الجمهورية الجزائرية الديمقراطية الشعبية
people's Democratic Republic of Algeria

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

DR.YAHIA FARESS UNIVERSITY OF MEDEA

FACULTY OF TECHNOLOGY

Electrical Engineering Department



Thesis of Master

Presented to obtain the master's degree
In: Telecommunication
Option : Telecommunication systems

THEME

Steganalysis of the RLPFS method using support vector machine.

Supervised by :

Dr. KHEDDAR Hamza

Presented by :

MADANI Amani Hanaa

GHERIBI Faitma zahra

Year: 2020

Acknowledgment

First, we would like to thank God for his merciful and his continuous help over all the period of our lives.

We would like to express our great appreciation to our supervisor Mr.Kheddar Hamza for his countless help, constructive suggestion and guidance in every step of our research.

We also wish to express our deepest gratitude to the members of the committee for spending their precious time on reading our thesis and giving us encouragement and constructive comments.

We would like to thank all the teachers who collaborated in our training since the beginning of our study, we also thank all Technology Faculty members at Yahia Fares Medea University.

Finally, and most importantly, we must thank our dearest family for giving us the support and encouragement that only a family can give.

Dedication

Every challenging work needs self efforts as well as guidance of elders
especially those who were very close to my heart.
I dedecate my humble effort to my sweet and loving

Father and Mother

Whose affection, love, encouragement and prays of day and night
make me able to get such success and honor,
Along with all my family, friends
And all hard and working and respected teachers.

Amani Hanaa

Dedication

First and foremost, I have to thank my parents for their love and support throughout my life. Thank you both for giving me strength to reach for the stars and chase my dreams.

To all my family members

Sisters and Brothers

For all my friends

Thank you all for being there for me

Fatima zohra

ملخص

تُعد تقنيات إخفاء المعلومات (الستيغانوغرافيا) وتحليل الإخفاء من تقنيات معالجة الإشارة المدمجة في خدمات أمن المعلومات، التي حظيت حديثا باهتمام وافر، وكما هو الحال بالنسبة لنظيراتها من التقنيات في التشفير، فهما في صراع دائم، إذ تسعى الستيغانوغرافيا لحجب الرسائل السرية داخل وسائط رقمية في حين تحاول الطرق المنتهجة في تحليل الإخفاء بكسر خوارزميات إخفاء المعلومات والكشف عن وجود الرسائل المحجوبة. تهدف الدراسة الحالية لتحليل اداء التقنية الستيغانوغرافية الحديثة المسماة *RLPFS* باستعمال الة المتجهات الداعمة *SVM*.

الكلمات المفتاحية

علم إخفاء المعلومات، تحليل الإخفاء، الة المتجهات الداعمة.

Abstract

Recently, merging signal processing techniques with information security services has drawn a lot of attention. Steganography and steganalysis are among those trends. Like their counterparts in cryptology, steganography and steganalysis are in a constant battle. Steganography methods try to hide the presence of covert messages in innocuous-looking data, whereas steganalysis methods try to break steganography algorithms and reveal the existence of hidden messages. The aim of our project is to analysis the performance of new steganographic technique called random least significant bits of pitch and fourier magnitude steganography (RLPFS) using support vector machin (SVM).

Keywords

Steganalysis, steganography, MFCC, SVM.

Résume

Dernièrement, l’intégration des techniques de traitement du signal dans les services de sécurité de l’information a largement attiré l’attention des chercheurs, dont la stéganographie et la stéganalyse font partie de ces tendances. Tout comme leurs homologues de la cryptologie, la stéganographie et la stéganalyse sont en lutte constante. Les méthodes de stéganographie tentent de masquer la présence de messages secrets dans des données d'apparence inoffensive, tandis que les méthodes de stéganalyse tentent de briser les algorithmes de stéganographie et de révéler l'existence de messages cachés. L’objectif de notre projet consiste à analyser les performances de la nouvelle technique stéganographique *RLPFS* en utilisant SVM.

Mots clés

Steganalyse , steganographie, MFCC, SVM.

Table of Contents

Table of contents I
List of figures V
List of tables VI
Acronyms VII
Introduction 1
Chapter I : State-of-the-art of speech steganalysis
Introduction 4
I.1 Steganography 4
I.1.1 Definition 4
I.1.2 The components of steganographic system 4
I.1.3 Steganographic techniques 5
I.1.3.1 Text steganography 5
I.1.3.2 Image steganography 5
I.1.3.3 Audio steganography 5
I.1.3.4 Video steganography 5
I.1.4 Steganograpy categories 5
I.1.4.1 Pure steganography 5
I.1.4.2 Secret key steganography 6
I.1.4.3 Public key Steganography 6
I.1.5 Characteristics of steganograhic system 6
I.1.5.1 Secrecy 6
I.1.5.2 Imperceptibility 6
I.1.5.3 Capacity 6
I.1.6 Methods of hiding 6
I.1.6.1 Injection 7
I.1.6.2 Substitution 7
I.1.6.3 Generation 7
I.1.7 Audio Steganography Algorithms 7
I.1.7.1 Low-bit Encoding 8
I.1.7.2 Phase Coding 8
I.1.7.3 Spread Spec trum Coding 8
I.1.7.4 Echo Hiding 8

I.1.7.5 Hiding in Silence Interval 9
I.1.7.6 Amplitude Coding 9
I.1.7.7 Discrete Wave Transform 9
I.2 Steganalysis 9
I.2.1 Definition 9
I.2.2 Types of steganalytic warden 10
I.2.2.1 Passive warden 10
I.2.2.2 Active warden 10
I.2.2.3 Malicious warden 10
I.2.3 Steganalysis techniques classification 10
I.2.3.1 Universal methods 11
I.2.3.2 Targeted methods 12
I.2.4 Audio steganalysis algorithms 13
I.2.4.1 Phase and echo steganalysis 13
I.2.4.2 Universal steganalysis based on recorded speech 14
I.2.4.3 Use of statistical distance measures for audio steganalysis 14
I.2.4.4 Audio steganalysis based on Hausdorff distance 14
I.2.4.5 Audio steganalysis for high complexity audio signals 15
I.2.5 Attacks used by the steganalyst 15
I.2.5.1 Stego-only attack 15
I.2.5.2 Known cover attack 15
I.2.5.3 Known message attack 15
I.2.5.4 Chosen stego attack 15
I.2.5.5 Chosen message attack 16
I.2.5.6 Known stego attack 16
I.2.6 Data quality 16
I.2.6.1 Accuracy 16
I.2.6.2 Consistency 16
I.2.6.3 Minimize false-positives 17
Conclusion 17
Chapter II : Theory of classification using support vector machine (SVM)
Introduction 19
II.1 Machine learning 19
II.1.1 Types of machine learning algorithms 19
II.1.1.1 Supervised learning 20

II.1.1.2 Unsupervised learning 20
II.1.2 Machine learning models 21
II.1.2.1 Logistic regression 21
II.1.2.2 Linear regression 22
II.1.2.3 K-nearst neighbor 23
II.1.2.4 Decision trees and random forests 24
II.1.2.5 K-Means clustering 24
II.1.2.6 Artificial neural networks 25
II.2 Support vector machine classification 25
II.2.1 Basic understanding to support vector machine 26
II.2.1.1 Feature extracting phase 27
II.2.1.2 Training phase 27
II.2.1.3 Testing phase 27
II.2.2 Support vector and marging in SVM 27
II.2.2.1 Linear SVM 27
II.2.2.2 Non-linearly SVM 29
II.2.3 Support vector machine classifiers 29
II.2.3.1 Maximal margin classifier 29
II.2.3.2 Soft margin classifier 30
II.2.3.3 Kernel support vector machine 31
II.3 Applications of SVM 32
II.4 Advantages and disadvantages of SVM 32
Conclusion 32

Chapter III: Steganalysis of the RLPFS method using SVM classifier and the MFCC features

Introduction 34
III.1 MELP coder 34
III.2 RLPFS method 34
III.3 Feature selection and extraction 37
III.3.1 Mel frequency cepstral coefficients 37
III.3.1.1 Framing 38
III.3.1.2 Windowing 38
III.3.1.3 Fast Fourier Transform 38
III.3.1.4 Mel spectrum 38
III.3.1.5 Mel filter bank 38

III.3.1.6 Discrete cosine transform 39
III.3.2 Deltas 39
III.4 Classifier training and testing 40
III.4.1 Training 40
III.4.2 Testing 41
III.4.3 Functions used for training and classification 41
III.5 Classifier 42
III.5.1 Kernel properties 42
III.6 The proposed model 42
III.7 Evaluation metrics 42
III.8 Results and discussion 43
III.8.1 Experimental work 44
III.8.1.1 Training and testing steps 44
III.8.1.2 Results and discussion 45
III.8.2 ANN and SVM comparison 53
Conclusion 54
General conclusion 55
References 56

List of figures

Figure I.1: a generic steganography system 4
Figure I.2: steganalysis techniques classification 11
Figure II.1: machine learning types 19
Figure II.2: logistic regression 22
Figure II.3: evaluation in linear regression 22
Figure II.4: based dynamic KNN 23
Figure II.5: standard algorithm of k-means clustering 24
Figure II.6: artificial neural network architecture 25
Figure II.7: support vector machine (SVM) 26
Figure II.8: SVM process overview 27
Figure II.9: linear SVM. (a) for linearly separable data. (b) for linearly non-separable data .. 28
Figure II.10: Non-linear SVM classification 29
Figure II.11: (a) maximal margin classifier, (b) soft margin classifier 31
Figure III.1: general scheme of hiding secret speech 35
Figure III.2: block diagram for MFCC feature extraction 37
Figure III.3: triangular filter bank 39
Figure III.4: generic diagram of the classifier training stage 40
Figure III.5: overview of the classification process of a test sample 41
Figure III.6: stages of the experimental work 45

List of tables

Table II.1: advantages and disadvantages of svm 32
Table III.1: confusion matrix for binary classification 43
Table III.2: confusion matrix results of linear kernel 45
Table III.3: accuracies of RLPFS steganalysis in linear kernel 46
Table III.4: confusion matrix results of rbf kernel 46
Table III.5: accuracies of RLPFS steganalysis in rbf kernel 47
Table III.6: confusion matrix results of mlp kernel 47
Table III.7: accuracies of RLPFS steganalysis in mlp kernel 48
Table III.8: confusion matrix results of quadratic kernel 48
Table III.9: accuracies of RLPFS steganalysis in quadratic kernel 49
Table III.10: confusion matrix results using delta mfcc in linear kernel 49
Table III.11: accuracies of RLPFS steganalysis with delta mfcc in linear kernel 50
Table III.12: confusion matrix results using delta mfcc in rbf kernel 50
Table III.13: accuracies of RLPFS steganalysis using delta mfcc in rbf kernel 51
Table III.14: confusion matrix results using delta mfcc in mlp kernel 51
Table III.15: accuracies of RLPFS steganalysis using delta mfcc in mlp kernel 52
Table III.16: confusion matrix results using delta mfcc in quadratic kernel 52
Table III.17: accuracies of RLPFS steganalysis using delta mfcc in quadratic kernel 53
Table III.18: Accuracies of the s method using ANN 53

ACRONYMS

AI	Artificial intelligence
ANN	Articial neaul network
ANOVA	Analyse Of Variance
AQM	Audio quality metrics
AVI	Audio video interleave
AR-TDNN	Autoregressive time delay neural network
AU	Audio
CELP	Certificate of language proficiency
CIAQM	Constant referencing
DCT	Discrete cosinus transform
EOF	End-of-file marker
FBM	Format-Based Method
FB	Forward–backward
FFT	Fast Fourier Transform
GMM	Gaussian mixture model
GGD	Generalised Gaussian distribution
GPC	Generalized predictive control
HAS	Human auditory system
IDS	Intrusion detection systems
KNN	K-nearst neighbor
LSB	Least significant bit
LPC	Linear Predictive Coding
MELP	Mixed-excitation linear predictive
MFCC	Mel-frequency cepstrum coefficients
MN	Mode number
MPEG	Moving picture expert group
MP3	Mpeg audio layer III
MSVQ	Multi-stage vector quantiser
NEB	Number of embedded bits
PCA	Principal components analysis
PDF	Portable document format

PN	Positive-negative
RBF	Radial basic function
RLPFS	Random least significant bits of pitch and fourier magnitude steganography
R-LSB	Random least significant bit
RSM	Random and Statistical Method
RM	Real media
SFS	Sequential floating search
SPNR	Signal-to-perceived noise ratio
SS	Spread spectrum
STFT	Short time fourier transform
SVM	Support vector machine
VoIP	Voice over internet protocol
WAV	Waveform audio file format

Introduction

The development of digital media and communication networks has facilitated the sharing and transfer of digital data, thus introducing new forms of document piracy and new security challenges. In addition, the problem of protection of the content of a digital multimedia medium does not yet have solutions satisfactory. It has become easy to modify or reproduce media and even claim its exploitation rights.

Information privacy has always been an issue that concerns everyone. In order to reduce the copying of multimedia works, and ensure the confidentiality of a transmission, new methods have been developed. These are the methods of concealment of information (data hiding).

The art of hiding information without arousing suspicion is called steganography. As a technology of covert communication, steganography can embed secret to be the open cover in public communication mode. It is not only hiding the information contents but also concealing the existence of this behavior. The intention of steganography is to provide the secret transmission of data. Hence, it is difficult for the third person to realize about the existence of the hidden message in the cover media, audio steganography is particular important resulted from the prevailing presence of audio or speech signals in our human society. There are techniques allowing to discover the steganography media, this is the case of steganalysis which also called steganographic analysis.

Steganalysis attempts to discover hidden signals in suspected carriers or at the least detect which media contain hidden signals. Therefore, an important consideration in steganography is how robust to detection is a particular technique.

Audio steganalysis techniques are actively investigated within the last decade. This interest is attributed to the growing number of steganography algorithms and the threats they represent. In practice, the work of a steganalyst relies on finding any unnatural modification resulting from the embedding process that may exist within the suspected audio file. To date, there's no known steganographic system that hides data in a perfect secure and undetectable way. All embedding algorithms leave a fingerprint in the stego-speech unless a really small embedding capacity is achieved. Although some research works had managed to reliably detect the presence of hidden data in audio signals. To this point all of them depend on the change of the intrinsic properties (features) of the audio signals to differentiate between stego-speech and cover-speech signals. Once the features are extracted, most of the steganlysis methods apply a learning process to distinguish between the cover- and the stego-audio signals.

The learning process is done by training a machine learning on a dataset fed with statistical properties (features) extracted from the cover and stego-audio signals. The right choice of these features reinforces the discriminatory power between cover- and stego-audio signals.

We review the existing steganography and steganalysis techniques and discuss their limitations and some possible research directions. This work provides the following chapters:

Chapter 1 includes literature review of both steganography and steganalysis in voice over IP (VoIP).

Chapter 2 describes the machine learning process and the theory of classification.

Chapter 3 details the experimental setup, which includes dataset and embedding method used. We also describe the proposed steganalysis method, including feature extraction and classification process with results analysis and discussion. Finally, we finish our work by a conclusion.

Chapter one

State-of-the-art of speech steganalysis

Introduction

As many techniques used from the ancient period up to the recent times failed to establish secret data communication, Cryptography and steganography came into light and blow the lid off. They have now become the unbeatable mode of communication of a secret information from source to destination. Cryptography initially eased the problem of one to one communication whereas steganography is to make peer to peer communication in a more sophisticated way.

I.1 Steganography

I.1.1 Definition

The word steganography is a composite of the Greek words steganos, which means "covered", and graphia, which means "writing". In other words, steganography is the art of concealed communication where the very existence of a message is secret [1]. In Steganography the confidential information (*secret message*) is hidden into a *cover* file in order to produce a secret cover media known as *stego* media. So, the secret information is not recognized or recovered by the attackers. High level of security and secrecy must be provided by the steganography.

