreign key admin_id links it to the table 1.2. Additional fields like alert_msg, and alert_timestamp helps

identify unresolved alerts and support administrators in taking appropriate action as the alert shown in App 1.5.

| Column Name | Data Type | Constraints |
|-----------------|-------------------|---------------------------|
| alert_id | Int (Primary Key) | Auto_Increment |
| admin_id | Int | Foreign Key (ins_admin) |
| alert_msg | Text | Not Null |
| alert_timestamp | Timestamp | Default Current_Timestamp |

Table 1.7 Alert Table

1.4.4.8 Notification Table

The Table 1.8 handles notifications and messages sent to admin as described in App 1.18 and App 1.19. Its primary key is notif_id, and recipient_id acts as a foreign key that could point to the admin that is Table 1.2. Other fields include email and Notif_msg, which determine the nature and content of the message sent.

| Column Name | Data Type | Constraints |
|----------------|-------------------|---------------------------|
| notif_id | Int (Primary Key) | Auto_Increment |
| receiptent_id | Int | Foreign Key (ins_admin) |
| notif_msg | Text | Not Null |
| sent_timestamp | Timestamp | Default Current_Timestamp |

Table 1.8 Notification Table

1.4.5 UML Design

UML diagrams help in modeling the structure and dynamic behavior of the system across modules like user authentication, document repository, adversary detection, and alert generation. These diagrams guide both developers and stakeholders in understanding the flow of data, user

interactions, and system processes clearly and accurately. In DARD system, UML design has been applied to represent the relationship between entities, the sequence of actions taken during user and adversary interactions, and the structural composition of modules.

1.4.5.1 Use Case Diagram

The use case diagram provides a high level functional overview of the DARD system. Figure 1.2 outlines the interactions between different actors namely the Admin, User, and Adversary and the system's core functionalities. Each actor has specific responsibilities.

The Admin actor performs key administrative tasks such as login, user authentication, user management that includes adding or removing users, manage document uploads and deletions which are described in App 1.2 to App 1.4. In addition, admins have exclusive access to monitor ongoing activities for any potential attacks as in App 1.5, and receives system generated notifications about adversary behavior as visualized in App 1.19.

The Adversary actor attempts to request sensitive documents using automated querying techniques as shown in App 1.6. Once such suspicious activity is identified through anomaly detection, the system triggers manipulation processes that is described in App 1.9, include document shuffling or topic alterations. These manipulated files are then sent in response as illustrated in App 1.13. Simultaneously, alerts are generated and sent to the Admin, keeping them informed in real time. The User actor has activities respectively primarily logging in, getting authenticated, and accessing authorized files securely that are shown in App 1.15 to App 1.17.

The Figure 1.2 offers a comprehensive view of how legitimate users and adversaries interact with the system and how the DARD system maintains integrity through intelligent threat detection and response. This visual representation helps understand the core functionalities and responsibilities of each actor within the system.

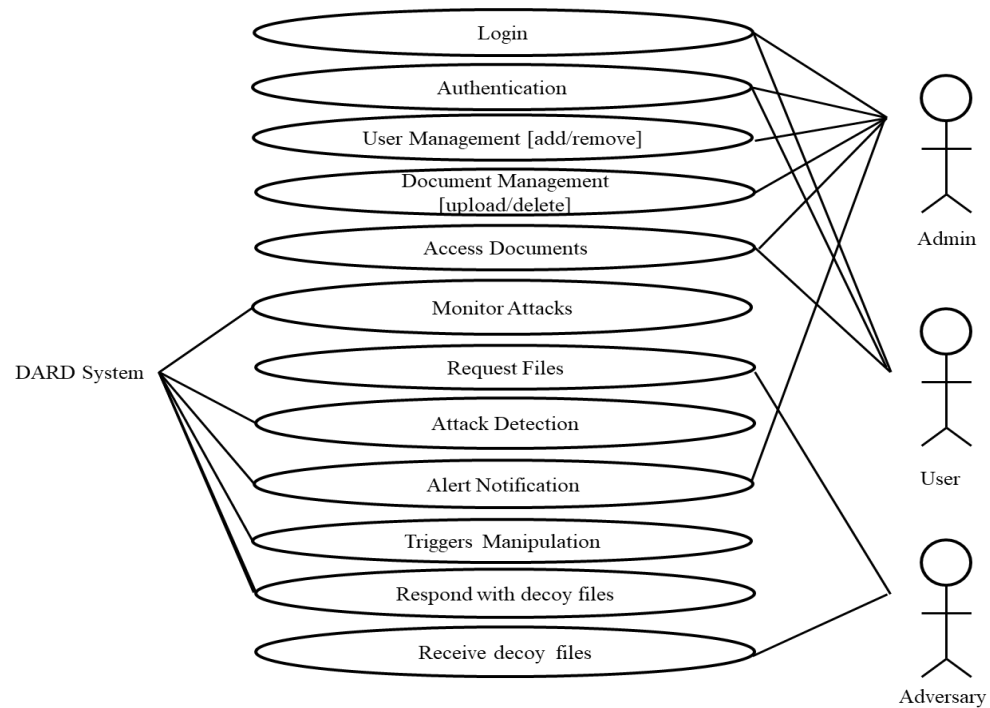


Figure 1.2 Use case Diagram

1.4.5.2 Class Diagram

The class diagram shown in Figure 1.3 represents the object oriented structure of the system by modeling its core entities User, Admin, Adversary, and DARD System along with their attributes and operations. The User class includes attributes like user_id, username, and password, which uniquely identify each end user and provide authentication credentials that are illustrated in Table 1.1. The method view_userdoc() allows a user to access documents based on access privileges. On the

administrative side, the Admin class has similar structural fields `admin_id`, `username`, and `password` and is responsible for the `user_management()` and `doc_management()` operations as illustrated in Table 1.2, indicating administrative privileges such as managing user accounts and uploading documents to the repository.

The Adversary class is designed to simulate potential threats to the system. It contains attributes like `attack_id` described in the Table 1.4 and `doc_id` used to track the adversary's interaction with specific documents shown in Table 1.3.

The operations defined in this class `attack_query()` and `doc_cluster()` reflect querying behavior and document request processing, often used in automated IP theft. This mirrors the adversarial component of the architecture, where unauthorized access attempts are detected and responded to accordingly.

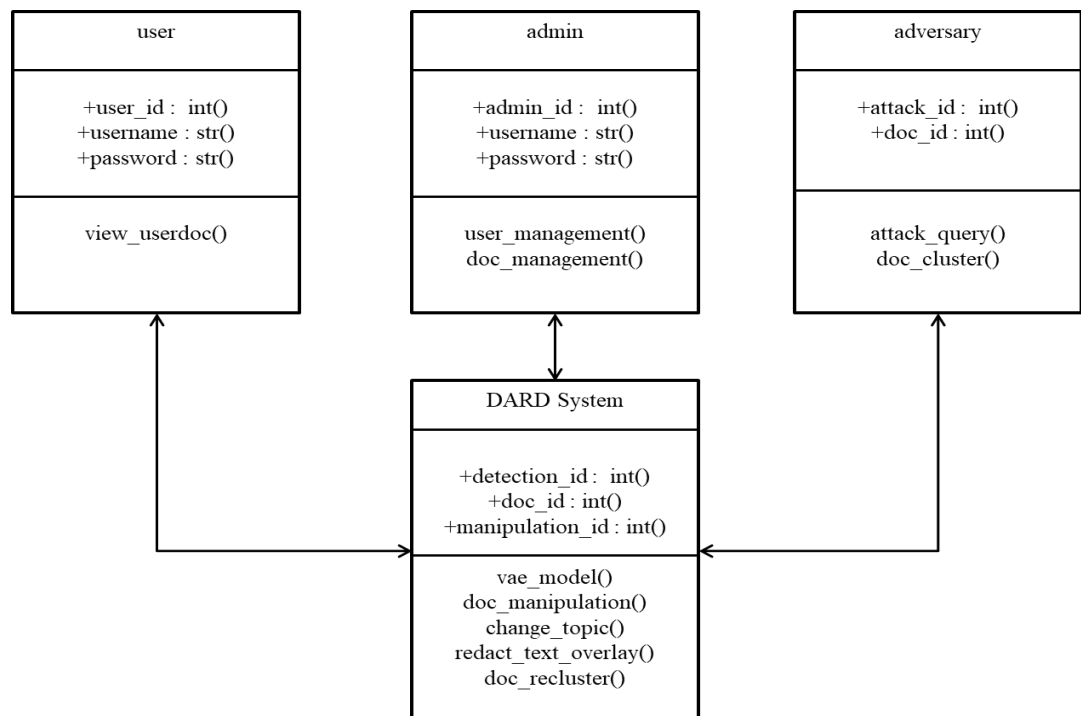


Figure 1.3 Class Diagram

At the core of the solution lies the DARD System class. It includes critical fields such as `detection_id`, `doc_id`, and `manipulation_id` that track adversary activity, targeted documents as illustrated in Table 1.5 and manipulation strategies used. The DARD System uses `vae_model()` to detect the adversary activities and also exposes several manipulation methods that are `doc_manipulation()`, `shuffle_reduction()`, `change_topic()`, and `redact_text_overlay()`.

After the manipulation the decoy documents are modeled using re topic modeling and `doc_cluster()` method clusters the modified documents to mislead the adversary clusters as described in Table 1.6.

These functions are executed dynamically once an adversary is detected, ensuring the decoy approach is applied to mislead the attacker while preserving system integrity. Together, these classes represent the control flow of authentication, adversary detection, document manipulation, and secure document access within the system.

1.4.5.3 Collaboration Diagram

The collaboration diagram illustrated in Figure 1.4 highlights how actors such as the Admin, User, and Adversary interact with the Document Repository and the DARD System, using numbered messages to indicate the sequence of actions.

In the initial step 1, the Admin interacts with the Document Repository to upload and manage documents. These documents include sensitive or intellectual property files that need to be securely stored. The Admin is responsible for adding, categorizing, and organizing these files. In step 2, the Admin configures the DARD System, specifying rules for

adversary detection, manipulation strategies, and alert thresholds. These configurations allow the system to automatically act upon suspicious activities without manual intervention.

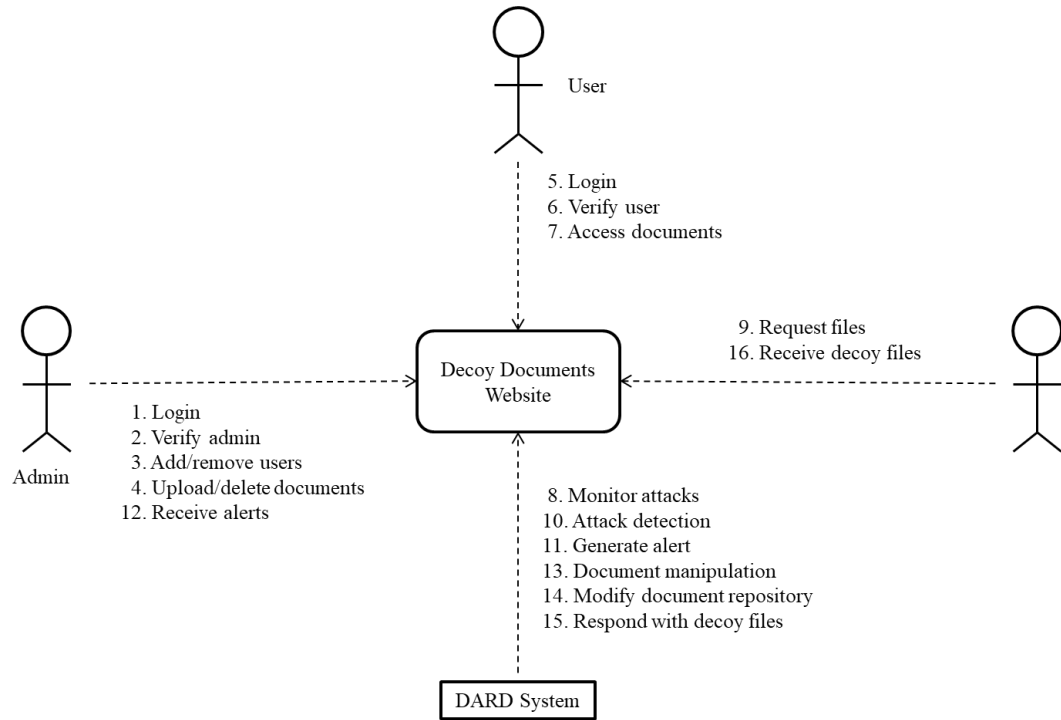


Figure 1.4 Collaboration Diagram

In step 3, a User logs into the system and accesses the document repository. If the user is verified and acts within normal behavioral patterns, the request proceeds directly to the repository in step 4, and the requested file is delivered without interference. Meanwhile, step 5 represents a critical point: an Adversary attempts to access files from the document repository. While the system treats this request like any other, it secretly forwards this behavioral activity to the DARD System in step 6.

