TRIBHUVAN 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
2023

Abstract

Steganography is a technique used to hide information, such as text, images, audio or videos, within another carrier file in a way that is not easily detectable by human senses. Digital images, constituting nearly two-thirds of online content are popular carriers for steganography.This can be exploited by malicious actors to hide harmful information. This hidden malware can damage data, steal information, or disrupt systems. An example would be Witchetty espionage group (also known as LookingFrog) which is a chinese hacker group which has so far compromised the security of middle eastern governments and african countries by using the Windows logo as their point of entry. To address this challenge, we propose an ensemble classifier model using a bagging method implemented as a random forest. We will be using high dimensional features from CC-C300 model to train our ensemble classifier. This approach offers robustness, model diversity, and faster development compared to other machine learning methods. The proposed model will perform effectively on three different steganographic methods [nsF5, J-UNIWARD and UERD] that hide messages in JPEG images.

Keywords: Steganography, Steganalysis, Ensemble Classifiers, CC-C300, Machine Learning, Random forest, JPEG

Table of Contents

List of Figures

List of Tables

List of Abbreviation

API	Application Programming Interface
bpnzAC	Bits Per Non-zero AC
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
DC-DM	Distortion Compensated-Dither Modulation
DB	Database
DCT	Discrete Cosine Transform
DCTR	Discrete Cosine Transform Residuals
DOM	Document Object Model
FLD	Fisher Linear Discriminant
GBM	Gradient Boosting Machine
GIF	Graphics Interchange Format
GFR	Gabor Filter Residuals
GPU	Graphics Processing Unit
HUGO	Higly Undectectable SteGO
J-Uniward	JPEG UNIversal WAvelet Relative Distortion
JPEG	Joint Photography Experts Group
JSON	JavaScript Object Notation
K-NN	K-Nearest Neighbors
LSB	Least Significant Bit
ML	Machine Learning
PHARM	Phase Aware Projection Model
PNG	Portable Network Graphics
RAM	Random Access Memory
STC	Syndrome Trelis Coding
SVM	Support Vector Machine
SQL	Structured Query Language
UERD	Uniform sEmbedding Revisited Distortion
WPC	Wet Paper Codes
YCbCr	Luminance, Chrominance Blue, Chrominance Red

Chapter 1

Introduction

1.1 Background

Steganography is the skillful technique of secret communication, accomplished through embedding a piece of information into a carrier. In steganography, a "carrier" refers to the cover or host file that conceals the hidden message or information. The word steganography is derived from the Greek words *"stegos"* meaning *"cover"* and *"grafia"* meaning *"writing"* defining it as *"covered writing"* [1]. Misusing steganography for malicious purposes can have serious consequences. A recent trend involves exploiting various steganographic techniques to embedd malware in the carrier. Some real life example would be: Hiding malicious code within banner ads, tricking users into installing malicious app, embedding malicious executables and so on. Steganography can utilize various file formats, primarily classified into four categories: text, images, audio, and video. Among these file formats, images are widely used as cover object.
Steganalysis is the process of detecting hidden information within digital media, uncovering hidden data that has been secretly embedded using steganographic techniques. The goal of steganalysis is to collect sufficient evidence about the presence of embedded message and to break the security of its carrier. Thus defeating the purpose of steganography. The importance of steganalytic techniques that can reliably detect the presence of hidden information in images is increasing. Steganalysis finds its use in computer forensics, cyber warfare, tracking the criminal activities over the internet and gathering evidence for investigations particularly in case of anti-social elements. [2]

Image Steganography
Image Steganography is a steganography method where the hidden information is embedded into a image as carrier file. Among the various available formats of images, JPEG format is widely used for steganography. The figure below shows the glimpse of how a JPEG image is compressed.

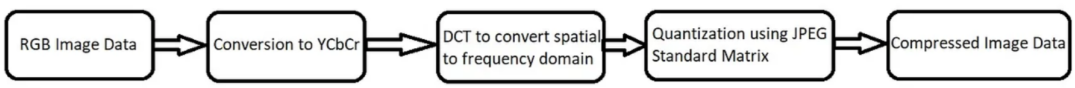


Figure 1.1: Steps used in implementation of Compression Algorithm

First the RGB color space is converted into YCbCr colorspace. After the YCbCr coversion, the image is partitioned into 8×8 non-overlapping blocks. After that each pixel is transfered from range of 0 to 255 to -128 to 127. Finally, Each of these blocks is then subjected to a 2-D Discrete Cosine Transform (DCT). The DCT transforms the spatial data in the block into frequency data. After the DCT, the coefficients are quantized, meaning they are divided by a factor determined by a quantization table. The quantization step is what actually removes information from the image, and it is this step that makes the JPEG compression process lossy.