I.1.2 The components of steganographic system

In steganography, the possible cover carriers are innocent looking carriers (images, audio, video, text, or some other digitally representative code) which will hold the hidden information. A message is the information hidden and may be plaintext, cipher text, images, or anything that can be embedded into a bit stream. Together the cover carrier and the embedded message create a stego-carrier. Hiding information may require a stego key which is additional secret information, such as a password, required for embedding the information. For example, when a secret message is hidden within a cover audio, the resulting product is a stego-audio [2].

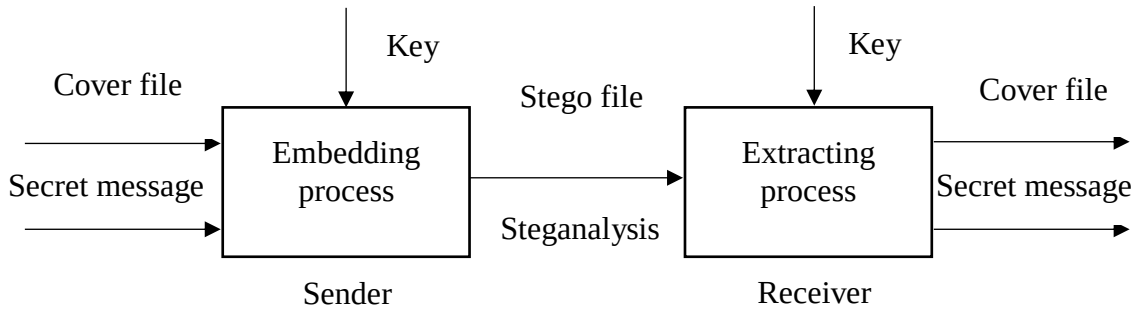


Figure I.1: a generic steganography system [2].

I.1.3 Steganographic Techniques

I.1.3.1 Text steganography

Inside the text files, it comprises of masking data. Behind every nth letter of every word of the text message, the confidential information is concealed. There are some methods that are used to conceal information in the text file and defined as fellow: Format-Based Method (FBM), Random and Statistical Method (RSM), Linguistics Method [3].

I.1.3.2 Image steganography

Image steganography is to as concealing the information by taking the secret image and cover image and inside the cover image the secret image is embedded. To conceal the information inside the cover object the pixel intensities of the secret image is used. In digital steganography, the cover source is used by images because a number of bits are present in a digital representation of an image [3].

I.1.3.3 Audio steganography

As the name implies, audio file stores the concealed information. For concealing information, it uses sound files like waveform audio file format (WAV), audio (AU), mpeg audio layer III (MP3), etc. Audio steganography consists of several methods. Some of methods are: least significant bit (LSB) ,phase coding ,spread spectrum (SS), echo hiding [3].

I.1.3.4 Video steganography

In order to conceal information, this technique uses a digital video format. For concealing the information carrier is used which is a combination of pictures (case video). Discrete cosinus transform (DCT) alter the values which are used to conceal information in each of the images in the video and the human eye cannot recognize it. Video steganography formats are H.264, mp4, moving picture expert group (MPEG), and audio video interleave (AVI) [3].

I.1.4 Steganograpy categories

There are basically three types of steganographic protocols used [4]:

I.1.4.1 Pure steganography

It is defined as a steganographic system which does not require the exchange of a cipher such as a stego-key. This method of is not much secured because the sender and receiver can rely only upon the assumption that no other parties are aware of the secret message.

I.1.4.2 Secret key steganography

It is defined as a steganographic system that requires the exchange of a secret key (stego-key) previous to communication. Secret key steganography takes a cover message and embeds the secret message inside of it by using a secret key (stego-key). Only the parties who know the secret key can reverse the process and read the secret message. Unlike pure steganography where a perceived invisible communication channel is present, secret key steganography exchanges a stego-key, which makes it more vulnerable to interception. The benefit to secret key steganography is even if it is intercepted, only parties who know the secret key can extract the secret message.

I.1.4.3 Public key Steganography

It is defined as a steganographic system that uses a public key and a private key to secure the communication between the parties wanting to communicate secretly. Sender will use the public key during the encoding process and only the private key, which has a direct mathematical relationship with the public key, can decipher the secret message.

I.1.5 Characteristics of steganograhic system

An effective steganographic scheme should satisfy the following desired evaluation criteria [5]:

I.1.5.1 Secrecy

A person should not be able to extract the hidden data from the host medium without the knowledge of the proper secret key used in the extracting procedure.

I.1.5.2 Imperceptibility

The medium after being embedded with the hidden data should be indiscernible from the original medium. One should not become suspicious of the existence of the covert data within the medium.

I.1.5.3 Capacity

The maximum length of the covert message that can be embedded should be as long as possible ($|cover| > |secret|$).

I.1.6 Methods of hiding

Fundamentally, audio steganography is the art and science of hiding digital data such as text messages, documents, and binary files into audio files such as WAV, MP3, and real media (RM) files. The output audio file is called the carrier file and is the only intermediate to be sent to the receiver. Characteristically, anyone who is taping the communication wire would not

notice anything suspicious being transmitted, except for a normal audio file [6]. This property of steganography is called imperceptibility and it refers to the fact that no one apart from the original sender and the intended receiver can suspect the presence of secret data into the carrier file being communicated. Steganography can be achieved by means of three types of techniques: injection, substitution, and generation.

I.1.6.1 Injection

The injection technique implants the data to hide in the insignificant part of the carrier file, which is normally ignored by operating systems and software applications. For example, most computer files comprise what so called an end-of-file marker (EOF), which indicates that no more data can be read from a data source. Likewise, an executable file usually ends with an EOF marking the end of binary instructions. Another example is the portable document format (PDF) file which ends with an EOF indicating to the reader application that no more pages are to be fetched and that the file has ended. Steganography by injection exploits the EOF section and injects secret data after the EOF marker which eventually has no side effect on the carrier file and is often disregarded by the execution environment.

I.1.6.2 Substitution

The substitution technique substitutes the insignificant bits in the carrier file with the bits of the data to hide. Insignificant bits are those bits that can be modified without damaging the quality or destroying the integrity of the carrier file. For example, in audio files, every unit of sound is made up of a sequence of bits. If the least significant bit of this sequence is modified, its impact is minimal on the perceptible sound so much so that the human ear cannot tell the difference between the original version and the altered one. This technique takes advantage of the limited capabilities of the human auditory system (HAS) which cannot recognize two sounds that are slightly not alike.

I.1.6.3 Generation

The generation technique reads the data to be hidden and generates out of them a new set of data. It is a dynamic method of creating a carrier file based on the information contained in the data to hide. For example, one generation technique will take the message to hide and turn its characters into matching audio frequencies that can ultimately make up an audio file.

I.1.7 Audio Steganography Algorithms

In this section, we first describe the major audio steganography algorithms: The disadvantages associated with these algorithms can be exploited for steganalysis [7].

I.1.7.1 Low-bit Encoding

In Low-bit encoding (like the method in [6]) , the binary version of the secret data message is substituted with the LSB of each sample of the audio cover file. Though this method is simple and can be used to embed larger messages, the method cannot protect the hidden message from small modifications that can arise as a result of format conversion or lossy compression.

I.1.7.2 Phase Coding

Phase coding is based on the fact that the phase components of sound are not as perceptible to the human ear as noise is. Message bits are encoded as phase shifts in the phase spectrum of a digital signal. This leads to inaudible encoding in terms of the signal-to-perceived noise ratio (SPNR) and the secret message gets camouflaged in the audio signal, not detectable by the steganalysis methods based on SPNR. Thus, phase coding addresses the disadvantages of the noise-inducing methods of audio steganography [8].

I.1.7.3 Spread Spectrum Coding

The basic SS coding method (like the method in [9]) randomly spreads the bits of the secret data message across the frequency spectrum of the audio signal. However, unlike LSB coding, the SS coding method spreads the secret message using a code that is independent of the actual cover signal. The SS coding method can perform better than LSB coding and phase coding techniques by virtue of a moderate data transmission rate coupled with a high level of robustness against steganalysis techniques. However, like the LSB coding method, the SS method can introduce noise to the audio file. This vulnerability can be tapped for steganalysis.

I.1.7.4 Echo Hiding

With echo hidng (like the method in [10]), information is embedded by introducing an echo into the discrete audio signal. Like SS coding, echo hiding allows for a higher data transmission rate and provides superior robustness when compared to the noise-inducing methods. To successfully hide the data, three parameters of the echo need to be altered: amplitude, decay rate and offset (delay time) from the original signal. The echo is not easily resolved as all the three parameters are set below the human audible threshold limit. Also, the offset is altered to represent the binary message to be hidden. The first offset value represents a one (binary), and the second offset value represents a zero (binary).

I.1.7.5 Hiding in Silence Interval

This technique inserts a silence interval in the original audio signal to embed the secret data. The values that represent the length of the silence intervals are decreased by some value such as $0 < value < 2^n$ where n is the number of bits required to represent an element from the data to hide. The carrier audio file is then sent to the receiver having the new lengths for its silence intervals [11].

I.1.7.6 Amplitude Coding

This technique conceals secret data in the magnitude speech spectrum while not distorting the carrier audio signal. It is based on searching for safe spectral regions in the signal whose magnitude speech spectrum is below a certain value. Besides, the carrier locations are selected based on how much they can badly affect the audio signal.

I.1.7.7 Discrete Wave Transform

In this technique secret data are embedded in the least significant bits of the wavelet coefficients of the audio signals. Often, secret data are chosen to be hidden in the integer wavelet coefficients and not in silent sections of the audio signal so as to promote the imperceptibility of the audio file. The disadvantage of DCT is that secret data can be lost during the recovering process as this technique is not that accurate.

The objective of steganography is to enable convert communication by hiding data in digital covers such as images, audios and videos, etc. Various steganography methods and software have been widely applied. Correspondingly, steganalysis techniques are developed to detect the existence of hidden information [12]. steganalysis aims to identify if a carrier (image, text, audio or video) has been manipulated by embedding a secret message using some embedding technique. Since there are different possible carrier types, and many different embedding techniques, designing a reliable universal steganalyzer is extremely challenging [13].

I.2 Steganalysis

I.2.1 Definition

Steganalysis is the discovery of the existence of hidden information, therefore, like cryptography and cryptanalysis, the goal of steganalysis is to discover hidden information and to break the security of its carriers [14].

Steganalysis is the countermeasure of the warden against such hidden messages and at the same time, it can also help with improving the security of the existing steganography methods [15].

I.2.2 Types of steganalytic warden

There exist three types of warden which are [16]:

I.2.2.1 Passive warden

It does not interfere with the content on the transmission channel. The steganalysis system goal in this case is to detect the mere presence of the secret communication between A and. The warden in this situation does not have the ability to destroy or modify the secret message that is detected by the steganalysis algorithm.

I.2.2.2 Active warden

The objective of the active warden is to detect and slightly modify the communicated objects and then send virtually any content to the receiver side.

I.2.2.3 Malicious warden

The warden in this situation has the information about the key. The secret message can be modified in order to impersonate the sender and trick the receiver.

I.2.3 Steganalysis techniques classification

The existing steganalysis techniques are classified under two categories, specific and universal steganalysis. The specific steganalysis techniques are designed for targeted embedding techniques. Thus, they yield very accurate decisions when they are used against the particular steganographic technique. In universal steganalyzers, dependency on the behavior of the individual embedding techniques is removed by determining and collecting a set of distinguishing statistics that are sensitive to a wide variety of embedding operation. Figure I.2 summarizes the existing steganalysis methods.

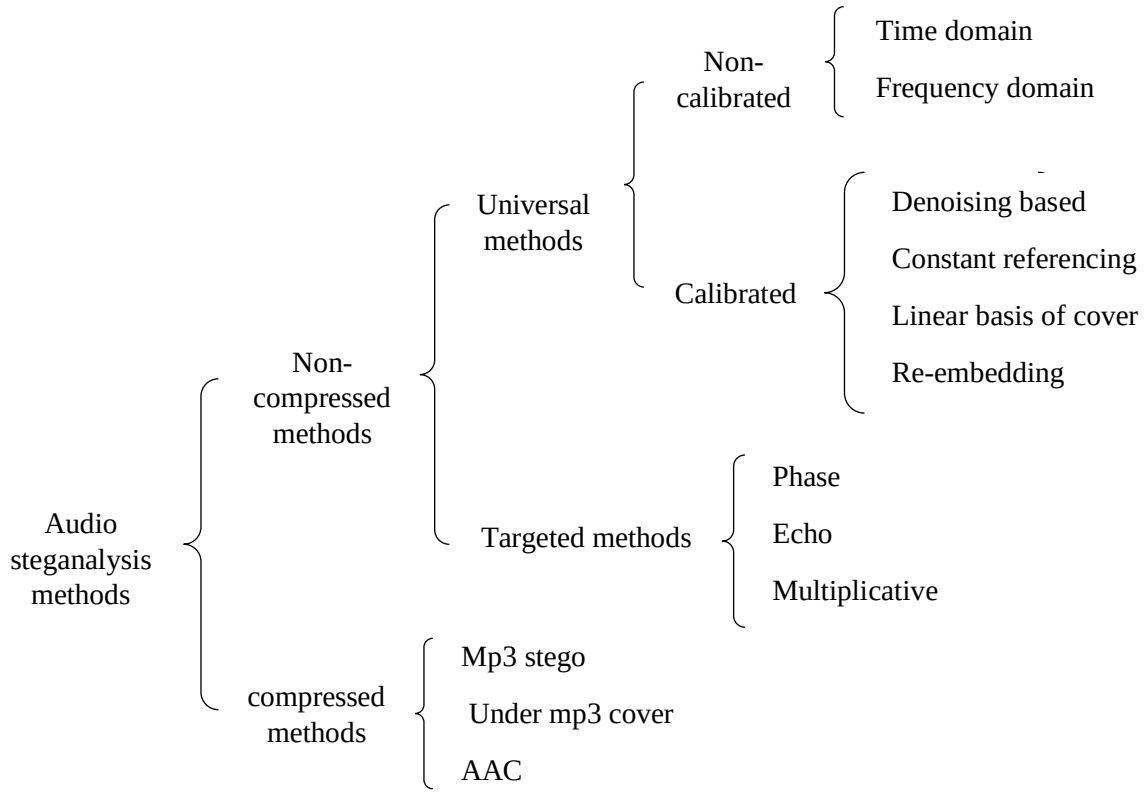


Figure I.2: steganalysis techniques classification [17].

I.2.3.1 Universal methods

Universal audio steganalyses are broadly divided into non calibrated and calibrated methods. In the non-calibrated case, steganalysis features are directly extracted from the audio signal. On the other hand, in the calibrated case, steganalysis features are extracted by comparing the audio signal with its estimated cover or stego signal [17].

∗ ***Non-calibrated methods:*** In these methods, features are extracted directly from the signal and they can be categorised according to their feature domain:

- Time domain features: These methods primarily use time domain features for distinguishing between covers and stegos, like Intrusion detection systems (IDS), Linear prediction residue, Statistical model of the histogram (PCA), Chaotic-based steganalysis. Features based on Markov process and Autoregressive time delay neural network (AR-TDNN).
- Frequency domain features: These methods primarily use frequency domain features for distinguishing between covers and stegos, like Steganalysis based on Mel-frequency cepstrum coefficients (MFCC): MFCC are one of the most well-known features in speech processing applications. Cepstrums are the frequency components of the logarithm of the magnitude of the spectral power of the signal and it can be interpreted as the rate of power

changes in different frequency regions. MFCC is a modified version of ceps trum and it reflects some of the characteristics of the HAS and the second method which is reversed-psychoacoustic model of human hearing.

