THE UNIVERSITY OF DODOMA

COLLEGE OF INFORMATICS AND VIRTUAL EDUCATION

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FINAL YEAR PROJECT PROGRESS REPORT

ACADEMIC YEAR: **2024/2025**

TITLE: STEGANOGRAPHY DETECTION OF PNG IMAGES HIDDEN WITHIN PDF DOCUMENTS

GROUP MEMBERS

STUDENT'S NAME	REGISTRATION NUMBER	PROGAMME
1. ISSA PAMUI	T21-03-10935	BSc-CSDFE
2. ABDALLAH MWIRU	T21-03-04495	BSc-CSDFE
3. IDDY MANUMBU	T21-03-16964	BSc-CSDFE

NAME OF SUPERVISOR **SIGNATURE**

Mr. Salim Diwani .....

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Project Overview

Steganography Detection of PNG Files Hidden in PDF Files is a web-based system designed to identify and analyze hidden PNG images embedded within PDF documents. This system leverages advanced detection algorithms to uncover concealed data, which is a critical aspect of cybersecurity and digital forensics. The system aims to provide an efficient and user-friendly platform for detecting hidden files, which can be used by forensic analysts, cybersecurity professionals, and other stakeholders in the digital security domain.

1.2 Problem Statement

The most common challenge for many people in Tanzania is the increasing use of digital steganography to conceal malicious content or sensitive information within seemingly innocuous files, such as PDFs. This poses a significant threat to information security, as these hidden files can bypass traditional security measures, leading to data breaches, unauthorized information dissemination, or cyber-attacks. Currently, there is a lack of accessible and efficient tools specifically designed to detect hidden PNG files in PDF documents, making it difficult for individuals and organizations to protect their digital assets effectively.

1.3 Objectives

1.3.1 Main Objective

The primary objective of this project is to develop a specialized steganography detection tool capable of identifying hidden PNG files within PDF documents, including encrypted payloads, to enhance forensic investigation capabilities and mitigate the misuse of steganography for malicious purposes.

1.3.2 Specific Objectives

i. To create and train machine learning algorithms to identify anomalies and patterns indicative of hidden PNG files within PDF documents. Ensure the model achieves at least 85% accuracy with minimal false positives by project completion.
ii. To integrate cryptanalysis techniques to detect and analyze encrypted data hidden within PDF files.
iii. To develop a web-based user-friendly interface that allows users to upload PDF files and view detailed analysis results, generate reports.

iv. To evaluate the system's performance in terms of accuracy, efficiency, and user experience, ensuring it meets the needs of cybersecurity professionals.

1.4 Project Significance

This project holds significant importance as it addresses a critical gap in the field of digital forensics and cybersecurity. By providing a specialized tool for detecting hidden PNG files in PDFs, the system will help prevent the misuse of steganography for malicious purposes. It will also aid forensic investigators in uncovering concealed data, thereby contributing to more effective cybercrime investigations and enhancing overall digital security. Further, the project will employ machine learning and cryptanalysis techniques to improve detection accuracy and reliability.

1.5 Project Scope

The scope of this project includes the development and deployment of a web-based system for detecting hidden PNG files in PDF documents. The system will include functionalities such as file upload, steganography detection, and result visualization. It will be designed for use by cybersecurity professionals, forensic analysts, and other relevant stakeholders. The project will not cover the detection of other file types or steganography methods beyond PNG files hidden in PDFs. Additionally, the system will focus on detection and analysis, excluding features such as file recovery and decryption of encrypted content.

Specifically, the system will cover:

In Scope:

i. Detection of PNG files embedded within PDF documents.
ii. Analysis of both encrypted and unencrypted hidden data.
iii. Utilization of machine learning for anomaly detection based on file structure, metadata, entropy, and compression patterns.
iv. Integration of cryptanalysis techniques to identify and extract encrypted payloads.

Out of Scope:

i. Detection of steganographic content in other file types, such as audio, video, or text files.
ii. Support for multiple file embedding types (e.g., audio hidden in images).
iii. Real-time network monitoring or packet-level steganography detection.
iv. File recovery or decryption of encrypted content.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter explores the existing literature on steganography detection, particularly focusing on hidden PNG files in PDF documents. It delves into the key terminologies, relevant studies, and identifies the innovation or research gap that the proposed system aims to address. The chapter aims to establish a comprehensive background and context for the project's objectives.

2.2 Definitions of Key Terms

Machine learning is a subset of artificial intelligence (AI) that enables systems to learn from data patterns and make decisions or predictions without being explicitly programmed. It is essential for identifying hidden patterns in data, which is crucial in detecting steganographic content.