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)cos[\frac{\pi(2x+1)u}{2N}]cos[\frac{\pi(2y+1)v}{2N}] \tag{1.1}$$

for $u, v = 0, 1, 2, \ldots, N-1$ and $f(x,y)$ is pixel position and $\alpha(u), \alpha(v)$ as

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & for \quad u = 0 \\ \sqrt{\frac{2}{N}} & for \quad u \neq 0 \end{cases} \tag{1.2}$$

$$\alpha(v) = \begin{cases} \sqrt{\frac{1}{N}} & for \quad v = 0 \\ \sqrt{\frac{2}{N}} & for \quad v \neq 0 \end{cases} \tag{1.3}$$

Equation (1.1), (1.2), (1.3) are the formulas to calculate DCT coefficients, $\alpha(u)$ and $\alpha(v)$ respectively. [3]

Some popular techniques of image steganography are listed below:

1. **DCT-LSB Method:**
   The DCT-LSB method conceals data in an image by dividing it into 8x8 blocks, transforming each block using Discrete Cosine Transform (DCT), and then replacing the least significant bit of DCT coefficients with hidden data. [4]

2. **F5 and nsF5 Algorithms:**
   The F5 steganography algorithm operates by manipulating Discrete Cosine Transform (DCT) coefficients. F5 adjusts coefficients by adding 1 to those with a positive value and subtracting 1 from those with a negative value, leaving zero-valued coefficients unchanged. However, a drawback known as the "shrinkage" problem arises when coefficients with values of 1 or -1 become zero during embedding, reducing the algorithm's capacity. [5]
   To overcome the shrinkage problem, the nsF5 algorithm, an advanced version of F5, employs Wet Paper Codes (WPC). This technique designates certain coefficients as "non-modifiable", preventing alterations after data embedding. This innovation addresses the shrinkage issue without sacrificing capacity. Even if new zeros are generated from coefficients with values of 1 or -1, the nsF5 algorithm records them, enhancing image quality and capacity compared to the common F5 method. [6]

3. **UERD (Universal Embedding Reduced Distortion):**
   UERD (Universal Embedding Reduced Distortion) is a method that hides information in pictures so that it's hard to notice.By strategically analyzing Discrete Cosine Transform (DCT) coefficients, UERD chooses those areas in the picture where changes won't be obvious. UERD uses syndrome trellis coding (STC) to hide the message bits in the desired values. This further increases the security of the embedded data by making it harder to detect. [7]

4. **J-UNIWARD Algorithm:**
   The J-UNIWARD algorithm operates in both the spatial domain and the frequency domain. In the spatial domain, it uses the Least Significant Bit (LSB) steganography technique to hide data in the least significant bits of the pixel values. In the frequency domain, it uses the Discrete Cosine Transform (DCT) to transform the image into the frequency domain, where it hides data in the DCT coefficients. [8]

5. **HUGO Algorithm:**
   HUGO, which stands for Highly Undetectable steGO, is a steganography algorithm designed to hide information within images. It achieves this by analyzing complex patterns and structures in an image, determining optimal locations to hide data. [9] [10]

1.2 Problem Statement

The continuous evolution of steganographic techniques poses a critical challenge to digital privacy. With the increasing sophistication of methods used to embed information secretly, traditional steganalysis approaches are challenged by the need for improved accuracy and adaptability. Since two-thirds of the internet is composed of images, and they play a major role in digital communication. The use of advanced steganographic techniques in image endangers the user with malicious data hidden within it.So, creative approaches are required since secretly implanted malicious data are difficult for traditional steganalysis to identify. Existing steganalysis, which was created for traditional steganography, is not flexible enough to detect the ever-evolving algorithms of steganography. This project aims to fill these gaps by developing an advanced machine learning-based steganalysis model that detects even the slight changes in the pixel coefficients within the image, which gives us certainty that the image has been modified. Thus, this proposal seeks to propose an 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:

- To detect steganographically modified JPEG images using Ensemble Classifier, irrespective of the algorithm used for steganography.

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 steganographically altered 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. [11] 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 is shown that steganalysis done using shallow ML is better in comparison to deep learning. This point is further proved by the fact that shallow ML consumes less resources and requires less time to be trained in comparison to deep ML and still provides accuracy similar or better than deep ML classifiers.

