T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SIRÖRTÜLÜ SES DOSYALARININ YAPAY ZEKA YÖNTEMLERİ YARDIMIYLA ÇÖZÜMLENMESİ

YÜKSEK LİSANS TEZİ

Ali DURDU

Enstitü Anabilim Dalı : ELEK. VE BİLG. EĞİTİMİ

Tez Danışmanı : Yrd. Doç. Dr. A. Turan ÖZCERİT

Temmuz 2010

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SIRÖRTÜLÜ SES DOSYALARININ YAPAY ZEKAY YÖNTEMLERİ YARDIMIYLA ÇÖZÜMLENMESİ

YÜKSEK LİSANS TEZİ

Ali DURDU

Enstitü Anabilim Dalı : **ELEK. VE BİLG. EĞİTİMİ**

Enstitü Bilim Dalı : **ELEK. VE BİLG. EĞİTİMİ**

Bu tez .. / .. /2010 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

Yrd.Doç.Dr. A. Turan ÖZCERİT	Yrd.Doç.Dr. Hayrettin EVİRGEN	Yrd.Doç.Dr. Özdemir ÇETİN
Jüri Başkanı	Üye	Üye

ÖNSÖZ

Teknolojinin gelişmesiyle iletişimde güvenlik unsuru daha da önem kazanmıştır. Artık dünyanın her yeriyle rahatlıkla iletişim kurulmakta ve her türlü bilgi paylaşımı yapılabimektedir. Teknolojinin bu kadar ilerlemesi ile iyi bir iletişim imkânı sağlanırken bu iletişimin güvenli olması da ayrıca önem taşır. Güvenli iletişim için farklı yöntemler geliştirilmiştir. İletişim kanalındaki verilere veri gizleme imkânı doğmuş ve böylece yeni bir iletişim kanalı oluşmuştur. Buradaki iletişim iyi niyetli kullanılabileceği gibi kötü niyetli de kullanılabilir.

Veri gizleme uygulamaları her türlü dosya türüne yapılabilmektedir. Veri gizleme yöntemlerinin gelişip kullanılmasıyla birlikte gizli veri analizi kavramı ortaya çıkmıştır.

Bu tezde ses dosyalarındaki gizli veri analizi üzerinde çalışılmış, sıraçma yöntemleri ile örnek bir uygulama geliştirilmiştir.

Tez çalışmasında değerli görüşleriyle yardımlarını esirgemeyen danışman hocam Yrd.Doç.Dr. Ahmet Turan ÖZCERİT'e, tezin her aşamasında hiçbir zaman desteğini eksik etmeyen eşim Latife'ye, her türlü yaramazlık yaparak beni çalıştırmayan kızım Ayşe Nehir'e, üzerimde büyük emekleri olan annem Atike DURDU, babam Niyazi DURDU'ya, ablam Fatma ŞENER'e ve kardeşim Ahmet DURDU'ya teşekkür ve şükranlarımı sunarım.

Bu çalışma SAÜ Bilimsel Araştırma Projeleri Komisyonu tarafından desteklenmiştir. (Proje No: 2010–50–01–010)

Ali DURDU

İÇİNDEKİLER

ÖNSÖZ ii

İÇİNDEKİLER iii

SİMGELER VE KISALTMALAR LİSTESİ v

ŞEKİLLER LİSTESİ vii

TABLOLAR LİSTESİ ix

ÖZET x

ABSTRACT xi

BÖLÜM 1. GİRİŞ 1

1.1. Sırörtme ve Yapılan Çalışmalar 1

1.2. Sıraçma ve Yapılan Çalışmalar 5

BÖLÜM 2. SIRÖRTME – SIRAÇMA KAVRAM VE TEKNİKLERİ 9

2.1. Sırörtme (Steganografi) 9

2.1.1. Sırörtmenin başlangıcı 10

2.1.2. Literatürde yapılmış sırörtme çalışmaları 10

2.1.3. Sırörtmenin sayısal ortamda kullanım alanları 13

2.1.3.1. Metin tabanlı sırörtme 14

2.1.3.2. Resim tabanlı sırörtme 15

2.1.3.3. Ses tabanlı sırörtme 20

2.1.3.4. Video tabanlı sırörtme 23

2.1.3.5. HTML tabanlı sırörtme 24

2.1.3.6. XML tabanlı sırörtme 25

2.1.3.7. EXE dosya tabanlı sırörtme 26

2.2. Sıraçma (Steganaliz) 26

2.2.1. Sıraçma Yöntemleri 27

2.2.1.1. x^2 (ki-kare) testi 28
2.2.1.2. Görsel tespit 30
2.2.1.3. Histogram analizi 32
2.2.1.4. RQP analizi 33
2.2.1.5. RS analizi (ikili istatistik yöntemi) 33
2.2.1.6. JPEG sıraçma 34
2.2.1.7. Evrensel tespit 34

BÖLÜM 3. SIRÖRTME UYGULAMASI 37

3.1. Wav Dosya Yapısı 27
3.2. Sırörtme İşlemi ve Detayları 41
3.3. Yapılan Testler ve Sonuçlar 49

BÖLÜM 4. Kİ-KARE YÖNTEMİ İLE SIRAÇMA UYGULAMASI 54

4.1. Kullanılan Yapay Zekâ Yöntemi 56
4.2. Sıraçma İşlemi ve Detayları 59
4.3. Yapılan Testler ve Sonuçlar 65

BÖLÜM 5. SONUÇLAR VE TARTIŞMA 79

KAYNAKLAR 81

EKLER 86

EK-A Veri Gizleme Uygulaması Matlab Kodları 86
EK-B Veri Gizleme Uygulaması Versiyon 2 Matlab Kodları 89
EK-C Sıraçma Uygulaması Matlab Kodları 92
EK-D Toplu Veri Gizleme Uygulaması Matlab Kodları 94
EK-E Toplu Sıraçma Uygulaması Matlab Kodları 95
EK-F PNN Yapay Sinir Ağı Eğitimi Matlab Kodları 97
EK-G PNN Yapay Sinir Ağı Testi Matlab Kodları 98

ÖZGEÇMİŞ 99

SİMGELER VE KISALTMALAR LİSTESİ

ASCII	: Karakter Kodlama Standardı(American Standard Code for Information Interchange)
AU	: Ham Ses Standardı(Audio Unit)
AVI	: Ham Görüntü Standardı(Audio Video Interleave)
BMP	: Ham Resim Standardı(Bitmap)
BPCS	: Bit Plane Complexity Segmentation
DCT	: Ayrık Kosinüs Dönüşümü(Discrete Cosine Transform)
DÇ	: Değer Çiftleri(Pairs of Values)
DFT	: Ayrık Fourier Dönüşümü(Discrete Fourier Transform)
DIV	: Ses İçerisinde Veri (Data in Voice)
EXE	: Windows Çalıştırılabilir Dosya Standardı(Executable)
FLD	: Fisher Doğrusal Ayırma(Fisher's Linear Discriminant)
GIF	: Sıkıştırılmış Resim Standardı (Graphics Interchange Format)
HSL	: Renk tonu, Doyum, Açıklık(Hue, Saturation, Lightness)
HSV	: Renk Tonu, Doyum, Değer (Hue, Saturation, Value)
HTML	: İnternet Sayfası Standardı(Hyper Text Markup Language)
ICQ	: Chat Programı(I Seek You)
İDO	: İnsan Duyu Organları
JPEG	: Sıkıştırılmış Resim Standardı(Joint Photographic Experts Group)
JPG	: Sıkıştırılmış Resim Standardı(Joint Photographic Group)
KB	: Kilo Bayt(Kilo Byte)
LSB	: En Önemsiz Bit(Least Significant Bit)
MAT	: Matlab Matematiksel Veri İçeren Dosya (Matlab Mat File)
MIDI	: Ses Formatı, Müzik Aleti Sayısal Arayüzü(Musical Instrument Digital Interface)
MPEG	: Video Sıkıştırma Standardı(Moving Picture Experts Group)

MP3	: Sıkıştırılmış Ses Standardı(MPEG-1 Audio Layer-3)
MSN	: Chat Programı, Microsoft Arama Ağı(Microsoft Search Network)
PNG	: Taşınabilir Ağ Grafikleri(Portable Network Graphics)
PNN	: Olasılıksal Sinir Ağı(Probabilistic Neural Network)
PoVs	: Değer Çifti (Pair of Values)
RIFF	: Ham Ses Standardı(Resource Interface File Format)
RGB	: Renk Kodlaması Kırmızı-Yeşil-Mavi(Red-Green-Blue)
RS	: Sıraçma Yöntemi, Düzenli, Tekil(Regular, Singular)
RQP	: Sıraçma Yöntemi(Raw Quick Pairs)
TXT	: Metin Dosya Standardı
VB	: Ve Benzeri
WAV	: Ham Ses Standardı(Waveform)
WIPO	: World Intellectual Property Organization
YCbCr	: Renk Kodlaması(Luma Blue-Difference, Red-Difference),
YSA	:Yapay Sinir Ağı

ŞEKİLLER LİSTESİ

Şekil 2.1. Johannes Trithemius'un Steganografi ile ilgili bir kitabında yer alan sayı çizelgelerinden bir örnek 11
Şekil 2.2. Sayısal resmin yapısı 16
Şekil 2.3 Gri ton seviye (gray level) resimlerin renk örneği 17
Şekil 2.4. Lena'nın resmi 19
Şekil 2.5. Lena'nın diğer resmi 19
Şekil 2.6. a.html ve b.html dosyaları ekran çıktıları 25
Şekil 2.7. Resim içine gizli mesaj gömülmeden önceki ve sonraki renk dağılım durumları 28
Şekil 2.8. Orijinal resim ve görsel atak sonucu resim 30
Şekil 2.9. 1Kb veri gizlenmiş resim dosyası ve görsel atak sonucu resim 31
Şekil 2.10. 5Kb veri gizlenmiş resim dosyası ve görsel atak sonucu 31
Şekil 3.1. WAV dosya yapısı 38
Şekil 3.2. WAV dosyası örneği 41
Şekil 3.3. Veri saklama uygulaması akış şeması 43
Şekil 3.4. Veri saklama uygulaması kod parçası 1 44
Şekil 3.5. Veri saklama uygulaması kod parçası 2 44
Şekil 3.6. Veri saklama uygulaması kod parçası 3 45
Şekil 3.7. Veri saklama uygulaması kod parçası 4 45
Şekil 3.8. Veri saklama uygulaması kod parçası 5 46
Şekil 3.9. Veri saklama uygulaması kod parçası 6 47
Şekil 3.10. Veri saklama uygulaması kod parçası 7 48
Şekil 3.11. Veri saklama uygulaması kod parçası 8 48
Şekil 3.12. Veri saklama uygulaması kod parçası 9 49
Şekil 3.13. Ortu1.wav dosyasına veri gizlenmesi 51
Şekil 3.14. Ortu3.wav dosyasına veri gizlenmesi 52
Şekil 3.15. Ortu2.wav dosyasına veri gizlenmesi 53