In step 7, the DARD System uses a trained Variational Autoencoder model, which evaluates this behavior based on a synthetic dataset to detect anomalies. If adversarial behavior is detected, the system triggers

manipulation protocols. Then, in step 8, it manipulates the documents using keyword shuffling or topic changes. In step 9, instead of the actual file, the adversary receives a decoy document. Simultaneously, the system sends an alert to the Admin in step 10, informing them of the adversary's activity.

1.4.5.4 Sequence Diagram

A sequence diagram used to represent the interaction between objects or components in a system in a time ordered sequence. It shows how messages are passed between different system parts to carry out a specific function or process. The diagram typically consists of vertical lifelines for each object and horizontal arrows indicating the messages or method calls exchanged between them.

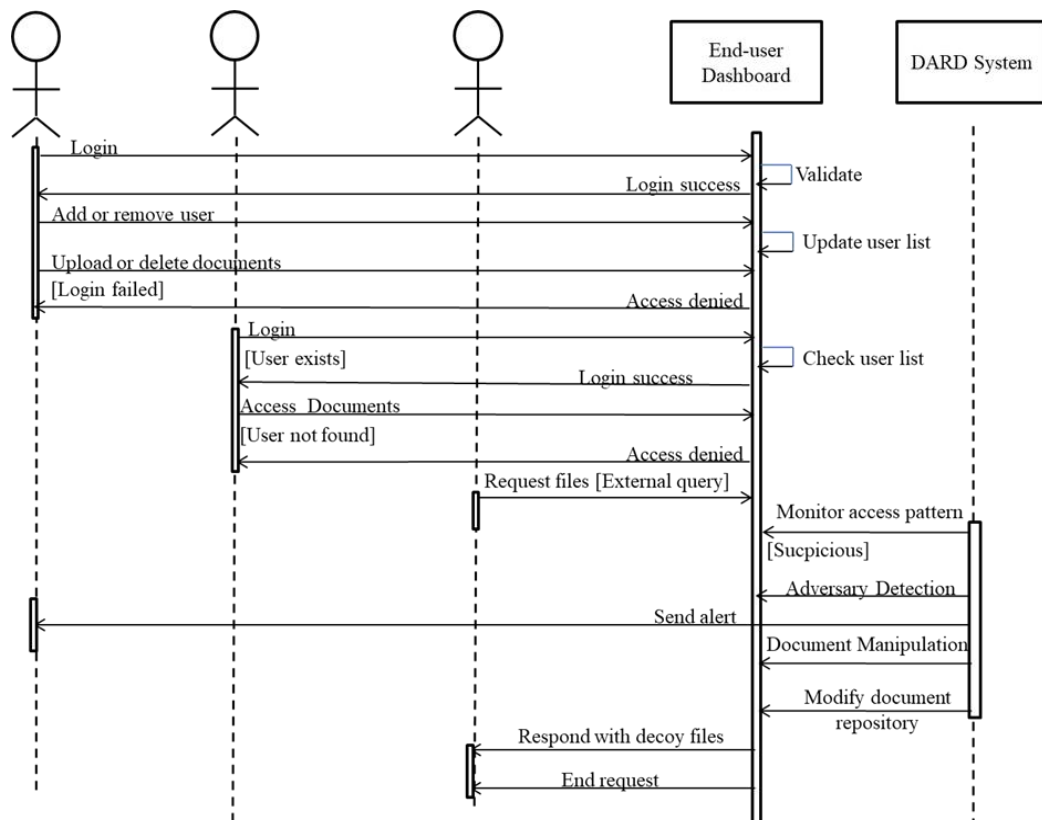


Figure 1.5 Sequence Diagram

The Figure 1.5 depicts as the process initiates when a User or Adversary attempts to authenticate themselves by submitting their login credentials. These credentials are verified by the Admin component, which controls authentication and access permissions. Once authenticated, the user navigates to the End user Dashboard , from which they can request access to documents. The dashboard processes this request and communicates it to the backend system, prompting the DARD System to analyze the nature of the request in real time.

If the behavior of the requester aligns with patterns associated with adversarial activity, the DARD System powered by a Variational Autoencoder model trained on a synthetic dataset detects the threat. Rather than serving the genuine document, the system triggers document manipulation techniques to generate a convincing decoy file. This decoy is then returned to the adversary via the End user Dashboard as visualized in App 1.13, misleading them and safeguarding the actual intellectual property. Meanwhile, an alert is generated for the Admin, enabling appropriate action in response to the detected threat.

1.4.5.5 Activity Diagram

An activity diagram that represents the flow of activities or operations in a system. It describes the dynamic aspects of a system by showing the sequence of actions and the flow of control from one activity to another. Activity diagrams are useful for visualizing the logic of complex processes. The activity diagram shown in Figure 1.6 illustrates the sequence of activities of the DARD system, mapping out various operations carried out by the Admin, User, and Adversary entities.

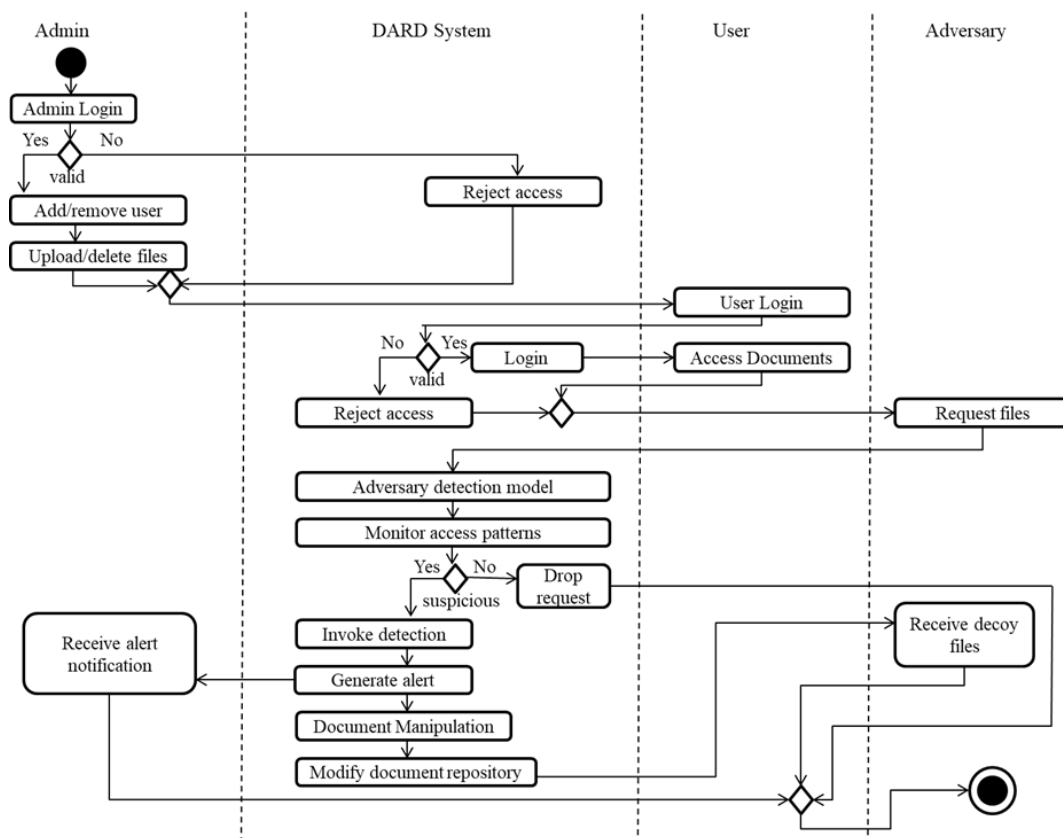


Figure 1.6 Activity Diagram

The activity begins with the Admin login, followed by optional admin operations such as adding/removing users and uploading/deleting files to the system. valid users are authenticated, and their requests to access or retrieve files are routed through the adversary detection model. This model, which uses a Variational Autoencoder trained on synthetic datasets, continuously monitors access patterns to detect anomalies.

When suspicious activity is identified, the system invokes detection logic, drops the adversary request, and triggers alert notifications for the admin. Subsequently, the DARD system engages in document manipulation techniques such as topic shifting, redaction, or term substitution. This manipulation ensures that adversaries receive decoy files instead of original documents, these activities are described in App 1.1 to App 1.19.

1.4.5.6 Deployment Diagram

The deployment diagram shown in Figure 1.7 illustrates the physical architecture of the DARD system, showcasing how different software components are deployed across various nodes. Deployment diagrams are essential in modeling the physical aspects of a system, such as how software artifacts run on hardware components.

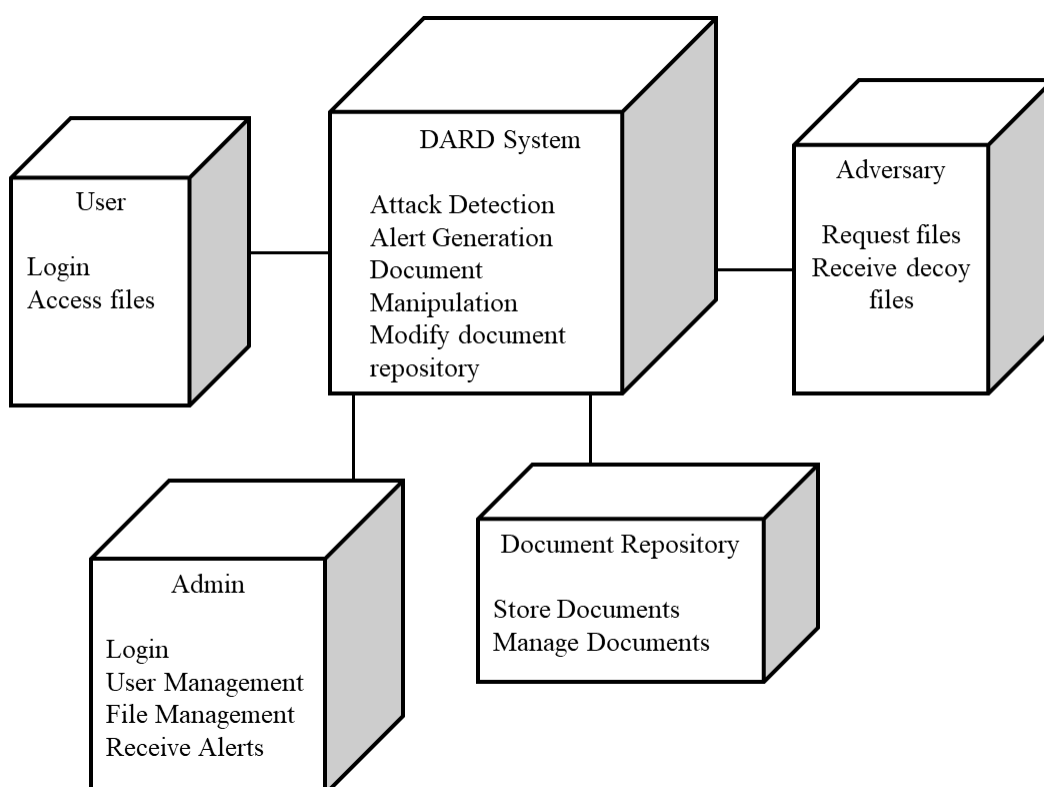


Figure 1.7 Deployment Diagram

In the diagram, the DARD System node represents the core intelligence module of the architecture. It houses functionalities such as Attack Detection, Alert Generation, Document Manipulation, and Repository Modification. This module is deployed on a secure server that continuously monitors the activity logs, identifies adversarial behaviors using machine learning models, and dynamically generates decoy

documents. This node plays the central role in orchestrating all protective mechanisms and interacting with other components like the document repository and user interfaces.

Other nodes include the Admin, User, and Adversary components, each having their respective deployment responsibilities. The Admin node supports functions such as Login, User Management, File Management, and Receiving Alerts. It is typically hosted on a secure administrative machine or a dedicated backend panel. The User node, on the other hand, is deployed on the user's device, enabling Login and Access Files operations. The Adversary node simulates a malicious actor requesting files, which upon detection receives Decoy Files instead of real ones. Meanwhile, the Document Repository is an independent node responsible for Storing and Managing Documents, acting as the central storage for sensitive files, whether original or manipulated. All these nodes collectively represent how the system is deployed in a real world scenario, ensuring security, manageability, and operational clarity for IP protection.

1.4.6 ER Diagram

An Entity Relationship diagram is a type of flowchart that shows how entities like people, objects, or concepts relate to each other within a system. It is used in database design to visually represent the structure of a database, including the entities, their attributes, and the relationships between them. The Entity Relationship diagram shown in Figure 1.8 depicts the structured database schema for the DARD system, which is designed to secure intellectual property through document access monitoring and decoy based response mechanisms.