∗ ***Calibrated methods:*** Finding features that are independent of the contents of the signal and only reflects the presence of a hidden message is one of the most challenging problems in steganalysis. Apparently, if the cover is known such features can be extracted very easily. However, in the realistic scenarios, this assumption does not hold. Therefore, researchers have proposed estimation as a practical solution to this dilemma [17]:

- Self-generated estimation of cover: Basically, these methods estimate the cover by applying a noise removal procedure on the signal. Also, it is expected that different amount of noise is extracted from stegos and covers.These methods have employed different approaches to quantify such discrepancies like : steganalysis based on audio quality metrics (AQM), steganalysis based on Hausdorff distance, Modelling the noise based on Gaussian mixture model (GMM) and generalised Gaussian distribution (GGD).
- Constant referencing (CIAQM): two fixed reference signals were selected, one of which was a cover and the other one was its stego version.
- Estimation of cover space model: While previous works measured discrepancies between a reference and the signal for making the decision, another approach is possible if the high dimensional model of cover space is known. In this fashion, we can check every signal $x[n]$ with this model if it fits the model, it is a cover and otherwise, it is a stego-speech.
- Re-embedding calibration: Previous calibration methods are similar in the sense that estimation of covers was used for calibration. Calibrating features based on the characteristics of stegos is another possibility that was pursued in [18]. That paper addressed both targeted and universal cases. In the targeted paradigm, the embedding algorithm is known; therefore, signals were embedded with the same algorithm and a random message. Then, the difference between features extracted from the original signal and its embedded version was used for steganalysis.

I.2.3.2 Targeted methods

Specific type of steganalysis techniques or embedding specific steganalysis is also termed as Targeted steganalysis and are designed for a particular steanographic algorithm. This technique is more robust, since it has good detection accuracy for that specific technique when they were used against the particular steganographic technique [19].

∗ ***Phase coding methods:*** Phase coding is a coding schemes that introduces least perceptible noise to the host. The off-set of the phase of a sound is irrelevant to recognition by the HAS. The HAS has very low sensitivity to changes to the phase components of an audio signal. By using the phase components of the sound segment as a data space, a fairly large amounts data can be coded into the host signal. The embedded data is fairly transparent to the HAS if the relative relations between the phase components of preceding segments are well preserved. The modification of the off-set of all the phase components results in no distortion to the sound signal. phase coding is robust to small amounts of additive noise, since the noise won't affect to the distortion of the phase in most of the frequency slots. The wave form of the signal is more important than the absolute value of each data point in phase coding.

∗ ***Echo embedding methods:*** Echo hiding methods have a bank of kernels and depending on the bit of message; one of them is selected and is convolved with different segments of cover [15]. Therefore, kernels play an important role in the characteristics of echo hiding methods. For example, it is possible to design kernels that achieve better robustness. Positive–negative (PN) and forward–backward (FB) are examples of kernels that have been designed for such purposes.

∗ ***Multiplicative embedding methods:*** There are two methods of embedding secret messages into signal's amplitude. One is the additive embedding method. The other is the multiplicative embedding method, where the amplitude of the secret message is changed as the amplitude of the stego-signal [20].

I.2.4 Audio steganalysis algorithms

Audio steganalysis is very difficult due to the existence of advanced audio steganography schemes and the very nature of audio signals to be high-capacity data streams necessitates the need for scientifically challenging statistical analysis [21].

I.2.4.1 Phase and echo steganalysis

Zeng et. Al [22] proposed steganalysis algorithms to detect phase coding steganography based on the analysis of phase discontinuities and to detect echo steganography based on the statistical moments of peak frequency. The phase steganalysis algorithm explores the fact that phase coding corrupts the extrinsic continuities of unwrapped phase in each audio segment, causing changes in the phase difference. A statistical analysis of the phase difference in each

audio segment can be used to monitor the change and train the classifiers to differentiate an embedded audio signal from a clean audio signal [23].

I.2.4.2 Universal steganalysis based on recorded speech

Johnson et. Al [24] proposed a generic universal steganalysis algorithm that bases it study on the statistical regularities of recorded speech. Their statistical model decomposes an audio signal (i.e recorded speech) using basis functions localized in both time and frequency domains in the form of short time fourier transform (STFT). The spectrograms collected from this decomposition are analyzed using non-linear support vector machines to differentiate between cover and stego audio signals. This approach is likely to work only for high-bit rate audio steganography and will not be effective for detecting low bit-rate embeddings [23].

I.2.4.3 Use of statistical distance measures for audio steganalysis

H. Ozer et. al [25] calculated the distribution of various statistical distance measures on cover audio signals and stego audio signals vis-a-vis their versions without noise and observed them to be statistically different. The authors employed audio quality metrics to capture the anomalies in the signal introduced by the embedded data. They designed an audio steganalyzer that relied on the choice of audio quality measures, which were tested depending on their perceptual or non-perceptual nature. The selection of the proper features and quality measures was conducted using the (i) ANOVA test to determine whether there are any statistically significant differences between available conditions and the (ii) sequential floating search (SFS) algorithm that considers the inter-correlation between the test features in ensemble. Subsequently, two classifiers, one based on linear regression and another based on support vector machines were used and also simultaneously evaluated for their capability to detect stego messages embedded in the audio signals [23].

I.2.4.4 Audio steganalysis based on Hausdorff distance

The audio steganalysis algorithm proposed by Liu et. al [26] uses the Hausdorff distance measure to measure the distortion between a cover audio signal and a stego audio signal. The algorithm takes as input a potentially stego audio signal x and its de-noised version x as an estimate of the cover signal. Both x and x are then subjected to appropriate segmentation and wavelet decomposition to generate wavelet coefficients at different levels of resolution. The Hausdorff distance values between the wavelet coefficients of the audio signals and their de-noised versions are measured. The statistical moments of the Hausdorff distance measures are

used to train a classifier on the difference between cover audio signals and stego audio signals with different content loadings [23].

I.2.4.5 Audio steganalysis for high complexity audio signals

More recently, Liu et. al propose the use of stream data mining for steganalysis of audio signals of high complexity. Their approach extracts the second order derivative-based Markov transition probabilities and high frequency spectrum statistics as the features of the audio streams. The variations in the second order derivative-based features are explored to distinguish between the cover and stego audio signals. This approach also uses the Mel-frequency cepstral coefficients, widely used in speech recognition, for audio steganalysis [23].

I.2.5 Attacks used by the steganalyst

To date, in VoIP, a considerable number of steganalysis algorithms, have been developed ,which were surveyed in [14]. In general, they can be divided into six methods:

I.2.5.1 Stego-only attack

In a stego-only attack the steganalyst does not have any other information available apart from the stego medium investigated. It is similar to the ciphertext only attack and it is the hardest scenario for the cryptanalyst. Realistically, the only way a steganalyst would be able to attack it is by trying every possible known attacks on current steganographic algorithms [27].

I.2.5.2 Known cover attack

The original cover-object is compared with the stego-object and pattern differences are detected. For example, the original image and the image containing the hidden information are available and can be compared.

I.2.5.3 Known message attack

A known message attack is the analysis of known patterns that correspond to hidden information, which may help against attacks in the future. Even with the message, this may be very difficult and may be considered the same as a stego-only attack.

I.2.5.4 Chosen stego attack

The steganography tool (algorithm) and stego-object are known. For example, the software and the stego-carrier and hidden information are known.

I.2.5.5 Chosen message attack

The steganalyst generates a stego-object from some steganography tool or algorithm of a chosen message. The goal in this attack is to determine corresponding patterns in the stego-object that may point to the use of specific steganography tools or algorithms.

I.2.5.6 Known stego attack

The stegonography tool (algorithm) is known and both the original and stego-object are available.

I.2.6 Data quality

The goal of steganalysis is to identify suspected information streams, determine whether or not they have hidden messages encoded into them, and, if possible, recover the hidden information [28].

I.2.6.1 Accuracy

The degree to which the data item correctly describes the object in context of appropriate real-world context and attributes. The real-world context may be identified as a single version of established truth and used as a reference to identify the deviation of data items from this reference. Specifications of the real-world references may be based on business requirements and all data items that accurately reflect the characteristics of real-world objects within allowed specifications may be regarded as an accurate piece of information. Data accuracy directly impacts the correctness of decisions and should be considered as a key component for data analysis practices.

I.2.6.2 Consistency

This measure represents the absence of differences between the data items representing the same objects based on specific information requirements. The data may be compared for consistency within the same database or against other data sets of similar specifications. The discrete measurement can be used as an assessment of data quality and may be measured as a percentage of data that reflect the same information as intended for the entire data set. In contrast, inconsistent data may include the presence of attributes that are not expected for the intended information. For instance, a data set containing information on app users is considered as inconsistent if the count of active users is greater than the number of registered users.

I.2.6.3 Minimize false-positives

During analysis of a stego-file the tool for the analysis wrongly indicates that the non-stego-file is a stego-file (false positive) this happens due to the wrong prediction, and to minimize the possibility of getting the false positive there are some metrics in machine learning that would help.

Conclusion

In order to understand steganalysis it is essential to have an understanding of steganography, the powerful tool which enables people to communicate without possible eavesdroppers even knowing there is a form of communication in the first place, so we approached its techniques, categories, characteristics and algorithms that's why we started the chapter by a brief description of this technology.

We then presented the state of art of audio steganalysis, its definition, techniques classification, methods, steganalytic algorithms, attacks and data quality of steganalysis.

Chapter II

Theory of classification using support vector machine (SVM)

Introduction

Steganography, the art of concealing information in digital media such as audios, videos. Steganalysis, its countermeasure attempting to detect the presence of secret information in a seemingly innocent document, are frequently reported as attractive security techniques. in the recent years, machine learning has emerged in the competition between steganography and steganalysis, appearing to be an interesting alternative in the context of steganalyzers.

One of the modern machine learning approach used in the steganalysis or steghanographic security is called support vector machine (SVM).

II.1 Machine learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or in struction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

II.1.1 Types of machine learning algorithms

Machine learning uses two types of techniques: ***supervised learning***, which trains a model on known input and output data so that it can predict future outputs, and ***unsupervised learning***, which finds hidden patterns or intrinsic structures in input data [29].

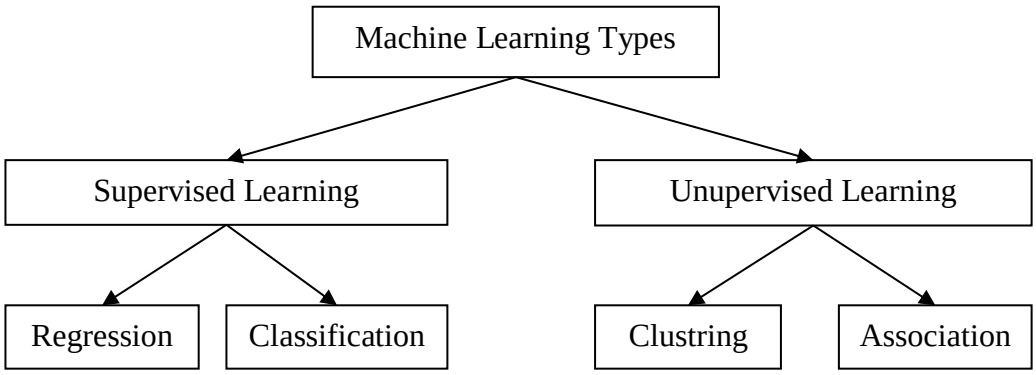


Figure II.1: machine learning types.

II.1.1.1 Supervised learning

The learning algorithm uses samples that are labeled. For example, the samples are microarray data from cells, and the labels indicate whether the sample cells are cancerous or healthy. The algorithm takes these labeled samples and uses them to induce a classifier. This classifier is a function that assigns labels to samples, including those that have never previously been seen by the algorithm.

Regression and classification are two major prediction problems which are usually dealt with data mining and machine learning categorized under the same umbrella of supervised machine learning. Both share the same concept of utilizing known datasets (referred to as training datasets) to make predictions, here is a short study on these two categories:

Regression: in machine learning, regression algorithms attempt to estimate the mapping function (f) from the input variables (x) to numerical or continuous output variables (y) .In this case, y is a real value, which can be an integer or a floating-point value. Therefore, regression prediction problems are usually quantities or sizes.

Classification: on the other hand, classification algorithms attempt to estimate the mapping function (f) from the input variables (x) to discrete or categorical output variables (y). In this case, y is a category that the mapping function predicts. If provided with a single or several input variables, a classification model will attempt to predict the value of a single or several conclusions.

difference between regression and classification: the main difference between them is that the output variable in regression is numerical (or continuous) while that for classification is categorical (or discrete) [30].

II.1.1.2 Unsupervised learning

Unsupervised learning is the training of machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data.

Unlike supervised learning, no supervisor is provided that means no training will be given to the machine. Therefore machine is restricted to find the hidden structure in unlabeled data by our-self [5]. In this scenario, the task is to find structure in the samples. For instance, finding clusters of similar instances in a growing collection of text documents reveals topical changes

across time, highlighting trends of discussions, and indicating themes that are dropping out of fashion.

Unsupervised learning classified into two categories of algorithms:

Clustring: a clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.

Association: an association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

II.1.2 Machine learning models

Statistical and mathematical models have multiple purposes, ranging from descriptive to predictive to prescriptive analytics. The goal of developing models in machine learning is to extract insights from data that you can then use to make better decisions. Algorithmic models tell you which outcome is likely to hold true for your target variable, based on your training data. They construct a representation of the relationships and extract patterns between all the different features in your dataset that you can apply to similar data you collect in the future, this allows users and researchers to make decisions based on those patterns and relationships.

II.1.2.1 Logistic regression

The Logistic regression is named for the function used at the core of the method the *logistic function*, also called the *sigmoid function*. It was developed by statisticians to describe properties of population growth in ecology, that rise quickly and achieve maximum at the carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

$$\frac{1}{(1+e^{-value})} \tag{II.1}$$

Where e is the base of the natural logarithms and value is the actual numerical value wanted to transform. Below is a plot of the numbers between -5 and 5 boundered into the range 0 and 1 using the logistic function.

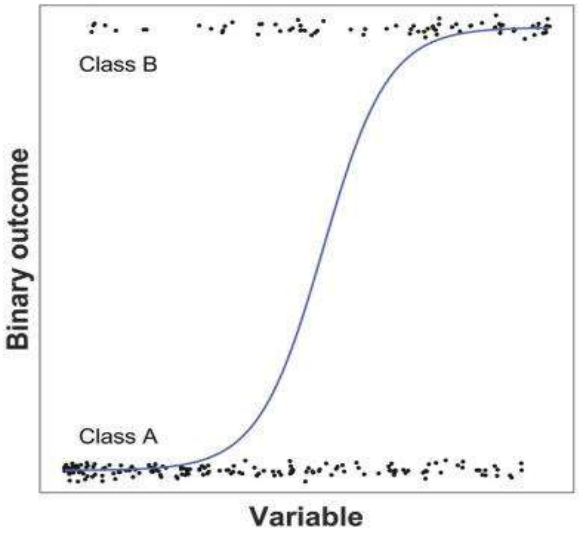


Figure II.2: logistic regression.

II.1.2.2 Linear regression

The representation of linear regression is a equation that describes a line that best fits the relationship between the input variables (x) and the output variables(y), by finding specific weightings for the input variables called coefficients(B).

For example:

$$y = B_0 + B_1 \times x \quad \text{(II.2)}$$

We will predict y given the input x and the goal of the linear regression learning algorithm is to find the values for the coefficients B_0 and B_1.

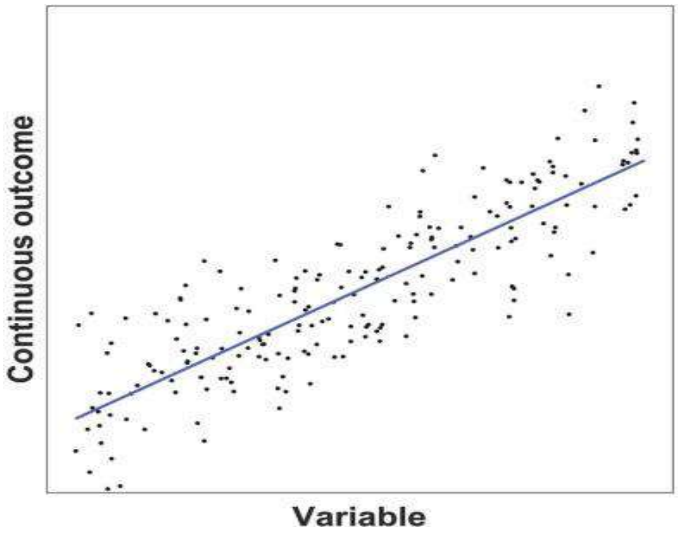


Figure II.3: evaluation in linear regression.