Şekil 4.1. Pnn ağının Matlab'da kullanımı 57
Şekil 4.2. P Matris gösterimi 57
Şekil 4.3. Sıraçma uygulaması akış şeması 59
Şekil 4.4. Sıraçma uygulaması kod parçası 1 60
Şekil 4.5. Sıraçma uygulaması kod parçası 2 61
Şekil 4.6. Sıraçma uygulaması kod parçası 3 61
Şekil 4.7. Sıraçma uygulaması kod parçası 4 62
Şekil 4.8. Sıraçma uygulaması kod parçası 5 62
Şekil 4.9. Sıraçma uygulaması kod parçası 6 63
Şekil 4.10. Sıraçma uygulaması kod parçası 7 63
Şekil 4.11. Sıraçma uygulaması kod parçası 8 64
Şekil 4.12. Sıraçma uygulaması kod parçası 9 64
Şekil 4.13. Sıraçma uygulaması kod parçası 10 64
Şekil 4.14. Stego1 grubu dosyaları için sıraçma saldırı sonuçları 66
Şekil 4.15. Stego2 grubu dosyaları için sıraçma saldırı sonuçları 68
Şekil 4.16. Stego3 grubu dosyaları için sıraçma saldırı sonuçları 69
Şekil 4.17. Stego4 grubu dosyaları için sıraçma saldırı sonuçları 71
Şekil 4.18. Ortu4.wav(A) ve Ortu2.wav(B) dosyalarına veri gizlenmesi 72
Şekil 4.19. Stego5 grubu dosyaları için sıraçma saldırı sonuçları 74
Şekil 4.20. Benzer çalışma [63] hata analizi 2 sonuçları 78
Şekil 4.21. PNN ağı (B) hata analizi 2 sonuçları 78

TABLOLAR LİSTESİ

Tablo 2.1 RGB renk tablosu..... 17
Tablo 3.1. RIFF yığın tanımlayıcısı yapısı..... 39
Tablo 3.2. “fmt” alt yığını yapısı 40
Tablo 3.3. “data” alt yığını yapısı 41
Tablo 4.1. Stego1 gurubu dosyaları için PNN ağı olasılık sonuçları 67
Tablo 4.2. Stego2 gurubu dosyaları için PNN ağı olasılık sonuçları 69
Tablo 4.3. Stego3 gurubu dosyaları için PNN ağı olasılık sonuçları 70
Tablo 4.4. Stego4 gurubu dosyaları için PNN ağı olasılık sonuçları 73
Tablo 4.5. Stego5 gurubu dosyaları için PNN ağı olasılık sonuçları 74
Tablo 4.6. Ki-kare saldırısı ile PNN ağı olasılık sonuçları 75
Tablo 4.7. Ki-kare saldırısı ile PNN ağı olasılık sonuçları hata analizi 1 76
Tablo 4.8. Ki-kare saldırısı ile PNN ağı olasılık sonuçları hata analizi 2 76
Tablo 4.9. Benzer çalışma (A) ile PNN ağı (B) hata analizi 2 sonuçlarının karşılaştırılması 77

ÖZET

Anahtar Kelimeler: Steganaliz, Steganografi, Sıraçma, Sırörtme, Sayısal Ses, Veri Gizleme, Gizli Veri Sezme

Bu çalışmada bugüne kadar yapılmış olan sırörtme çalışmalarının aksine sıraçma teknikleri üzerine yoğunlaşılmıştır. Sıraçma konusunda resim dosyaları üzerine birçok çalışma yapılmıştır. Fakat ses dosyaları üzerine çok fazla çalışma bulunmamaktadır. Bu tezde ses dosyalarında sıraçma işlemleri üzerinde durulmuştur. Sıraçma saldırısında gizleme algoritmasının bilindiği sıraçma saldırı yöntemi kullanılmıştır. Bu yöntem ses dosyalarına LSB sırörtme yöntemi kullanılarak oluşturulmuş sırlı nesnelere yönelik bir saldırı şeklidir. Geliştirilen sıraçma yönteminde, ses dosyalarının son bitlerine gömülmüş veriler analiz edilerek veri çıkartma işlemi yapılmaya çalışılmıştır. Genelde sıraçma yöntemlerinde sezme (detection) yani gizli verinin varlığını anlama işlemi yapılabilmektedir. Oysa geliştirilen yöntemde gizli veri içeren dosyalar için "dosyadaki gizli veri oranı" sorusuna cevap aranmaktadır.

ANALYSIS OF STEGANOGRAPHED AUDIO FILES WITH THE GUIDE OF ARTIFICIAL NEURAL NETWORKS

SUMMARY

Key Words: Steganalysis, Steganography, Digital Voice, Data Detection, Data Hiding

In this study, we have focused on steganalysis in contrast to steganography literature. There have been many studies on image driven steganalysis, but a very few on audio-file driven steganalysis. In this thesis, we have focused on audio-file driven steganalysis studies. We have used stego-object steganalysis known attack methods through our attack algorithm. This method of audio file created using LSB steganography method is a form of an attack on the stego object. In our steganalysis methods, there has been made an analysis of data extraction process for data embedded in the last bit of audio files.

Steganalysis detection methods in general (insight) that can be done there, or do not have the data in this file can be hidden. Here, we have investigated the answers for question of 'What percentage of hidden data is in such a file'.

BÖLÜM 1. GİRİŞ

Teknolojinin son derece hızlı geliştiği günümüzde neredeyse tüm işlemler bilgisayarlar üzerinden yapılmaya başlandı. Günlük yaşantımızın bir parçası olan bilgisayarlar sürekli birbirleri ile iletişim halinde. Bugün her türlü banka, alım satım vs. gibi önemli işlemlerimizi Internet üzerinden yapmamız mümkün hale geldi. Bu tür önemli işlemlerde bilgi güvenliği ön plana çıkmaktadır. Veri transferlerinde bilgiler 3. şahıslar tarafından çeşitli yöntemler ile ele geçirilerek kötü maksatlı kullanılabilir. Bunun gibi güvenlik sorunları oluşturacak durumlara karşı şifreleme ve veri gizleme yöntemlerine başvurulmuştur. Fakat kötü amaçlı saldırılar bu ve bir takım korunma yöntemlerini de delerek güvenliği tehdit etmeye devam etmektedir.

Korunma yöntemleri içerisinde sıklıkla kullanılan şifreleme teknolojisi iletilen veriyi belirli bir yapıda bozarak 3. şahısların anlamayacağı şekle dönüştürürler. Fakat bu gibi yöntemlere karşı şifre çözme algoritmaları geliştirilmiş ve bunun yanı sıra şifreli iletişim fark edildiğinde iletişimi engelleme yollarına giden kötü amaçlı casus yazılımlar, şifreleme yönteminin de bilgi güvenliği için yetersiz olduğunu göstermiştir. Bu noktada veri gizleme (sırörtme/stegonagrafi) yöntemleri ön plana çıkmaktadır.

1.1. Sırörtme ve Yapılan Çalışmalar

Sırörtme, gizli mesajı gömme yoluyla bilgi saklama sanatı ve bilimidir. Bu yaklaşım, bir nesnenin içerisine bir verinin gizlenmesi olarak da tanımlanabilir. Bu yaklaşımla ses, sayısal resim ve video görüntüleri üzerine veri saklanabilir. Görüntü dosyaları içerisine saklanacak veriler metin dosyası olabileceği gibi, herhangi bir görüntü içerisine gizlenmiş başka bir görüntü dosyası da olabilir. Bu yaklaşımda içine bilgi gizlenen ortama örtü verisi (cover-data), oluşan ortama da sırlı-metin (stego-text) veya sırlı-nesne (stego-object) denmektedir.

Sırörtmenin geçmişi binlerce yıl öncesine dayanmaktadır. Steganografi kelimesinin kökleri "στεγαυος" ve "γραΦειν"'den gelen Yunan alfabesinden türetilmiştir [1]. Kelime anlamı olarak gizli yazı veya örtülü yazı anlamına gelmektedir [2]. Amacı gizli iletişime olanak sağlamaktır. Sırörtmede, sayısal veri başka bir sayısal veri içerisine fark edilmeyecek şekilde gömülmektedir. Mesala gizli bir metin resim dosyasının içerisine gömülmekte ve gömülü resim orjinalinden ayırt edilemeyecek hale getirilmektedir. Bu şekilde haberleşen iki kişi arasında bilgi iletişimi son derece gizli olmakta ve bu iletişimi gözetleyen kişiler, arada bir iletişim kanalı olduğunun farkına bile varmayacaklardır. Bu şekilde görünmez bir iletişim kanalı oluşmaktadır.

Sırörtme her türlü dosya ortamına uygulanabilmektedir. Bu bir resim dosyası olabileceği gibi ses dosyası ve video dosyası da olabilir. Burada kısıtlama yoktur. Veri gömme yöntemlerinin temel mantığı, sayısal dosyalarda ihmal edilebilecek önemsiz verilerin yerine gizli iletişimde kullanılacak olan veri parçalarının gömülmesini esas almaktadır. Böylece bu küçük değişiklikler kullanıcılar tarafından fark edilemez ölçüde olacaktır. Bu gibi sistemler insan duyu organlarının zafiyetinden yararlanılarak düşünülmüş ve geliştirilmiştir.

İnsan Duyu Organları (İDO) içerisinde görme sistemi resim dosyaları üzerindeki küçük ihmal edilebilecek farklılıkları algılamamaktadırlar. Örneğin bir JPEG dosyasındaki her bir piksel aslında bir veri barınağı olarak kullanılabilir. JPEG dosyasını oluşturan piksellerin resmin bütünlüğü açısından küçük olmalarına rağmen önem taşımaktadırlar. Pikseller boyutlarına göre çok büyük olan JPEG resminin temelini oluştururlar. Her bir pikseldeki ihmal edilebilecek ölçüdeki bitlere saklanacak olan verinin bitleri gömülmek suretiyle orjinalinden boyut ve görünüm olarak hiçbir fark olmayan resim dosyaları elde edilebilinir. Bunun yanında resim dosyaları temel alınarak geliştirilen video dosyalarına da sırörtme uygulamak mümkündür. Video dosyalarında her bir çerçeve aslında bir resim dosyasını barındırmaktadır. Bu da her bir çerçevedeki resim dosyasına gömülecek veri anlamına gelmektedir. Yani video dosyasında ne kadar çok çerçeve var ise o kadar çok veri gömülebileceği anlamına gelir. Video dosyaları veri barındırma açısından çok zengin durumdadırlar. Son yıllarda bu dosyalara da veri gömme yöntemleri denenmiş ve olumlu sonuçlar elde edilmiştir [3].

Yine İDO işitme sistemi görme sisteminde olduğu gibi sesteki ufak çaptaki değişimleri fark edemez. Örneğin sayısal ses örneklerindeki küçük yükseliş ve alçalışları algılayamaz. Sırörtme teknikleri yine işitme sistemindeki bu zafiyetten yararlanarak, sayısal ses örneği değerlerini, veri gizlemek amacıyla değiştirebilirler. Ayrıca sayısal sesin sıkıştırılarak başka bir formata çevirilmesi sırasında (örneğin MP3) ses örneklerinin en az önemli bitlerinin ihmal edilerek, gizli verinin bitlerini gömmek süretiyle veri gizleme yapılabilir. Ses sırörtmesinde WAV dosyaları en çok kullanılan dosyalardandır. Mono ve Stereo olmak üzere iki farklı yapıdadır ve mono sesler 8-bit, stereo sesler 16-bit ile temsil edilir. Günümüzde kullanılan WAV dosyaları genellikle 16-bitliktir. Bu ses örneklerinin son bitlerinindeki artış veya azalışların sebep olacağı farklılık işitme sistemi tarafından algılanamaz. Bu durum, WAV dosyalarına veri gizleme imkânı sunmaktadır.

1996 yılına kadar hem akademik hem de endüstriyel çevrelerde sırörtme, şifrelemeye göre daha az ilgi çekmiştir. Ancak 1996'da organize edilen ilk sırörtme konferansıyla birlikte bu durum hızla değişmeye başlamıştır [4]. Bu kazanılan ivme ile sırörtme çalışmaları birçok sayısal verilere uygulanmaya başlanmıştır. Resim, ses dosyalarının yanı sıra video dosyalarına da son derece iyi bir performansla veri gömülebilinmektedir [3]. Literatürde resim ve ses dosyaları üzerinde birçok sırörtme çalışmaları yapıldığı gözlenmiştir.

Kerckhoffs A. (1883) çalısmasında, çogunlukla kriptografi üzerinde durulmuş ve sırörtme sistemlerinin tasarımı ile ilgili bilgiler verilmiştir[5]. Briquet C. (1907) damgalamanın tarihsel sözlüğüdür [6].