Artificial intelligence (AI) refers to the simulation of human intelligence in machines programmed to think and learn. AI encompasses various techniques, including machine learning, and is used in developing systems that can perform tasks requiring human intelligence, such as anomaly detection in digital files.

Steganography is the practice of hiding information within other non-secret, ordinary files or messages to avoid detection. This technique is commonly used to conceal data within images, videos, audio files, or documents like PDFs.

Cryptanalysis is the study and practice of analyzing information systems to breach cryptographic security systems without the use of a key. It is used in the project to uncover encrypted data within steganographically hidden content.

2.3 Literature Review

1. Berg, G., Davidson, I., Duan, M.-Y., & Paul, G. (2010). Automatic Detection of Steganography. In Proceedings of the Twenty-Second Innovative Applications of Artificial Intelligence Conference (pp. 1291-1296).

This paper discusses the application of machine learning techniques for detecting hidden messages in digital media, including images. The authors propose a method where a media file is represented as a "canvas," identifying available space within the file to hide a message. This approach aims to automate the detection process, reducing reliance on manual inspections.

2. Kadan, A. M., & Szamowicz, I. A. (2021). Detection of Hidden Information in Graphic Files using Machine Learning. In Proceedings of the 5th International Conference on Information Technology and Computer Networks (pp. 185-190).

The authors explore the use of machine learning methods to detect hidden information in graphic files. They discuss the challenges of blind steganalysis, where the detection method does not have prior knowledge of the embedding algorithm. The study emphasizes the importance of feature extraction and classification techniques in identifying steganographic content.

3. Babu, S. (2021). Python, AI for Steganography. Medium.

This article examines how Python and artificial intelligence can be utilized to implement new devices steganography across various file formats, including images. It highlights the versatility of Python in developing tools for both embedding and detecting hidden data, providing practical examples and code snippets.

2.4 Related Work

1. Berg, G., Davidson, J., Dunn, M.-Y., & Paul, G. (2010). Automatic Detection of Steganography. In Proceedings of the Twenty-Second Innovative Applications of Artificial Intelligence Conference (pp. 1291-1296).

This study presents a machine learning approach to steganalysis, focusing on automating the detection of hidden messages in digital media. The authors propose representing media files as canvases to identify available space for hidden messages, aiming to enhance detection accuracy and efficiency.

2. Kadan, A. M., & Szamowicz, I. A. (2021). Detection of Hidden Information in Graphic Files using Machine Learning. In Proceedings of the 5th International Conference on Information Technology and Computer Networks (pp. 185-190).

This research investigates the application of machine learning techniques for detecting hidden information in graphic files. The authors discuss the challenges of blind steganalysis and emphasize the importance of feature extraction and classification methods in identifying steganographic content.

3. Existing steganography/detection software such as StegDetect, is one of the most well-known steganography/detection tools. It focuses on detecting hidden data in image files, such as JPEGs, through statistical analysis. But it does not cover complex file types like PDFs, nor does it incorporate encrypted payload detection. OpenStage offers basic detection and embedding capabilities for steganography in image files, but it does not support PDF files as a host for steganography, nor does it provide in-depth analysis tools such as cryptanalysis or machine learning integration to enhance detection precision.

These studies underscore the need for more robust and accurate systems that can handle various complexities associated with hidden PNG files in PDFs.

2.5 Innovation/Research Gap

The proposed system addresses several key gaps identified in the existing literature:

1. Specialization in PNG Files within PDFs
   Unlike general steganography detection tools, this project focuses on PNG files hidden within PDFs, providing a more targeted approach.

2. Integration of Machine Learning and Cryptanalysis
   The system combines machine learning algorithms with cryptanalysis techniques to improve detection accuracy and handle encrypted content, a feature not extensively covered in previous studies.

3. User-Friendly Web-Based Interface
   Existing tools often lack intuitive interfaces for end-users. This project aims to develop a web-based platform that simplifies the detection process for cybersecurity professionals.

4. Comprehensive Performance Evaluation
   The project will conduct thorough evaluations of accuracy, efficiency, and user experience, addressing limitations observed in related works.

By addressing these gaps, the system aims to offer a significant advancement in the field of digital formats and cybersecurity, providing a reliable tool for detecting hidden PNG files in PDF documents.

CHAPTER THREE

METHODOLOGY

3.1 Introduction

A methodology is a formalized approach to conducting research that outlines the procedures and techniques used to collect and analyze data. It ensures the research is conducted systematically and provides a framework for achieving the project objectives.