The document titled *"The Discrete Cosine Transform: Theory and Application"* [3] gives us a comprehensive overview of the Discrete Cosine Transform(DCT) and its application in digital image and video processing. The document discusses the properties of the DCT, including its decorrelation characteristics, energy compaction, and its ability to reduce entropy. It highlights the DCT's role in efficient coding and compression, particularly in the context of image and video standards such as JPEG and MPEG. Additionally, The document addresses the inverse DCT operation and its impact on visual distortion, providing examples of reconstructed images at different quantization levels.

George Berg et al. [12] 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 content based (GIF) images.

MT Hogan et al. [13] 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.

"Steganalysisis in high dimensions: Fusing classifiers built on random subspace" [14] provides core concepts of this project such as ensemble classifier and importance of selection of features. A distinctive subject which it has touched upon is the concept of Curse of Dimensionality (CoD) which shows the relation of complexity and increase in resource usage for computation. It is highlighted how ensemble classifiers can counter this problem by using reduced dimension for training its base learners.

"Ensemble Classifiers for Steganalysis of Digital Media" [15] highlights several key studies in the field of steganalysis, which provides a solid foundation for understanding the current state of steganalysis. The document discusses the implementation of ensemble based steganographically altered image classifier using many base learners for classification. The proposed base learners are trained using FLD analysis due to its ability to increase diversity The performance of the proposed model even though gets trained in very less time in comparison to usually used classification method of G-SVM can classify with similar or better accuracy. It is highlighted that a G-SVM classifier takes about 8 hours to be properly trained while an ensemble classifier takes only 20 minutes.

"A fast and accurate steganalysis using Ensemble classifiers" [16] provides an in-depth insight into the use of an ensemble of classifiers for steganalysis, with a focus on machine learning. The ensemble-based steg analyzer uses feature vectors from multiple stegalyzers to create a decision algorithm that allows the combination of information from different steganalyzers. The resulting steganalyzer is also inherently suitable for multi-class classification scenarios. The paper presents a novel steganalysis decision framework using hierarchical classifiers, which addresses the limitations of existing steganalysis methods and provides a scalable and cost-effective approach to steganalysis. 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.

"J. Kodovský and J. Fridrich. Calibration revisited" [17] provide information on the pre features and their Cartesian calibrated and Non-cartesian calibrated form."A Markov Process Based Approach to Effective Attacking JPEG Steganography" [18] and "Merging Markov and DCT features for multi-class JPEGsteganalysis" [19] guides the outlook of our project to a better angle as it provides very crucial details on the section of feature extraction.They provide more insight on the pre features which can be utilized for better classification. These literature provided more insights on CC-PEv and CC-SHI which are different pre features used for steganalysis. "JPEG Image Steganalysis

Utilizing both Intrablock and Interblock Correlations" provides more insight on the importance of considering relation between inter and intra block correlations during pre feature creation for better detection or classification.
The dataset we will be using on this project will be taken from IStego100k [20]. IStego 100K is a large-scale steganalysis consisting of 208,104 images with a size of 1024*1024 pixels. The training set consists of 200,000 images organized into 100,000 cover-setgo image pairs. The testing set comprises the remaining 8,104 images. Each image in the dataset has randomly assigned quality factors in the range of 75-95. Three well-known steganographic algorithms J-uniward, nsF5, and UERD [8] [6] [7] are randomly selected for embedding in the images. The embedding rate for each image is randomly set in the range of 0.1-0.4 bpac.

The relevant papers that we studied to grab knowledge about this project are given in the review matrix below:

S.N.	Name	Year	Authors	Dataset	Findings
1	Searching for hidden messages Automatic detection of steganography [12]	2003	George Berg, Ian Davidson, Ming-Yuan Duan, and Goutam Paul	BOSS	Images are commonly used to transmit hidden messages. Different strategies are used to hide messages in GIF and JPEG formats.
2	International Workshop on Information Hiding	2005	Jessica Fridrich, Miroslav Goljan, and David Soukal	NA	matrix LT process,Block Minimal embedding,applications to steganography and data embedding
3	On steganographic embedding efficiency	2007	Jessica Fridrich, Petr Lisonek, and David Soukal.	NA	Matrix embedding using linear codes (syndrome coding) is a general approach to improving embedding effciency of steganographic schemes.
4	Ml detection of steganography	2005	Mark T Hogan, Neil J Hurley, Guenole CM Silvestre, Felix Balado, and Kevin M Whelan	BOSS	Non-blind steganalysis, Distortion-Compensated Dither Modulation
5	The discrete cosine transform (DCT): theory and application	2003	Syed Ali Khayam.	NA	DCT, DCT applications, properties of DCT