Sırörtme uygulamalarının yapılabilmesi için mutlaka taşıyıcı veri kullanılması gerekmektedir. Bunlar ses, resim, video vb. olabilir. Bunlara olarak, Adli ve Nakao (2005), MIDI ses dosyaları içerisine, LSB yöntemiyle, tekrarlanan komut kodları algoritması ve sistem harici kodları algoritmasını kullanarak veri saklayan bir çalışma sunmuşlardır [7]. Xu ve arkadaşları da sıkıştırılmış video görüntülerine sırörtme algoritması önermişlerdir [8].

Christian Kratzer, Jana Dittmann, Thomas Vogel, Reyk Hillert 2008 yılında yayınladıkları makalede, iki kullanıcı arasında sesli görüşme yapılırken sessizlik

algılaması (Silence Detection), şifreleme ve sırörtme yöntemlerini kullanarak gizli bilgi gönderme işlemini gerçekleştirmişlerdir. Gönderilecek bilgiler, ses bilgilerinin son bitlerine (LSB) gömülmüştür [9].

LiWu Chang and Ira S. Moskowitz, ses içerisine veri gömme uygulaması yapmak için 4 farklı yöntem incelemişler ve bu yöntemler arasında birim zamanda gönderilecek veri miktarı bakımından en iyi yöntemin düşük değerlikli bitlere veri gömülmesi olduğunu göstermişlerdir. Ancak düşük değerlikli bitlere veri gömmenin kısmi güvenlik sağlayacağını göstermişlerdir [10].

Muzak şirketinin, 1954 yılında müzik kayıtları içerisine sahiplik bilgisini içeren kod yerleştirmek için almış oldugu patentle birlikte, telif haklarının korunmasına yönelik ses bilgileri içerisine veri gömme teknigi üzerine çalışmalar yoğunlaşmıştır. Bu durumun sadece kayıtlı ses verilerine değil, gerçek zamanlı ses verilerine de uygulanabileceği tartışılmaktadır. Örneğin hava trafik kontrolünde daha güvenli iletişim için ses bilgileri içerisine veri gömülmesinden bahsedilmekte ve bu da Data in Voice (DIV) olarak adlandırılmaktadır [11].

1990'ların başında imge damgalama kavramı gelişmiş; Tanaka ve arkadaşları faks gibi ikili imgelerin korunması kavramını ortaya atmışlardır [12]. 1993 yılında Tirkel ve arkadasları gerçekleştirdikleri veri gömme tekniğine, daha sonra "watermark" olarak birleştirilecek olan "water mark" ismini vermişlerdir [13].

Müzik dosyalarının telif haklarının korunması için "Audio Watermarking" adı altında yapılan çalışmalar gömülü verinin sezilememesini amaçlamıştır. Dünyaca telif haklarının korunması ve düzenlenmesi ile ilgili çalışmalar yapan ve hükümetler üstü bir kuruluş olan WIPO (World Intellectual Property Organization) sayısal veri gömme sistemlerinin yasal alanlarıyla ilgili çalışmalarını sürdürmektedir [14].

Sağıroğlu ve Tunçkanat (2002), gri seviyeli bitmap resimleri içerisine, görsel olarak fark edilmeksizin, en önemsiz 4. bit seviyesine kadar, LSB modifikasyonu yöntemiyle veri saklanabileceğini gösteren Türkçe bir yazılım geliştirmişlerdir [15].

1.2. Sıraçma ve Yapılan Çalışmalar

Sıraçma ise sırörtmenin gelişmesi ile birlikte ortaya çıkan bir kavramdır. Sırörtme yöntemiyle yapılan gizli iletişimin çözülme isteği doğmuştur. Bu nedenle sıraçma kavramı ön plana çıkmaktadır. Sıraçmanın amacı sırörtme yöntemi ile gizlenen veriyi tespit etmeye çalışmaktır. Sıraçma yöntemi taşıyıcı dosya ile orjinal dosyayı birbirinden ayırt etmeye çalışır.

Sıraçma çalışmalarının temelinde her ne kadar İDO tarafından ayırt edilemez olsa da aslında sırörtme sonrasında değişen bitlerin bir parmak izi misali iz bıraktığını düşünerek bu izleri takip etme ve bu izlerden istatiksel veya analitik bir sonuç çıkararak gerçek nesne ile örtülü nesneyi birbirinden ayırt etme düşüncesi yatmaktadır.

Literatürde resim ve ses dosyalarına birçok sıraçma çalışması yapıldığı tesbit edilmiştir.

Cheng ve arkadaşları elektronik metin resimleri için ilgili resim içerisinde veri gömülü olup olmadığını sezen bir sıraçma tekniği geliştirmişlerdir [16].

Fridrich ve ark. (2000), 24-bitlik renkli resimlerde LSB yöntemi kullanılarak gerçekleştirilen veri saklamayı tesbit edebilmek için, birbirine yakın renk değeri çiftlerinin istatistiksel analizine dayanan bir çalışmada, tekil renk oranının toplam piksel sayısının %30 undan az olduğu resimlerde güvenilir sonuçlar verebilen bir yöntem sunmuşlardır. Bu yöntemin sıkıştırılmamış yüksek çözünürlüğe sahip, yani günümüzdeki sayısal fotoğraf makinelerinin çektiği resimlerde başarı sağlanamayacağını ve ayrıca gri seviyeli resimlere bu yöntemin uygulanamayacağını bildirmişlerdir [17].

Farid (2001), sırlı resimlerden oluşan bir veritabanı üzerinde yaptığı analizlerden ve özellik çıkarımlardan sonra sırörtme algoritmasının bilinmesine gerek kalmadan tahminde bulunan evrensel tespit çalışmasını sunmuştur [18].

Provos ve Honeyman (2001), JPEG resimlerde sıraçma üzerine kapsamlı bir çalışma yapmışlardır ve geliştirdikleri yazılımla Internet üzerinde binlerce şüpheli resmi tespit edebilmişlerdir [19].

Özer ve Sankur (2004), algısal kıyım fonksiyonu ile ses dosyalarında sezme işlemi yapmaya çalışmışlardır [20].

Fridrich (2006), veri saklama sonucu oluşan değişim ile değişim sayısı arasındaki ters orantıyı analiz ettiği çalışmasında, herhangi bir tespit edilebilirlik profilinde, (tespit edilebilirlik profili gömme etkisinin dağılımını ifade etmektedir) sadece en küçük gömme etkisine sahip piksellerin kullanılmasının uygun olmadığını belirlemiştir [21].

Özer ve ark. (2006), veri gizleme algoritmasına ihtiyaç duyulmaksızın, ses dosyaları içerisinde gizli verinin varlığını evrensel bir şekilde tespit etmeye yönelik yaptıkları çalışmada %75 ile %90 arasında değişen bir başarı yüzdesi elde etmişlerdir [22].

Avcıbaş (2006), içeriğe bağımsız bozulma ile seste sıraçma adı altında ses dosyaları üzerinde sezme işlemi yapmaya çalışmıştır [23].

Ru ve ark. (2006), mevcut bazı sırörtme yazılımlarıyla WAV dosyaları içerisine yapılan gömme işlemini tespit etmeye yönelik bir yöntem önermişlerdir [24].

Qingzhong Liu ve ark.(2006) WAV ses dosyalarındaki gizli bilgi sezimi ile ilgili bir çalışma yapmışlardır [25]. Yine Qingzhong Liu ve ark.(2009) WAV ses dosyalarına spektrum sıraçmayla gizli bilgi sezimi ile ilgili bir çalışma yapmışlardır [26].

Görüldüğü gibi geçmişte ses dosyaları üzerine veri saklama teknikleri üzerinde çalışılmış ([6],[7],[8],[9],[10],[11]) ve bu gibi çalışmaların sayısı son yıllarda artış gösterse de, resim dosyaları içerisine veri saklayan çalışmaların sayısına göre daha azdır. Ülkemizde de sırörtme alanında çalışmalar yapılmaktadır. Gri seviyeli bitmap resim dosyaları içerisine veri saklayan Türkçe yazılım [15], renkli bitmap resim dosyaları içerisine veri saklayan ve bu hizmeti Internet üzerinden sunan Türkçe web sitesi [27] ve ses dosyalarına veri gizleyen Türkçe bir yazılım [65] bulunmaktadır.

Ülkemizde Internet kullanımı, gelişen Internet altyapı sistemleri ile yaygınlaşmaktadır. Bu gelişmiş teknolojinin sunduğu imkânlar doğrultusunda ses dosyaları ile iletişim de çok fazla kullanılır hale gelmiştir. Bu maksatla, ses dosyalarının içerisine gizli veri saklama yöntemlerinin kullanım alanları artmaktadır. Ülkemizde bu yönde çalışmaların sayısı kısıtlıdır. Bu tezle bu alandaki boşluğun doldurulması amaçlanmıştır.

Tezin amacı; sıraçma tekniklerini incelemek ve ses dosyaları üzerine yapılan sırörtme çalışmalarının analizini yaparak tesbitlerde bulunmak, yapılan tesbitlerin doğruluğunu ve başarımlarını ölçmek, elde edilen tesbitlerin güvenlik açısından nasıl kullanılabileceğine dair bilgiler sunmaktır. Ülkemizde ses dosyaları üzerine sırörtme uygulamaları henüz yeterli derecede olmadığı görülmüş fakat gelişen Internet altyapısının sonucu olarak ileride bu alanlarda çalışmalara ihtiyaç duyulacağı düşünülerek ses dosyaları üzerinde sıraçma yapan bir yazılım geliştirilmiş ve bu yazılımın sonuçları sunulmuştur.

Tez çalışmasında LSB yöntemi ile ses dosyalarının son bitlerine gizli mesaj gömülmüş ve Ki-kare yöntemi ile gizli verinin varlığı analiz edilmiştir. Ki-Kare'nin vermiş olduğu sonuçlar daha önce eğitilmiş olan PNN yapay sinir ağına verilerek ses dosyasına hangi oranda veri gizlenildiği bulunmuştur.

Bu tez çalışması beş bölümden oluşmaktadır. Aşağıda tez organizasyonu anlatılmaktadır.

Birinci bölümde üzerinde durduğumuz konu hakkında ön bilgiler, kavramlar, geçmişte yapılan çalışmalar ve tezin yapılma nedenleri üzerinde durulmuştur.

İkinci bölümde sırörtme ve sırörtme alanının aksi yönünde araştırmalar yapılan sıraçma konusunun tanımı, geçmişi, yöntemleri ve dünyadaki bu iki alanda yapılan çalışmalar anlatılmaktadır.

Üçüncü bölümde, yapay zekâ yöntemleri kullanılarak sıraçmak amacıyla geliştirilen uygulamaya veri sağlamak için kullanılacak olan WAV ses dosyalarına gizli veri gömme yöntemimiz ve veri gizleme uygulamasının kodları açıklanmıştır.

Dördüncü bölümde, tezin asıl amacı olan yapay zekâ yöntemleri ile sıraçma uygulamamız detaylı olarak anlatılmakta ortaya çıkan sonuçlar üzerinde durulmaktadır. Ayrıca uygulama kodları da açıklanarak anlatılmıştır.

Beşinci bölümde, tesbit edilen sonuçlar irdelenmiş ve gelişmeye açık olan hususlar belirtilmiştir.