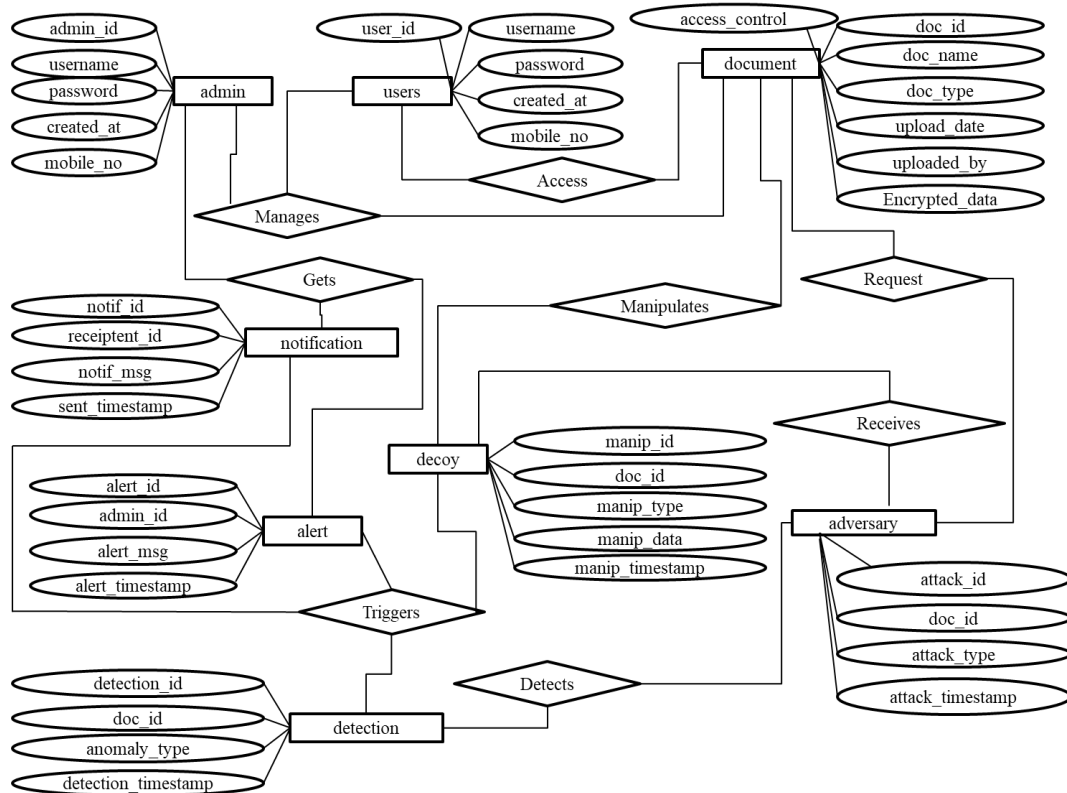


Figure 1.8 ER Diagram

The Admin entity includes fields such as admin_id, username, password, created_at, and mobile_no outlined in Table 1.2, it manages the Users who also have similar fields shown in Table 1.1, the Admin manage users and can upload or delete documents, stored in Documents entity with attributes outlined in Table 1.3, Users access these documents, while both users and admins interact with the Notification entity shown in Table 1.8 which includes notif_id, recipient_id, notif_msg, and sent_timestamp.

When abnormal access patterns are detected, the Detection entity shown in Table 1.5 logs the event with attributes like detection_id, doc_id, anomaly_type, and detection_timestamp. It triggers an Alert entity shown in Table 1.7 which is linked to the admin entity. Upon detection, the system initiates Decoy entity shown in Table 1.6 which stores manipulated

versions using `manip_id`, `doc_id`, `manip_type`, `manip_data`, and `manip_timestamp`. These manipulated or decoy files are then received by the Adversary entity shown in Table 1.4 which records `attack_id`, `doc_id`, `attack_type`, and `attack_timestamp`. Each relationship in the diagram ensures proper linkage between processes such as access control, threat detection, and decoy delivery forming the backbone of the DARD system's secure operations.

1.5 System Specification

This part defines the required hardware and software components needed for smooth functioning of the application.

1.5.1 Hardware Specification

Client Side

- Processor : Intel® Core™ i3 processor
- RAM : Minimum 4GB RAM
- Storage : At least 50mb of free space for installation

Developer Side

- Processor : Intel® Core™ i3 processor
- RAM : Minimum 4GB RAM
- Storage : 50 GB

1.5.2 Software Specification

Client Side

- Operating System : Windows 10/11.
- Browser : Chrome
- Server : Wampserver 2i

Developer Side

- Programming : Python 3.7.4(64-bit) or (32-bit)
- Operating System : Windows 10/11.
- Web Design : HTML, CSS, Bootstrap
- IDE : IDLE Python
- Web Framework : Flask 1.1.1
- Database : MySQL 5.
- Web Server : Wampserver 2i
- Packages : Pandas, NumPy, Scikit Learn, NLTK

PROJECT DESCRIPTION

CHAPTER II

2. PROJECT DESCRIPTION

The system aims to develop a robust document access framework that protects sensitive intellectual property from automated cyberattacks using deception and anomaly detection techniques. By integrating a Variational Autoencoder for detecting abnormal user behavior and deploying decoy based manipulation strategies, the system intelligently disrupts adversarial document analysis.

2.1 System Description

This system is an intelligent security framework designed to safeguard sensitive intellectual property documents, particularly within the insurance sector[1][9]. The system addresses the increasing risk of both external cyber threats and internal misuse by incorporating a proactive defense mechanism that blends behavioral anomaly detection with deceptive content manipulation[6]. All critical documents such as insurance policies, claims, and financial data are stored in a secure IP repository that is tightly controlled and accessible only to authenticated users[7].

DARD uses Variational Autoencoders to analyze and track user behavior, identifying deviations in access patterns such as irregular login times or abnormal file request volumes[6][9]. Once suspicious activity is detected, the system automatically engages its Document Manipulation Module, which transforms the actual document content by shuffling sections, altering keywords, reducing information density, or modifying thematic structures[6]. These transformations are strategically implemented to disrupt machine learning techniques like document clustering and topic

modeling often employed by attackers[6][8]. To further obscure the document's real intent and structure, the system reapplies KMeans clustering and Latent Dirichlet Allocation to the manipulated content, effectively generating misleading document groupings and semantic themes[6][9].

In parallel, real time alerts are sent to administrators via a dedicated Alert and Notification Module, enabling immediate action against the detected threat[5][4]. Through this combination of secure storage, machine learning based anomaly detection, deception tactics, and responsive alerting, the DARD system offers a modern, layered defense against IP theft ensuring data integrity, confidentiality, and trust in digital insurance environments[2][4][6].

This method confuses malicious tools by altering document content, modifying topic distributions, and injecting convincing decoys all without disrupting legitimate access[6]. By blending deception with intelligent detection and manipulation, the DARD system provides a smarter, more adaptive way to protect intellectual property in the face of evolving cyber threats[3][5].

2.2 Module Description

This part provides a brief explanation of each module included in the system. It highlights the functionality and role of each module in achieving the overall system objectives. Each module operates independently yet contributes to the unified workflow of the system. Understanding each module individually helps in grasping how they work together to enable seamless detection of adversaries and misleading them by serving decoy.

2.2.1 End User Dashboard

2.2.2 Adversary Model

2.2.3 Adversary detection

2.2.4 Document Manipulation

2.2.5 Document Repository Modification

2.2.6 Alert Generator

2.1.1 End User Dashboard

This module allows two types of users to interact with the system that is shown in App 1.1 and also holds the document repository where all the documents are stored and managed securely. The end users who interact with the system are Admin and User.

2.1.1.1 Admin

The Admin facilitates user management, document uploading, and access control. Admins can add new documents, categorize them, monitor the system's health, track the progress of security measures, and manage user permissions as described in App 1.2 to App 1.4. Admin gets alerts upon abnormal access of the adversaries as visualized in App 1.18 and App 1.19. Only the users added by the admin can login and access the documents they want to secure the valuable documents.

2.1.1.2 User

The User module allows legitimate users to log in securely and access only the documents they are authorized to view. The system ensures that users can seamlessly retrieve and work with the documents as portrayed in App 1.15 to App 1.17 while maintaining stringent security protocols to prevent unauthorized access.

2.1.1.2 Insurance IP Repository

The Insurance IP Repository module is designed to securely store and manage all sensitive intellectual property documents related to the insurance industry. These documents may include customer policy information, claim records, financial data, medical reports, and more. The repository ensures that these documents are stored in a centralized, well organized manner, allowing easy access for legitimate users while protecting them from unauthorized access or theft.

2.2.2 Adversary Model

This module simulates adversarial attacks on the IP repository to test the robustness of the system as visualized in App 1.6. It uses advanced data mining techniques like document preprocessing, feature extraction, and clustering to simulate how an adversary would target the IP documents[8][9]. First, the documents are preprocessed to remove irrelevant information, such as numbers, punctuation, and stop words[6]. Then, relevant terms are extracted using the TF-IDF technique to represent the documents more effectively. The KMeans algorithm is employed to cluster the documents based on these features, and Latent Dirichlet Allocation is used for topic modeling, where the adversaries attempt to identify key themes and sensitive data within the documents[6][9].

2.2.2.1 Document Preprocessing

This involves removing irrelevant or non informative content such as numbers, punctuation marks, special characters, and common stop words such as "and", "the", "of" that do not contribute meaningfully to document understanding. By performing tokenization, lowercasing, and stemming or lemmatization, adversaries reduce noise and ensure the textual data is in a

simplified, structured format. This preprocessing stage is essential for enhancing the quality of features extracted in the next phase and for allowing algorithms like clustering or topic modeling to work efficiently.

2.2.2.2 Feature Extraction Using TFIDF

After cleaning the document content, adversaries proceed with feature extraction to convert textual data into numerical form. One commonly used technique in this stage is Term Frequency Inverse Document Frequency which helps in identifying words that are most relevant to individual documents while reducing the weight of commonly used terms across the entire corpus. TFIDF scores reflect how important a term is in a specific document relative to its frequency in the entire dataset. By transforming the documents into a high dimensional matrix of weighted term scores, adversaries can better understand document semantics, which aids in document classification, clustering, and topic extraction.

2.2.2.3 Clustering Using KMeans Algorithm

With the documents represented as TFIDF vectors, adversaries apply the KMeans clustering algorithm to group similar documents together. KMeans operates by partitioning the feature space into 'K' clusters, where each document is assigned to the cluster with the nearest mean. In the adversary's strategy, clustering helps in organizing large volumes of retrieved documents based on content similarity.

2.2.2.4 Topic Modeling Using Latent Dirichlet Allocation

Following the clustering phase, adversaries apply Latent Dirichlet Allocation to perform topic modeling on the grouped documents. LDA is a generative probabilistic model that identifies latent topics within a corpus by

analyzing word co occurrence patterns. It assumes that each document is a mixture of various topics and each topic is a distribution over words. By running LDA, adversaries aim to uncover hidden semantic structures and dominant themes present in the document set such as customer information, financial terms, or policy keywords in an insurance domain. This helps them infer the purpose and sensitivity of the documents without needing full contextual understanding.

2.2.3 Adversary Detection

The Adversary Detection module is responsible for monitoring user behavior and access patterns to identify any malicious activities. By leveraging Variational Autoencoders, this module can detect unusual access patterns as shown in App 1.5, such as abnormal login attempts, rapid or bulk document retrieval, and other suspicious behaviors that are typical of an adversary attempting to steal intellectual property. Once the system detects such anomalies, it flags the activity as potentially adversarial and triggers the protective measures implemented by the DARD system.

2.2.3.1 Dataset Generation and Preprocessing

Before training the model, the system must simulate a wide range of access patterns to represent both normal and potentially malicious user behavior. This is done using a custom dataset generation function called `generate_data()`. The dataset is extracted from a backend table named `ins_data` within a MySQL database. This table likely contains behavioral features like login time, session duration, document count accessed, time between requests, and IP variance. Using Python's CSV and MySQL libraries, the function exports this table into a local file named `data.csv`

located in the dataset/directory. The script ensures clean formatting, removes any empty records, and rewrites the dataset into a normalized, machine readable format.

This curated synthetic dataset becomes the foundational input for training the VAE model. It is structured with 5 distinct numerical features respectively login time, session duration, document count accessed, time between requests, and IP variance per data sample, and 100 synthetic samples are generated to simulate both normal and potentially suspicious behaviors.

These features are numerically encoded and normalized, making them suitable for input into a neural network model. This preparation is crucial to ensure the model can distinguish between legitimate and adversarial patterns.