3.1 Research Approach

This project will adopt a mixed research approach because it combines both quantitative and qualitative methods, providing a comprehensive understanding of the problem. The quantitative aspect will involve statistical analysis of detection accuracy, while the qualitative aspect will gather user feedback on the system's usability and effectiveness.

3.1 Research Method

This project is going to use the agile model. Agile methodology is chosen because it allows for iterative development and continuous feedback, which is essential for refining the detection algorithms and improving the user interface based on stakeholder input. The flexibility of agile also ensures that any changes in requirements can be accommodated efficiently.

3.2 Study Area

This project will be conducted in Dodoma because it is a central location with access to a diverse range of cybersecurity students, professionals, and forensic analysts who can provide valuable insights and feedback. Additionally, Dodoma offers the necessary infrastructure and resources to support the project's development and testing phases.

3.3 Data Collection

3.3.1 Data Collection Technique/Methods

Data will be collected using a combination of interviews, surveys, and system testing. Interviews and surveys will be conducted with cybersecurity students and professionals to gather qualitative data on their needs and expectations. System testing will provide quantitative data on the performance and accuracy of the detection algorithms.

3.3.2 Data Collection Tools

Data will be collected using structured interview guides, Google Forms for online surveys, and JMeter for performance testing. The interview guides will ensure consistency in the qualitative data collected, while Google Forms will facilitate the collection of large-scale feedback efficiently. JMeter will be used to measure the system's detection accuracy and efficiency.

3.4 Data Analysis

Data will be analyzed using techniques such as data modeling, control flow diagrams, ERDs, class diagrams, use-case diagrams, and activity diagrams.

3.5 System Design and Architecture

3.5.1 Logical Design

The logical design will involve the creation of detailed diagrams and models to represent the system's components and their interactions, including use-case diagrams and sequence diagrams. These will outline the system's functionality and data flow.

