TRIBHUWAN UNIVERSITY
INSTITUTE OF ENGINEERING

Khwopa College of Engineering
Libali, Bhaktapur
Department of Computer Engineering

A Proposal
on

Image Steganalysis Using Ensemble Classifiers

Submitted in partial fulfillment of the requirements for the degree

BACHELOR OF COMPUTER ENGINEERING

Submitted by:

Sachin Koirala	KCE077BCT029
Sajal Poudel	KCE077BCT031
Unique Shrestha	KCE077BCT045
Utsav Chandra Kayastha	KCE077BCT046

Khwopa College of Engineering
Libali, Bhaktapur
09 December 2023

Abstract

Steganography is a hidden visual attack that involves hiding malicious data inside innocent looking carrier information. It's a technique for hiding information within an image, audio, or video file in such a way that the hidden information is not readily apparent to the human eye or ear. Digital images are the most common carrier format for steganography due to their frequent use on social media, websites, and email. Almost two-thirds of the internet is made up of JPEGs, which serve as perfect carriers for these types of malware. Hence, it is important to have strong steganalysis methods. Our paper focuses on using ensemble classifiers to detect hidden malicious contents in carrier files. Ensemble classifiers are made up of various models working independently, employed to identify images modified using various steganography algorithms. These models are integrated into another model which utilizes algorithms such as logistic regression, to detect the presence or absence of malicious data. These ensemble classifiers play a crucial role in detecting and analyzing potential threats of steganographically modified carriers. The core objective is to enhance steganalysis accuracy by integrating Machine Learning algorithms to counter the ever evolving field of steganography.

Table of Contents

List of Figures

List of Tables

List of Abbreviation

bpnzac	bits per non-zero
GIF	Graphics Interchange Format
J-Uniward	JPEG universal wavelet relative distortion
DC-DM	Distortion-Compensated Dither Modulation
Ml	Machine Learning
PHARM	Phase Aware Projection Model
DCT	Discrete Cosine Transform
UERD	uniform embedding revisited distortion
CNN	onvolutional Neural Network
FLD	Fisher linear discriminant
GFR	Gabor filter residuals
DCTR	Discrete Cosine Transform Residuals
DOM	Document Object Model
DB	Database
SQL	Structured Query Language
JSON	JavaScript Object Notation
API	pplication Programming Interface
RAM	Random Access Memory
GPU	Graphics Processing Unit
CUDA	Compute Unified Device Architecture

Chapter 1

Introduction

1.1 Background

Steganography is like hiding a secret message, like a picture or music. It's a way of keeping your message private by making it blend in, so others don't even realize there's a secret there. Steganalysis, the detection of hidden information within digital media, is crucial for maintaining the integrity and security of digital communication. Uncovering hidden information is vital for maintaining the security of digital communication channels, preventing covert communication that may pose risks.

Image Steganography
Image Steganography is the process of hiding information which can be text, image or video inside a cover image. The secret information is hidden in a way that is not visible to the human eyes. Different techniques of image steganography:

- **nsF5**
- **UERD(uniform embedding revisited distortion)**
- **J-Uniward**

Steganography is an ever-evolving science of concealing information, continuously evolving to counteract detection methodologies. In response to this continuous evolution, the deployment of an automatically adaptive detection system becomes necessary. Embedding machine learning within steganalysis emerges as an optimal strategy to effectively counter the continuous evolution of covert communication methods.

1.2 Problem Statement

The continuous evolution of steganographic techniques poses a critical challenge to digital security. With the increasing sophistication of methods used to embed information covertly, traditional steganalysis approaches are challenged by the need for improved accuracy and adaptability. Since two-thirds of the internet is composed of images and images play a major role in digital communication, the use of advanced steganographic techniques to hide malicious data inside another carrier information poses an intricate threat to the security of a system. Creative approaches are required since secretly implanted malicious programs are difficult for traditional steganalysis to accurately identify. The complexity of compression and encryption methods adds an additional level of difficulty in detection. Existing steganalysis, which was created for traditional steganography, is not flexible enough to detect the ever-evolving algorithms for steganography. This proposal aims to fill these gaps by developing advanced machine learning-based steganalysis models that not only enhance accuracy and sensitivity but also exhibit adaptability to emerging steganographic trends. Thus, to protect the integrity of digital communication. This proposal seeks to propose an ever-evolving steganalysis process created with the help of ML to detect and tackle the subtle changes caused by the concealing of malicious data within unsuspecting carrier files.