2.2.3.2 VAE Model Architecture and Training

The core of the adversary detection engine is the VAE model, defined in the VAE class using PyTorch. It consists of three main parts: an encoder, a reparameterization layer, and a decoder.

Model Structure

The encoder network maps the 5 dimensional input down to a latent space with only 2 dimensions via two separate linear layers that calculate the mean and log variance of the latent variables. This compact representation captures the distribution of normal behaviors.

Reparameterization is used to sample from the learned distribution, allowing for gradient descent optimization. This is done via the `reparameterize()` method.

The latent vector is passed into the decoder network, which tries to reconstruct the original input from the compressed form. If the behavior is normal, the reconstruction should be accurate; if it is unusual, the model will struggle to replicate it.

Training Process

The model is trained on the synthetic dataset using Mean Squared Error for reconstruction loss and KL divergence as a regularization term to encourage a smooth latent space. An Adam optimizer is used with a learning rate of 0.001, and training proceeds for 50 epochs, ensuring the model captures a generalizable representation of the data. During training, the model prints the loss value in each epoch to track convergence. This combined loss ensures both accurate reconstructions and meaningful latent representations. Finally, the trained model is saved for deployment in the live detection system.

2.2.3.3 Anomaly Detection and Testing

After training, the VAE model is used to detect adversarial behavior in real time. Every time a user accesses a document, the system captures a snapshot of their behavior. The new data will pass through the trained VAE model. If the reconstructed output is very similar to the input, it indicates normal behavior. However, if the reconstruction loss is high, the system flags it as an anomaly. A threshold is established during training by analyzing the reconstruction losses of known normal samples. Any behavior with a reconstruction error above this threshold is considered adversarial.

Once flagged, the system responds by, Activating the Document Manipulation Module shown in App 1.9 to serve a decoy document.

Triggering real time alerts via SMS and email to notify administrators visualized in App 1.18 and App 1.19. This approach provides a proactive defense layer, stopping potential breaches even before the adversary completes their data exfiltration.

2.2.4 Document Manipulation

The Document Manipulation module plays a critical role in confusing adversaries by manipulating the IP document repository. When adversary behavior is detected, this module alters the documents to make it difficult for adversaries to extract meaningful information. The goal is to alter the structure and content of documents in a way that deceives as shown in App 1.9 machine learning algorithms while maintaining plausible readability, thereby misleading attackers and protecting intellectual property. For authenticated users and legitimate requests, the original document remains unchanged and accessible. The manipulation engine dynamically differentiates between genuine users and suspected adversaries, ensuring zero disruption to authorized workflows.

2.4.1 Manipulation Techniques

This module employs manipulation techniques that work individually or in combination to reduce the document's analytical value for attackers. These methods are especially effective against adversarial systems that rely on keyword frequency, structure, or semantic relationships.

2.4.1.1 Basic Shuffle

This technique involves randomly rearranging keywords and phrases within a document. Since many machine learning models including KMeans clustering and TFIDF based topic modeling rely heavily on term position

and contextual flow, shuffling disrupts this natural ordering as described in App 1.9. As a result, the semantic meaning becomes diluted or lost, making it harder for adversaries to interpret the document's intent.

2.4.1.2 Shuffle Increment

In this method, extra decoy keywords are inserted into the document. These keywords are carefully chosen to appear contextually plausible but are semantically unrelated to the actual topic as portrayed in App 1.10. This increases the noise to signal ratio, confusing topic modeling tools like Latent Dirichlet Allocation and making it difficult to extract accurate themes from the document. The attacker receives a bloated version of the file with numerous misleading cues.

2.4.1.3 Shuffle Reduction

This process removes certain key terms from the original document as shown in App 1.11. By eliminating specific words that are central to the document's subject matter, this manipulation technique erodes the accuracy of feature extraction and clustering algorithms. The attacker ends up with an incomplete dataset, leading to false inferences or failed data mining attempts.

2.4.1.4 Change Topic

The most deceptive method, topic change involves replacing original keywords with unrelated or misleading terms as described in App 1.12. While the document still appears structurally sound, its semantic core is misaligned. This makes topic detection models completely ineffective, directing adversaries to irrelevant clusters that contains the deceptive content generated to safeguard the valuable documents.

2.4.1.5 Redaction and Visual Masking

In addition to the above strategies, the module incorporates a visual redaction mechanism. This technique adds a redaction layer over the original text, either by blacking out sensitive words or replacing them with opaque boxes or placeholder characters. This is especially useful when adversaries attempt to visually scan or optically recognize the content.

2.2.5 Document Repository Modification

Upon the document manipulation, the system applies Reclustering and Retopic Modeling techniques to ensure that the adversary cannot generate correct clusters or topics from the modified content as shown in App 1.13. The KMeans algorithm is reapplied to the altered documents, producing misleading document groupings. Similarly, the LDA model is used to assign inaccurate topics to the manipulated clusters, confusing any attempts to analyze the true content of the documents. This module ensures that the adversary's automated classification methods fail and that they are misled by incorrect outputs, making it extremely difficult for them to access critical intellectual property.

2.2.5.1 Reclustering with KMeans Algorithm

Once the documents have been manipulated using techniques like keyword shuffling, topic replacement, or redaction, the system reapplies the KMeans clustering algorithm to the altered dataset. Normally, KMeans is used to group documents based on feature similarity, such as common terms or semantic patterns. However, in this module, the manipulated documents are purposefully fed into the algorithm to produce misleading clusters. These clusters do not reflect the original categories or themes of the

documents, but rather, reflect the intentionally distorted content. For example, insurance related documents that once belonged to categories like "Claims," "Policies," or "Payouts" might now appear to belong to entirely different or overlapping categories due to the inserted or removed keywords. This creates false clusters that confuse adversarial machine learning models attempting to infer relationships or document taxonomy. As a result, attackers receive meaningless or disorganized groupings, undermining their efforts to extract actionable insights.

2.2.5.2 Retopic Modeling Using Latent Dirichlet Allocation

After reclustering, the system proceeds to retopic modeling using the Latent Dirichlet Allocation algorithm[6][9]. In typical use cases, LDA identifies hidden topics within a corpus of text by analyzing patterns of word occurrence[8]. However, when applied to manipulated documents, the model derives topics that are aligned with the altered and misleading textual data[6]. This means that even if an adversary successfully performs LDA topic modeling on the accessed documents, the topics extracted will be semantically incoherent or irrelevant as described in App 1.14. A document that originally pertained to "Policy Renewal Guidelines" may now be associated with a fabricated topic like "Astronomy Concepts" or "Fashion Trends," depending on the injected or shuffled keywords[6]. This misguides the attacker's understanding of the documents' core meaning, rendering their analytical tools ineffective[5].

2.2.6 Alert Generator

The Alert Generator module is responsible for notifying administrators and security personnel about potential adversarial activities and other critical incidents within the system. Once suspicious behavior is

detected by the Adversary Detection Module, the Alert Generator immediately triggers alerts based on predefined criteria, such as high frequency document access. These alerts are generated in real time as visualized in App 1.5 and App 1.9 to 1.13, ensuring that the system's administrators can respond to potential threats swiftly.

Notification

The Notification Module works hand in hand with the Alert Generator to provide clear communication of security events and system status updates to the relevant stakeholders. Whenever a significant event occurs, such as an adversary detection, document manipulation, or system status change, the Notification Module ensures that the administrators are kept informed as shown in App 1.18 and App 1.19.

2.3 Software Description

This section outlines the various software tools and technologies used in the development of the system. It includes programming languages, frameworks, libraries, and platforms that support functionalities such as web development, database management and machine learning.

2.3.1 Python

Python is currently the most widely used multipurpose, high level programming language. Python allows programming in Object Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python is a general purpose interpreted, interactive, object oriented, and high level programming language.

Pandas

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high level building block for doing practical, real world data analysis in Python.

NumPy

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. NumPy is a general-purpose array-processing package. It provides a high performance multidimensional array object, and tools for working with these arrays.

Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

Scikit Learn

Scikit learn is a Python module for machine learning built on top of SciPy and is distributed under the 3 license. Scikit learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, kmeans. It is designed to interoperate with python numerical scientific libraries.

Natural Language Tool Kit

Natural Language Toolkit is a leading platform in Python for working with human language data. It provides tools and libraries for text processing tasks such as tokenization, stemming, lemmatization, part of speech tagging, parsing, and semantic reasoning. NLTK comes with a large collection of corpora and lexical resources like WordNet, which support a wide range of natural language processing functions. It is widely used in academic research, education, and NLP applications for preprocessing textual data and extracting linguistic features.

2.3.2 MySQL

MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open source and free software. It is supported by Oracle Company. MySQL database that provides for how to manage database and to manipulate data with the help of various SQL queries. These queries are: insert records, update records, delete records, select records, create tables, drop tables, etc. MySQL is currently the most popular database management system software used for managing the relational database.

2.3.3 Wampserver

WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP, and a MySQL database. Alongside, PhpMyAdmin allows you to easily manage your database. WampServer is a reliable web development software program that lets you create web apps with MySQL database and PHP Apache2. With an intuitive

interface, the application features numerous functionalities and makes it the preferred choice of developers from around the world. The software is free to use and doesn't require a payment or subscription. It is commonly used for testing and developing websites locally before deploying them to a live server.

2.3.4 Bootstrap 4

Bootstrap is a free and open source tool collection for creating responsive websites and web applications. It is the most popular framework for developing responsive, mobile first websites. It solves many problems which we had once, one of which is the cross browser compatibility issue. Nowadays, the websites are perfect for all the browsers and for all sizes of screens.

2.3.5 Flask

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web based calendar application or a commercial website.

Flask is often referred to as a micro framework. It aims to keep the core of an application simple yet extensible. Flask does not have built in abstraction layer for database handling, nor does it have formed a validation support. Instead, Flask supports the extensions to add such functionality to the application. Although Flask is rather young compared to most Python frameworks, it holds a great promise and has already gained popularity among Python web developers.

2.4 Features and Advantages

Features

- **Secure IP Repository:** A centralized and access controlled storage system for sensitive documents such as insurance claims, policies, and financial records.
- **User Role Management:** Distinct roles for Admin and Users, where Admins handle user registration and document management, and Users can access only authorized documents.
- **Anomaly Based Adversary Detection:** Uses a Variational Autoencoder model trained on behavioral patterns to detect abnormal user activity that may indicate adversarial access.
- **Document Manipulation Engine:** Applies manipulation techniques like keyword shuffling, decoy insertion, content reduction, and topic alteration to mislead attackers.
- **Dynamic Redaction with Overlay:** Sensitive information is visually masked using redaction and overlay techniques before delivering content to suspected adversaries.
- **Reclustering and Retopic Modeling:** Reapplies KMeans and LDA on manipulated documents to confuse classification and topic detection algorithms used by attackers.
- **Real Time Alerts and Notification:** Automatically informs administrators via SMS and email when suspicious activity is detected, enabling quick response.
- **Decoy Based Document Delivery:** Adversaries are silently served manipulated or decoy documents, avoiding alerting them while disrupting their data theft attempts.

Advantages

- **Proactive Threat Defense:** Rather than passively monitoring or blocking access, the system actively confuses and disrupts attackers using deception strategies.
- **Resistance to Automated Attacks:** By targeting the core tools used in machine learning-based document analysis, such as clustering and topic modeling, the system breaks down adversarial automation.
- **Low False Positives for Legitimate Users:** The manipulation and defense mechanisms are selectively triggered based on behavior analysis, ensuring regular users access accurate, unaltered documents.
- **Enhanced IP Protection:** Sensitive information remains safe even after potential unauthorized access, thanks to intelligent document manipulation.
- **Scalable and Modular Design:** The system can be adapted to different domains and extended with more advanced detection or manipulation methods.
- **Supports Regulatory Compliance:** Ensures sensitive data is not leaked or exposed, supporting compliance with industry data protection standards.
- **Fast Incident Response:** With integrated alerting mechanisms, administrators are instantly notified, reducing response time and impact.
- **Minimal Disruption to Operations:** Legitimate users continue to interact with the system as usual, with no knowledge of the deception layers working in the background.

TESTING METHODOLOGIES

CHAPTER III

3. TESTING METHODOLOGIES

Software testing in the DARD project ensures the system functions as intended, meets user requirements, and is free of vulnerabilities that could compromise the security of intellectual property. The testing process will encompass multiple stages, including unit, integration, system, performance, security, and user acceptance testing.

3.1 Types of Testing

To ensure comprehensive quality assurance, different testing types were carried out throughout the development process.

3.1.1 Unit Testing

Focuses on testing individual components of the system, such as the Adversary Detection and Document Manipulation modules, to ensure they work as expected in isolation.