BÖLÜM 2. SIRÖRTME - SIRAÇMA KAVRAM VE TEKNİKLERİ

2.1. Sırörtme (Steganografi)

Sırörtme, gizli veriyi ihmal edilebilecek küçük bilgi parçacıklarına saklama sanatı ve bilimidir. Bu kavram, bir verinin içerisine başka bir verinin gizlenmesi olarak da tanımlanabilir. Bu yöntemle birçok farklı ortama veri saklanabilinmektedir. Resim, ses ve video ortamlarına son derece güvenli veri saklayabilen bu yöntemde, resim dosyaları içerisine saklanacak veriler metin dosyası olabileceği gibi, herhangi bir resim dosyası içerisine gizlenmiş başka bir resim dosyası da olabilir. Bu yaklaşımda, içine bilgi gizlenen ortama örtü verisi (cover-data), oluşan ortama da sırlı-metin (stego-text) veya sırlı-nesnesi (stego-object) denmektedir. Lau Stephan'ne göre sırörtme, "görünüşte zararsız olan haberleşmenin içine gizli haber ilave ederek haberleşmektir" [28].

Sırörtmede amaç, bir mesajın varlığını gizlemek ve bir örtülü kanal (covert channel) ya da örtülü iletişim ortamı oluşturmaktır. Bu yüzden sırörtme, mesajın içeriğini saklamaya çalışan şifrelemenin (kriptoloji) bir parçası olarak görülebilir. Aslında şifreleme mantığına hiç de benzemeyen bir sistemde çalışır. Şifrelemede verinin varlığı ortadadır ve bu veri her türlü saldırıya maruz kalabilir. Yani açık erişime sahip bu veriye erişmek isteyen şifre kırıcı, bu verinin üzerinde denemeler yaparak şifreyi kırmaya çalışabilir. Oysaki sırörtmede gizli verinin varlığı bilinmemektedir. Yani gizli verinin varlığı herhangi bir çalışma yapılmadan bilenemez. Bu da gizleme yöntemini kırmak isteyenler için çok büyük bir engeldir. Gizli veriye ulaşmak için öncelikle elinizde gizli veri olup olmadığını bilmeniz gerekir. Bununla birlikte şifreleme algoritmaları gizlenecek olan veriye uygulanarak daha da güvenli bir mekanizma oluşturmak mümkündür. Böylece hem gizli veri ortada yoktur hem de gizli veriye ulaşılsa bile şifreli veriye ulaşılmış olunucaktır. Böylece gizli veriye erişmek daha da güçleştirecektir.

2.1.1. Sırörtmenin başlangıcı

Steganografi kelimesi Yunanca “steganos: gizli, saklı” ve “grafi: çizim ya da yazım” kelimelerinden gelmektedir. Yunanca bir kelime olan steganografinin tam karşılığı olarak “covered writing (gizlenmiş yazı)” diyebiliriz.

Sırörtme, Antik Yunan zamanına kadar uzanan oldukça eski bir veri gizleme yöntemidir ve bugün kullanılan pek çok orijinal özelliği Antik Yunan medeniyetinden gelmektedir.

İran savaşları sırasında, Herodot, kafasını kazıtıp kafa derisinin üzerine, gizli mesajın dövmesinin yapılmasına izin veren bir ulaktan bahseder. Mesaj yazıldıktan sonra ulak saçının uzamasını bekler. Daha sonra mesajı bekleyen kişiye ulaşır ve kafasını tekrar tıraş eder. Böylece gizli mesaj ortaya çıkar. Bu, sırörtmenin tarihte ilk kullanımıdır [29].

Sparta ve Xerxes arasındaki savaş esnasında, Dermetaus, Xerxes’in işgal için beklemede olduğunu Sparta’ya haber etmek istedi. Bunun yapabilmek için de, tahta tabletlerin üzerine mesajını yazarak bunları balmumu ile kapladı. Balmumu ile kaplı olan tahtalardan hiçbir şey gözükmediği içinde, nöbetçiler hiç kuşkulanmadı. Bundan sonraki zamanlarda da özellikle savaşlarda bu sırörtme tekniğinden oldukça yaralanılmıştır [30].

2.1.2. Literatürde yapılmış sırörtme çalışmaları

İskoçya kraliçesi Mary Queen mektuplarını gizleyebilmek için şifreleme ve sırörtme tekniklerini kullanmışlardır. Yazdığı mektupları bir bira fıçısının deliğinde gizlemiş ve kendisi o sayede hapishaneden kurtulmuştur [32].

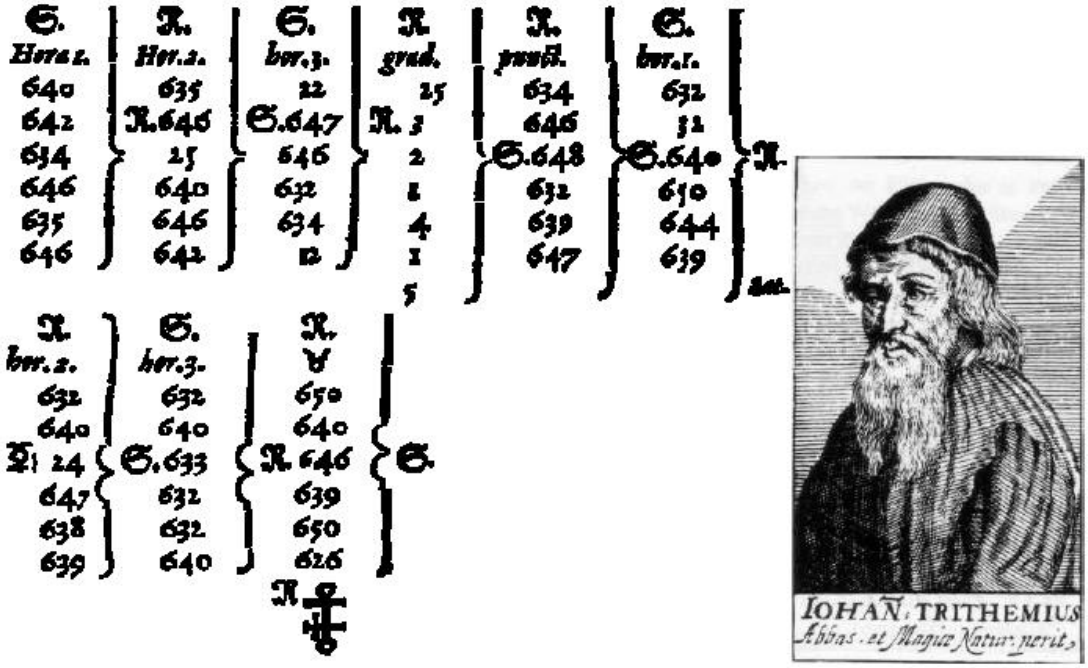
II. Dünya Savaşı’nda Amerika Birleşik Devletleri Deniz Kuvvetleri Navajo şifre konuşucularını kullandı. Şifre konuşucular basit bir şifreleme tekniği kullanarak, mesajları açık bir metin içerisinde göndermekteydi.

Aşağıda, II. Dünya Savaşında kullanılan bir sırörtme örneği verilmiştir [31]. “Apparently neutrals protest is thoroughly discounted and ignored. Isman hard hit.

Blockade issue affects pretex for embargo on by-products, ejecting suets and vegetable oils."

Yukarıda verilen paragrafta her kelimenin ikinci harfleri yan yana getirildiğinde "Pershing sails from NY June 1." mesajı ortaya çıkmaktadır.

Gina Kolata "Thomas Ernst adlı araştırmacı, Şekil 2.1'de verilen kitabın gizli bir şeyler bardındırdığını düşünerek bir çalışma yaptı. Bu konuda 200 sayfalık Almanca bir rapor yayınlamış, 1996'da Hollanda dergisi Daphnis'te yayımlanmış fakat çok az ilgi görülmüştür" [34]. Jim Reeds, AT&T laboratuarlarında bir matematikçi iken Trithemius'un bu kitabı üzerinde çalışmaya başlamıştır. Trithemius'un çalışmaları üzerine bilgi toplarken Ernst'ın raporunu ortaya çıkarmıştır.



Şekil 2.1. Johannes Trithemius'un sırörtme ile ilgili 3.kitabında yer alan sayı çizelgelerinden bir örnek [33].

Bu iki araştırmacı, kitabın içerisinde gizli mesajların olduğunu keşfettiler. Mesajlardan bir tanesinde "Hızlı kahverengi tilki tembel köpeğin üzerinden zıpladı" denilmektedir. İkinci mesaj, "Bu mektubu taşıyan kişi dolandırıcı ve hırsızdır Kendini ona karşı koru. Sana bir şeyler yapmak istiyor." ve üçüncü mesaj ise İncilin 23.bölümünün birinci kısmını içermektedir [64].

Bender ve arkadaşlarının (1996) kaleme aldığı makalede, resim, ses ve metin gibi dosya türleri içerisine veri saklama teknikleri detaylı bir şekilde açıklanmıştır [35]. Ses içerisine, yayılmış spektrum (spread spectrum) faz kodlaması (phase coding), düşük bit kodlaması (Low Bit Encoding) ve yankı veri saklaması (echo data hiding) yöntemleriyle veri saklanması ve metin içerisine boşlukların kullanımı, konuşma dilinin yapısı ve eşanlamlı kelimelerden faydalanarak veri saklama yöntemleri bu çalışmada detaylı olarak ele alınmıştır.

2000'li yıllardan sonra sırörtme, LSB (en önemsiz bit) yaklaşımıyla [36], dönüştürme tekniğiyle ([37],[38]), permutasyon tekniğiyle [39] ve BPCS (Bit Plane Complexity Segmentation) yaklaşımıyla [40], resim içerisine veri saklama çalışmaları gerçekleştirilmiştir.

Lee ve Chen (2000), LSB yöntemiyle ve anahtar kullanılarak, gri seviyeli resimlerde, piksel değerini oluşturan bitlerin ilk dördünün modifikasyonu ile %50 ye yaklaşan kapasiteyle veri saklamışlardır [36].

Noda ve ark.(2002), kayıplı sıkıştırma gerçekleştiren resimler üzerinde BPCS yöntemini kullanarak veri gizleyen diğer bir çalışmadır [37]. Bu çalışmada, sıkıştırma işlemi sırasında wavelet katsayılarının niceleme (quantization) işlemiyle bit düzlemine çevirilmiş hali üzerinde BPCS yöntemiyle veri saklama yapılmıştır ve %9 ile %15 arasında değişen kapasitelerde veri gizlenebilme başarımı elde edilmiştir.

Niimi ve ark. (2002), BPCS yöntemini temel alarak palet tabanlı resimler içerisine veri saklayan ve paletteki renk vektörlerinin sırasına bağlı olmayan bir metot öne sürmüşlerdir [40].

Akleylek ve Nuriyev 2005 yılında geliştirdikleri çalışmalarında, açık anahtar ve gizli anahtar çiftiyle çalışan bir şifreleme sistemi oluşturmuşlar ve bu sistemle resim dosyalarının LSB bitlerini değiştirerek veri saklayan bir yöntem geliştirmişlerdir. "AS knapsack" ismini verdikleri şifreleme sistemi sayesinde göndericinin alıcıya yolladığı veriyi kabul etmemesi durumu engellenmiş ve alıcının göndericinin ilettiği gerçek veriyi görebilmesi sağlanmıştır. Fakat saklama sonrası oluşan resim ile orijinal resim arasındaki fiziksel boyutun farklı olduğunu belirtmişlerdir [38].

Shahreza (2006), cep telefonlarındaki kısa mesaj servisini kullanarak burada kullanılan siyah beyaz resimlere metin gizleyerek SMS yoluyla gizli mesajlaşmayı sağlayan bir çalışma sunmuştur [41]. Siyah beyaz resimlerin veri saklama güvenliği açısından renkli resimlere oranla hem daha az güvenli hem de az veri barındırma olumsuzluklarına rağmen SMS servisi herkes tarafından rahatlıkla kullanıldığı için, İngiltere'de yakın zamanda hayata geçirilen gizli metin SMS fonksiyonuna göre, alınan mesaj 40 saniye içerisinde okunarak silinmesi ve bu servisi kullanmak için WAP servisine ihtiyaç duyulması gibi maliyetli ve bir o kadar da külfetli olduğundan siyah beyaz resimlere veri gizleme tekniği çok daha tercih edilir bir yöntemdi.