II.1.2.3 K-nearst neighbor

The model representation for k-nearst neighbor (KNN) is the entire training dataset, predictions are made for a new data point by searching through the entire training set for the *K* most similar instances (the neighbors) and summarizing the output variable for those *K* instances. For regression this might be the mean output variable, in classification this might be the mode, or most common, class value.

The trick is in how to determine similarity between the data instances. The simplest technique, if your attributes are all of the same scale (all in inches for example), is to use the Euclidean distance, a number you can calculate directly based on the differences between each input variable.

The idea of distance or closeness can break down in very high dimensions (lots of input variables), which can negatively affect the performance of the algorithm on your problem. This is called the curse of dimensionality. Is suggested that users only use those input variables that are most relevant to predicting the output variable.

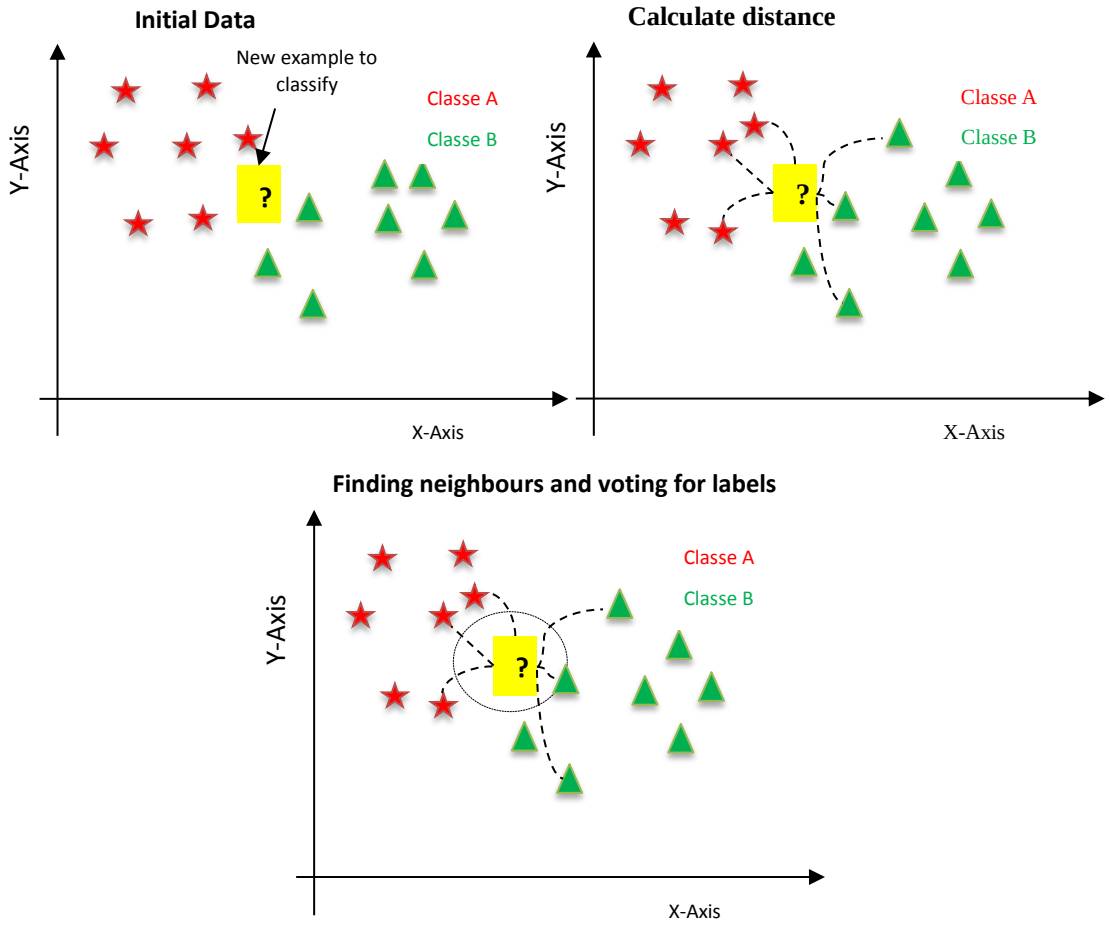


Figure II.4: based dynamic KNN.

II.1.2.4 Decision trees and random forests

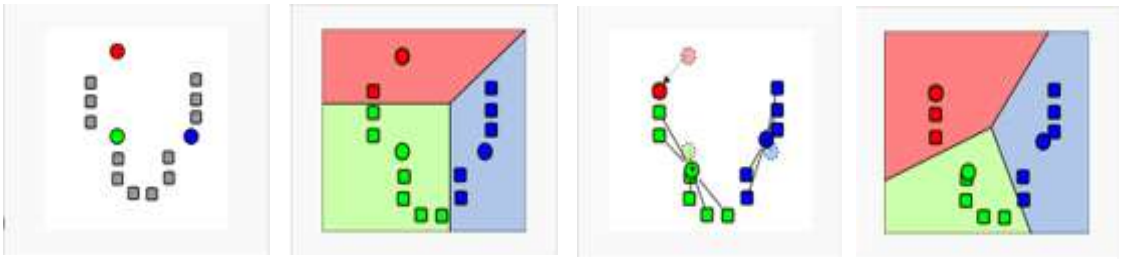
A decision tree is a flowchart-like structure made of nodes and branches. At each node, a split on the data is performed based on one of the input features, generating two or more branches as output. More and more splits are made in the upcoming nodes and increasing numbers of branches are generated to partition the original data. This continues until a node is generated where all or almost all of the data belong to the same class and further splits (branches) are no longer possible.

This whole process generates a tree-like structure. The first splitting node is called the root node. The end nodes are called leaves and are associated to a class label. The paths from root to leaf produce the classification rules. If only binary splits are possible, they are binary trees.

Many is better than one. This is, simply speaking, the concept behind the random forest algorithm. That is, many decision trees can produce more accurate predictions than just one single decision tree by itself. Indeed, the random forest algorithm is a supervised classification algorithm that builds N slightly differently trained decision trees and merges them together to get more accurate and stable predictions.

II.1.2.5 K-Mean clustering

k-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. It partitions n observations into k clusters in which each observation belongs to the cluster with the nearest mean. In terms of finding nearest mean, it uses squared *Euclidean distance*. Mathematically, this means partitioning the observations according to the *Voronoi diagram* generated by the means.



1. k initial ‘’means ‘’ (in this case k=3) are randomly generated within the data domain (shown in color)

2. $k$clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

3. The centroid of each of the k clusters becomes the new mean.

4. Steps 2 and 3 are repeated until convergence has been reached.

Figure II.5: standard algorithm of k-means clustering.

II.1.2.6 Artificial neural networks

Artificial neural network**s** (ANN) is a model based on a collection of connected units or nodes called "*artificial neurons*", which loosely model the *neurons* in a biological *brain*. Each connection can transmit information, a "signal", from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. Typically, artificial neurons are aggregated into layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first layer (the input layer) to the last layer (the output layer), possibly after traversing the layers multiple times.

The original goal of the ANN approach was to solve problems in the same way that a *human brain* would. However, over time, attention moved to performing specific tasks, leading to deviations from *biology*. Artificial neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board, video games and medical diagnosis.

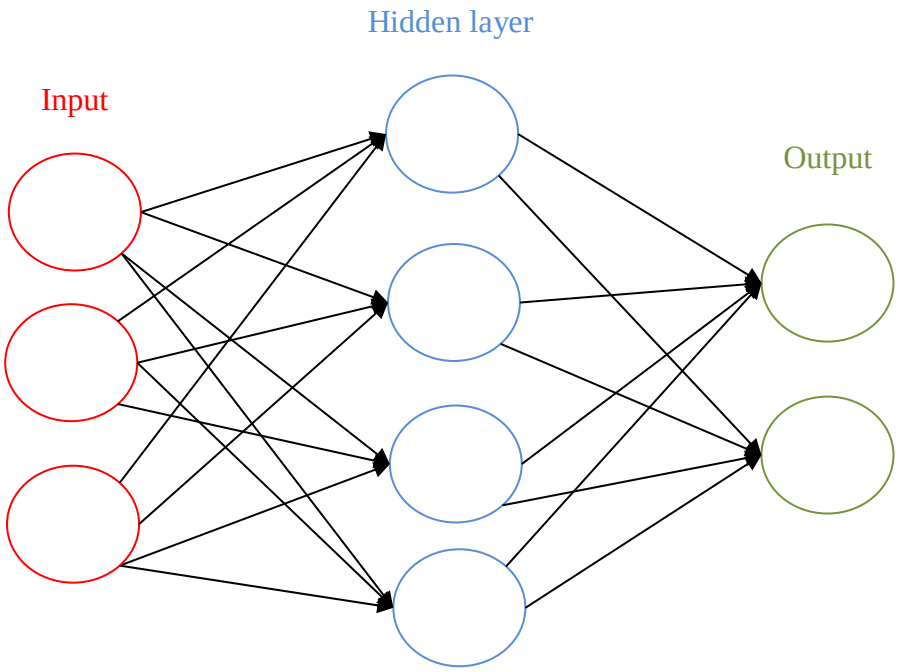


Figure II.6: artificial neural network architecture.

II.2 Support vector machine classification

SVMs are one of the most popular and talked about machine learning algorithms. A hyperplane is a line that splits the input variable space. In SVM, a hyperplane is selected to best separate the points in the input variable space by their class, either class 0 or class 1. In two-

dimensions you can visualize this as a line and let's assume that all of our input points can be completely separated by this line.

The SVM learning algorithm finds the coefficients that results in the best separation of the classes by the hyperplane. The distance between each hyperplane and the closest data points is referred to as the margin. The best or optimal hyperplane that can separate the two classes is the line that as the largest margin. Only these points are relevant in defining the hyperplane and in the construction of the classifier. These points are called the support vectors, which define the hyperplane. In practice, an optimization algorithm is used to find the values for the coefficients that maximizes the margin.

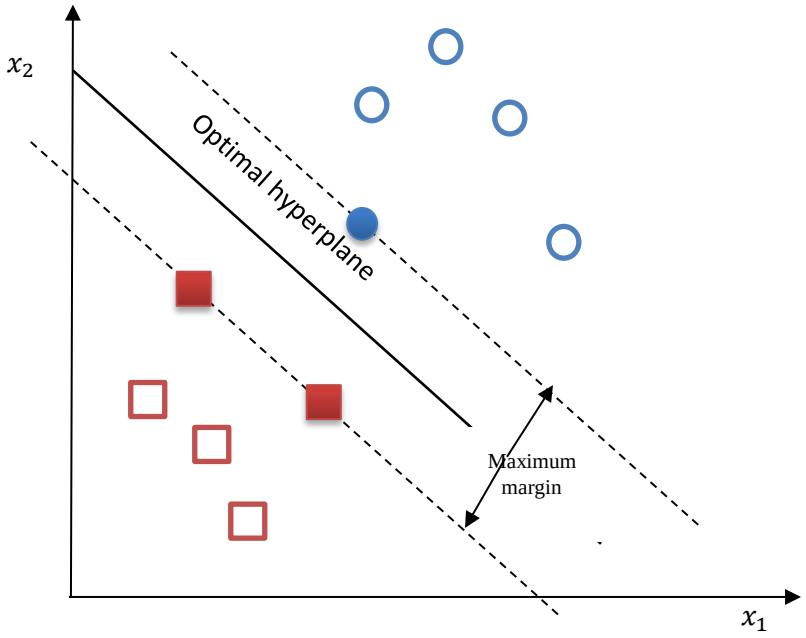


Figure II.7: support vector machine (SVM).

II.2.1 Basic understanding to support vector machine

Classification in SVM is an example of supervised learning. Known labels help indicate whether the system is performing in a right way or not. This information points to a desired response, validating the accuracy of the system, or be used to help the system learn to act correctly. A step in SVM classification involves identification as which are intimately connected to the known classes. This is called feature selection or feature extraction. Feature selection and SVM classification together have a use even when prediction of unknown samples is not necessary. They can be used to identify key sets which are involved in whatever processes distinguish the classes [32].

The next figure shows the svm's process overview:

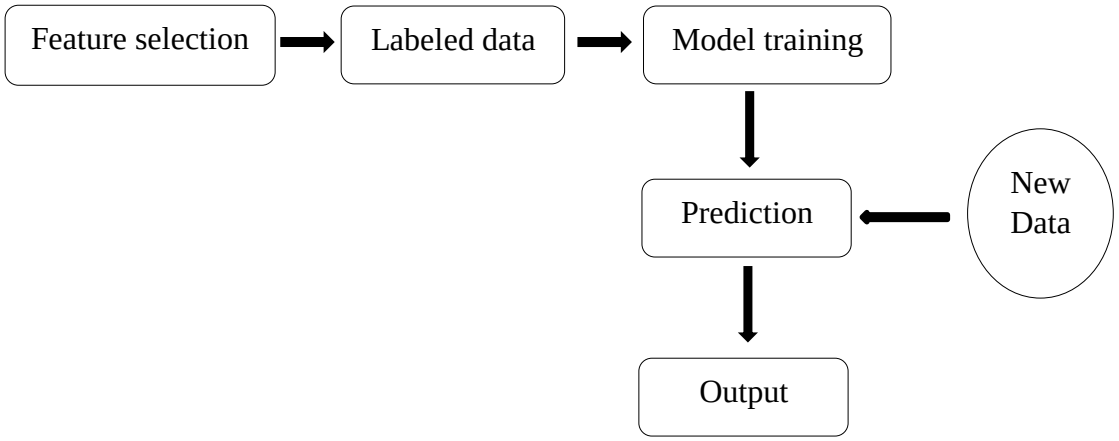


Figure II.8: SVM process overview.

II.2.1.1 Feature extracting phase

Feature selection is an important issue in building classification systems. It is advantageous to limit the number of input features in a classifier to have a good predictive and less computationally intensive model.

II.2.1.2 Training phase

Every input learning vector is represented by vectors. The SVM classifier consist of the input data from distance to border vectors. Vector V_i with label L_i. $L_i = \pm 1, 1 \leq i \leq S$, where S is the number of samples [33].

II.2.1.3 Testing phase

The resulting classifier is functional to unlabelled to make a decision whether is positive or the negative category. The label of L is basically obtained by scrutinize as category 1 if the calculation is positive and category 2 otherwise [33].

II.2.2 Support vector and marging in SVM

After predicting data classes or categories we can represent it by a plot where the model separate data into class one and class two by a hyperplane or decision boundary, which is choosen in order to data type.