Test Case ID: TC001

- **Input:** User tries to access more documents than allowed in a short time (flooding behavior).
- **Expected Result:** VAE model detects unusual behavior and flags it as potentially adversarial.
- **Actual Result:** VAE detected the anomaly and flagged it.
- **Status:** Pass

Test Case ID: TC002

- **Input:** Perform document manipulation shuffle or change topic.
- **Expected Result:** Document integrity should be maintained, and the manipulations should be reversible.

- **Actual Result:** Data integrity was maintained, and manipulation was correctly applied.
- **Status:** Pass

3.1.2 Integration Testing

Ensures that the different system components, such as the Document Repository and Adversary Detection modules, interact seamlessly without errors.

Test Case ID: TC003

- **Input:** Trigger adversary detection with abnormal login attempts or rapid document retrieval.
- **Expected Result:** An alert should be generated to notify the admin of suspicious activity.
- **Actual Result:** Alert successfully generated and notified the admin.
- **Status:** Pass

3.1.3 Performance Testing

Assesses the system's performance under load, including handling large volumes of document access and manipulation without degradation.

Test Case ID: TC004

- **Input:** Trigger adversary detection, manipulate document content using the Basic Shuffle technique.
- **Expected Result:** Document content should be shuffled, making it difficult for adversaries to extract meaningful information.
- **Actual Result:** Document was successfully shuffled and analysis of the content showed confusion in clustering.
- **Status:** Pass

Test Case ID: TC005

- **Input:** Apply KMeans clustering again after document manipulation.
- **Expected Result:** New clusters should be created that mislead the adversary, and incorrect topic assignments should be made.
- **Actual Result:** New misleading clusters were generated, and incorrect topics were assigned to documents.
- **Status:** Pass

3.1.4 Security Testing

Focuses on identifying vulnerabilities and ensuring the system is secure against potential attacks, especially concerning unauthorized access to sensitive documents.

Test Case ID: TC006

- **Input:** A non admin user attempts to access a restricted document.
- **Expected Result:** The system should prevent access and displays an error message.
- **Actual Result:** Access was denied, and error message displayed.
- **Status:** Pass

Test Case ID: TC007

- **Input:** Admin receives an alert for detected adversarial behavior.
- **Expected Result:** Admin receives an email/SMS notification with details of the alert.
- **Actual Result:** Email/SMS notification delivered successfully to the admin.
- **Status:** Pass

3.1.5 Validation Testing

Verifies that the DARD system meets all functional requirements and behaves as intended under various input scenarios. It ensures the system

correctly handles valid, invalid, and boundary inputs related to user actions, document processing, and access control.

Test Case ID: TC008

- **Input:** Username: admin, Password: correct_password
- **Expected Result:** Admin should be logged in successfully and redirected to the Admin Dashboard.
- **Actual Result:** Admin logged in and redirected to Admin Dashboard.
- **Status:** Pass

Test Case ID: TC009

- **Input:** Upload a valid document.
- **Expected Result:** Document should be uploaded successfully to the IP repository and visible in the system.
- **Actual Result:** Document uploaded and visible in the repository.
- **Status:** Pass

Test Case ID: TC010

- **Input:** Username: user, Password: correct_password
- **Expected Result:** User should be logged in and redirected to the user interface where they can access authorized documents.
- **Actual Result:** User logged in and redirected to the authorized documents page.
- **Status:** Pass

Test Case ID: TC011

- **Input:** Username: user1, Password: wrong_password
- **Expected Result:** Login should fail with an error message.
- **Actual Result:** Login failed, error displayed.
- **Status:** Pass

Test Case ID: TC012

- **Input:** Try uploading a .exe file.
- **Expected Result:** System should reject the file and show a supported file type error.
- **Actual Result:** File rejected, correct error message shown.
- **Status:** Pass

Test Case ID: TC013

- **Input:** Admin enters valid user details.
- **Expected Result:** New user is successfully created.
- **Actual Result:** User creation successful.
- **Status:** Pass

Test Case ID: TC014

- **Input:** Logged in user requests an authorized document.
- **Expected Result:** Document retrieved and displayed correctly.
- **Actual Result:** Document successfully retrieved.
- **Status:** Pass

Test Case ID: TC015

- **Input:** Submit login form with empty username and password.
- **Expected Result:** System should prevent submission.
- **Actual Result:** Error shown, form not submitted.
- **Status:** Pass

Test Case ID: TC016

- **Input:** Upload a .pdf document close to the maximum allowed size.
- **Expected Result:** Document should be uploaded successfully.
- **Actual Result:** Upload succeeded within limits.
- **Status:** Pass

SYSTEM IMPLEMENTATION

CHAPTER IV

4. SYSTEM IMPLEMENTATION

System implementation is the process of integrating all designed components and modules of the proposed DARD system into a fully functioning and secure environment. This phase ensures that each module such as user management, adversary detection, document manipulation, and alert generation works cohesively within the system framework. Implementation involves setting up the software environment, training and deploying machine learning models, configuring the backend and frontend, and linking the database with secure access controls.

4.1 Introduction

This system is designed to detect and deceive adversaries attempting to extract confidential data from a secured IP repository by integrating behavior analysis, document deception, and real time monitoring[6][9]. The core of the system lies in its modular architecture, where each module functions independently yet contributes collectively to safeguarding document access[5]. In addition, the system applies reclustering and retopic modeling to further degrade the effectiveness of attacks like document classification and topic modeling, typically performed using KMeans and LDA[6][8]. Real time alerts are sent to admins via SMS, email, and the dashboard whenever adversarial behavior is flagged[4]. The system has been implemented using Python Flask for the backend, MySQL for database management, and HTML/CSS/JavaScript for the user interface. Encryption and role based access control are integrated throughout to ensure data confidentiality and access integrity[2][4].

4.2 Pre Implementation

Before deploying the DARD system, several essential steps were completed during the pre-implementation phase to ensure the system's effectiveness, usability, and reliability in a real world environment.

Dataset Creation and Preprocessing

A synthetic dataset simulating login and access behavior was generated using torch to train the Variational Autoencoder. The dataset included 100 samples with 5 behavioral features such as access frequency, login time, and user ID correlation. Data was normalized and cleaned to remove inconsistencies before training.

Model Training and Evaluation

The VAE model was trained using PyTorch with a custom architecture to learn behavioral anomalies. The model was evaluated using loss convergence and reconstruction error metrics. Once trained, it was exported and saved for integration within the adversary detection pipeline.

Module Planning and Design

Each module was carefully planned from document handling by the admin to adversarial manipulation. Manipulation rules such as Basic Shuffle, Shuffle Increment, Shuffle Reduction, and Topic Change were defined in alignment with NLP processing methods. A method called `redact_text_overlay()` was also implemented to mask keywords visually from suspected adversaries.

System Architecture and Environment Setup

The project environment was prepared using Flask, MySQL, and frontend frameworks. Data flow diagrams, component diagrams, and ER

diagrams were finalized. User interface prototypes for Admin and User roles were created. Python libraries like NLTK and Seaborn were configured for preprocessing and visualization, respectively.

4.3 Post Implementation

After successful integration and testing of modules, the DARD system was deployed and evaluated in the post-implementation phase. This phase involved the following steps,

Model Deployment and Integration

The trained VAE model was integrated into the Flask server and connected to the adversary behavior monitoring pipeline. Upon detecting anomalies, the system automatically redirected adversaries to manipulated document outputs as shown in App 1.14.

Document Manipulation and Defense Activation

The document manipulation logic was triggered in real time upon suspicious activity. This included injecting decoy keywords, redacting sensitive information using overlays, and altering document topics as described in App 1.9 to App 1.12. KMeans and LDA were reapplied to ensure the manipulated documents created misleading clusters and topics.

User and Admin Role Verification

Role based access control was strictly enforced. Only admins could upload, delete, or categorize documents as illustrated in App 1.4. Users approved by the admin could access documents shown in App 1.17, while others were denied and flagged. Admins received real time alerts as visualized in App 1.18 and App.1.19 through email, SMS notifications.

Testing and Optimization

Unit testing and user acceptance testing were conducted to verify model behavior, access flows, and system stability. Performance optimizations were applied to ensure minimal latency in document delivery and manipulation response time.

Monitoring and Updates

Logs from user interactions were periodically reviewed to fine tune detection sensitivity. Admin dashboards were updated to display visual statistics using Seaborn based charts and analytics. Future enhancements, including deeper NLP models or integration with external alerting systems, were proposed.

This post deployment process ensured that the DARD system not only functioned reliably in detecting adversaries but also delivered intelligent responses without disrupting the experience of legitimate users.

RESULT AND CONCLUSION

CHAPTER V

5. RESULT AND CONCLUSION

The DARD system effectively secured sensitive insurance documents by detecting adversarial behavior using a Variational Autoencoder model. When suspicious activity was identified, the system applied document manipulation techniques like keyword shuffling and topic changes to confuse attackers. Reclustering and retopic modeling further misled adversaries, while real time alerts notified admins instantly. The system maintained document integrity and access for valid users, proving to be a reliable defense against automated IP theft.

5.1 Conclusion

In conclusion, the project represents a significant advancement in the protection of intellectual property from automated cyberattacks. By incorporating innovative decoy based techniques, the system ensures that sensitive IP documents remain secure and inaccessible to unauthorized adversaries. The use of Variational Autoencoders for adversary detection, combined with document manipulation methods like shuffle techniques and topic alteration, creates an effective defense against sophisticated threats.

This multi layered approach ensures that adversaries are consistently misled, preventing them from successfully extracting valuable information. Furthermore, the real time monitoring and alert system enhances responsiveness, enabling swift action to protect data integrity. The system's ability to dynamically adjust document content and deceive automated clustering algorithms ensures that IP theft is thwarted at multiple levels. Moving forward, DARD can be refined and expanded with new AI driven

techniques to further enhance its effectiveness. The success of this project lays the groundwork for the next generation of IP protection systems, offering a proactive, automated solution to safeguard critical digital assets. This marks a crucial step towards ensuring robust security in an increasingly digital and data driven world.

5.2 Future Enhancement

The future scope of the project is expansive, with potential for continual improvement and adaptation to evolving technological landscapes.

- **Introduce Blockchain:** Use blockchain to create a tamper proof log of all access and changes to IP documents, ensuring better transparency and traceability.
- **Introduce Multi Factor Authentication:** Introduce multi factor authentication to provide an additional layer of security for accessing the IP repository, reducing the risk of unauthorized access.
- **Cloud Integration:** Enable integration with various platforms to allow secure access and management of documents across different devices and environments.