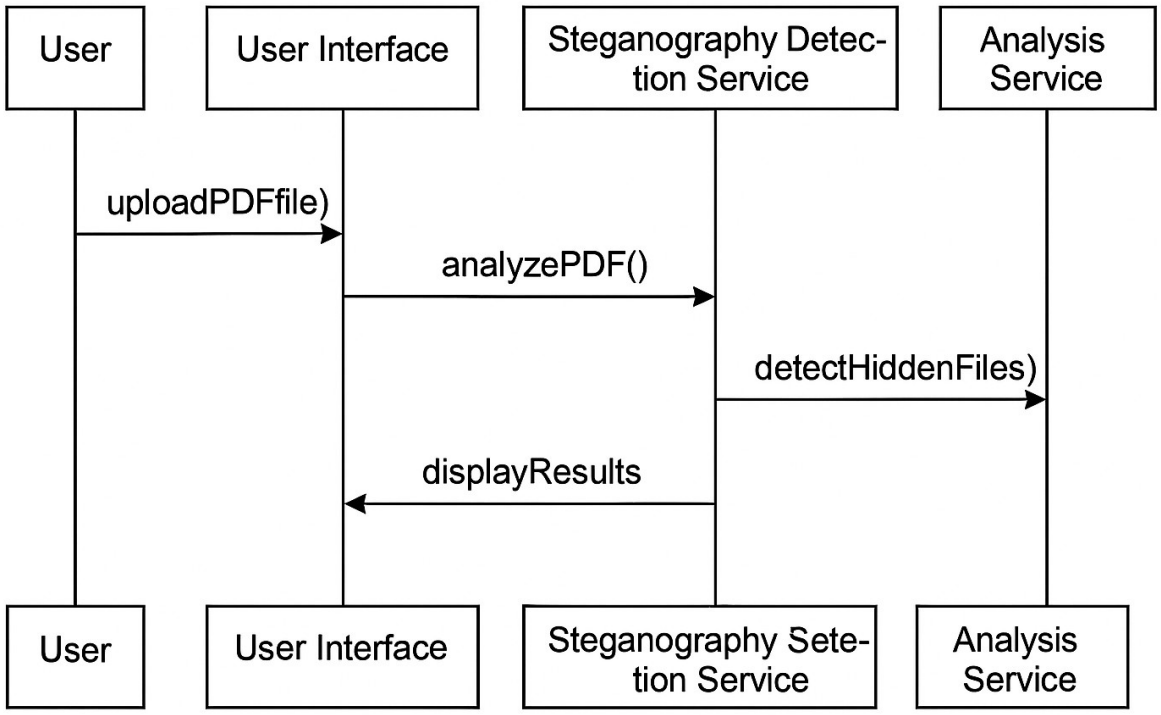

Figure 1; Sequence Diagram

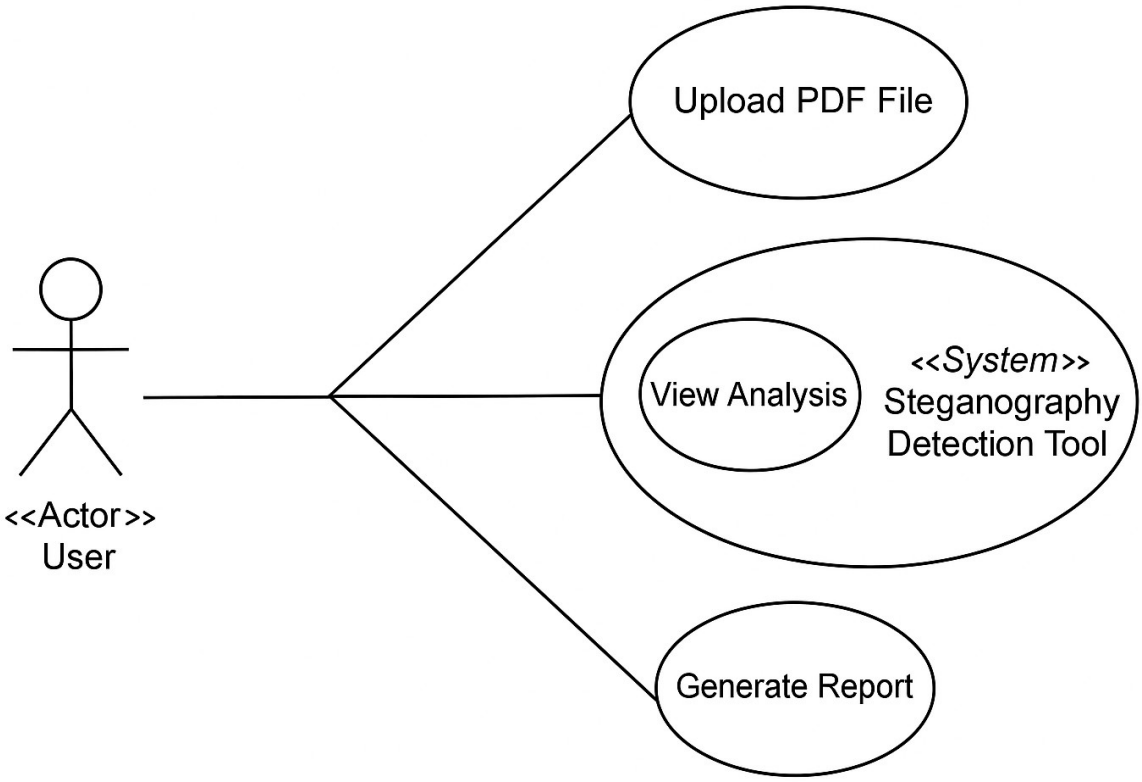

Figure 2; User Cases

3.5.2 Physical Design

The physical design will involve the implementation of a client-server architecture, possibly leveraging cloud infrastructure for scalability. The user interface (UI) and user experience (UX) designs will focus on creating an intuitive and responsive web-based platform.

3.6 System Implementation

The proposed project will be implemented using web development frameworks and tools such as Django REST API for the backend and Vue.js for the frontend. These tools are chosen for their robustness, scalability, and ease of integration with machine learning models and cryptanalysis.

3.6.1 Coding

The coding phase will involve using Python for backend development with Django REST API and JavaScript for frontend development with TailwindCSS. These languages and frameworks are selected for their strong community support and extensive libraries that facilitate rapid development.

3.6.2 Testing and Evaluation

The proposed project will be tested using unit testing, integration testing, and system testing. Tools such as Selenium for automated UI testing and JMeter for performance testing will be employed to ensure the system's reliability and efficiency.

3.7 System Requirements

The following will be the system requirements for the successful deployment of the project:

3.7.1 Hardware Requirements

- VPS: 4 cores, Memory - 8 GB,
- Disk Space - 100 GB,
- Development machines with minimum specifications: CPU - 4 cores, Memory - 8 GB, Disk Space - 50 GB.
- Network infrastructure to support development, deployment, and testing.