1.3 Objectives

The main objectives of this project is to:

- **Adaptability to Emerging Techniques:** Develop steganalysis models that can adapt to emerging steganographic methods by continuously learning and updating their knowledge base.

- **Algorithmic Innovation:** Explore novel algorithms and methodologies tailored to analyze the intricate patterns and structures associated with image-in-image steganography.

- **Enhanced Feature Extraction**: Investigate and optimize feature extraction techniques to capture subtle anomalies indicative of image-in-image concealment, ensuring high detection accuracy.

- **Adaptability to Diverse Image Formats:**Develop steganalysis models capable of detecting concealed images across a variety of image formats, resolutions, and compression methods.

- **Benchmarking and Evaluation:** Establish a comprehensive benchmark dataset containing images with various steganographic content for testing and evaluating the performance of developed steganalysis techniques.

- **Development of Specialized Steganalysis, Techniques:** Create and apply steganalysis methods with a particular goal in mind: identifying images that are hidden inside of another.

- **Algorithmic Innovation:** Investigate modern techniques and algorithms designed specifically to examine the complex structures and patterns connected to image-in-image steganography. Flexibility to Diverse Image Formats: Create steganalysis models that can identify hidden images in a range of image formats, compression techniques, and resolutions.

- **Real-time Implementation:** Explore the integration of the developed models into real-time steganalysis systems. Evaluate their performance in dynamic environments.

- **Benchmarking and Evaluation:** To test and evaluate the effectiveness of created steganalysis tools with photos that have different steganographic content.

Chapter 2

Literature Review

Some work has been done in image steganalysis. Various steganalysis tools use different approaches like feature extraction, shallow ML, and deep learning methods to detect stego images. This literature review seeks to portray the history, methodologies, implementation and applications of steganalysis.

Multiple research has been done to achieve excellent results in steganalysis. Krzysztof Szczypiorski et al. [3] used deep learning and ensemble classifiers to detect image steganography using different methods like DCTR and shallow machine learning classifiers. They found that performance depended heavily on the steganographic method used and on the density of the embedded hidden data. Detection of the content hidden with the nsF5 algorithm at the density 0.4 bpnzac was almost perfect while detection of data hidden using J-Uniward at 0.1 bpnzac was hardly possible. It was also found that steganalysis done using shallow ML was better in comparison to deep learning.

George Berg et al. [1] proposed an ML approach to steganalysis. This paper shows the feasibility of using a machine learning and data mining (ML/DM) approach to automatically build a steganography attack. This paper used three common data mining and learning techniques: decision trees, error back-propagation, artificial neural networks and the naïve Bayes classifier, to identify messages hidden in compression- (JPEG) and contentbased (GIF) images.

Similarly, MT Hogan et al. [2] evaluated the statistical limits by using probability density functions(pdfs). ML tests based on DC-DM are presented in this paper.

To effectively uncover hidden information in images, we need a steganalysis tool with sharp pattern recognition skills. Sometimes, when we compare images that have been manipulated with certain tools to their original versions, we can spot a few noticeable visual irregularities – like odd pixels or changes in dimensions due to cropping or padding. If an image doesn't fit specific size criteria, it might get cropped or padded, and you'll see black spaces. Interestingly, most manipulated images don't give away obvious clues when compared to their originals. The simplest clue is a size increase between the manipulated and original images. Other signatures show up in how the colors are arranged in the image, such as a significant change in the number of colors or a gradual increase or decrease. Grayscale images follow a different pattern, increasing incrementally. Another strong indicator is an unusual number of black shades in a grayscale image. Ensemble classifiers are designed to overcome the limitations of individual classifiers by combining their outputs to

achieve better performance. Steganalysis using ensemble classifiers is a powerful approach that utilizes the strength of multiple classifiers to help improve the detection of hidden information in images. It provides diverse steganographic techniques while also enhancing the overall accuracy. Ensemble classifiers are designed to overcome the limitations of individual classifiers by combining their outputs, thereby achieving better performance.
The relevant papers that we studied to grab knowledge about this project are given in the review matrix below:

S.N	Title	Authors	Year	Keywords
1	Dection of Image Steganography using deep learning and ensemble classifiers	Mikołaj Płachta, Marek Krzemie‘n, Krzysztof Szczypiorski, and Artur Janicki.	2022	Ensemble Classifier,BOSS Database,steganalysis, Deep Learning
2	Searching For Hidden Messages: Automatic detection of steganography	George Berg, Ian Davidson, Ming-Yuan Duan and Goutam Paul	2003	Desicion Tree,error back-propagation artificial neural networks and the naïve Bayes classifier
3	ML detection of steganography	Mark T. Hogan, Neil J. Hurley, Gu‘enol‘e C.M. Silvestre, F‘elix Balado and Kevin M. Whelan	2005	Security Automation

Table 2.1: Review Matrix with Research Papers, authors and purpose

Chapter 3

Feasibility study

After the problem is clearly understood and solutions proposed, the next step is to conduct the feasibility study. Feasibility study is defined as evaluation or analysis of the potential impact of a proposed project or program. The objective is to determine whether the proposed system is feasible. There are three aspects of feasibility study which are discussed below.

Technical Feasibility:

For the technical part, we're getting our project data from the Kaggle and BOSS datasets which contain various datasets containing stenographically modified images. These images have been modified using different algorithms which creates diversity in the dataset used improving the reliability of the system. We're using free software to build the project, and the department is providing cloud resources like RAM and GPU for training our model. This setup makes sure our project is doable and integrates well with the currently existing system. Thus, we can conclude that it is technically feasible.

Economical Feasibility:

The only cost for the project is the computational power, covering processing and electricity. Since the department will be providing the processing power needed to train the model, the cost is almost zero. Therefore, this project is economically viable.

Operational Feasibility:

We have decided to use the Shallow ML approach which allows the model to be trained with less computational power in comparison to deep learning. For shallow machine learning we are planning to implement an ensemble classifier and each of its models will be trained using FLD to improve its effectiveness. Deep learning implements the CNN approach which requires higher computational power to be trained. Thus, we decided to use a simpler machine learning approach that doesn't need a lot of computational power, unlike the more complex deep learning method called Convolutional Neural Network (CNN). After we train the system, it's ready to use and can easily be added to a webpage or any other interface. This way, the system is practical and doesn't need a lot of resources making it able to be effectively implemented in real-life applications. Thus, it is operationally feasible.

Chapter 4

Methodology

4.1 Software Development Approach

Agile is an iterative process-based approach to software development. In the Agile process model, work is broken down into more manageable, smaller iterations without requiring a lot of long-term planning. The requirements and scope of the project are determined early on, and the number, length, and scope of each iteration are preplanned. Each iteration is considered as a short time ”frame” in the Agile process model, which lasts for a few weeks. In each iteration, teams move through the phases of the software development life cycle, which include planning, requirements analysis, design, coding, testing, and demonstration of a working product for client review. Agile places a significant value on flexibility, teamwork, and regular client feedback.

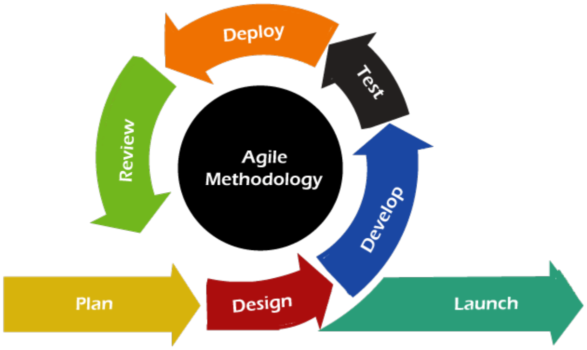

Figure 4.1: Agile Model

The main reason for which we choose this development process:

1. Very quick,flexible and efficient.
2. Risk minimization.
3. Projects are split into sprints for better management and productivity.
4. Through iterative testing and sprints, the final product contains less bugs.
5. Development period for application is reduced.