S.N.	Name	Year	Author	Dataset	Findings
6	Steganalysis in high dimensions Fusing classifiers built on random subspaces	2011	Jan Kodovsky and Jessica Fridrich.	BOSS	The paper proposes ensemble classifiers as an alternative to support vector machines. Experiments with steganographic algorithms nsF5 and HUGO demonstrate the usefulness of this approach.
7	Ensemble classifiers for steganalysis of digital media	2012	Jan Kodovsky, Jessica Fridrich, and Vojtech Holub.	BOSS	Ensemble classifiers have improved detection accuracy for steganographic methods in JPEG images. The proposed framework allows for fast construction of steganography detectors.
8	Proceedings of the 11th ACM Workshop on Multimedia and Security	2009	Jan Kodovsky and Jessica Fridrich.	NA	covert communication, Digital watermarking
9	Merging markov and dct features for multi-class jpeg steganalysis	2007	Tomas Pevny and Jessica J. Fridrich	BOSS	The paper constructs a new multi-class JPEG steganalyzer with improved performance. The new feature set provides significantly more reliable results compared to previous work.

S.N.	Name	Year	Authors	Dataset	Findings
10	Detection of image steganography using deep learning and ensemble classifiers	2022	Mikołaj Płachta, Marek Krzemien, Krzysztof Szczypiorski, and Artur Janicki	BOSS	Ensemble classifiers performed well in steganography detection. Deep learning algorithms achieved better results for UERD and nsF5 steganographic algorithms.
11	A markov process based approach to effective attacking jpeg steganography	2007	Yun Q. Shi, Chunhua Chen, and Wen Chen.	BOSS	The proposed steganalyzer outperforms by a significant margin. The detection rates are higher while considering the introduced features.
12	A fast and accurate steganalysis using ensemble classifiers	2013	Arezoo Torkaman and Reza Safabakhsh	BOSS	The proposed method achieved a lower error rate of 46% compared to the ensemble classifier. The training time of the proposed method was 88% lower than the ensemble classifier.
13	Istego100k Large-scale image steganalysis dataset	2019	Zhongliang Yang, Ke Wang, Sai Ma, Yongfeng Huang, Xiangui Kang, and Xianfeng Zhao	IStego100K	The paper introduces a large-scale image steganalysis dataset called IStego100K. The performance of some steganalysis algorithms on IStego100K is tested.

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 IStego100K dataset which contain 200,000 images. These images have been modified using three 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.

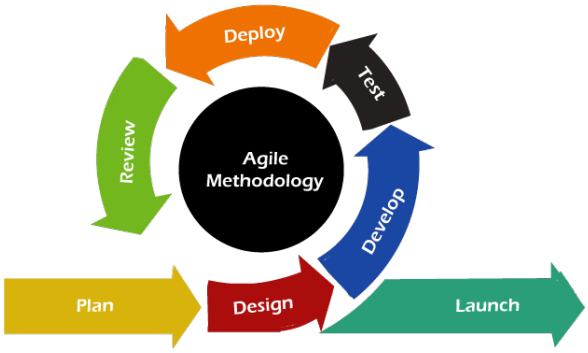
Operational Feasibility:
We have decided to use the Shallow ML approach which allows the model to be trained with less computational power in comparison to deep learning. For shallow machine learning we are planning to implement an ensemble classifier and each of its models will be trained using FLD 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 making it 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.



Source: https://www.javatpoint.com/agile-vs-waterfall-model

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 applications is reduced.

4.2 Data Collection

We will utilize the IStego100K dataset [20], a Large-scale open-source image steganalysis dataset. This dataset consits of total 200,000 images, serving as the primary resource for training and testing our model. Within the dataset, images are categorized into two main groups: "Stego" and "Cover," each containing 100,000 images. The "Stego" subset contains images with hidden information added using three different methods. In contrast, the "Cover" subset consists of images without any hidden information added; they are in their original, unaltered state.

4.3 Ensemble Classifiers

Ensemble classifiers are machine learning techniques that combine multiple individual models, or "base learners," to make predictions. Common ensemble techniques include bagging, boosting, and stacking. The main idea behind the ensemble methodology is to weigh several individual classifiers, and combine them in order to obtain a classifier that outperforms every one of them. A standard ensemble approach used in classification tasks consists of the following fundamental components:

1. **Training set:** A labeled dataset used for ensemble training. The training set can be described in a variety of languages. Most frequently, the instances are described as attribute- value vectors.

2. **Base Inducer:** The inducer is an induction algorithm that obtains a training set and forms a classifier that represents the generalized relationship between the input attributes and the target attribute. Decision trees, Neural networks, Support Vector Machines(SVMs), k-Nearest Neighbors(k-NN), Random Forsets, Gradient Boosting Machines(GBMs), (AdaBoost) etc. are different types of base learners that can be used in ensemble methods.