Resim dosyalarında yapılan birçok çalışma bulunmasına rağmen ses dosyaları üzerinde daha az çalışma yapılmıştır. Fakat son yıllarda ses dosyalarına veri gizleme uygulamaları artmaktadır. Bu uygulamalar veri gizleme tenikleri LSB modifikasyonu ile ([32],[43]), dönüştürme tekniğiyle, tekrarlanan komutların varlığından faydalanarak ve sıkıştırma esnasında ses içerisine veri gizleme gibi yöntemleri kullanılarak geliştirilmiştir.

Cvejic ve Seppanen (2002), LSB yöntemi ile ses dosyalarına veri saklama kapasitesini % 33 oranında artış sağlayan bir teknik geliştirmişlerdir [42].

Gopalan (2003), yine LSB yöntemini ile ses dosyaları içerisine veri saklanması üzerine çalışma yapmış ve kokpit sesi gibi ses dosyalarının içerisine daha fazla veri saklanabileceğini tesbit etmişleridir [43].

2.1.3. Sırörtmenin sayısal ortamda kullanım alanları

Sırörtme yukarıda da bahsedildiği gibi çok fazla alanda kullanım imkânı olan bir bilim dalıdır. İnsan yaşamının her kesiminde sırörtme kullanılabilir. Bu insanoğlunun hayal gücüne bağlıdır. Zaten sırörtmenin temel çıkış noktası da insandır. İnsanoğlu düşünme becerisi ile yeni iletişim kanalları ararken sırörtmenin temelleri oluşmuş ve bunu örnek alan diğer insanlar da metodu farklı alanlara uyarlarak gelişmesine katkıda bulunmuşlardır.

Ülkemizde sırörtme henüz çok fazla bilinmemektedir. Bu tezin oluşumunda çoğu kişinin bu konu hakkında ortalama bir bilgiye sahip olmadıkları görülmüştür.

Şifreleme gibi konular ile ilgilenen kişiler tarafından bilinen veya az çok güvenlik ile bilgisi olan kişiler tarafından bilinen bir konudur. Ülkemizde bu alanın daha çok uygulama yapılarak daha da yaygınlaşması sağlanabilir. Devletin önemli mercilerinde de bu konuda çalışmalar yapılarak devlet yararına gizli bilgi taşıma amaçlı bu yöntem kullanılabilinir.

Bilgisayar teknolojisinin gelişmesiyle birlikte bilgisayar her alanda kullanılmaya başlanmıştır. Bunun yanı sıra çeşitli bilim dalları da bu durumdan nasibini almış ve bilgisayar ortamına girmiştir. Sırörtme alanı da sayısal ortamlarda yerini almıştır. Bu kısımda sırörtmenin sayısal ortamlarda nerelerde kullanıldığa değinilecektir.

2.1.3.1. Metin tabanlı sırörtme

Metin içerisine birçok farklı yöntemlerle gizleme işlemi yapılabilir. Gizleme yöntemleri 3 ana çatıda toplanmıştır. Bunlar aşağıda listelenmiştir;

- Açık Alan Yöntemleri (Open Space Methods)
- Yazımsal Yöntemler (Syntactic Methods)
- Anlamsal Yöntemler (Semantic Methods)

Bu yöntemlerden Açık Alan Yöntemleri kelimeler arasındaki boşluklardan ve satır sonlarındaki boşluklardan yararlanarak geliştirilen tekniklerdir. Bu tekniğin üç farklı uygulama şekli vardır bunlar:

- Kelime Kaydırma Kodlaması
- Satır Kaydırma Kodlaması
- Gelecek Kodlaması

Kelime kaydırma kodlaması yönteminde, metnin satırları yatay olarak kaydırılarak dokümanın tek olarak kodlanması sağlanır. Gömülmüş kelime yine metin dosyası ya da BMP dosyası olarak açılabilir. Bu yöntem, dokümana uygulandığında yakın kelimeler arasında çok ta fark edilmeyen boşluklar ortaya çıkmaktadır. Bu oluşan

boşluklardan dolayı dokümanın kodunun çözülmesi için eski belgeye de ihtiyaç vardır.

Satır kaydırma kodlaması yönteminde metin satırları düşey olarak kaydırılarak gömülecek mesajın kodlanması sağlanır. Gömülmüş kelime yine metin dosyası ya da Windows Bitmap (BMP) dosya olarak açılabilir. Kaydırma esnasında bu durumdan faydalanarak gizlenecek metin 0 ve 1 ile kodlanarak metin içersine gömülür.

Gelecek kodlaması yönteminde ise hem metin belgelerine hem de bitmap dosyalara uygulanabilmektedir. Burada kelimelerin yerleri ve bazı harflerin boylarıyla oynanmakta ve ASCII kodlarında değişiklik yapılmaktadır.

Metin içerisine veri gizleme yöntemlerine sadece bunlarla sınırlı değildir, bir örnek verecek olursak bir metindeki kelimeler, ilk harfleri yan yana geldiğinde anlamlı bir mesaj oluşturacak şekilde sıralanarak da mesaj saklanabilir. Bununla ilgili olarak, II. Dünya Savaşında gercekleşmiş bir örnekten daha önce bahsedilmişti. Metin içerisine veri saklanması ile ilgili örnek olarak spam mimic adlı bir web sitesi verilebilir [44]. Bu sitede gizlenme istenen mesaj spam maile dönüştürülmektedir.

Spam maillerin azizliğine yakalansak ta spam mailler aracılığı ile de bilgi gizleme yöntemi kullanılabiliyorsa bu durumda faydaları da yadsınamaz. Saklanan cümlenin karakterlerinin Ascii kodlarının bit değerlerine göre spam metin içerisindeki boşlukların ayarlanmasıyla saklama işlemi gercekleştirilmektedir. Fakat metininin ne olursa olsun çıktı benzer olmaktadır. Bu da sitenin gizleme amaçlı olarak hep aynı metni kullandığını ortaya çıkarıyor. Bu yöntemde gösteriyor ki sırörtülü uygulamalarda spam e-postaların iyi bir taşıyıcı olduğu değerlendirilmektedir [45].

Yukarıda anlatılanlardan anlaşılacağı üzere metin dosyalarına veri gizleme yöntemleri oldukça çeşitlidir. Fakat veri gizlenebilirlik konusunda düşük seviyede kaldığı yadsınamaz bir gerçektir.

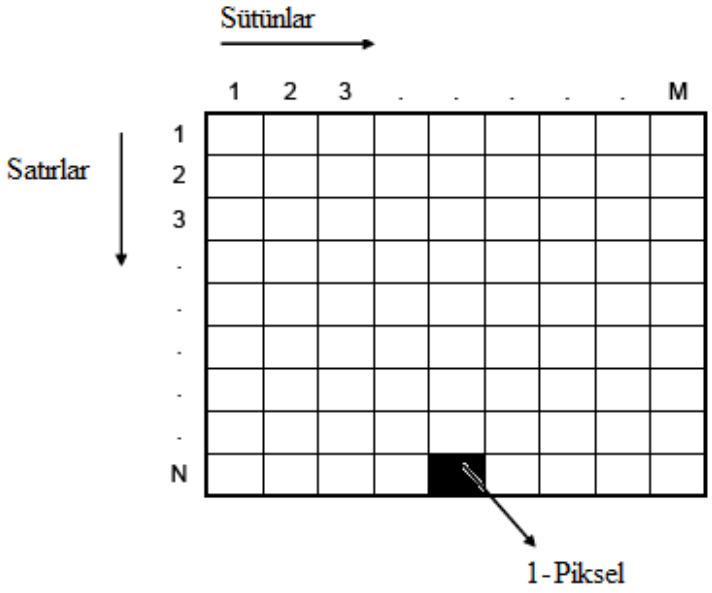
2.1.3.2. Resim tabanlı sırörtme

Resim dosyaları Internet ve gelişen teknolojik iletişim kanalları ile çok fazla kullanılan dosya tiplerindendir. Resim dosyaları, farklı formatlarına farklı sırörtme

yöntemleriyle veri gizlenebilmesinin yanı sıra sırörtmenin en çok uygulandığı ortamlardandır. Bu nedenle, sırörtme konusunda yapılan çalışmalar ve geliştirilen teknikler ağırlıklı olarak resim sırörtme çerçevesinde yer almaktadır.

Görüntü dosyalarının içerisine bir metin gizlenebileceği gibi bir resim dosyasının içine bir başka resmi de gizlemek mümkündür. Gizli bilgiyi bir resme gömme (yada gizleme) işleminde iki dosya söz konusudur. Kapak resim ya da örtü verisi (cover image) olarak adlandırılan ilk dosya, gizli bilgiyi saklayacak resim dosyasıdır. İkinci dosya ise gizlenecek bilgi olan mesajdır. Bu mesaj da sır olarak isimlendirilmektedir. Mesaj, açık metin (plain text), şifreli metin (chipher text), resim veya bit dizisi içinde saklanabilecek herhangi bir sayısal veri olabilir. Gömme işlemi sonucunda kapak resim ve gömülü mesajın oluşturduğu dosyaya "sırlı resim" adı verilir.

Sayısal resimler aslında çok küçük noktalardan(piksel) oluşur. Bu noktaların bütünlüğü ile bir resim görüntüsü elde edilir. Bu noktalar aynı zamanda içerisinde gerekli renk bilgilerini barındırırlar. Bunların sayısı ne kadar çok olursa görüntü o kadar ayrıntılara sahip olur. Sayısal fotoğraf makineleri gibi kayıt cihazlarında çözünürlük, ne kadar piksel algılandığını, ekranlarda ise, gösterilebilen piksel sayısı değeridir. Örnegin: Sayısal fotoğraf makinelerinde 1440x900 piksel, ekranlarda 1024x768 piksel gibi.



Şekil 2.2. Sayısal resmin yapısı

Sayısal resimler Şekil 2.2'de gösterildigi gibi N satır ve M sütunluk bir dizi ile temsil edilir. Genellikle satır ve sütun indeksleri y ve x veya r ve c olarak gösterilebilir. Renk bilgilerini taşıyan pikseller siyah beyaz resimlerde 0 ile 255 arasında değer alırlar. Bu değer 1- byte anlamına gelmektedir. Yani bir piksel 1 byte ile gösterilir. Bu piksellerden oluşan resimlere ikili (binary) resim denir. İkili seviyede değerler 1 ve 0 ile temsil edilir 1 ve 0 degerleri sırasıyla aydınlık ve karanlık bölgeleri veya nesne ve zemini (nesnenin önünde veya üzerinde bulundugu çevre zemini) temsil ederler [46]. Şekil 2.3'de de görüldügü gibi 0 ile 255 arası tüm değerler gri seviyesindedir. Fakat en uç noktalar da durum farklıdır. 0 siyah 255 ise beyazdır. Bu iki değer arasındaki tüm renkler gridir.

0	1	2	…	…	…	126	127	128	…	…	…	253	254	255

Şekil 2.3. Gri ton seviye (gray level) resimlerin renk örneği

Gri resimlerden farklı olarak renkli resimler de her piksel üç ana renk tonundan oluşan renkleri barındırır. Bu üç ana renk tonu kırmızı, yeşil ve mavidir. Her pikselde üç ayrı renk bilgisi bulunmaktadır. Bu renk bilgilerinin seviyelerine göre sonuçta farklı bir renk oluşur. Renklerin her biri için N x M lik bir diziye ihtiyaç duyulur. Bu üç renkte sayısal deger büyüdükçe renk daha koyulaşmaktadır. Üç temel renkten faydalanılarak istenilen renk oluşturulmaktadır. Her bir renk 0 ile 256 arasında değerler içerir.Bu 3 renk grubu RGB (Red-Green-Blue) olarak isimlendirilir. RGB (Kırmızı-Yesil-Mavi) renk modelinde 256 x 256 x 256 = 16.777.216 adet degişik tonda renk oluşturulabilir. Böylece belirli bir pozisyondaki pikselin Tablo 2.1'de görüldügü resmin bileşenlerinin şiddetini belirler.