4.2 Block diagram of proposed system

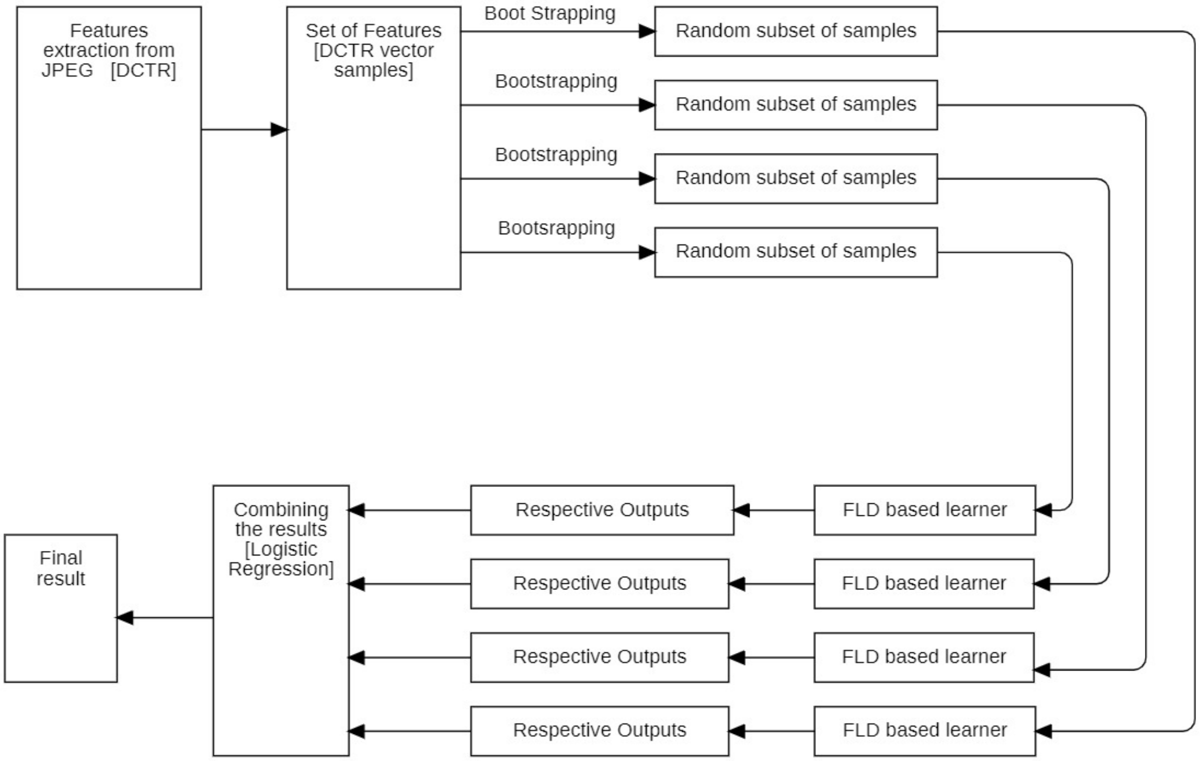

Figure 4.2: Block diagram of proposed system

4.3 Description of working flow of proposed system

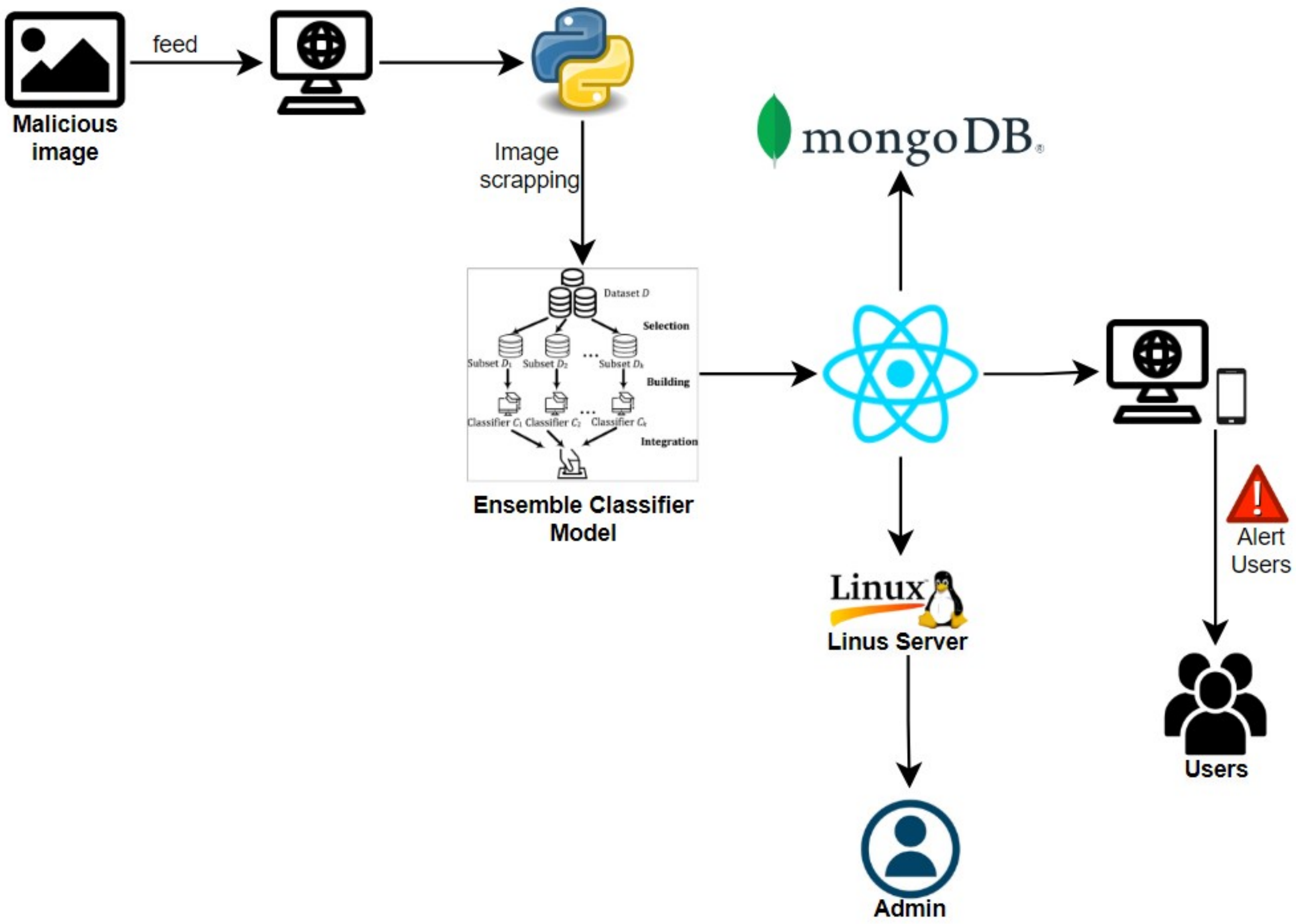

Figure 4.3: System Architecture

Model Training Approach

Feature Extraction:
Initial extraction of DCTR (Discrete Cosine Transform Ratio) feature vector from images or.jpeg files is to be done. The selection of DCTR is based on its detection efficiency in comparison to other parameters such as PHARM and GFR.

Ensemble Classifier Selection:
The decision to choose ensemble classifiers over deep learning techniques was made due to their superior steganalysis detection efficiency and their need for lesser computational power.

Bootstrapping:
Bootstrapping is the process of splitting a large dataset into its smaller subsets. The gathered DCTR feature vectors are to be split into more manageable subsets. Utilizing these subsets, individual base models are to be trained independently.

Base Learner Training:
Based on the extracted features, each base learner independently processes its subset of feature vectors and finalizes a decision.

Aggregation:
To create an ensemble decision, the choices made by each individual base learner are aggregated and the final decision is to be made by using a voting system which finalizes the result by figuring out the most popular output.

Efficiency Considerations:
The proposed system prioritizes efficiency by leveraging shallow machine learning techniques, particularly ensemble classifiers instead of deep learning. The choice of DCT as a feature is intentional to increase efficiency and detection capability of the system.