II.2.2.1 Linear SVM

Consider a linear classification problem aiming to find optimal separating hyperplane with maximum margin, Let $\overrightarrow{x}_i$ denote the feature vector, and let y_i denote its class label (e.g. y_i = +1, if $\overrightarrow{x}_i$ corresponds to a clean audio signal, and y_i = −1, if $\overrightarrow{x}_i$ corresponds to a stego-speech signal). In a linear SVM, users seek a linear decision function $f(\cdot)$ determined by a unit vector $\overrightarrow{w}$ and an offset b as [34]:

$$f\ (\vec{x}) = sgn(\vec{w} \cdot \vec{x}\ - b) \tag{II.3}$$

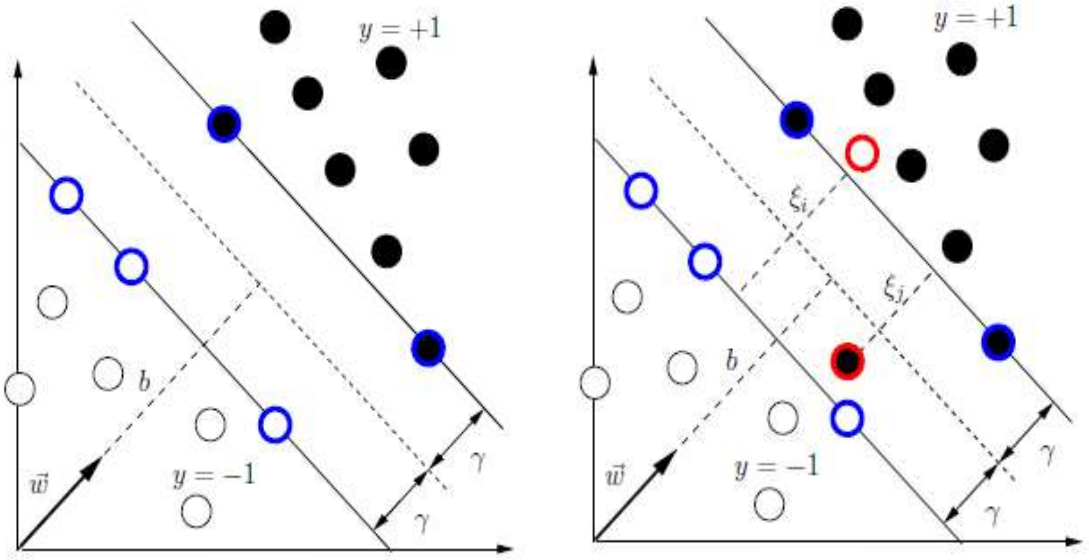


Figure II.9: linear SVM. (a) For linearly separable data. (b) For linearly non-separable data [34].

Linearly separable data: SVM classification seeks the surface (dashed line) that maximizes the classification margin, where $f\ (\vec{x})$ outputs +1 for positive-labeled data points and, -1 for negative-labeled data points. The decision function $f\ (\cdot)$ is estimated by maximizing the classification margin γ subject to the following constraints:

$$\vec{w} \cdot \vec{x} - b \geq \gamma \ \ if \ \ y_i = +1\,,$$

$$\vec{w} \cdot \vec{x} - b \leq -\gamma \ \ if \ \ y_i = -1\,, \tag{II.4}$$

$$\|\vec{w}\| = 1.$$

Linearly non-separable data: in the case where the data is not linearly separable, the optimization problem is adjusted to tolerate some classification errors, as shown in Figure (b). Specifically, slack variables ξ_i are introducedfor each data point $\overrightarrow{x_i}$ to indicate its violation from a linear separation. The constraints of equation (II.4) are changed accordingly to:

$$\vec{w} \cdot \vec{x} - b \geq \gamma - \xi_i \ \text{ if } \ y_i = +1\,,$$

$$\vec{w} \cdot \vec{x} - b \leq -\gamma + \xi_i \text{ if } y_i = -1, \quad \text{(II.5)}$$

$$\|\vec{w}\| = 1.$$

$$\xi_i \geq 0.$$

II.2.2.2 Non-linearly SVM

The linear SVM can also be performed in a non-linearly mapped space to achieve a non-linear separation of the data. First, the data points are mapped by a non-linear function $\varphi(\cdot)$ into a new space H. Then, a linear SVM algorithm is run in H to find the linear decision function from equation (II. 3). A linear decision function in H corresponds to a non-linear classification surface in the original space. For computational efficiency, a kernel function that is equivalent to computing inner products of two mapped data points in H is used in the optimization algorithm [34].

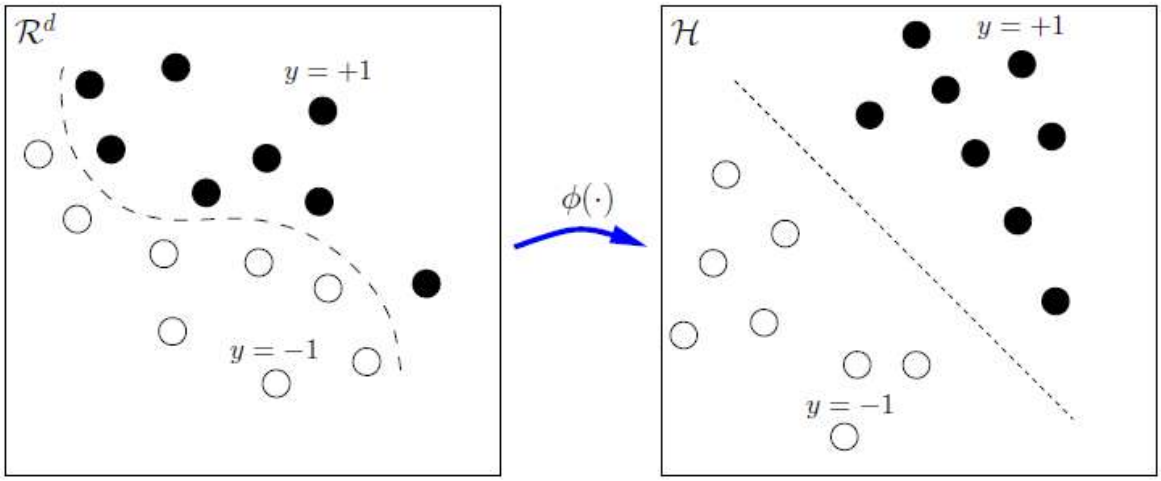


Figure II.10: non-linear SVM classification [34].

II.2.3 Support vector machine classifiers

II.2.3.1 Maximal margin classifier

Maximal margin classifier is the simplest form of SVM classifiers that solves the problem of finding a separating hyperplane with the maximum margin. The underlying optimization problem for the maximal margin classifier is only feasible if the two classes of pattern vectors are linearly separable. However, most of the real-life classification problems are not linearly separable. Nevertheless, the maximal margin classifier encompasses the fundamental methods

used in standard SVM classifiers. The solution to the optimization problem in the maximal margin classifier minimizes the bound on the generalization error [35].

The basic premise of this method lies in the minimization of a convex optimization problem with linear inequality constraints, which can be solved efficiently by many alternative methods [36]. We start our review with the definition of a hyperplane, $(w \cdot x) + b = 0$, which is represented as the normal vector w and the offset parameter b. There is inherent degree of freedom in specifying a hyperplane as (λ_w, λ_b). A canonical hyperplane is the one from which the closest pattern vector has a distance $\frac{1}{\|w\|}$ i.e., $min_{i=1,\dots,n}|(w \cdot x_i) + b| = 1$. Now, consider two pattern vectors x^+ and x^-, belonging to classes S^+ and S^-, respectively and they are the closest pattern vectors to a canonical hyperplane, such that $(w \cdot x^+) + b = 1$ and $(w \cdot x^-) + b = -1$. It is easy to show that the margin between these pattern vectors and the hyperplane are both equal to $\frac{1}{\|w\|}$. Maximizing the margin to $\frac{1}{\|w\|}$. For the canonical hyperplane is equivalent to minimizing $\|w\|$ or $\|w\|^2$. In the following optimization problem, each point x_i with a label y_i satisfies the canonicality condition in the constraints equation II.6 (b) while maximizing the margin by minimizing$\|w\|$, as shown in Figure II.10.

$$\begin{cases} \min(\frac{1}{2}\|w\|^2) & (a) \\ y_i(w \cdot x_i + b) \geq 1 \qquad i = 1 \dots, n. & (b) \end{cases} \tag{II.6}$$

From the solution to 1, a new pattern vector x^*can be classified as positive if $(w \cdot x^*) + b > 0$, and negative otherwise [35].

II.2.3.2 Soft margin classifier

Most real-life problems are composed of non separable data which is generally due to the noise. In this case slack variables ξ_i are introduced for each pattern vector x_i in the training set. The slack variables allow misclassifications for each pattern vector; however, they are subject to a penalty of $C/2$. In figure II.10 soft margin is demonstrated that incurs penalty for misclassified pattern vectors. The maximum margin formulation can be augmented to soft margin formulation as follows [35].

$$\begin{cases} \min(\frac{1}{2}\|w\|^2) + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 \\ y_i(w \cdot x_i + b) \geq 1 - \xi_i \qquad i = 1 \dots, n. \end{cases} \tag{II.7}$$

In (II.7), the nonnegativity of the slack variables are implicitly implied since the solution cannot be optimal when $\xi_i < 0$ for any pattern vector.

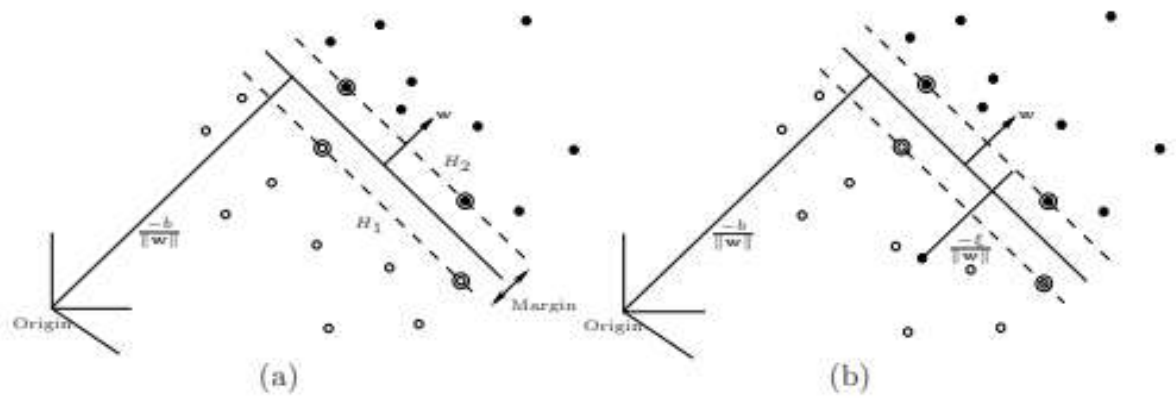


Figure II.11: (a) maximal margin classifier, (b) soft margin classifier [35].

II.2.3.3 Kernel support vector machine

A kernel is a function that takes two vectors x_i and $x_j$as arguments and returns the value of the inner product $\Phi(x_i)$ and $\Phi(x_j)$, the function $\Phi(x)$ is a kernel-induced *implicit* mapping. Mathematically, kernel function is $k(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$.The training vectors x_i are mapped into a higher (probably infinite) dimensional space by the function Φ. There are following four basic kernels: linear, polynomial, radial basic function (RBF), and sigmoid [37]:

Linear:

$$k(x_i, x_j) = x_i{}^T x_j \tag{II.8}$$

Polynomial: the polynomial kernel of degree d is of the form:

$$k(x_i, x_j) = (x_i, x_j)^d \tag{II.9}$$

RBF: the Gaussian kernel, known also as the radial basis function, is of the form

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \tag{II.10}$$

Where σ stands for a window width.

Sigmoid: the sigmoid kernel is of the form

$$k(x_i, x_j) = tanh(k(x_i, x_j) + r) \tag{II.11}$$

Where r is kernel parameter.

When the sigmoid kernel is used with the SVM, it can be seen as a two-layer neural network.

II.3 Applications of SVM

SVMs have a number of applications in several fields. Some common applications of SVM are: face detection, text and hypertext categorization, classification of images, bioinformatics, handwriting recognition and generalized predictive control (GPC).

II.4 Advantages and disadvantages of SVM

Table II.1: advantages and disadvantages of svm.

Advantages of SVM	**Disadvantages of SVM**
-SVM Classifiers offer good accuracy and perform faster prediction. -SVM can be used for both linearly separable as well as non-linearly separable data. Linearly separable data is the hard margin however, non-linearly separable data poses a soft margin. -It is effective in high dimensional spaces. -SVM works well with a clear margin of separation and with high dimensional space.	-SVM is not suitable for large datasets because of its high training.

Conclusion

Machine learning is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention. In this chapter, we took a look on this data analysis method, we discussed its types of learning and some of its algorithms in general,

In the second part of this chapter we presented the SVM classification in detail. We have discussed the concept behind support vector machines, how it works. Lastly, we came across the advantages and the drawback of SVM along with various applications of support vector machines.

Chapter III

Steganalysis of the RLPFS method using SVM classifier and the MFCC features

Introduction

This chapter provides the description of one of the steganographic methods implemented with the mixed-excitation linear predictive (MELP) coder called random least significant bits for pitch and fourier magnitude steganography (RLPFS) which will be subject to our steganalysis method. Our work provides also a detailed description of MFCC features and classification steps. The latter procedure aims to detect the existence of the hidden message.

III.1 MELP coder

The MELP standard belongs to parametric voice coders family [38]. It was developed by texas instruments and was proposed as a new military and federal standard for 2.4 kbps to replace the federal standard FS-1016 CELP [38]. The MELP coder produces a coded speech frame every 22.5 ms (44 frames/s) at a sampling rate of 8 kHz to get 54 bits per frame for the 2.4 kbps data rate [38]. MELP is basically an improvement with respect to LPC [38], where additional parameters are introduced to add naturalness, smoothness and adaptability to more diverse signal conditions. Some of parameters involved to hide secret speech are: pitch period, gain, bandpass voicing and fourier magnitudes (FMs).

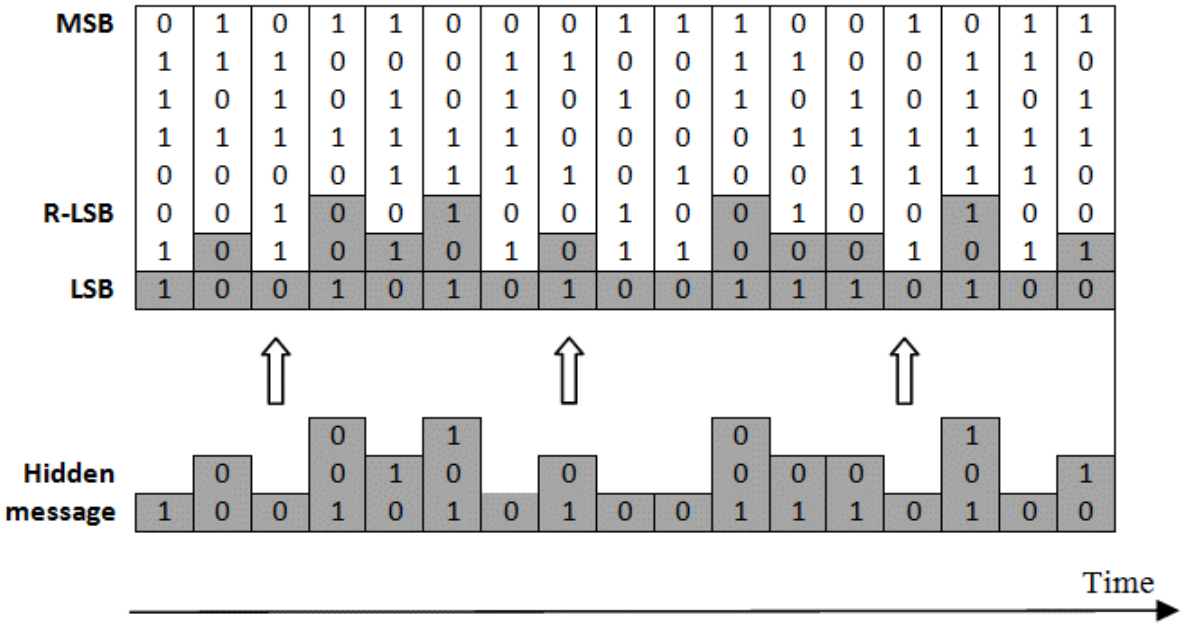
III.2 RLPFS method

In RLPFS, the secret speech and the cover speech signals are coded with the 2.4 kbps MELP coder. In fact, the bitstream is made up of necessary and sufficient coefficients (54 bits per frame) taken from the output of the MELP coder. This is explained by the fact that the speech coding is a procedure to represent a digitised speech signal using as few bits as possible, maintaining at the same time a reasonable level of speech quality. Let $x_1, \dots, x_n$ be a sequence of scalar values (cover speech), where x_i is encoded into M bits, and $M = 8$ bits in MELP coding. Let $b_1, \dots, b_n$ be the data to be embedded (secret speech), where $b_i \in \{0, 1\}$ [39].