Tablo 2.1. RGB renk tablosu

Renk	**R**	**G**	**B**
Kırmızı	255	0	0
Yeşil	0	255	0
Mavi	0	0	255
Fosfor	204	255	0

Resimlere farklı yöntemler ile veri gizlemek mümkündür. En önemsiz bite veri gömme yöntemi olan LSB yöntemi sırörtmede kullanılan en basit ve en yaygın yöntemdir. Mesaj, resmin her pikselinin son bitlerine gömülmek vasıtasıyla saklanır. Saklama işlemi ardışık olabileceği gibi rastgele hatta bir anahtarla da yapılabilir.

Resim içerisine LSB yöntemi ile veri saklamaya aşağıda bir örnek verilmiştir. Gizlenecek verinin 5 bitten oluştuğunu ve aşağıdaki değerlere sahip olduğunu düşünürsek:

1 – 0 – 1 –1 – 0

Verilerimizi gizleyeceğimiz örtü resmi 24-bitlik renkli Bitmap (bmp) olsun. Buna göre 5 baytlık bir veri saklama kapasitesi gereklidir. Her baytın en son biti kullanılacağından ve bir pikselde (R-G-B) 3 baytlık veri olduğundan, 1 piksele 3 bitlik veri saklanabilir. Bununla beraber, 5 bit veri gizlemek için 2 piksel yeterli olacaktır. Taşıyıcı resim dosyasının en az önemli olan bitleri sırasıyla şu şekilde olsun:

0 – 1 – 0 –1 – 1

LSB yöntemi uygulandıktan sonra bitlerin değerleri aşağıdaki gibi değişecektir:

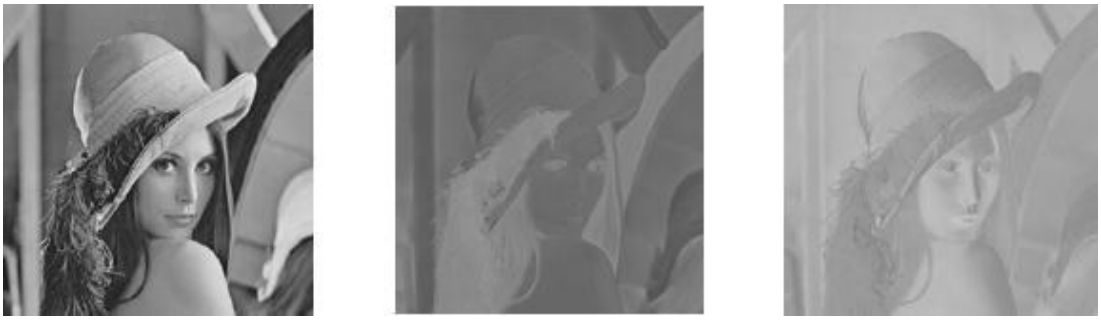
<u>1</u> – <u>0</u> – <u>1</u> –1 – <u>0</u>

Görüldüğü gibi son bitler gizlenmek istenen bitler olacak şekilde değiştirildi. Bu değişim sonucunda 4 bit değişmiştir. Altı çizili bitler değişen bitleri göstermektedir. Fakat bu değişim insan gözü ile görünüşte ayırt edilebilecek en ufak bir değişiklik yapmayacaktır. Geri getirme işleminde yeni piksel değerlerinin en az önemli bitleri sırayla okunup yan yana dizildiğinde saklanan mesaj yeniden elde edilecektir.

Bu tür yöntemler resim dosyalarındaki renklerin değerleri ile ilgilenirler. Resim dosyaları temelde resimleri sıkıştırıp sıkıştırmadıklarına göre iki gruba ayrılabilirler. Ayrıca, her resim dosyası türü renk değerlerini farklı şekilde saklı tutabilirler. En yaygın türler parlaklık bilgisinin ön planda tutulduğu YCbCr ve HSV (HSL), renklerin ön planda tutulduğu RGB'dir. Yine bu türlerin her bir renk bileşenini nasıl tuttuğu ayırt edici olabilir. Örneğin üç renk değerinin de tek bir sekizlide tutulduğu

RGB resim dosyaları ile her bir renk değerini ayrı üç adet sekizlide saklayan RGB resim dosyaları çeşitli yönden farklıdır. En önemli fark, temsil edilebilecek renk sayısıdır. Belirtilen ikinci temsil yönteminde çok daha fazla renk değerini barındıran resim dosyalarını saklamak mümkündür. Ancak, resim dosyasının bu bilgileri saklamak için çok daha fazla yere ihtiyaç duyacağı da açıktır [47].

Yer değiştirmeye dayalı sırörtme yöntemlerinin çalışma ilkesi oldukça basittir. Renk değerinin tek bir sekizlide tutulduğunu varsayalım. Bu durumda 256 adet farklı renk değerini resim dosyasında temsil edebiliriz. Adı geçen yöntemde renk değerinin en düşük anlamlı biti ile gizli verinin bitleri değiştirilir. Sonuçta ortaya çıkacak resim dosyasındaki renk değerleri ya olduğu gibi kalır ya bir artar ya da bir azalır. Ancak, her üç durumda da söz konusu olan, bu durumun insan gözü tarafından algılanamamasıdır. Şekil 2.4 ve Şekil 2.5'de Lena'nın resminin YCbCr ve HSV renk katmanları verilmiştir.



(a) (b) (c)

Şekil 2.4. Lena'nın resmi a) Y bileşeni b) Cb bileşeni c) Cr bileşeni [47]

(a) (b) (c)

Şekil 2.5. Lena'nın diğer resmi [47]
a) H bileşeni (HSV) b) S bileşeni (HSV) c) V bileşeni (HSV)

Yer değiştirmeye dayalı yöntemlerde, yer değişimi için kullanılacak bitlerin sayısı bir bitten daha fazla olabilir. Ancak, karar vermede, kullanılan resmin özelliği belirleyicidir. Resim dosyasında düz alanlar fazlaysa yapılan yer değiştirme işlemi çıplak gözle bile fark edilebilir. Öte yandan, sırörtmede unutulmaması gereken en önemli ilke, taşıyıcı dosyanın bir kere kullanılması ve sonra yok edilmesidir [47]. Yer değiştirmeye dayanan yöntemler temelde aynı ilke ile çalışmakla beraber birden fazla şekilde gerçeklenebilirler. Her bir renk değerinin değiştirildiği gerçeklemelerle beraber renk değerlerinin gruplanarak, eşlik değer olarak gizli veri bitinin kullanıldığı ve bu grubun eşlik bitine uyum sağlaması için gruptaki tek bir renk değerinin değiştirildiği yöntemler örnek olarak sayılabilir. Aralarındaki tek fark, saklanacak veri miktarıdır. Yer değiştirme yöntemine dayanan gerçeklemeler oldukça kolay bir şekilde programlanabilir. Sunmuş oldukları yüksek veri saklama miktarı ile gizli iletişim için oldukça ilgi çekicidirler. Ancak, bu yöntemlerin karşılarında iki önemli engel vardır: bunlardan ilki resim dosyaların Internet trafiğinde yaratmış oldukları yükü azaltmak için sıkıştırılmaları ve yine bu yöntemlerin sıkıştırma başta olmak üzere en hafif resim işleme yöntemlerine karşı oldukça zayıf olmaları. Gizli veri bitlerinin taşıyıcı resme gürültü olarak eklendiğini hatırlayacak olursak, parlaklığın bile değiştirilmesi bu bilgilerin tamamen yok olmasına neden olacağı açıktır.

2.1.3.3. Ses tabanlı sırörtme

Resim dosyalarına yapılan sırörtmeden sonra ses dosyaları popüler hale gelmiştir. Resim dosyalarına benzer şekilde ses içerisine veri saklama yöntemleri de insan işitme sisteminin zafiyetinden yararlanılarak geliştirilmiştir.

Daha önce bu dosyalar üzerinde pek fazla sırörtme çalışması mevcut değilken, 2000'li yılların başından itibaren bu alanda çalışmalar artmıştır. Bu konuda yapılan çalışmalarda genelde LSB yöntemi ([42],[43]) ve dönüştürme teknikleri [48] kullanılmıştır. Ses sinyallerine veri gizlemek resim piksellerine oranla daha karmaşık işlemler gerektirir. Ses içerisine veri gizleme yöntemleri: aşama kodlaması (phase coding), düşük bit kodlaması (Low Bit Encoding), yankı veri saklaması (echo data hiding) ve tayf yayılması(spread spectrum) olarak sınıflandırılmaktadır [35].

Aşama kodlaması yönteminde, ses dosyası parçalara ayrılmakta ve bu parçalara ait faz değeri, veriyi saklayacak şekilde yeniden oluşturulmaktadır. Bu yöntem, dönüştürme tekniğinde kullanılan yönteme benzemektedir. Gömme işleminde ses dosyası küçük segmentlere bölünür ve her segmente ait aşama (faz) gizlenecek veriye ait aşama referansı ile değiştirilir. Aşama kodlaması işlem sırası aşağıdaki gibidir [35]:

- Ses verisi N adet kısa segmente bölünür.
- Her segmente Discrete Fourier Transform (DFT) uygulanarak aşama ve büyüklük (magnitude) matrisleri yaratılır.
- Komşu segmentler arasındaki aşama farklılıkları hesaplanır.
- Her segment için yeni bir aşama değeri bilgi gizlenerek oluşturulur.
- Yeni aşama matrisleri ile büyüklük matrisleri birleştirilerek yeni segmentler elde edilir.
- Yeni segmentler birleştirilerek kodlanmış çıkış elde edilir.

Düşük bit kodlaması, ses örneklerinin son bitleri ile gizlenecek verinin son bitlerini değiştirme işlemi yani LSB yöntemidir. Sesin analog ortamlara girmeksizin, tamamen sayısal ortamlarda transferi durumunda kullanılabilir. İşlem sonrasında oluşan gürültüler ses dosyasında bozulmalara neden olur. Ayrıca sıraçma saldırılarına karşı dayanıksız bir yapısı vardır. Tekrar örnekleme veya kanalda oluşabilecek gürültü ile mesaj zarar görebilir veya yok edilebilir.

Yankı veri saklaması yönteminde ses sinyali üzerine yankı sesi eklenmekte ve yankının farklı gecikme değerleri kodlanarak veri saklanabilmektedir. Bu yöntem sesin kalitesini bozma konusunda sıkıntılar doğurabilir[35].

Tayf yayılması yönteminde ise sese ait frekans tayfı üzerinde veri gizlenmektedir. Bu yöntemde seste istenmeyen gürültüler oluşabilir. LSB tekniğinde bu gibi durumlar yoktur. En önemsiz bitteki değişmeler seste herhangi bir bozulma yapmaz[35].

Farklı ses dosya türlerine sırörtme teknikleri uygulanabilinmektedir. Bunlardan en çok kullanılanı WAV ve MP3 ses dosyalarıdır. Bugüne kadar yapılan çalışmalar genellikle WAV dosyaları üzerinde yoğunlaşmıştır. Çünkü WAV dosyaları ham yani işlenmemiş verileri barındırmaktadır. Herhangi bir dönüşüm işlemi bu dosyalarda

daha kolay olmaktadır. Oysa MP3 gibi özel formatlara sahip dosyalar işlenmiş ve sıkıştırılmış verileri içerir. Bu gibi dosyalarda yapılacak dönüşüm işlemleri öncesinde açma (decompres) işlemlerinin uygulanması gerekmektedir. Fakat MP3 dosyaları, kullanımı giderek yaygınlaşan sıkıştırılmış ses dosyalarından birisidir. Bu açıdan sırörtme uygulamarlında tercih edilebilir. MP3 dosyaları içerisine veri saklayabilen az sayıda uygulama mevcuttur. Bunların en bilineni Mp3Stego yazılımıdır[66]. Bu yazılım sıkıştırma esnasında MP3 dosyaları içerisine veri saklayabilmektedir.