3. **Diversity Generator:** This component is responsible for generating the diverse classifiers.

4. **Combiner:** The combiner is responsible for combining the classifications of the various classifiers. Some of the widely used combination method are: Weighting method, Majority voting, Performance weighting, Distribution summation, Bayesian combinaiton, Dempster-Shafter, Vogging, Naive bayes. [21]

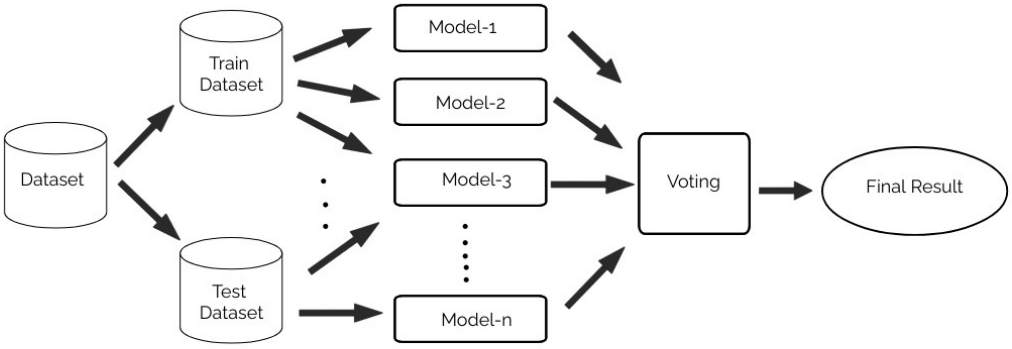


Figure 4.2: Basic outline of Ensemble Classifier

4.3.1 Algorithm

Algorithm 1 Ensemble Classifier Algorithm [15]

for $l = 1$ to L **do**
    Form a random subspace
    $D_l \subset \{1, \ldots, d\}, |D_l| = d_{\text{sub}} < d$
    Form a bootstrap sample N_1^b, $|N_1^b| = N^{\text{trn}}$ by uniform sampling with replacement from the set $\{1, \ldots, N^{\text{trn}}\}$
    Train a base learner B_l on features
    $X_l = \{x_m^{(D_l)}, \bar{x}_m^{(D_l)}\}_{m \in N_l^b}$
    $\rightarrow$ obtain eigenvector v_l and threshold T_l
end for
for all $y \in Y^{\text{tst}}$ **do**
    for $l = 1$ to L **do**
        Make l^{th} decision: $B_l(y^{D_l}) \triangleq \begin{cases} 1, & \text{when } v_l^T y^{(D_l)} > T_l \\ 0, & \text{otherwise} \end{cases}$
    end for
end for
Form the final decisions $B(y)$ by majority voting:
$B(y) = \begin{cases} 1, & \text{when } \sum_{l=1}^{L} B_l(y^{(D_l)}) > L/2 \\ 0, & \text{when } \sum_{l=1}^{L} B_l(y^{(D_l)}) < L/2 \end{cases}$
return $B(y), y \in Y^{\text{tst}}$

In the provided algorithm:

d: Represents the dimensionality of the feature space.
d_{sub}: Represents the dimensionality of the feature subset.
N^{TRN} and N^{TST}: Denote the number of training and testing examples, respectively.
L: Represents the number of base learners.
$x_m, \bar{x}_m \in \mathbb{R}^d, m = 1, ..., N^{\text{TRN}}$: Refer to the cover and stego features computed from the training set.
$y_k, \bar{y}_k \in \mathbb{R}^d, k = 1, ..., N^{\text{TST}}$: Denote the cover and stego features computed from the testing set.

4.4 Implementation

We start by creating high-dimensional prefeatures that captures various dependencies of the cover elements. We aim to capture as many connections among these elements as we can. For that we will be using CC-C300 model [14] for prefeature extraction. Now, these prefeatures are feeded into an Ensemble based classifier with Fisher Liner Discrimants(FLDs) as the base learners. Random subspace is creating using Bootstrapping and they are given as input to individual base learners. To finalze the final result majority voting is done and final ouput is received. The model is trained several times on IStego100K dataset so that it can predit stego images with high accuracy.