Chapter 5

Implementation Plan

5.1 Gantt Chart

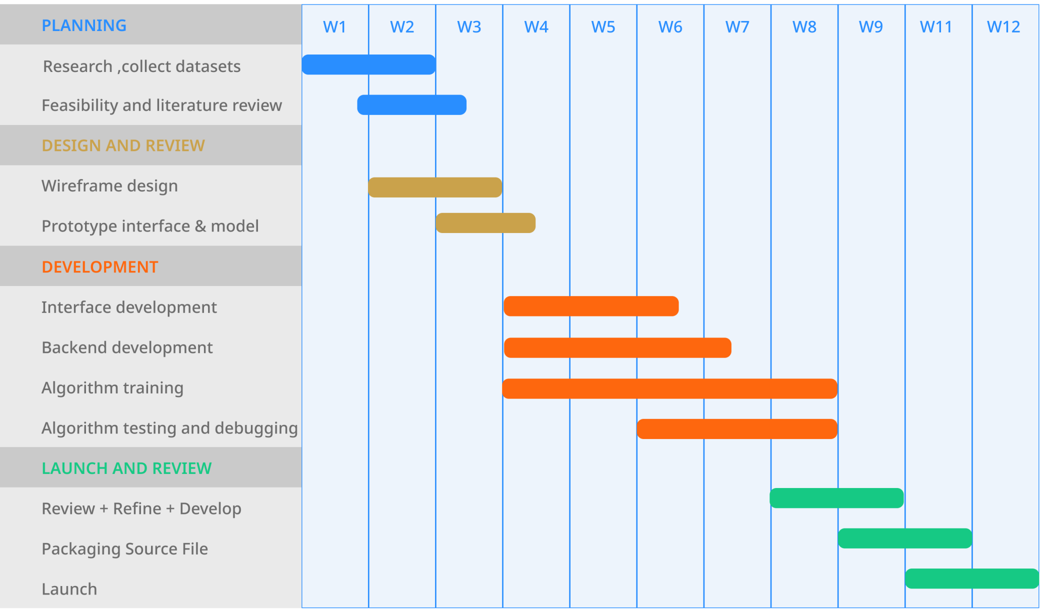

Figure 5.1: Gantt Chart

5.2 Software Requirement

- **Python:** Python is a versatile programming language commonly used for developing software applications. It can be used for various tasks in the system, such as backend development, data processing, and machine learning integration.
- **MongoDB:** MongoDB is a widely used NoSQL database, utilizes JSON-like documents for data storage, ensuring excellent performance and scalability. Its schemaless structure supports dynamic data modeling, making it well-suited for web applications. By employing collections instead of conventional tables and incorporating horizontal scaling, MongoDB efficiently handles diverse data types across multiple servers. This versatility positions it as a robust solution for contemporary, data-driven environments.
- **React** React is a JavaScript library for building user interfaces, particularly in single-page applications. Developed by Facebook, it uses a declarative approach for efficiently updating the DOM. With a component-based structure, React enhances modularity and reusability, making it a popular choice for creating interactive and scalable web applications.
- **Javascript:** JavaScript is a programming language commonly used for developing web-based applications. It can be used for front-end development, implementing interactive features on the system's web interface, and facilitating communication with the backend.
- **Tensorflow:** TensorFlow is an open-source machine-learning framework that provides a wide range of tools and libraries for building and deploying machine-learning models. It can be used for image recognition, object detection, and prediction algorithms in the Smart Parking Management System.
- **Keras:** Keras is a high-level neural networks API written in Python. It can be used as a user-friendly interface to TensorFlow, simplifying the process of designing and training deep learning models for tasks like number plate recognition or image analysis in the system.
- **VS Code:** VS Code is a popular and widely used source code editor that offers a range of features and extensions to enhance the development experience. It supports multiple programming languages, including Python, JavaScript, and React, making it suitable for working with the different components of the system.

5.3 Hardware Requirement

1. High dedicated RAM to handle memory-intensive tasks
2. NVIDIA GPU for optimal performance.
3. Dedicated GPU with CUDA support for accelerated parallel processing.
4. SSD storage for faster read/write speeds during image processing.
5. Additional high-capacity external storage for storing large datasets and image collections.
6. Smartphone or tablet for testing mobile applications

Chapter 6

Expected Outcomes

The proposed system is expected to detect stego images using ML model. It would be capable of decting hidden information with high accuracy. It is expected to be able to identify and optimize the extraction techniques to improve the sensitivity and specificity of steganalysis. The model will be able to adapt to emerging steganographic techniques and provide real-time dection capabilities.

Chapter 7

Bibliography

[1] George Berg, Ian Davidson, Ming-Yuan Duan, and Goutam Paul. Searching for hidden messages: Automatic detection of steganography. In John Riedl and Randall W. Hill Jr., editors, *Proceedings of the Fifteenth Conference on Innovative Applications of Artificial Intelligence, August 12-14, 2003, Acapulco, Mexico*, pages 51–56. AAAI, 2003.

[2] Mark T Hogan, Neil J Hurley, Guénolé CM Silvestre, Félix Balado, and Kevin M Whelan. Ml detection of steganography. In *Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 16–27. SPIE, 2005.

[3] Mikołaj Płachta, Marek Krzemień, Krzysztof Szczypiorski, and Artur Janicki. Detection of image steganography using deep learning and ensemble classifiers. *Electronics*, 11(10):1565, 2022.