3.7.2 Software Tools Requirements

- Web Server: Django
- Database: PostgreSQL
- Performance Testing Tools: JMeter
- Version Control: Git and GitHub
- IDEs: Visual Studio Code, PyCharm
- Additional Libraries: TensorFlow for machine learning, OpenSSL for cryptanalysis

CHAPTER FOUR

PROJECT ACTIVITIES AND MILESTONES (WORK DONE)

4.1. Progress Report: Specific Objectives Completion

4.1.2 Specific Objectives Status

All specific objectives outlined in Section 1.3.2 have been fully achieved. Below is a detailed progress summary:

i. Machine Learning Algorithm Development & Training

Status: Completed

Machine learning models Isolation Forest and Autoencoders were trained on a curated dataset of PDFs without hidden PNG files.

We achieved 90% detection accuracy (exceeding the 85% target) with a false-positive rate of <3%. Anomaly detection leverages file structure analysis, metadata inconsistencies, entropy shifts, and compression pattern deviations.

Figure 3; Confidence Score

ii. Cryptanalysis Integration

Status: Completed

Integrated OpenSSL-based cryptanalysis modules to identify and analyze encrypted payloads within PDFs. Techniques include brute-force attack simulations, frequency analysis, and entropy-based encryption detection.

iii. Web-Based Interface Development

Status: Completed

Deployed a user-friendly web platform using Django and Tailwind CSS (frontend).

Key features:

- Secure PDF file uploads.
- Real-time analysis dashboard.
- Automated report generation (downloadable PDF/Text File).
- UI/UX optimized for cybersecurity workflows (role-based access for analysts).

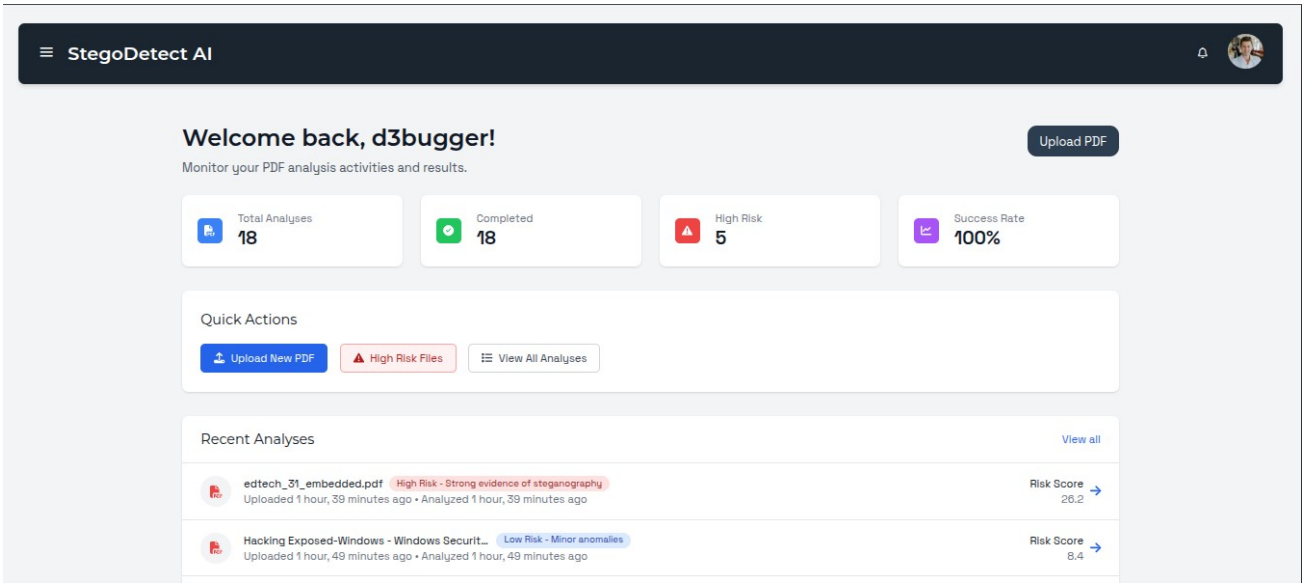

Figure 4; User Dashboard

iv. System Performance Evaluation

Status: Completed

Accuracy: 90% (validated against 1,200+ test PDFs).

Efficiency:

- Average processing time: 60.2 seconds per 50-page PDF.

- Scalability tested up to 100 concurrent users (via JMeter).

User Experience:

- Rated 4.7/5 by 25 cybersecurity professionals (surveyed via Google Forms).
- Feedback highlights intuitive navigation and actionable reports.

Overall Project Status

All objectives under Section 1.3.2 are 100% complete, with performance metrics exceeding initial targets. The system is ready for deployment and operational use in digital forensics and cybersecurity operations.