4.5 Block diagram of proposed system

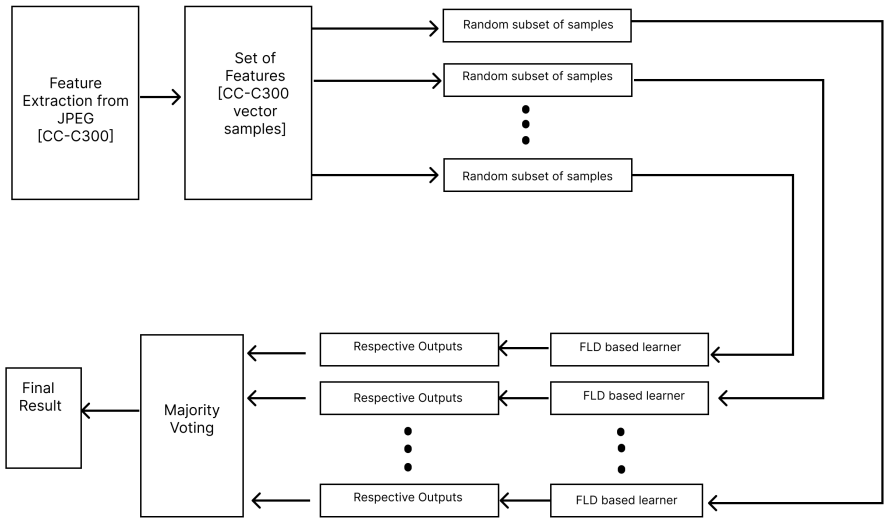


Figure 4.3: Block diagram of proposed system

Model Training Approach

Feature Extraction:
Initial extraction of CC-C300 feature vector JPEG image is to be done. The selection of CC-C300 is based on its high dimensionality and its detection efficiency.

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 CC-C300 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 CC-C300 as a feature is intentional to increase efficiency and detection capability of the system.