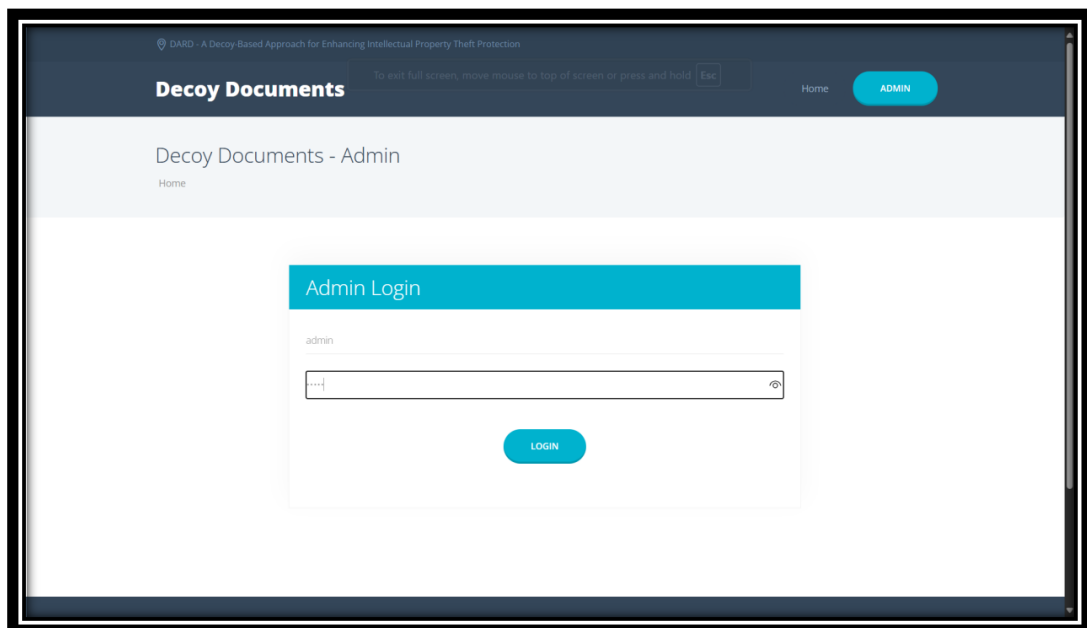
APPENDIX

APPENDIX

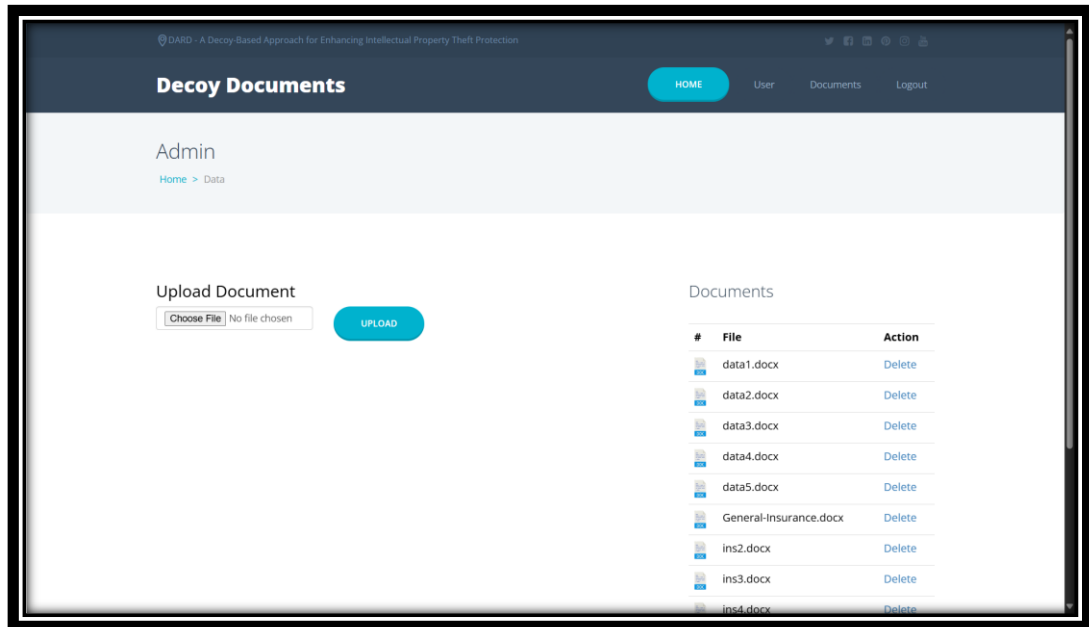
SCREEN SHOTS



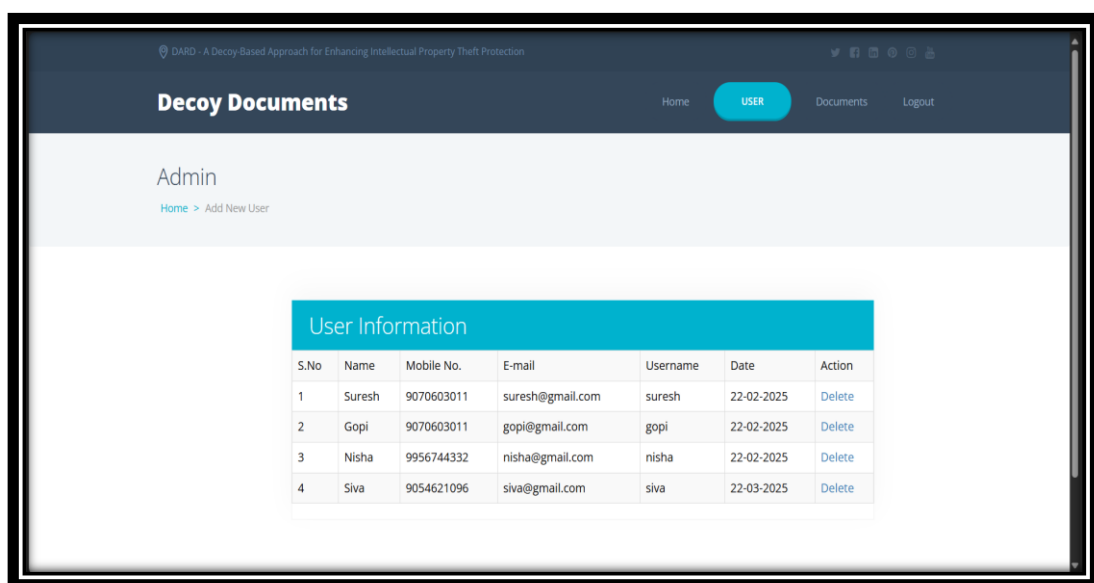
App 1.1 Home page



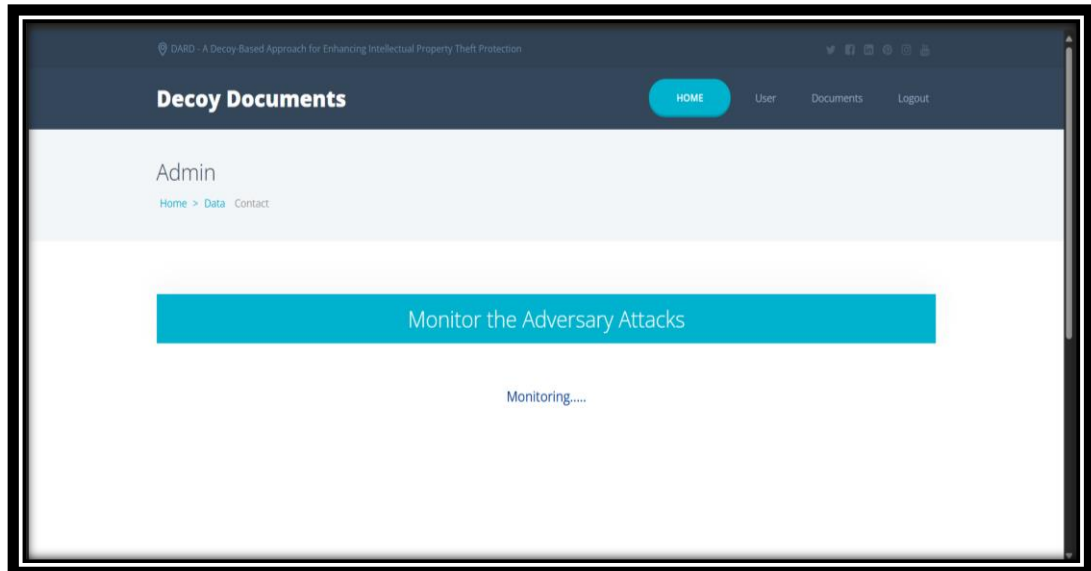
App 1.2 Admin Login



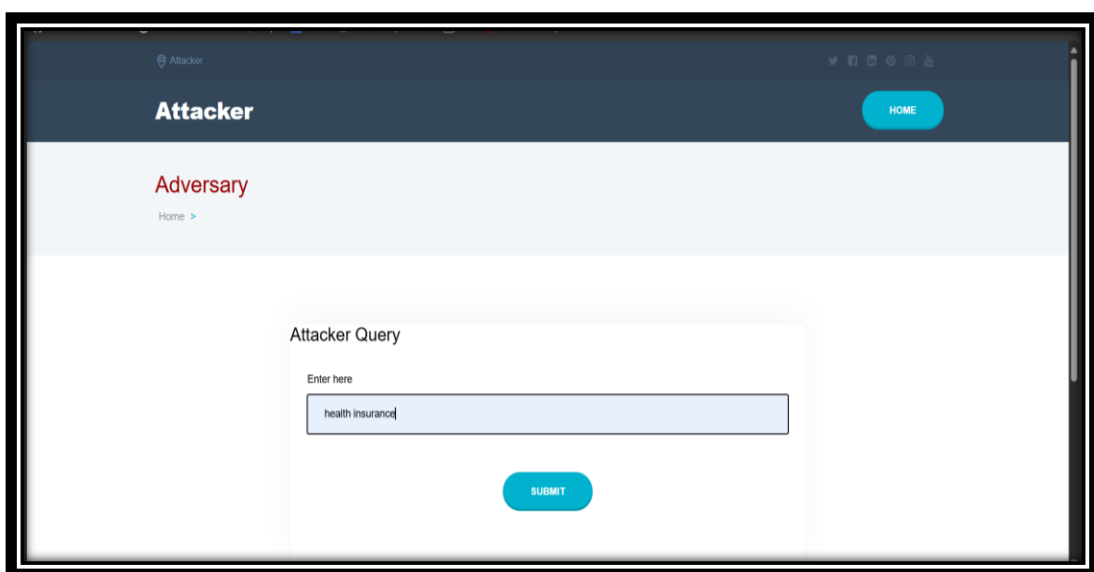
App 1.3 Document Management



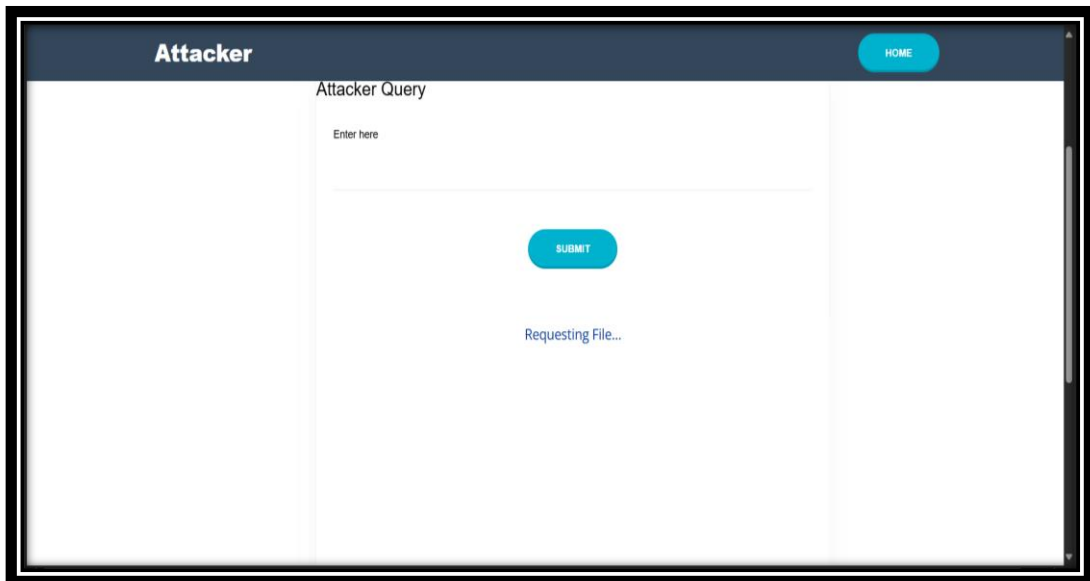
App 1.4 User Management



App 1.5 Adversary Monitoring

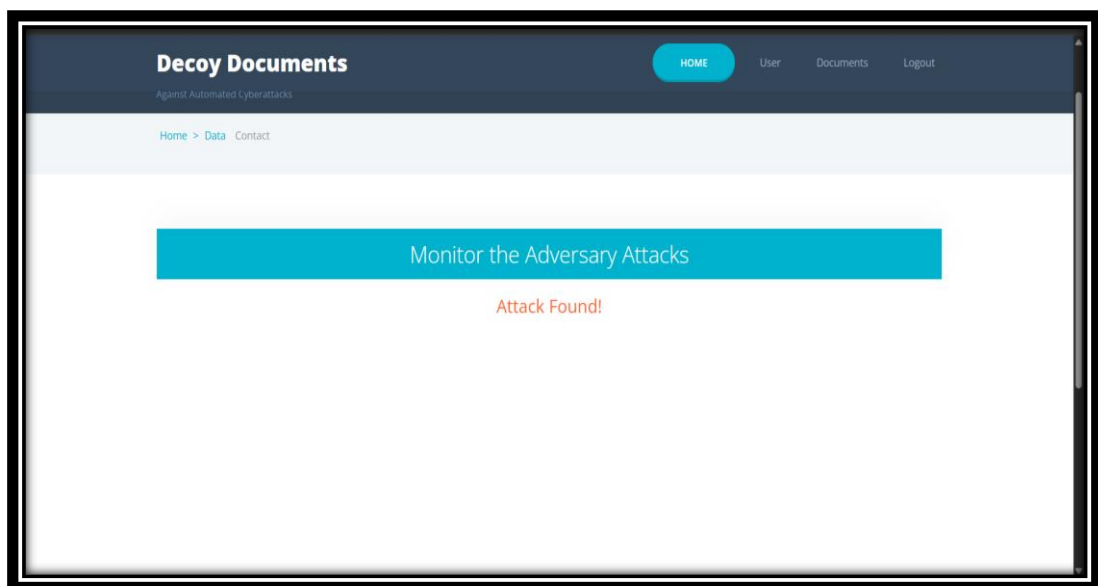


App 1.6 Attacker Query page



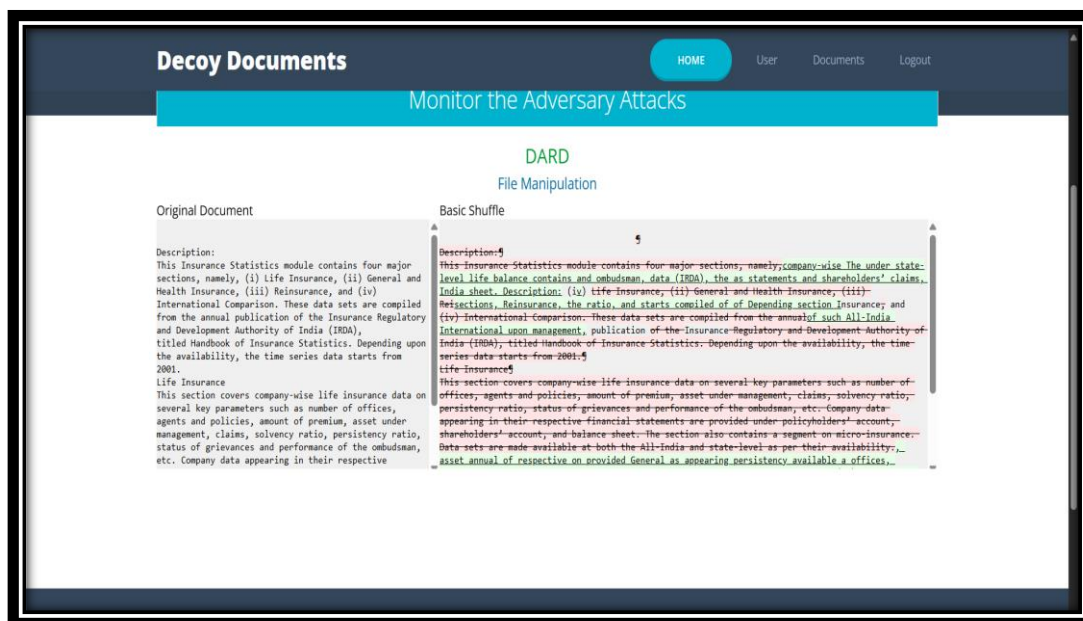
The screenshot shows a web application titled "Attacker". At the top right is a "HOME" button. The main content area is titled "Attacker Query" and contains a text input field with the placeholder "Enter here". Below the input field is a blue "SUBMIT" button. Underneath the button, the text "Requesting File..." is displayed.

App 1.7 Adversary File Request

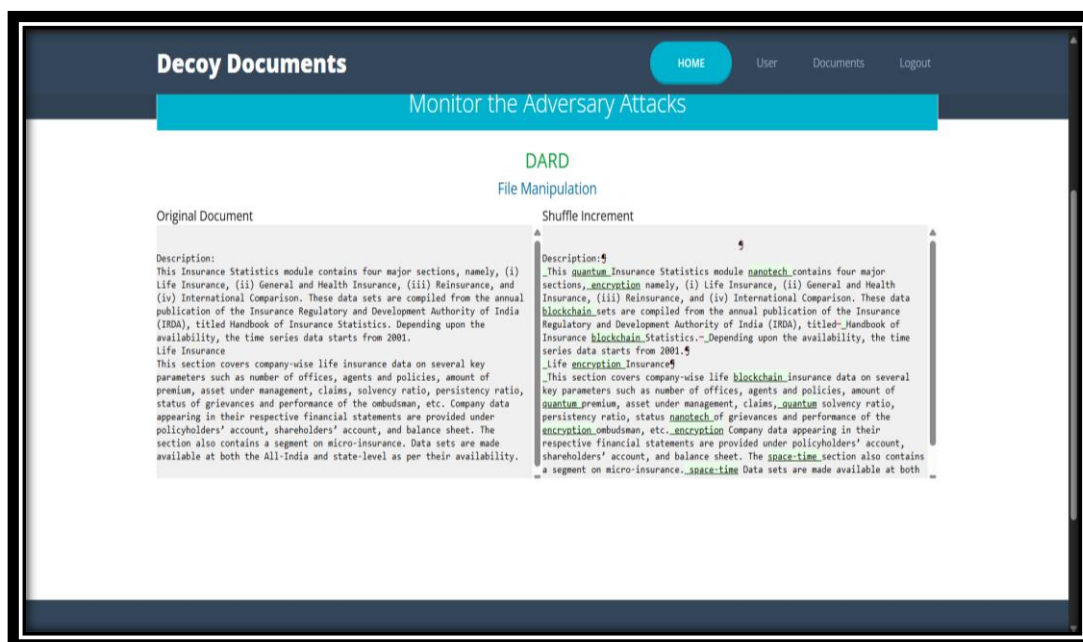


The screenshot shows a web application titled "Decoy Documents" with the subtitle "Against Automated Cyberattacks". The top navigation bar includes a "HOME" button and links for "User", "Documents", and "Logout". Below the navigation bar is a breadcrumb trail: "Home > Data > Contact". The main content area features a large blue button labeled "Monitor the Adversary Attacks". Below this button, the text "Attack Found!" is displayed in red.