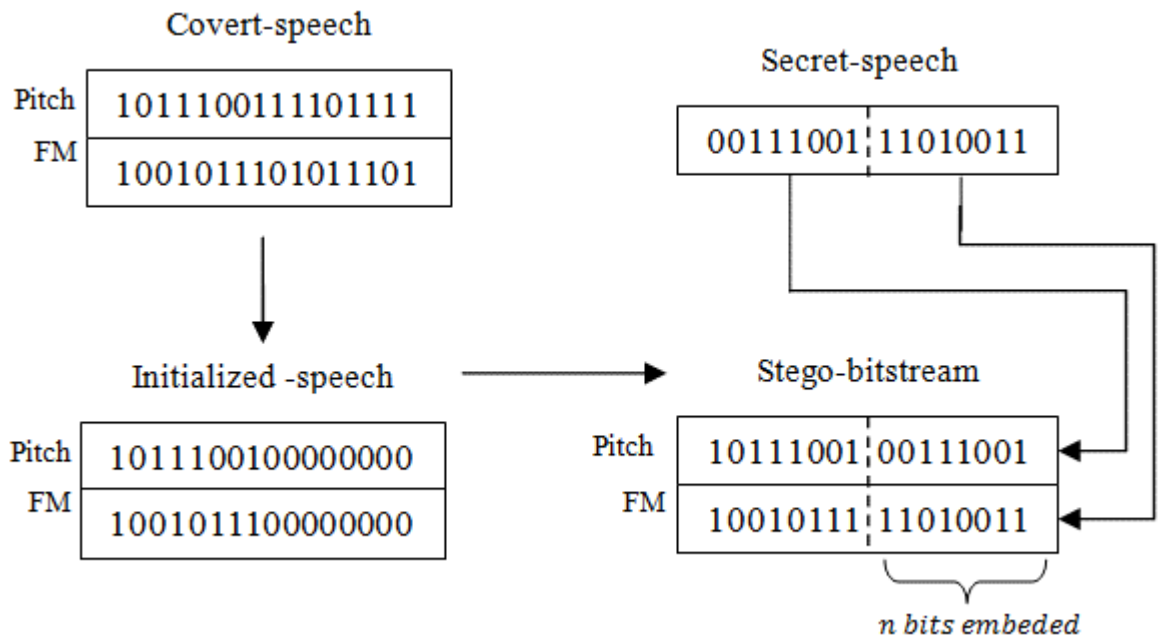
Second, the strategy to hide secret speech into cover speech is done by putting into effect the R-LSB method (Figure III.1(a)), which is an enhanced version of LSB.

Let $m(x)$ be a bit-mask function that clears the LSB of x. If M is sufficiently large, we can expect that there is no large difference between x and $m(x)$. Therefore, we can use the LSB of x_i as the payload for embedding information. Formally, y_i, which represents the *stego-speech*, can be written as follows [39]:

$$y_i = m(x_i) + b_i \quad \text{(III.1)}$$



(a) Principle of LSB and R-LSB to hide secret data.



(b) Principle of R-LSB method to embed data in pitch and FM.

Figure III.1: general scheme of hiding secret speech [39].

As mentioned in [40] and detailed in [41], the stego-speech can be modelled as additive white gaussian noise, as mainly utilised in information theory to resemble the impact of random natural processes. The additive property comes from the fact that this model is added for any noise, which may be inherent for information systems. The white property represents its

uniform power across the frequency band, while the gaussian property represents its normal distribution with zero average value of time domain. From the statistics point of view, the steganograms are random variables with the probability density function as follows [39]:

$$pdf(x) = \frac{1}{\sqrt{2\pi}\sigma}\ exp\left(-\frac{(m(x)+ b_i)^2}{2\sigma^2}\right) \quad \text{(III.2)}$$

i.e. steganograms are random variables that fulfil the normal distribution with the expectation $E(\xi) = 0$ and the variance $D(\xi) = \sigma^2$.

For R-LSB, $b_i \in \{0,1\}$ if $NEB = 1$, $b_i \in \{00,01,10,11\}$ if $NEB = 2$ or $b_i \in \{000,001,010,100,011,101,110,111\}$ if $NEB = 3$.

Then they used the data $y_1, \dots, y_n$ as new data where the information $b_1, \dots, b_n$ is embedded. The embedded information b_i can be recovered by:

$$b_i = y_i - m(y_i) \quad \text{(III.3)}$$

One problem of applying the simple LSB or R-LSB substitution method to MELP coded speech is that the distortion of the decoded speech depends on the magnitude of the input speech. When linear quantisation is applied, it is ensured that $0 \leq (x_i - y_i)^2 \leq 1$ for any x_i. On the other hand, a non-linear quantisation method such as multi-stage vector quantiser (MSVQ) used in MELP requires a non-linear transformation for restoring the speech. The formalism of non-linear distortion is given as follows [39]:

$$D = \frac{1}{n}\min_x \sum_{i=1}^{n} d(x_i, y_i) \quad \text{(III.4)}$$

The R-LSB method is applied on x_i, which can be the pitch, the FM or both. At the sender side, the pitch and the FM of the cover speech (real values) are converted to binary, and, then, the LSBs of these two parameters are initialised to zeros using bitmask function $m(x)$. First, the secret message is converted to binary and then, it is split into groups of bits b_i and b_{i+1} (number of bits per group is fluctuated and depends on how many bit can be embedded at a time). The addition of these results (secret and cover signals) is called stego bitstream (y_i). Figure III.1(b) illustrates these steps. In the stego bitstream, the cover speech takes the higher significant bits while the secret speech occupies the LSBs of less importance. In this method, a switch is used to randomly choose MN, $mode = 1$ means embedding in the pitch, $mode = 2$ means embedding in the FM and, finally $mode = 3$ means embedding in both pitch and FM simultaneously for each transmitted frame. The strategy of changing mode of embedding increases the security level against an attacker or steganalysis process. Varying the MN and NEB multiple times in one stego-speech file will prevent attackers to detect the hidden speech neither correctly nor permanently [39].

III.3 Feature selection and extraction

Dimensionality reduction as a preprocessing step to machine learning is effective in removing irrelevant and redundant data, increasing learning accuracy, and improving result comprehensibility. However, the recent increase of dimensionality of data poses a severe challenge to many existing feature selection and feature extraction methods with respect to efficiency and effectiveness. Feature extraction is an important audio analysis stage. In general, it is an essential processing step in pattern recognition and machine learning tasks. The goal is to extract a set of features from the dataset [42]. These features must be informative with respect to the desired properties of the original data. Feature extraction can also be viewed as a data rate reduction procedure because we want our analysis algorithms to be based on a relatively small number of features.

III.3.1 Mel frequency cepstral coefficients

Mel frequency cepstral coefficients (MFCCs) are short-term spectral features and are widely used in the area of audio and speech processing. The MFCCs have been applied in a range of speech recognition and have shown good performance compared to other features. The low orders MFCCs contain information of the slowly changing spectral envelope while the higher orders MFCCs explain the fast variations of the envelope. MFCC's are based on the known variation of the human ear's critical bandwidths with frequency, filters spaced linearly at low frequencies and logarithmically at high frequencies to capture the phonetically important characteristics of speech and audio. To obtain MFCCs, the speech signals are segmented and windowed into short frames of 20 msec. The block diagram for MFCC is shown in figure III.2 [43].

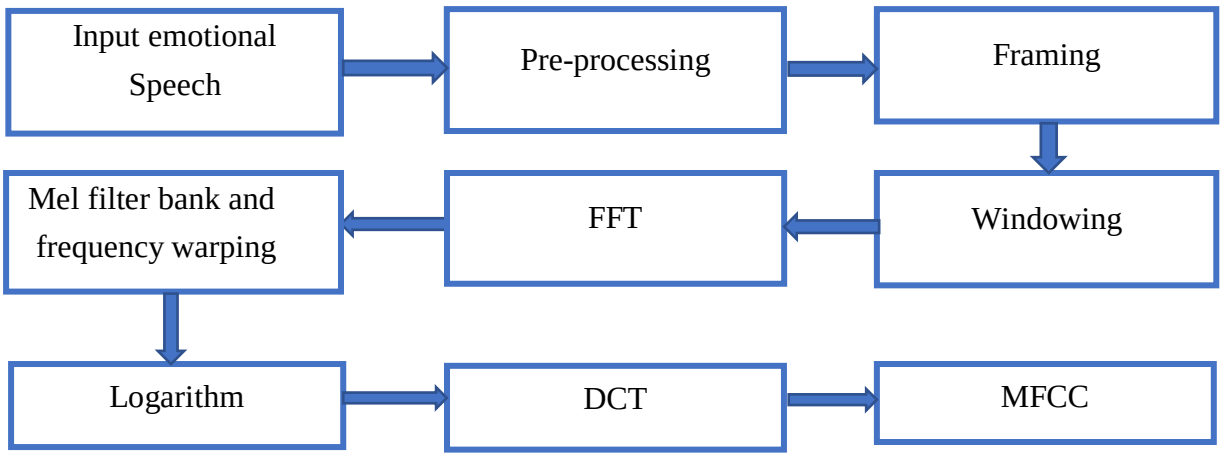


Figure III.2: block diagram for MFCC feature extraction [43].

III.3.1.1 Framing

In framing, the continuous input speech is segmented into N sample per frames. The first frame consists of N samples, second frame consists of M samples after N, and third frame contains $2M$ and so on [44].

III.3.1.2 Windowing

After framing, each frame is multiplied by a window function prior to reduce the effect of discontinuity introduced by the framing process by attenuating the values of the samples at the beginning and end of each frame. The Hamming window is commonly used, it decreases the frequency resolution of the spectral analysis while reducing the sidelobe level of the window transfer function [44].

III.3.1.3 Fast Fourier Transform

Fast Fourier Transform (FFT) algorithm is used for converting the N samples from time domain to frequency domain. It is used to evaluate frequency spectrum of speech [43].

III.3.1.4 Mel spectrum

Mel spectrum is computed by passing the Fourier transformed signal through a set of band-pass filters known as Mel-filter bank. A Mel is a unit of measure based on the human ears perceived frequency. It does not correspond linearly to the physical frequency of the tone, as the human auditory system apparently does not perceive pitch linearly. The Mel scale is approximately a linear frequency spacing below 1 kHz and a logarithmic spacing above 1 kHz. The approximation of Mel from physical frequency can be expressed as [45]:

$$F_{Mel} = 2595 log_{10}\left(1 + \frac{f}{700}\right) \quad \text{(III.5)}$$

where f denotes the physical frequency in Hz, and f_{Mel} denotes the perceived frequency.

III.3.1.5 Mel filter bank

Filter banks can be implemented in both time domain and frequency domain. For MFCC computation, filter banks are generally implemented in frequency domain. The center frequencies of the filters are normally evenly spaced on the frequency axis. However, in order to mimic the human ears perception, the warped axis, according to the nonlinear function given in equation (III.5) is implemented. The most commonly used filter shaper is triangular, and in some cases the Hanning filter can be found. The triangular filter banks with Mel frequency warping is given in Figure III.3 [45].

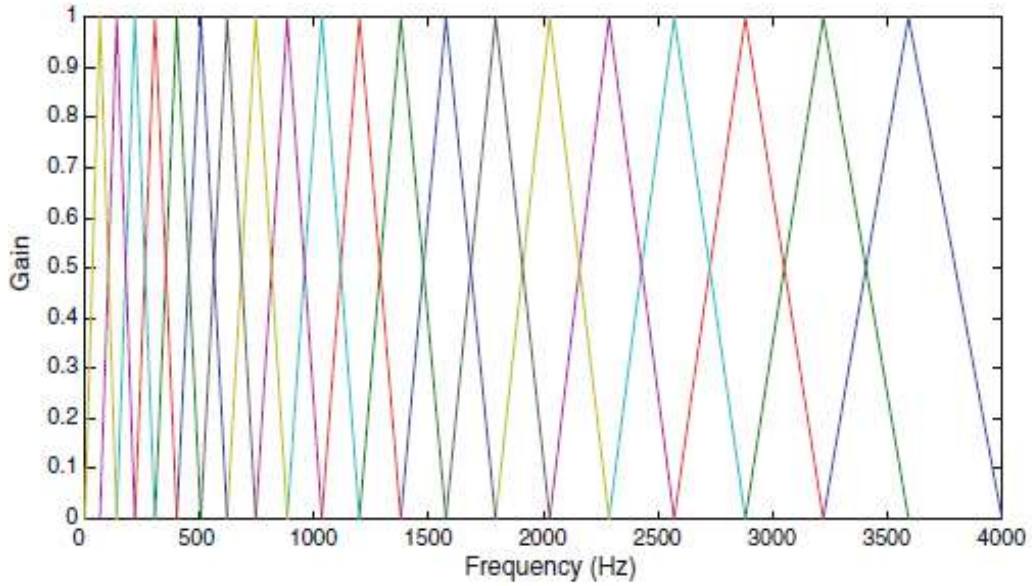


Figure III.3: triangular filter bank [45].

III.3.1.6 Discrete cosine transform

Since the vocal tract is smooth, the energy levels in adjacent bands tend to be correlated. The Discrete cosine transform (DCT) is applied to the transformed mel frequency coefficients produces a set of cepstral coefficients. Prior to computing DCT, the mel spectrum is usually represented on a log scale. This results in a signal in the cepstral domain with a quefrequency peak corresponding to the pitch of the signal and a number of formants representing low frequency peaks. Since most of the signal information is represented by the first few MFCC coefficients, the system can be made robust by extracting only those coefficients ignoring or truncating higher order DCT components. Finally, MFCC is calculated as [45]:

$$\mathrm{C(n)} = \sum_{m=0}^{M} log_{10}\big(s(m)\big) \cos\left(\frac{\pi n(m-0.5)}{M}\right);\ \mathrm{n} = 0, 1, 2, \dots, \mathrm{C} - 1 \qquad \text{(III.6)}$$

Where $\mathrm{c(n)}$ are the cepstral coefficients, and C is the number of MFCCs. Traditional MFCC systems use only 8–13 cepstral coefficients. The zeroth coefficient is often excluded since it represents the average log-energy of the input signal, which only carries little speaker-specific information.

III.3.2 Deltas

The MFCCs are static coefficients. They represent only the power spectral of the signal in a single frame, but it is also known that speech contains information in the dynamics, such as in the trajectory of the MFCCs over time. It appears that finding the MFCC trajectories and

adding them to the MFCC feature vector improve classification performance to some extent. For 12 MFCCs, we will get 12 deltas. Deltas are also called as differential coefficients. The following equation is used to get the delta coefficients:

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2} \quad \text{(III.7)}$$

Here, d_t is a delta coefficient, c_t is the static mel frequency cepstral coefficient of frame t. A conventional value for N is 2 [46].

III.4 Classifier training and testing

Before we proceed with the description of the chosen classifiers, which are readily applied to the analysis of audio segments, we briefly explain an important stage which is of paramount importance in the design of most classifiers, namely the training stage. Another important stage, the testing stage will be thoroughly treated [42].

III.4.1 Training

To begin with, it is assumed that the designer of the classifier has access to a set of training data, a set of feature vectors whose class labels are already known. The training dataset is usually the product of a time-consuming procedure which, in the case of audio signals, involves data gathering, listening tests, manual annotation (tagging), and organization of the annotated audio data to data structures in order to facilitate data exchange. In general, the quality of the training set determines the quality of the resulting classifier.

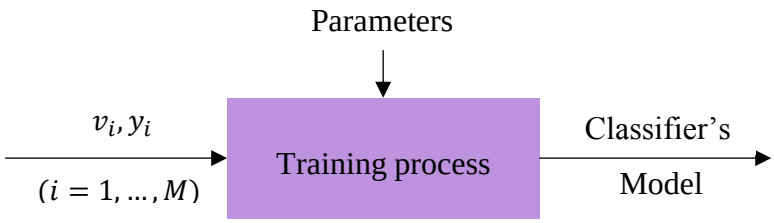


Figure III.4: generic diagram of the classifier training stage [42].