WAV, AU, AVI ve MPEG formatları gibi dosyaları kullanan çeşitli programlarda vardır. Bu uygulamalar verileri saklama, şifreleme, dosyalama işlemlerinden ibarettir ve bunlar dosyanın sonuna gizlenmiş bilgileri ekler.

WAV dosyaları sıklıkla kullanılan ses dosya türlerindendir. Ham veriler içerdiğinden özel bir çözücüye gereksinim duyulmaz. Bu özelliği her türlü ortamda rahatlıkla kullanılmasını mümkün kılar. Örneğin yeni çıkan dosya türlerinin çalıştırılabilmesi için özel kod çözücülere ihtiyaç duyar. WAV dosyaları ses örneklerinden oluşur ve bu örnekler doğası gereği, ses dalgasının belli bir andaki yaklaşık tahmini değerini ifade etmektedir. Bu durum ses örneklerinin en az önemli bitleri ile değişiklik yapmaya müsaade eder.

WAV dosyaları iki bölümden oluşur. Bunlar, başlık (head) ve içerik (data) olarak adlandırılır [49]. Başlık kısmındaki bilgiler WAV dosyası hakkında bilgi verir ve bu bilgiler WAV dosyasını tanımlayan temel bilgilerdir. Bu bilgilerde herhangi bir değişiklik WAV dosyasının doğrudan yapısını değiştirmektedir. Sırörtmede ise amaç önemsiz ihmal edilebilir küçük değişiklikler oluşturmaktır. Dolayısıyla başlık kısmında veri gizleme amaçlı kullanılabilecek boş bir alan yoktur. Başka bir deyişle ihmal edilebilecek bir şey yoktur. Fakat içerik kısmında WAV dosyalarının ses örneklerinin bilgileri olduğundan, bu bilgilerin en önemsiz bitleri üzerinde değişiklik yapılabilir. Burada yapılan değişikler seste çok küçük derecede artış veya azalmalar yapacaktır. Bu değişiklikler insan kulağı açısından ayırt edilecek bir değişiklik oluşturmayacaktır. WAV dosyalarının MP3 formatına donüştürülmesi sırasında da veri saklama işlemi gercekleştirilebilir. Bu değişim resim dosyalarında yapılan

işleme benzer ve ses örneklerinin değerini değiştirmez ve sıraçma saldırıları karşısında daha sağlamdır.

Bilinen bir diğer yöntem ise doğrudan MP3 dosyalarına veri saklama yöntemidir. MP3 dosyaları Internette çok popüler olduğundan dolayı çok fazla tercih edilmektedir. Boyutunun küçük olabilmesi nedeniyle Internet üzerinden trasferleri hızlı ve kolay olur. Internet kullanıcılarının birbirleri arasında dosya paylaşımı yapabilmek için özel programlar yapılmıştır. Bu programlara Ares, Emule, Torent vs. gibi programları örnek verebiliriz. Bu yüzden MP3, sırörtme yöntemleri için en elverişli dosya türlerinden biridir. Hem kullanımı yaygındır hem de veri gizleme kapasitesi açısından oldukça elverişlidir.

Bir diğer dosya türü MIDI dosyalarıdır. MIDI dosyaları, müzikal performans amacıyla kullanılmak için küçük komutlardan oluşan küçük boyutlara sahip dosya türleridir. Bir MIDI dosyasında en sık kullanılan komut olan "note on" komutunun sesi etkileyen parametresinin en az önemsiz bitlerinin değişmesi, ses üzerinde bir farklılık oluşturmamaktadır ve bundan ötürü sırörtme işleminde kullanılabilir [7]. Her komut, komut kodu ve parametreler içerir ve ard arda aynı komutun aynı parametreleri bulunmaktadır. Bu parametrelerden ilki hariç diğerlerinin olmaması sesin çalmasında herhangi bir farklılık oluşturmamaktadır. Bu açıktan da yararlanılarak sırörtme çalışmaları yapılabilir [7].

MIDI dosyaları dosyaları çok küçük dosyalar olmalarından ötürü Internetten transferleri kolay olur. Bu nedenle sırörtme çalışmalarında tercih edilebilir.

2.1.3.4. Video tabanlı sırörtme

Bir video görüntüsüne gizli bilgi, video görüntüsünün her bir resim çerçevesi kullanılarak gömülebilir. Video görüntüsü içerisine bilgi gömme yöntemi resimlere bilgi gömme yöntemleri ile tamamen benzer bir işleyişe sahiptir.

Hareketli görüntü üzerinde yapılan damgalama çalışmaları da kullanılan video'ya göre Ham–Video Damgalama (Raw–Video Watermarking) ve Sıkıştırılmış–Video Damgalama (Bit–Stream Watermarking) olarak iki ana sınıfa ayrılmaktadır [67].

2.1.3.5. HTML tabanlı sırörtme

HTML (Hyper Text Markup Language) dosya türleri sırörtme için çok müsait ortamlar oluşturmaktadır. Çünkü HTML dili çok katı kuralları olan bir dil değildir. HTML koddaki bazı hatalar web sayfasının görüntüsünde değişiklik oluşturmaz. HTML dilinde tag denilen komut parçaları kullanılır ve web sayfası görüntüleyici programlar aracılığıyla sayfanın görüntüsü kullanıcıya gösterilir. Mesala bir tag kullanılmış ve bunu kapatma tagı unutulmuş olunabilir. Bu durumda HTML çok katı bir şekilde uyarmaz. HTML bunu görmezden gelir ve sanki kapatılmış gibi davranır.

Sui ve Luo (2004), yardımlı metin (hypertext) içerisine biçimleme (markup) etiketlerinin konumlarını değiştirerek veri sakladıklarını rapor etmişlerdir [50].

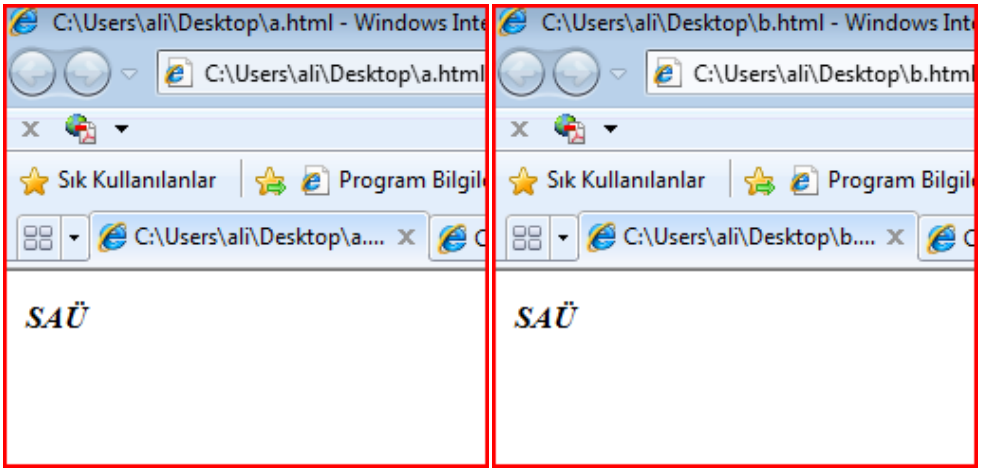
Aynı şekilde HTML'de etiketlerde (tag) aynı nesne için birden fazla etiket kullanıldığında bunların kapanış sıraları önemli değildir. Diğer bir ifadeyle bunların sırası oluşan görüntüyü değiştirmez. Örnek verecek olursak;

SAÜ (a.html)

Yukarıdaki satır normal olması gereken kullanımı göstermektedir. Aşağıdaki satır ise kapanış sırasına dikkat edilmeden yazılmış HTML kod parçasıdır.

SAÜ (b.html)

İki HTML kodu da birbiri ile aynı görüntüyü oluşturur. Html bu farklılığı görmezden gelir. HTML etiketlerinin bu özelliğinden faydalanılarak sırörtme uygulamaları geliştirilebilir. İki dosyanın ekran çıktısı Şekil 2.6'de gösterilmiştir. Aralarında ekran çıktılarından da anlaşılacağı üzere hiçbir fark olmadığı görülmektedir.



Şekil 2.6. a.html ve b.html dosyaları ekran çıktıları

2.1.3.6. XML tabanlı sırörtme

XML, HTML ile benzer bir dildir. Ancak HTML'den farklı olarak veri gösterme yerine veri saklama işlemini gerçeklemektedir. XML'de aynı işlevi gören etiketler ile veri saklanabilir. Örneğin kalın yazdırmak için kullanılan <meta> </meta> yazımı ile <meta/> yazımı aynı işlevi görmektedir. Bu durumda <meta> 1, <meta/> ise 0 olarak kodlanacak olursa aşağıda verilen XML kodu içerisine 1 0 1 1 0 1 0 0 1 bitlerinden oluşacak bir karakterlik bir veri saklanmış olacaktır.

<meta http-equiv="content-language" content="tr"></meta>

<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<meta name="medium" content="video"></meta>

<meta name="video_type" content="application/x-shockwave-flash" ></meta>

<meta name="video_height" content="250" />

<meta name="video_width" content="350" ></meta>

<meta name="keywords" content="sırörtme" />

<meta name="keywords" content="sıraçma" />

<meta name="keywords" content="audio" ></meta>

2.1.3.7. EXE dosya tabanlı sırörtme

Yukarıdaki anlatılanlara benzer bir şekilde aynı işlevi gerçekleştiren kodlar yardımı ile EXE dosyalara da veri saklanabilir. Hydan isimli program i386 komut setindeki tekrarlardan faydalanarak 1/110 kapasiteyle bir mesajı bir uygulama içerisine saklayabilmektedir [51].

2.2. Sıraçma (steganaliz)

Sıraçma ise sırörtmenin tersine gizlenmiş verileri tesbit etmeye çalışan bilim dalıdır. Sıraçmada dosya içerisinde gizli bilgi varlığı sezilmeye çalışılır. Bu işleme sezme (detection) işlemi denir. Bundan sonra ise gizli bilgiyi çıkarma (extraction) işlemi gelir. Bu işlem oldukça zordur, çünkü gizli veri birçok farklı yöntemlerle saklanmış olabilir. Bunun kestirilip gizli bilginin çözülmesi oldukça zahmetli bir iştir.

Sıraçma çalışmalarının temelinde, saklanan verinin taşıyıcı dosya üzerinde bir takım parmak izleri bıraktığı düşüncesi yatmaktadır. Veri gizleme sonrasında oluşan sırlı dosya görsel işitsel anlamda orjinalinden ayırt edilemese de birtakım istatiksel izler bırakmaktadır.

Resim ve ses dosya tabanlı sıraçma konusunda 2000'li yıllarda birçok çalışma yapılmıştır. Sırörtmenin gelişmesiyle sıraçma çalışmaları da gelişmiş ve bunun sonucunda sırörtme çalışmaları sıraçma çalışmalarına karşı daha sağlam yapılmaya çalışılmış böylelikle daha da gelişmişlerdir. Bu anlamda sıraçma sırörtmeyi desteklemiştir.

Sıraçma çalışmalarının temelinde her ne kadar insan duyu organları tarafından ayırt edilemez olsa da aslında sırörtme sonrasında değişen bitlerin bir parmak izi bıraktığını düşünerek bu izleri takip etme ve bu izlerden istatiksel bir sonuç çıkararak gerçek nesne ile örtülü nesneyi birbirinden ayırt etme düşüncesi yatmaktadır.