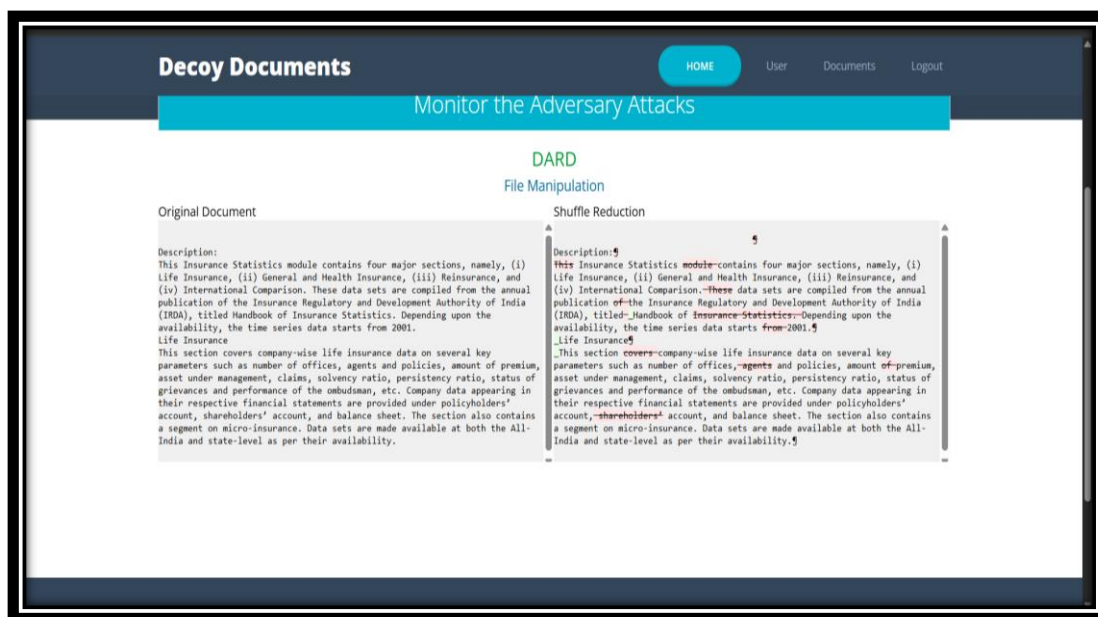
App 1.8 Adversary alert



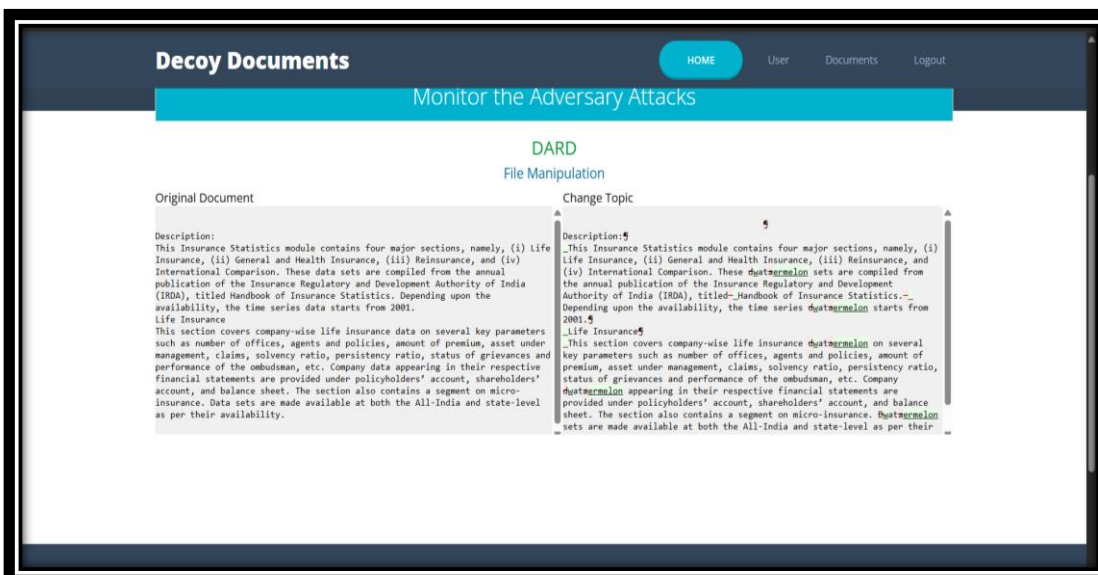
App 1.9 Basic Shuffle



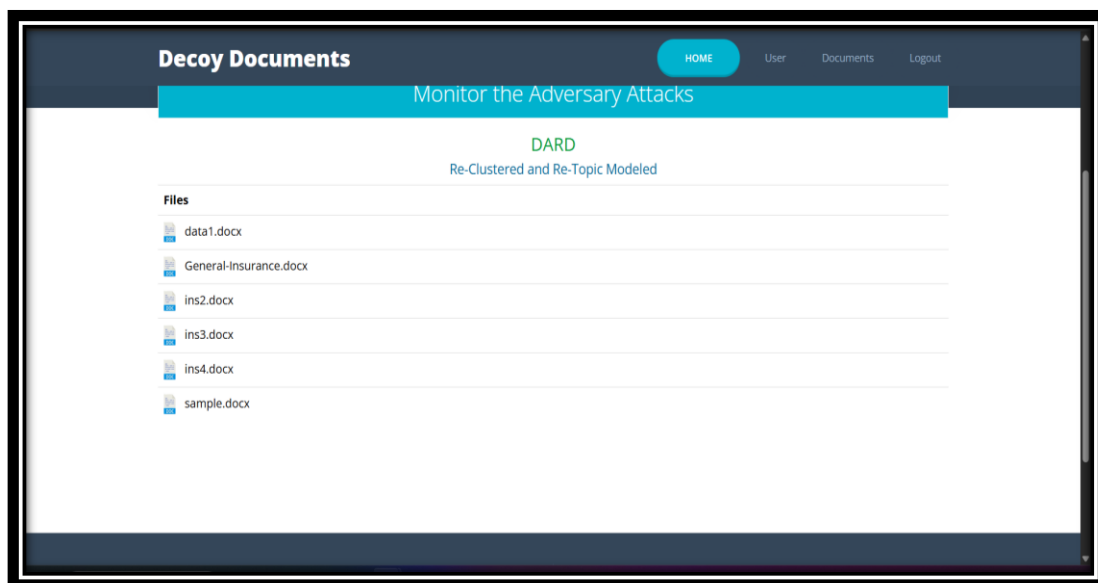
App 1.10 Shuffle Increment



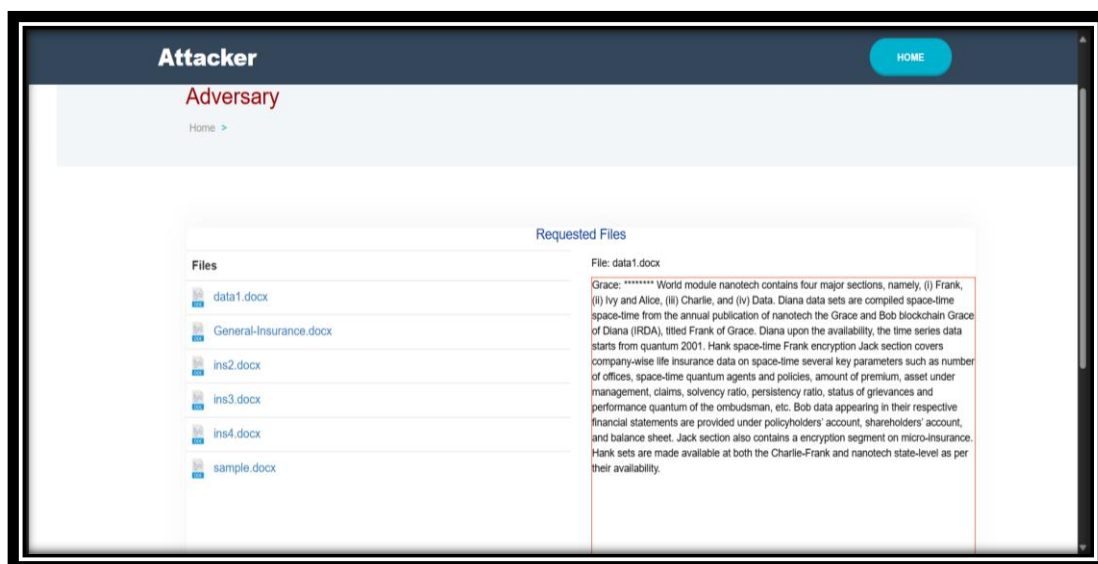
App 1.11 Shuffle Reduction



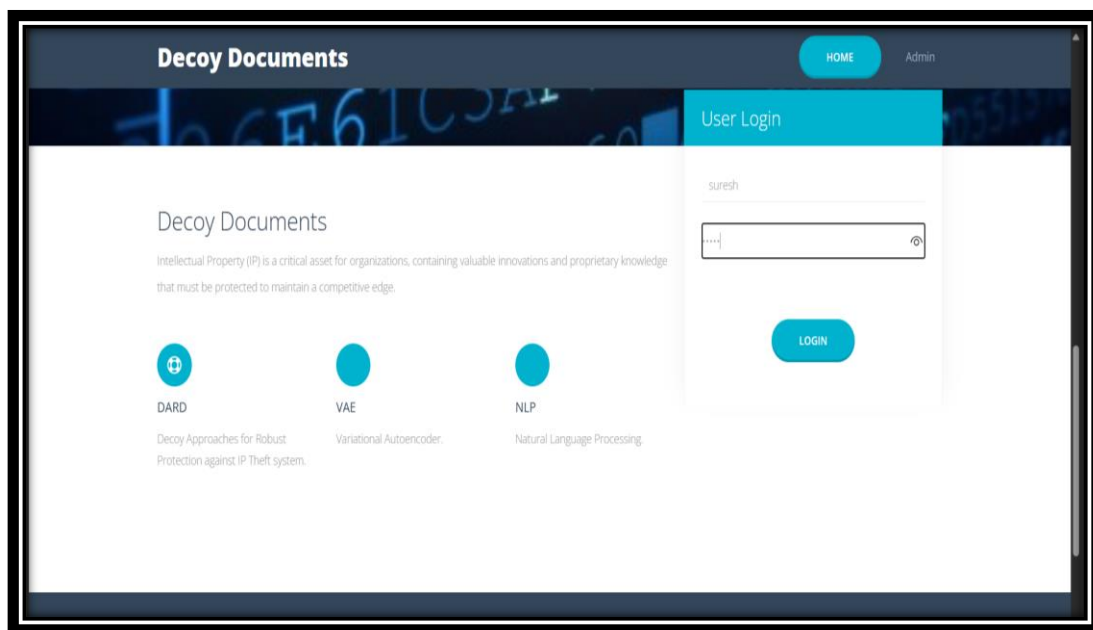
App 1.12 Change Topic



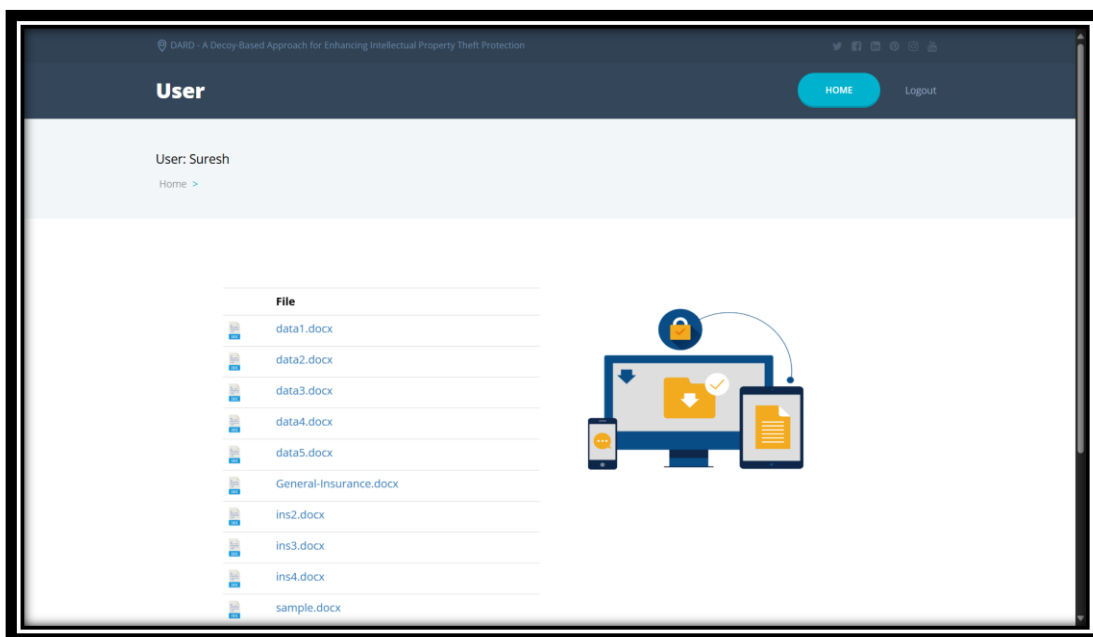
App 1.13 Decoy Serving



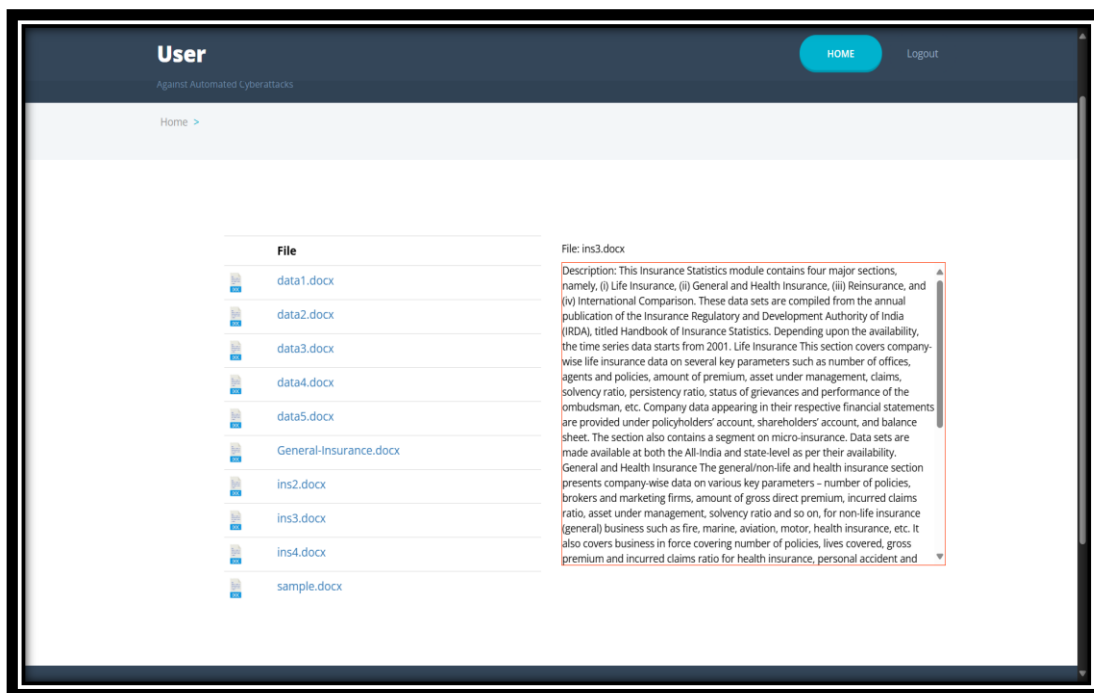
App 1.14 Decoy Files Preview



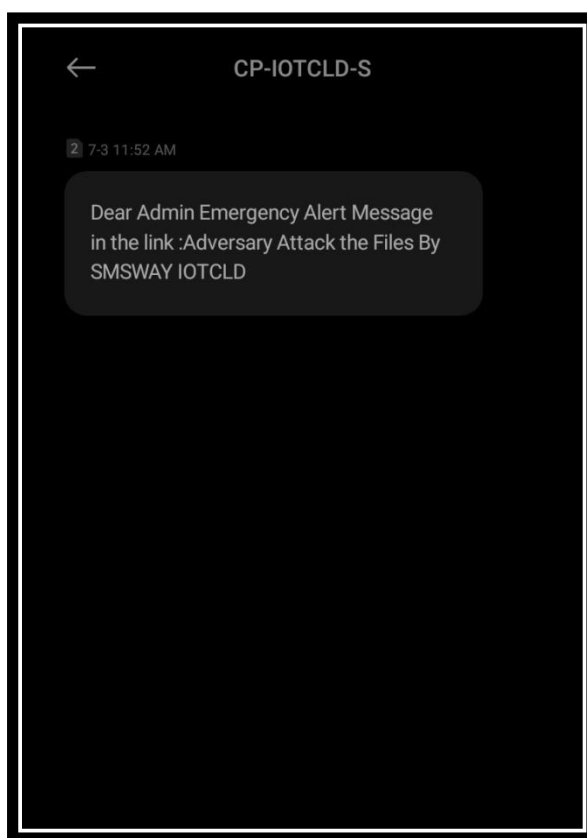
App 1.15 User login



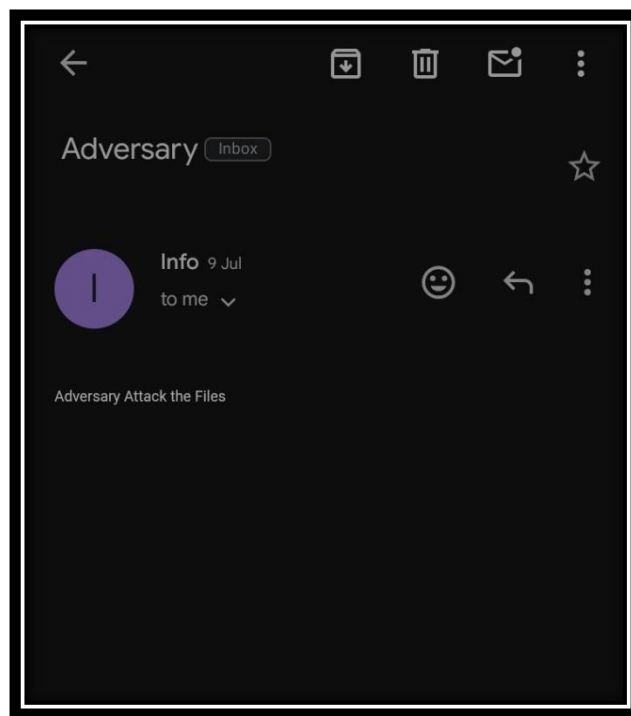
App 1.16 User Home page



App 1.17 User File access



App 1.18 SMS Notification



App 1.19 Email Notification

REFERENCES

REFERENCES

- [1] Andi Fariana and Syauqi Jinan, "The urgency of intellectual property rights in the digital era from the perspective of Sharia economic law in Indonesia", *Int. J. Res. Bus. Soc. Sci.* (2147-4478), vol. 12, no. 8, pp. 552-556, 2023.
- [2] P. Kumar, "Intellectual Property Rights (IPR): Nurturing Creativity Fostering Innovation", vol. 02, no. 02, pp. 32-38, 2024.
- [3] F. Indra and F. Santiago, "Intellectual Property Rights in Legal Perspective in Indonesia", *Proc. First Multidiscip. Int. Conf.*, 2022.