The feature vectors, v_i,where $i = 1, \ldots, M$, of the training dataset and the respective class labels, y_i , have to be available before the training procedure begins. Furthermore, the designer of the system usually needs to make certain decisions related to key parameters of the classifier. The output of the training stage depends on the type of classifier that has been adopted.

Once the training dataset has been made available, we adopt a classification method and try to tune its parameters so that the classifier 'learns' the training dataset. This is known as the training stage, a fundamental concept in machine learning. The number of free parameters which are involved in the training stage depends on the complexity of the classification task and the classification algorithm that has been adopted. For example, if a linear classifier is used in a 2-class task where the classes are separable, the goal of the training stage is to determine the hyperplane that separates the training data of the two classes in the feature space. Most classifiers may demand from the user to make various decisions with respect to the design of the classifier before the actual training procedure begins. Such decisions may refer to a distance metric, choice of a kernel function.

III.4.2 Testing

After the classifier has been trained, it can be used to classify features that belong to the testing dataset. These are features whose class label needs to be determined by the classifier, and the resulting classification decisions must be compared with the true class labels of the data. It is important to note that the training and test datasets must not overlap, otherwise the evaluation of the performance of the classifier cannot be trusted. Figure III.5 presents an overview of the classification process of a test sample. The input to the classification model is the feature vector that represents the unknown data. The output of this process is the estimated class label. Furthermore, depending on the classifier, estimates of the posterior probabilities, $P(\omega_i|x)$, where $i = 1,\ldots,N_c$,which are associated with the classifier's decision, may also be returned, when the number of classes is two $N_c = 2$ the classification is called ***binary***.

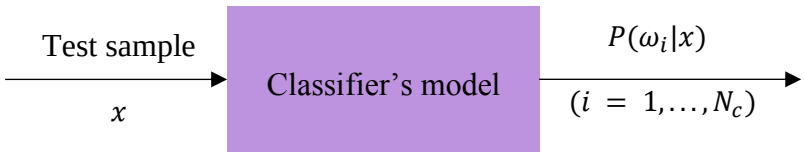


Figure III.5: overview of the classification process of a test sample [42].

III.4.3 Functions used for training and classification

The Bioinformatics Toolbox of MATLAB provides a set of functions for training and evaluating SVM classifiers [42]:

• **svmtrain**: It takes as input (a) a matrix with training vectors (one vector per row, one feature dimension per column), and (b) a vector of respective class labels.

• **svmclassify**: It receives as input (a) the SVM structure that has been returned by svmtrain, and (b) one or more samples to be classified.

III.5 Classifier

SVM's are state-of-the-art classifiers that have been successfully employed in numerous machine learning fields. They are based on the key observation that, for the simple case of linearly separable classes, the optimal decision hyperplane (line on the two-dimensional space) is the one that maximizes the margin among the training data of the two classes. To this end, supporting hyperplanes are used, parallel to the decision hyperplane, in order to define the margin that minimizes the classification error. This rationale is extended to cover the non-separable case by permitting misclassified training data. The way to the success of SVM is the kernel function which maps the feature vectors from the original space into a high dimensional feature space (kernel space).

III.5.1 Kernel properties

Depending on the selected kernel, certain kernel parameters need to be set for the training and evaluation of the SVM classifier, like the constraint parameter, C. It is related to the cost function of the SVM training procedure. As the value of C increases, the cost of misclassified samples also increases. Care must be taken so that overfitting is avoided.

III.6 The proposed model

The proposed steganalysis model is based on analyzing features of an audio, with the aim of detecting the existence of hidden data.

To realize the objectives of the proposed model, three processes are required: steganography which involves embedding of secret data inside cover audios, feature extraction from clean and stego audios and training and testing of a classifier based on the selected features and the classifier.

III.7 Evaluation metrics

In order to calculate the performance of the model some evaluation metrics are required, the metric used in this work is the confusion matrix table III.1, it is a performance measurement for machine learning classification. It contains information about actual and predicted classifications.

Table III.1: confusion matrix for binary classification

Predicted values		
	TP	FP
	FN	TN

Actual values

- True positive (TP): the number of samples correctly labelled as belonging to the steganographic class:

$$TP = \frac{TP}{TP+FN} \quad \text{(III.8)}$$

- True negatives (TN): the number of samples correctly labelled as belonging to the original class:

$$TN = \frac{TN}{TN+FP} \quad \text{(III.9)}$$

- False positives (FP): the number of samples mistakenly labelled as belonging to the steganographic class:

$$FP = \frac{FP}{FP+TN} \quad \text{(III.10)}$$

- False Negatives (FN): the number of samples mistakenly labelled as belonging to the original class:

$$FN = \frac{FN}{FN+TP} \quad \text{(III.11)}$$

- Detection accuracy: Accuracy (Acc) is the proportion of true results in all experimental samples, as given in the following equation:

$$Acc(\%) = \frac{N_{TP}+N_{TN}}{N_{TP}+N_{TN}+N_{FP}+N_{FN}} \cdot 100 \quad \text{(III.12)}$$

III.8 Results and discussion

The proposed model was implemented in MATLAB as a working system. The stego audios analyzed in this work are created using the steganographic method RLPFS. The experimental work included embedding of secret audios, features extractions, training and classification, and testing.

III.8.1 Experimental work

The proposed model aims to detect the existence of a hidden message that has been embedded inside an audio cover. The detection task is based on prior training of a classifier on the features of a dataset of clean and stego-speech, using supervised learning technique. The features set of the proposed model are used in the training and the testing phases.

III.8.1.1 Training and testing steps

To proceed with training and testing, the clean and stego audio datasets were created as in the following steps:

1- Dataset creation: dataset of 105 clean audios for training and testing.

2- Steganography set creation: the secret speeches were embedded in the cover audio dataset to create stego audio dataset which was created using the RLPFS technique, so we have 105 stego audios.

3-Feature extraction: the selected features were extracted from the stego and clean datasets using two methods, the first one using mfcc function and the second one using delta mfcc function, the feature sets are grouped into training datasets and testing.

a. Feature extraction for training datasets: the features extracted from clean and stego datasets of 105 audio file each were labeled and intended for training.
b. Feature extraction of testing dataset: the features of stego dataset were extracted using the same production of feature extraction for training.

4-Training process: the training process utilized the features extracted from clean and stego audios and their labels. These two parameters were trained by svm training function ***svmtrain*** in matlab. We changed the kernel in order to evaluate the method in other metrics.

5-Classification process: the trained data and one-fifth of the feature extracted for testing were classified using svm classifier.

6- Confusion matrix and evaluation of accuracy: after training and classification we calculate the accuracy, we classified the data by the one-fifth, so we have five vectors, we calculate the accuracy for every classification vector, then we calculate the average.

The experimental work is carried out in the stages shown in activity diagram, Figure III.6.

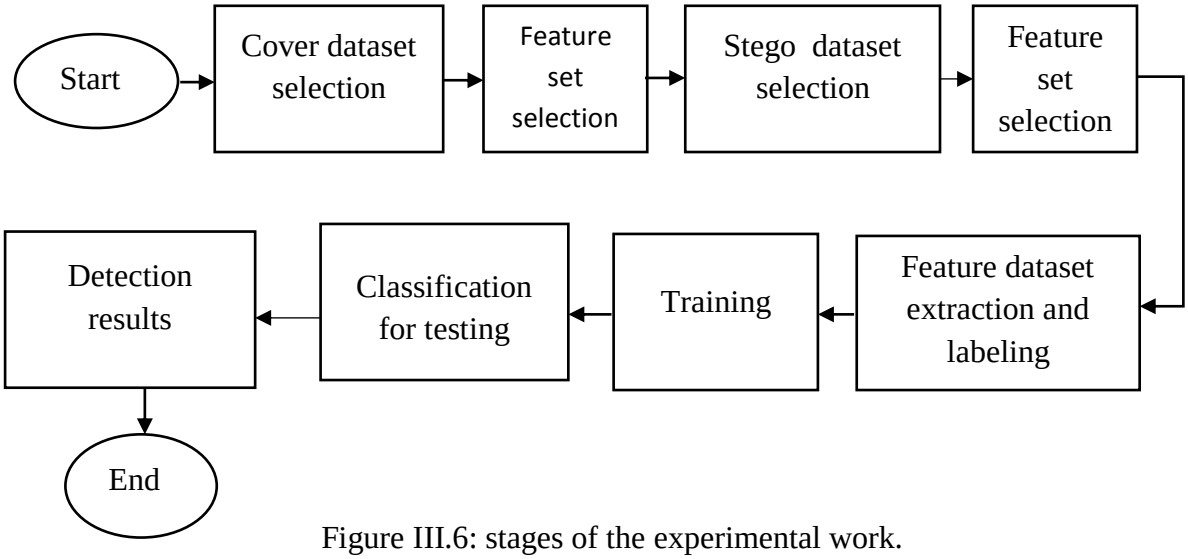


Figure III.6: stages of the experimental work.

III.8.1.2 Results and discussion

The detection accuracies and confusion matrix is done using different analysis windows size and kernels, we varied the window from 22.5 to 500 ms.

* ***Linear kernel:*** the detection accuracy and confusion matrix results for linear kernel are represented in the following tables:

Table III.2: confusion matrix results of linear kernel.

Tw (ms) / Classification	22.5		50		100		200		500	
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2777	0	1160	0	522	0	263	0	104
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2651	0	1262	0	673	0	319	0	110
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2897	0	1198	0	506	0	248	0	97
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2531	0	1224	0	689	0	334	0	117
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2973	0	1240	0	521	0	275	0	113
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2455	0	1182	0	674	0	307	0	101
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2716	0	1102	0	487	0	245	0	106
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2712	0	1320	0	708	0	337	0	108
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2707	0	1130	0	482	0	249	0	86
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2721	0	1292	0	713	0	333	0	128

Table III.3: accuracies of RLPFS steganalysis in linear kernel.

Tw (ms) / Classification	22.5	50	100	200	500
20%	48.83	52.10	56.31	54.81	51.40
20%	46.62	50.53	57.65	57.38	54.67
20%	45.22	48.80	56.40	52.74	47.19
20%	49.96	54.50	59.24	57.90	50.46
20%	50.12	53.34	59.66	57.21	59.81
Average (%)	48.15	51.85	57.85	56.01	52.71

For linear kernel, For all analysis window lengths, the average detection accuracy the average detection accuracy is 53.31 % (with the maximum value of 57.85 % and the minimum value of 48.15 %), the accuracy of the steganalysis is <80%. This indicates that the steganographic method can avoid being detected by the use of SVM classification based MFCC feature for all frame sizes.

- ***Rbf kernel:*** the detection accuracy and confusion matrix results for rbf kernel are represented in the following tables:

Table III.4: confusion matrix results of rbf kernel.

Tw (ms) / Classification	22.5		50		100		200		500	
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	661	0	771	0	365	0	193	0	25
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	1500	0	1651	0	839	0	389	0	189
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	658	0	770	0	376	0	180	0	17
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	1503	0	1652	0	828	0	402	0	197
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	652	0	757	0	374	0	184	0	26
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	1506	0	1665	0	821	0	398	0	188
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	640	0	735	0	344	0	172	0	18
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	1521	0	1687	0	851	0	410	0	196
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	648	0	732	0	355	0	174	0	28
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	1513	0	1690	0	840	0	408	0	186

Table III.5: accuracies of RLPFS steganalysis in rbf kernel.

Tw (ms) / Classification	22.5	50	100	200	500
20%	69.41	68.17	70.21	66.84	61.21
20%	69.55	68.21	69.29	69.07	63.08
20%	69.86	68.74	68.70	68.38	63.08
20%	70.38	69.65	71.21	70.45	67.76
20%	70.01	69.78	70.29	70.10	64.49
Average (%)	69.81	68.91	69.49	68.97	63.93

The average detection accuracy is 68.22 % (with the maximum value of 69.81% and the minimum value of 63.93%), the accuracy of the steganalysis is <80%, which means the steganographic process cannot be detected exactly.

* ***Mlp kernel:*** the detection accuracy and confusion matrix results for mlp kernel are represented in the following tables:

Table III.6: confusion matrix results of mlp kernel.

Tw (ms) / Classification	22.5		50		100		200		500	
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2676	0	1209	0	606	0	271	0	104
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2752	0	1213	0	589	0	311	0	110
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2779	0	1304	0	618	0	231	0	107
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2649	0	1118	0	577	0	261	0	107
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2897	0	1255	0	649	0	305	0	115
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2531	0	1167	0	546	0	277	0	99
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2694	0	1278	0	593	0	294	0	111
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2734	0	1144	0	602	0	288	0	103
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2660	0	1150	0	559	0	296	0	117
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2768	0	1272	0	636	0	286	0	97

Table III.7: accuracies of RLPFS steganalysis in mlp kernel.

Tw (ms) / Classification	22.5	50	100	200	500
20%	50.70	50.08	49.28	53.45	51.40
20%	48.80	46.16	48.28	44.84	50
20%	46.62	48.18	45.59	47.59	46.26
20%	50.36	47.23	50.37	49.48	48.13
20%	50.99	52.51	53.22	49.19	45.32
Average (%)	49.49	48.83	49.37	48.09	48.22

The average detection accuracy is 48.80% (with the maximum value of 49.49 % and the minimum value of 48.09 %), the accuracy of the steganalysis is <80%. This indicates that the steganographic method can avoid being detected by the use of SVM classification based MFCC feature for all frame sizes.

- ***Quadratic kernel:*** the detection accuracy and confusion matrix results for mlp kernel are represented in the following tables:

Table III.8: confusion matrix results of quadratic kernel.

Tw (ms) / Classification	22.5		50		100		200		500	
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	661	0	666	0	329	0	167	0	58
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	1780	0	1756	0	866	0	415	0	156
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	658	0	643	0	306	0	165	0	82
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	1783	0	1779	0	889	0	417	0	132
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	652	0	654	0	332	0	167	0	62
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	1789	0	1768	0	863	0	415	0	152
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	644	0	605	0	326	0	164	0	70
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	1797	0	1817	0	869	0	418	0	144
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	648	0	645	0	313	0	180	0	64
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	1793	0	1777	0	882	0	402	0	150

Table III.9: accuracies of RLPFS steganalysis in quadratic kernel.

Tw (ms) / Classification	22.5	50	100	200	500
20%	72.92	72.50	72.46	71.30	72.89
20%	73.04	73.45	74.39	71.64	61.68
20%	73.32	72.99	72.29	71.30	71.02
20%	73.61	75.02	72.71	71.82	67.28
20%	73.45	73.36	73.80	69.07	70.09
Average (%)	73.26	73.46	73.12	71.03	68.59

The average detection accuracy is 71,89% (with the maximum value of 73.46% and the minimum value of 68.59%), the accuracy of the steganalysis is <80%, which means the steganographic process cannot be detected exactly.

* ***Delta mfcc feature:*** as a second method we used delta mfcc as features for training and testing, the results for multiple kernels are represented in the following tables:

Table III.10: confusion matrix results using delta mfcc in linear kernel.

Tw (ms) / Classification	22.5		50		100		200		500	
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2754	0	1161	0	530	0	264	0	96
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2674	0	1261	0	665	0	318	0	118
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2907	0	1200	0	524	0	264	0	91
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2521	0	1222	0	671	0	318	0	123
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2950	0	1237	0	539	0	277	0	105
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2478	0	1185	0	656	0	305	0	109
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2702	0	1101	0	487	0	250	0	98
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2726	0	1321	0	708	0	332	0	116
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2743	0	1110	0	486	0	245	0	74
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2685	0	1312	0	709	0	337	0	140

Table III.11: accuracies of RLPFS steganalysis with delta mfcc in linear kernel.

Tw(ms) / Classification	22.5	50	100	200	500
20%	49.26	52.06	55.65	54.54	55.14
20%	46.44	50.45	56.15	54.64	57.48
20%	45.65	48.93	54.90	52.41	50.93
20%	50.22	54.54	59.25	57.04	54.21
20%	49.47	54.17	59.33	54.90	65.42
Average (%)	48.21	52.03	57.05	55.33	56.64

The method of delta MFCC, which has the average detection accuracy of 53.852% (with the maximum value of 57.05% and the minimum value of 48.21 This indicates that the steganographic method can avoid being detected by the use of SVM classification based MFCC feature for all frame sizes.