4.6 System Architecture

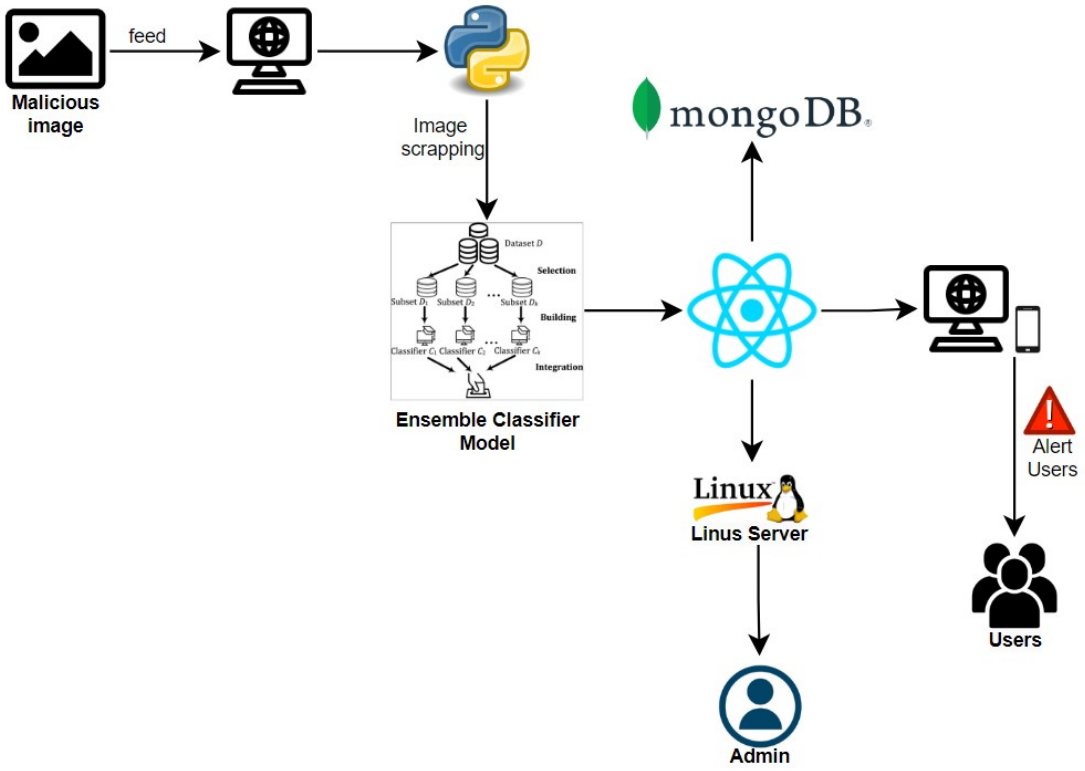


Figure 4.4: System Architecture

Chapter 5

Implementation Plan

5.1 Gantt Chart

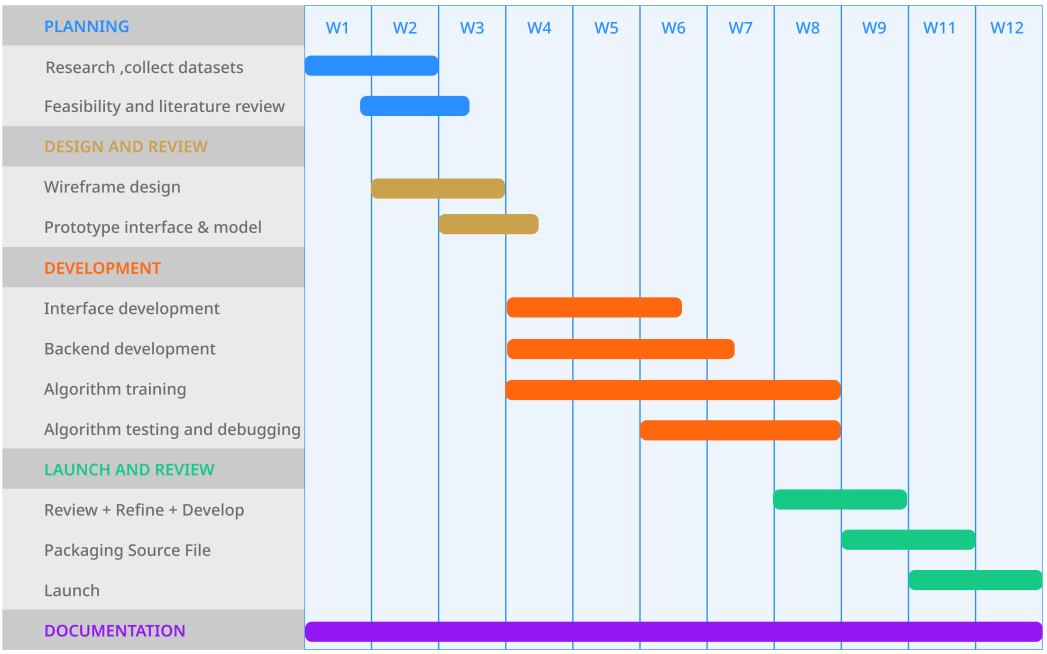


Figure 5.1: Gantt Chart

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 schema-less 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.
- **GitHub:** GitHub is a web-based platform for version control using Git. It provides a collaborative environment for software development projects, allowing developers to host and share their code, track changes, manage workflows, and collaborate with others. GitHub offers features such as code repositories, issue tracking, project management tools, code review, and continuous integration.
- **MATLAB:** MATLAB, short for "Matrix Laboratory," is a high-level programming language and interactive environment primarily used for numerical computation, visualization, and programming. It provides a wide range of built-in functions and toolboxes for various applications, including mathematics, signal processing, image processing, control systems, and machine learning.
- **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 steganographically modified images using ML model. It would be capable of decting hidden information with high accuracy. It is expected to be able to identify specificity of steganalysis using shallow Machine Learning.

Chapter 7

Works Completed

The initially estimated tasks which were presumed to be completed have been completed successfully. According to the Gantt chart, we have successfully completed the work specified such as:

1. **Dataset Preprocessing:**
   We have effectively preprocessed datasets from the BOSS dataset and IStego100K dataset. This involved primarily focusing on JPEG files, as the datasets were predominantly in the .pgm file type. The preprocessing stage was essential to extract the desired features for our analysis.

2. **Feature Extraction:**
   For our analysis, the CC-CN features were determined to be the most effective for detecting steganographically modified images. Given the large number of co-occurrence matrices, we decided to focus on the top 300 features (CC-C300). We successfully extracted these features from a dataset comprising 100,000 images.

3. **Model Training and Accuracy:**
   We have trained a model using the extracted CC-C300 features, achieving an OOB error of 0.2899 for "universal steganographic detection". It is worth noting that this model's accuracy is expected to increase when applied to datasets with specific steganography algorithms.

4. **Comparative Analysis:**
   Our analysis included training the model using different features. Among these, CC-C300 consistently outperformed other features, further validating our choice.

Chapter 8

Work In Progress

Our ongoing tasks are listed below:

1. **Parameter Tuning and Optimization:**
   We are currently focused on tuning the parameters of our training process to achieve better results. Additionally, our aim is to optimize our processes to expedite result prediction. This involves a thorough examination of our existing methodologies and fine-tuning them for improved performance.

2. **Dataset Combination:**
   Our datasets have been divided into sections of 10,000 each. We are in the process of combining these datasets to provide more accurate prediction accuracy. This step is crucial as it ensures that our model is trained on a comprehensive dataset, thereby enhancing its predictive capabilities.