- [4] X. Sun, X. Zhou, Q. Wang, P. Tang, E. L. C. Law and S. Cobb, "Understanding attitudes towards intellectual property from the perspective of design professionals", *Electron. Commer. Res.*, vol. 21, no. 2, pp. 521-543, 2021.
- [5] C. Novelli, F. Casolari, P. Hacker, G. Spedicato and L. Floridi, "Generative AI in EU law: liability privacy intellectual property and cybersecurity", 2024.
- [6] T. Chakraborty, S. Jajodia, J. Katz, A. Picariello, G. Sperli and V. Subrahmanian, "A fake online repository generation engine for cyber deception", *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 2, pp. 518-533, Mar./Apr. 2021.
- [7] "Intellectual Property: What is Intellectual Property? World Intellectual Property Organization", 2020, [online] Available: <https://www.wipo.int/about-ip/en/>.
- [8] S. M. H. Bamakan, N. Nezhadsistani, O. Bodaghi and Q. Qu, "Patents and intellectual property assets as non-fungible tokens; key technologies and challenges", *Scientific Reports*, vol. 12, no. 1, pp. 2178, 2022.
- [9] R. Guo, "Research on the protection of enterprise digital intellectual property rights", *Science of Law Journal*, vol. 3, no. 2, pp. 153-159, 2024.