* ***Rbf kernel with delta mfcc feature:*** the detection accuracy and confusion matrix results for rbf kernel are represented in the following tables:

Table III.12: confusion matrix results using delta mfcc in rbf kernel.

Tw (ms) / Classification	22.5		50		100		200		500	
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	984	0	884	0	453	0	232	0	93
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	1838	0	1538	0	742	0	350	0	121
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	928	0	828	0	464	0	209	0	93
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	1894	0	1594	0	731	0	373	0	121
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	948	0	868	0	35	0	219	0	87
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	1874	0	1554	0	760	0	363	0	127
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	942	0	842	0	439	0	208	0	101
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	1880	0	1580	0	756	0	374	0	113
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	978	0	843	0	431	0	215	0	106
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	1844	0	1579	0	764	0	367	0	108

Table III.13: accuracies of RLPFS steganalysis using delta mfcc in rbf kernel.

Tw (ms) / Classification	22.5	50	100	200	500
20%	65,13	63.50	62.09	60.13	56.54
20%	67,11	65.81	61.17	64.08	56.54
20%	66,40	64.16	63.59	62.37	59.34
20%	66,61	65.23	63.26	64.26	52.80
20%	65,34	65.19	63.93	63.05	50.46
Average (%)	66,12	64.78	62.81	62.78	55.14

The average detection accuracy is 62.32% (with the maximum value of 66.12% and the minimum value of 55.14%), the accuracy of the steganalysis is <80%, which means the steganographic process cannot be detected exactly.

* ***Mlp kernel with delta mfcc feature:*** the detection accuracy and confusion matrix results for mlp kernel are represented in the following tables:

Table III.14: confusion matrix results using delta mfcc in mlp kernel.

Tw (ms) / Classification	22.5		50		100		200		500	
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2671	0	1234	0	595	0	283	0	102
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2757	0	1188	0	600	0	299	0	112
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2637	0	1167	0	596	0	284	0	113
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2791	0	1255	0	599	0	298	0	101
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2795	0	1259	0	631	0	301	0	102
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2633	0	1163	0	564	0	281	0	112
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2846	0	1265	0	605	0	306	0	115
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2682	0	1157	0	590	0	276	0	99
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	2694	0	1158	0	579	0	304	0	117
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2734	0	1264	0	616	0	278	0	99

Table III.15: accuracies of RLPFS steganalysis using delta mfcc in mlp kernel.

Tw (ms) / Classification	22.5	50	100	200	500
20%	50.79	49.05	50.21	51.37	52.34
20%	51.42	51.82	50.13	51.20	47.20
20%	48.51	48.02	47.20	48.28	52.34
20%	47.57	47.77	49.37	47.42	46.26
20%	50.37	52.19	51.55	47.77	45.33
Average (%)	49.73	49.77	49.69	49.21	48.69

The average detection accuracy is 49.41 % (with the maximum value of 49.77 % and the minimum value of 48.69 %), the accuracy of the steganalysis is <80%, This indicates that the steganographic method can avoid being detected by the use of SVM classification based MFCC feature for all frame sizes.

- ***Quadratic kernel with delta mfcc feature:*** the detection accuracy and confusion matrix results for quadratic kernel are represented in the following tables:

Table III.16: confusion matrix results using delta mfcc in quadratic kernel.

Tw (ms) / Classification	22.5		50		100		200		500	
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	741	0	641	0	508	0	168	0	66
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2181	0	1781	0	687	0	414	0	148
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	744	0	644	0	508	0	165	0	85
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2178	0	1778	0	684	0	417	0	129
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	759	0	659	0	506	0	168	0	62
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2136	0	1763	0	689	0	414	0	152
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	761	0	641	0	486	0	159	0	58
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2161	0	1781	0	709	0	423	0	156
20%	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
	0	767	0	627	0	495	0	161	0	49
	FN	TN	FN	TN	FN	TN	FN	TN	FN	TN
	0	2155	0	1795	0	700	0	421	0	165

Table III.17: accuracies of RLPFS steganalysis using delta mfcc in quadratic kernel.

Tw(ms) / Classification	22.5	50	100	200	500
20%	74,64	73.53	57.48	71.13	69.16
20%	74,53	73.34	57.48	71.65	60.28
20%	74,02	72.27	57.65	71.13	71.03
20%	73,95	73.53	59.33	72.68	72.90
20%	73,75	74.11	58.33	72.34	77.11
Average(%)	74,18	73.35	58.10	71.79	70.09

The average detection accuracy is 69.50 % (with the maximum value of 74.18 % and the minimum value of 58.10 %), the accuracy of the steganalysis is <80%, which means the steganographic process cannot be detected exactly.

III.8.2 ANN and SVM comparison

In this section we are going to train data using another classifier which is ANN, using matlab toolbox ***nntool***. We took the same steps as we did in SVM befor generating the training and the classification with the ANN classifier, the detection results are presented in table III.18.

Table III.18: accuracies of the s method using ANN.

Tw (ms)	Average acc (%) with MFCC	Average acc(%) with Detla MFCC
10	50.37	48.59
22.5	53.50	48.59
50	51.16	49.46
100	45.23	49.46
200	46.83	50.34
500	50.47	44.39
Max	53.50	50.34
Min	45.23	44.39
Average	49.59	48.47

For all analysis window lengths, the average detection accuracy for linear kernel is 49.59 % (with the maximum value of 53.50% and the minimum value of 45.23%). This indicates that the proposed method can avoid being detected by the use of ANN classification based MFCC feature for all frame sizes. For delta MFCC, the average detection accuracy is 48.47% (with the maximum value of 50.34% and the minimum value of 44.39%), the accuracy of the steganalysis is around of 50, which means the steganographic process cannot be detected.

As we can see the accuracies depends on the kernel function used to well detect the hidden data. For the rbf and quadratic kernel, we found that the steganalysis process is approaching to detect the hidden data. After using SVM for classification we used the ANN classifier, we can conclude that the SVM classify data better then ANN classifier, and the steganographic method cannot be detected even using both classifiers.

Conclusion

In this chapter we presented all the experimental steps of our work, we also made a detailed study on the well-known feature used in speech recognition to describe the signal characteristics the MFCC features which were trained and classified by two classifiers SVM and ann. In the discussion part, we presented the results of accuracies after changing the kernel function and the MFCC to delta MFCC.

The results we came out with is that the RLPFS method is an efficient steganographic method that cannot be detected, we can say that it is a secure way of data hiding.

Conclusion

The growth of modern communication needs a special means of security especially on computer network. As there appears a risk that the sensitive information transmitted might be intercepted or distorted by unintended observers for the openness of the internet. So it has resulted in an explosive growth in secure communication and information hiding. Moreover, the information hiding technique can be used extensively in applications like business, military, commercials, anti-criminal and so on.

Nowadays there are many steganalysis methods claiming to be universal, however, the lack of standard datasets and settings makes it difficult to prove it. Another problem is the variety of steganographic methods. For example, a steganalysis method may be capable to detect embedding data from some steganographic techniques, but it may not detect the secret message from others.

After a detailed study on state of the art of speech steganalysis, the role of artificial intelligence and machine learning, we discovered the principal steps of a steganalysis method, how should we choose the classifier, which features to select before training and classification, how the kernel may effect on the detection accuracy?

In the experimental part of this thesis, speech steganalysis method have been proposed to analysis the performance of the data hiding technique RLPFS coded with MELP codec. The evaluation of detection accuracy using multiple kernels with two types of features the MFCC and D-MFCC which were classified by SVM and ANN classifiers. Results proved that our steganalysis method is not effective to detect the hidden data with RLPFS steganography, which make this steganographic method recommended to hide secret speech for very low speech coding.

References

[1] J. Fridrich, *Steganography in Digital Media*. 2007.

[2] B. Ahuja, "An Overview of Steganography" vol. 11, no. 1, pp. 15–19, 2013.

[3] K. Ray, T. K. Sharma, S. Rawat, · R K Saini, and A. Bandyopadhyay, *Advances in Intelligent Systems and Computing 742 Soft Computing: Theories and Applications Proceedings of SoCTA 2017*. 2017.

[4] J. Ashok, "STEGANOGRAPHY : AN OVERVIEW" vol. 2, no. 10, pp. 5985–5992, 2010.

[5] S. A. Patil, "Hiding Text in Audio Using LSB Based Steganography" vol. 2, no. 3, pp. 8–15, 2012.

[6] R. Sridevi, "Efficient method of audio steganography by modified LSB algorithm and strong encryption key with enhanced security" 2009.

[7] M. Arnold, S. Wolthusen, and M. Schmucker, *Techniques and Application of Digital Watermarkingand content Protection*, 2nd editio. Boston: Artech House, 2003.

[8] W. Bender, N. Morimoto, and A. Lu, "Techniques for data" vol. 35, pp. 313–336, 1996.

[9] D. Kirovski and H. S. Malvar, "Spread-Spectrum Watermarking of Audio Signals" vol. 51, no. 4, pp. 1020–1033, 2003.

[10] D. Huang and T. Y. Yeo, "Robust and Inaudible Multi-echo Audio Watermarking" pp. 615–622, 2002.

[11] Y. Bi, S. Kapoor, and B. Rahul, *Intelligent Systems and Applications: Extended and Selected Results from the the SAI Intelligent Systems Conference (IntelliSys) 2016*. Springer, 2016.

[12] Y. Tint and K. T. Mya, "Audio Steganalysis Using Features Extraction and Classification" vol. 3, no. 2, pp. 1593–1595, 2012.

[13] P. Cababie, A. Zweig, G. Barrera, T. Sobh, and K. Elleithy, *Lecture Notes in Electrical Engineering*, 1st ed., vol. 151. Springer, 2013.

[14] P. Richer, "Steganalysis: Detecting hidden information with computer forensic analysis" *SANS/GIAC Pract. Assign. GSEC Certif.*, vol. 6, 2003.

[15] H. Ghasemzadeh, "Toward a Robust and Secure Echo Steganography Method Based on Parameters Hopping" no. November, 2015, doi: 10.1109/SPIS.2015.7422329.

[16] Y. A. Fadil and Y. A. Fadil, "Security analysis of steganalyzers" 2018.

[17] H. Ghasemzadeh and M. H. Kayvanrad, "Comprehensive Review of Audio Steganalysis Methods" pp. 1–17, 2016, doi: 10.1049/iet-spr.2016.0651.

[18] H. Ghasemzadeh and M. K. Arjmandi, “Universal Audio Steganalysis Based on Calibration and Reversed Frequency Resolution of Human Auditory System 1” *Int. J. Eng. Technol.*, pp. 1–16, 2016.

[19] R. Kannan and F. Andres, *Data Engineering and Management : Second International Conference , ICDEM*, no. May 2016. 2012.

[20] X. Niu and Y. Yixian, “Performance analysis on the algorithms of information hiding and extracting” *Acta Electron. Sin.*, vol. 30, pp. 952–955, 2002.

[21] C. Kraetzer and J. Dittmann, “Pros and Cons of Mel-cepstrum Based Audio Steganalysis Using SVM Classification Pros and Cons of Mel-cepstrum based Audio Steganalysis using SVM Classification” no. November, 2014, doi: 10.1007/978-3-540-77370-2.

[22] W. Zeng, H. Ai, and R. Hu, “A novel steganalysis algorithm of phase coding in audio signal” in *Sixth International Conference on Advanced Language Processing and Web Information Technology*, 2007, pp. 261–264.

[23] S. Bhattacharyya and G. Sanyal, “A UDIO S TEGANALYSIS OF LSB A UDIO USING M OMENTS” vol. 3, no. 1, pp. 145–160, 2012.

[24] M. K. Johnson, S. Lyu, and H. Farid, “Steganalysis of recorded speech Steganalysis of Recorded Speech” no. May, 2014, doi: 10.1117/12.586941.

[25] H. Ozer, B. Sankur, and N. D. Memon, “Steganalysis of Audio Based on Audio Quality Metrics Steganalysis of Audio Based on Audio Quality Metrics ” no. August 2015, 2003, doi: 10.1117/12.477313.

[26] Y. Liu, K. Chiang, C. Corbett, R. Archibald, and D. Ghosal, “A Novel Audio Steganalysis Based on High-Order Statistics of a Distortion Measure with Hausdorff Distance” no. December, 2014, doi: 10.1007/978-3-540-85886-7.

[27] B. Laden, “Statistical Steganalysis 1 CHAPTER 1 Steganography Statistical Steganalysis 2” pp. 1–70.

[28] A. Kumar, “Steganography- A Data Hiding Technique” vol. 9, no. 7, pp. 19–23, 2010.

[29] P. K. Garg, R. D. Garg, G. Shukla, and H. S. Srivastava, *Digital Mapping of Soil Landscape Parameters: Geospatial Analyses using Machine Learning and Geomatics*. Springer Nature, 2020.

[30] M. J. Garbade, “Regression Versus Classification Machine Learning: What’s the Difference?” 2018. .

[31] Aimadhu, “Supervised and Unsupervised learning” 2014. www.geeksforgeeks.org/supervised-unsupervised-learning/.

[32] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and*

Other Kernel-based Learning Methods. Cambridge University Press, 2000.

[33] A. Nedumaran, R. G. Babu, and M. M. Kassa, "Machine level classification using support vector machine" vol. 020013, no. February, 2020.

[34] M. K. Johnson, S. Lyu, and H. Farid, "Steganalysis of Recorded Speech" vol. 1, no. c.

[35] O. Seref, O. E. Kundakcioglu, and P. Pardalos, "Selective linear and nonlinear classification" no. October 2017, 2008.

[36] K. P. Bennett, "Support Vector Machines: Hype or Hallelujah?" vol. 2, no. 2.

[37] M. Pirooznia and Y. Deng, "SVM Classifier: a comprehensive java interface for support vector machine classification of microarray data" vol. 7, 2006.

[38] CHU and C. WAI, *Speech coding algorithms: foundation and evolution of standardized coders*. Hoboken, New Jersey: John Wiley and Sons, 2004.

[39] H. Kheddar and M. Bouzid, "Pitch and Fourier magnitude based steganography for hiding 2 . 4 kbps MELP bitstream" 2019, doi: 10.1049/iet-spr.2018.5339.

[40] M. M. Salih and M. S. Atoum, "A pplying AWGN MP3 Steganography Attack in BiLSB and SLSB Techniques" no. February, 2018, doi: 10.1109/ACSAT.2015.35.

[41] P. Liu, S. Li, and H. Wang, "Steganography integrated into linear predictive coding for low bit-rate speech codec" *Multimed. Tools Appl.*, 2016, doi: 10.1007/s11042-016-3257-x.

[42] G. THEODOROS and P. AGGELOS, *Introduction to A MATLAB Approach AUDIO ANALYSIS :*, 1st ed. Academic Press, 2014.

[43] A. Sharma, R. Kumar, and V. Mansotra, "Proposed Stemming Algorithm for Hindi Information Retrieval" *Int. J. Innov. Res. Comput. Commun. Eng. (An ISO Certif. Organ.*, vol. 3297, no. 6, pp. 11449–11455, 2016, doi: 10.15680/IJIRCCE.2016.

[44] S. A. Majeed, H. Husain, S. A. Samad, and T. F. Idbeaa, "Mel frequency cepstral coefficients (Mfcc) feature extraction enhancement in the application of speech recognition: A comparison study" *J. Theor. Appl. Inf. Technol.*, vol. 79, no. 1, pp. 38–56, 2015.

[45] T. Mfcc, "MFCC Features" 2017, doi: 10.1007/978-3-319-49220-9.

[46] P. P. Das, S. M. Allayear, R. Amin, and Z. Rahman, "Bangladeshi dialect recognition using Mel Frequency Cepstral Coefficient, Delta, Delta-delta and Gaussian Mixture Model" *Proc. 8th Int. Conf. Adv. Comput. Intell. ICACI 2016*, pp. 359–364, 2016, doi: 10.1109/ICACI.2016.7449852.