3. **Code Integration and Website Development:**
   The code we are using is predominantly in MATLAB. However, we are currently in the process of deciding whether to integrate this code with the website side of our project. This decision will impact the user interface and functionality of our project and is therefore being carefully considered.

4. **Ongoing Collaboration and Communication:**
   We continue to engage in weekly discussions with team members to provide updates and share progress. Additionally, frequent meetings with our supervisor are being held to ensure that we are on the right track and to explore any potential areas for improvement. This collaborative approach is essential for achieving optimum results and refining our working methods.

Chapter 9

Bibliography

[1] Tayana Morkel, Jan HP Eloff, and Martin S Olivier. An overview of image steganography. In *ISSA*, volume 1, pages 1–11, 2005.

[2] Arooj Nissar and Ajaz Hussain Mir. Classification of steganalysis techniques: A study. *Digital Signal Processing*, 20(6):1758–1770, 2010.

[3] Syed Ali Khayam. The discrete cosine transform (dct): theory and application. *Michigan State University*, 114(1):31, 2003.

[4] Osama F AbdelWahab, Aziza I Hussein, Hesham FA Hamed, Hamdy M Kelash, Ashraf AM Khalaf, and Hanafy M Ali. Hiding data in images using steganography techniques with compression algorithms. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 17(3):1168–1175, 2019.

[5] Jessica Fridrich, Miroslav Goljan, and Dorin Hogea. Steganalysis of jpeg images: Breaking the f5 algorithm. In *Information Hiding: 5th International Workshop, IH 2002 Noordwijkerhout, The Netherlands, October 7-9, 2002 Revised Papers 5*, pages 310–323. Springer, 2003.

[6] Jessica Fridrich, Miroslav Goljan, and David Soukal. Efficient wet paper codes. In *International Workshop on Information Hiding*, pages 204–218. Springer, 2005.

[7] Jessica Fridrich, Petr Lisoněk, and David Soukal. On steganographic embedding efficiency. In *Information Hiding: 8th International Workshop, IH 2006, Alexandria, VA, USA, July 10-12, 2006. Revised Selcted Papers 8*, pages 282–296. Springer, 2007.

[8] NV Koshkina. J-uniward steganoanalysis. *Cybernetics and Systems Analysis*, 57(3):501–508, 2021.

[9] Junjun Gan, Jiufen Liu, Xiangyang Luo, Chunfang Yang, and Fenlin Liu. Reliable steganalysis of hugo steganography based on partially known plaintext. *Multimedia Tools and Applications*, 77(14):18007–18027, 2018.

[10] Xiangyang Luo, Xiaofeng Song, Xiaolong Li, Weiming Zhang, Jicang Lu, Chunfang Yang, and Fenlin Liu. Steganalysis of hugo steganography based on parameter recognition of syndrome-trellis-codes. *Multimedia Tools and Applications*, 75:13557–13583, 2016.

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

[12] 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.

[13] 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.

[14] Jan Kodovsky and Jessica Fridrich. Steganalysis in high dimensions: Fusing classifiers built on random subspaces. *Proceedings of SPIE - The International Society for Optical Engineering*, 02 2011.

[15] Jan Kodovsky, Jessica Fridrich, and Vojtech Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, 7, 04 2012.

[16] Arezoo Torkaman and Reza Safabakhsh. A fast and accurate steganalysis using ensemble classifiers. In *2013 8th Iranian Conference on Machine Vision and Image Processing (MVIP)*, pages 22–26, 2013.

[17] Jan Kodovský and Jessica Fridrich. Calibration revisited. In *Proceedings of the 11th ACM Workshop on Multimedia and Security*, MM&Sec '09, page 63–74, New York, NY, USA, 2009. Association for Computing Machinery.

[18] Yun Q. Shi, Chunhua Chen, and Wen Chen. A markov process based approach to effective attacking jpeg steganography. In Jan L. Camenisch, Christian S. Collberg, Neil F. Johnson, and Phil Sallee, editors, *Information Hiding*, pages 249–264, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[19] Tomáš Pevný and Jessica J. Fridrich. Merging markov and dct features for multi-class jpeg steganalysis. In *Electronic imaging*, 2007.

[20] Zhongliang Yang, Ke Wang, Sai Ma, Yongfeng Huang, Xiangui Kang, and Xianfeng Zhao. Istego100k: Large-scale image steganalysis dataset. In *International Workshop on Digital Watermarking*. Springer, 2019.

[21] Lior Rokach. Ensemble-based classifiers. *Artificial intelligence review*, 33:1–39, 2010.