Aşağıda sırörtme uygulamalarına karşı geliştirilen sıraçma saldırılarından bazılarına yer verilmiştir [24]:

- Sırlı dosyanın bilindiği saldırı (stego-only)
- Taşıyıcı dosya ile sırlı dosyanın ikisinin de bilindiği saldırı (known cover)

- Saklı bir mesajın bilindiği saldırı (known message)
- Algoritma ve sırlı dosyanın bilindiği saldırı (chosen stego)
- Algoritma ve saklı bir mesajın bilindiği saldırı (chosen message)
- Evrensel Tespit

Evrensel tesbit çalışmalarında saklama algoritmasına ihtiyaç duyulmaz ve bu tesbit her türlü saklama algoritması için geçerlidir. Bu şekildeki çalışmalar daha çok önem kazanmıştır [22].

Sıraçma ve sırörtme konusunda mevcut çalışmalar olsa da yeni yeni gelişmeler olacağı beklenmektedir. Genelde sırörtme yöntemleri LSB yöntemine göre çalışmakta ve mesaj bitlerinin sıralı bir şekilde gömmektedirler. Fakat bazı tekniklerde ise rastgele bir şekilde mesaj bitleri gömülmektedir. Aynı zamanda rastgele gizlenen bitler üzerinde şifreleme algoritmaları da kullanıldığında daha karmaşık bir yapı çıkmaktadır. Bu tür saldırılarda esas düşünce, sırlı dosya içerisindeki çok küçük yerlerde gizli olan mesajı belirginleştirerek insan gözünün görebileceği seviyeye taşımaktır. Böylelikle gizlenen mesaj rahatlıkla ayırt edilebilecektir.

2.2.1. Sıraçma yöntemleri

Sıraçmada sezme saldırıları yapılır. Bu saldırılar sırlı dosya içerisinde verinin mevcudiyetini kestirmeye çalışırlar. Sıraçma sezme yöntemleri şunlardır [53]:

- $\chi 2$ Testi (Ki-Kare)

- Görsel tespit (Görsel saldırılar)

- Histogram analizleri

- RQP Yöntemi.

- RS analizi (İkili istatistik yöntemi)

- JPEG Sıraçma

- Evrensel tespit sistemleri

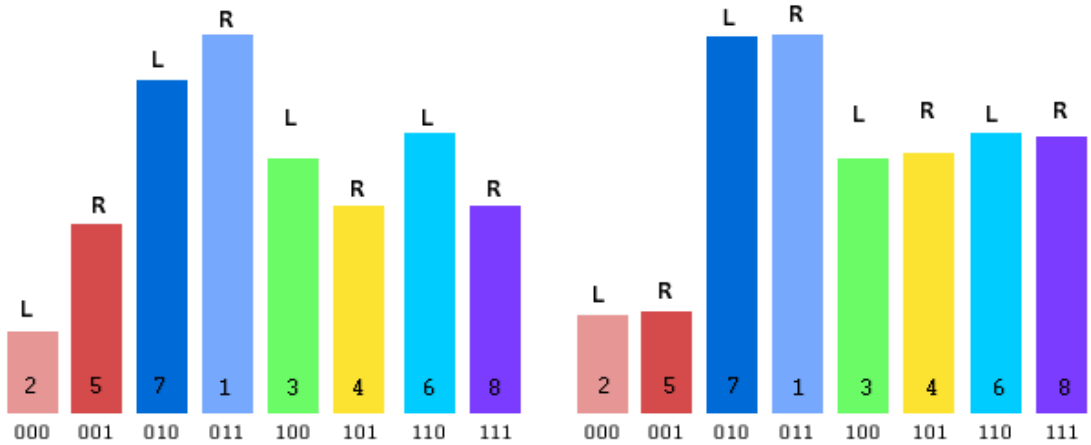
Bu sezme saldırıları aşağıda detaylı bir şekilde anlatılmaktadır.

2.2.2.1 x^2 (ki-kare) testi

Ki-kare testi Westfeld tarafından geliştirilmiştir ve resim içersinde PoVs (Pair of Values- Değer Çifti) değerlerinin istatiksel yaklaşımına dayanan bir istatiksel analiz yöntemidir. Taşıyıcı resim içerisine veri gizlendiğinde resmin içerindeki piksel PoVs değerleri gerçek değerlerinden farklı olmaktadır.

Ki-kare testi ile fotoğraf makinelerinin çektiği yüksek çözünürlüklü video ve sayısal resimlerin analizi yapıldığında değerler "0" düzeyinde çıkmaktadır. Düşük çözünürlüklü sayısal ortamlarda daha iyi sonuç vermektedir. Rastgele gizlenmiş verilerde bu yöntem pekiyi sayılmaz ancak sıralı bir şekilde veri gizlendiğinde sonuçlar daha olumlu ve elde edilecek değerler "0"dan farklı değerler olmaktadır. Ayrıca gizlenen verinin boyutunun da bulanabilmesi bu test ile mümkündür.

PoVs'ler piksel değerlerinden meydana getirilebilir, ölçülmüş DCT katsayıları veya palet indisleri sadece LSB'de değişmektedir. İçine veri gizlenmemiş görüntüler için PoVs'lerin frekansları düz bir şekilde dağılmamaktadır, fakat LSB gizleme sırörtme söz konusu olunca her PoVs' in frekansları eşit olmaktadır. Aşağıdaki Şekil 2.7'de mesajın gizlenmesinden önce ve sonraki renk histogramını göstermektedir.



Şekil 2.7. Resim içine gizli mesaj gömülmeden a) önceki ve b) sonraki renk dağılım durumları [52]

Bundan sonra Westfeld ve Pfitzmann şüpheli bir görüntüde ölçülen PoVs'lerin oluşumunu, istatistiksel rastgele testi ile karşılaştırmıştır. x^2 istatistik testin ayrıntıları adım adım aşağıda verilmiştir:

Adım 1: k kategoriler ve gözlemlerden oluşan rastgele bir örnekleme olduğunu varsayılsın. Her gözlem sadece ve sadece bir kategoriye düşmektedir. Şüpheli bilginin PoVs'lerinin tek değerlerine önem verilmektedir.

Adım 2: Düz bir şekilde dağılmış bir mesajın gizlenmesinden sonra, i kategoride teorik olarak beklenen frekansı böyledir:

$$n_i^* = \frac{|\{renk|(renk)'in\ sıralanmış\ indeksi\ \in \{2i, 2i+1\}\}|}{2}$$

Adım 3: Rastgele örneklemede, ölçülen vuku bulma frekansı aşağıdaki gibidir.

$$n_i = |\{renk\}|(renk'in\ sıralanmış\ indeksi\ \in \{2i, 2i+1\})|$$

Adım 4: x^2 istatistiği ise k-1 bağımsızlık dereceleri ile şu şekilde hesaplanır:

$$X_{k-1}^2 = \sum_{i=1}^{k} \frac{(n_i - n_i^*)^2}{n_i^*}$$

Adım 5: n_i ve n_i^* dağılımları eşit olduğu durumda, p mesaj gömme olasılığıdır. Bu olasılık yoğunluk fonksiyonun integrali alınarak hesaplanmaktadır (Γ , Euler'in gama fonksiyonudur)[54]:

$$P = 1 - \frac{1}{2^{\frac{k-1}{2}}\Gamma(\frac{k-1}{2})} \int_0^{x_{k-1}^2} e^{\frac{x^2}{2}} . x^{\frac{k-1}{2}-1} dx$$

Gama fonksiyonu matematikte faktoriyel fonksiyonun karmaşık sayılar ve tam sayı olmayan reel sayılar için genellemesi olan bir fonksiyondur [54].

x^2 istatistiksel analizi sıralı LSB gizleme sırörtmesinde başarılı sonuç vermiştir. Provos ise bu metodu test aralıkları ve değerleri yeniden değiştirerek

genişletmiştir[55]. Provos, test aralığı ve piksel değerlerini P ve (P +1) piksel çiftinden P ve (P−1) çifte kadar yeniden değiştirip, x^2 testini geliştirmiştir [55].

2.2.1.2 Görsel tespit

Mesajlar görüntü nesnelerinin son bitlerine çoğunlukla sıralı veya rastgele bir şekilde gizlenmektedir. Gizli veri barındıran resimlerde en yüksek katman olan MSB (en önemli bit) bitinde en düşük katman olan LSB (en önemsiz bit) bitine doğru gidildikçe gizli veri saklanımı kesin olarak kanıtlanır deliller içermektedir. Bu mantıkla görsel saldırılarda son katmandaki bilgiler kanıtlanır bir şekilde veri içerdiğini düşünülerek geliştirilmiş ve bu teknikte temel amaç göz önünde olmayan en önemsiz bitlerin değişmesiyle yapılan bu gizli bilgileri üst seviyeye çıkararak kullanıcının görebileceği seviyeye taşımaktır [33].

Görsel analizde sıralı olarak gizlenen mesajlar tesbit edilebilir. Çoğunlukla veriler sıralı olarak gizlenmektedir. Ayrıca BMP resimlere gizlenen verilerin analizinde kullanılmaktadır[56]. JPEG resimlerdeki gizleme tekniği 8x8 boyutundaki bloklar halinde olduğu için bu teknik JPEG resim dosyalarında kullanılmaz.

Görsel saldırılar basit olmasına rağmen, bunları otomatikleştirmek zordur ve güvenilirlikleri çokça sorgulanır.

Aşağıda Şekil 2.8'da içinde veri gizlenmemiş orijinal BMP dosyanın görsel saldırı sonucu elde edilmiş görüntüsü vardır [56]:

Şekil 2.8. Orijinal resim ve görsel saldırı sonucu resim [56]

Şekil 2.9 ve Şekil 2.10 ise sırasıyla içine 1 Kb 5 Kb büyüklüğünde veri gizlenmiş olan görüntü dosyalarını ve görsel saldırı sonucu elde edilmiş görüntüleri göstermektedir.

Şekil 2.9. 1Kb veri gizlenmiş resim dosyası ve görsel saldırı sonucu resim [56]

Şekil 2.9'da da göründüğü gibi içerisine veri gizlendikce görsel saldırı sonucu oluşan resim daha bozuk hale gelmekte ve içerisinde gizli veri barındırdığını göstermektedir. Aşağıdaki Şekil 2.10'da gösterilen resimde ise daha fazla veri gömülmesi sonucu dosyadaki görsel atak sonucu oluşan bozulma gözlenmektedir. Bu da gösteriyor ki bir dosyaya ne kadar çok veri gizlenirse analiz noktasında o kadar çok kendini ele vermeye yaklaşır. Bu da gizleme tekniğinin dezavantajıdır.

Şekil 2.10. 5Kb veri gizlenmiş resim dosyası ve görsel saldırı sonucu [56]

2.2.1.3. Histogram analizi

Histogram analizleri saldırı teknikleri içerisinde en çok kullanılanlar arasındadır. Bu teknik sabit PoVs (pairs of values-değer çiftleri) kümesi mesaj bitlerini gömmek için birbirinin içine karışmaktadır. Örneğin, PoVs piksel değerleri, kuantal DCT katsayıları veya LSB içinde farklılaşan palet indisleri ile oluşturulabilir. Orjinal görüntüye veriyi gömmeden önce her çiftteki değerlerin meydana çıkması, eşitlenme eğilimi gösterecektir. Bir değeri bir başkasının içine takas etmek görüntüdeki her iki rengin meydana çıkma toplamını değiştirmediğinden, bu durum istatistiksel Ki-kare testi tasarlamak için kullanılabilir. Her çiftteki her iki değerin meydana çıkmasının aynı olduğu gerçeğinin istatistiksel önemini test edebiliriz. Eğer, buna ek olarak, sırlı tekniği mesaj bitlerini takip eden piksellere/indislere/katsayılara sıralı olarak üst sol köşeden başlayarak gizlerse, mesajın sonu ile karşılaştığımızda, istatistiksel delilimizde ani bir değişiklik gözlemleyeceğiz.

Sıralı olarak gizlenmiş bir mesaj için, mesajın saklama sırası ile birlikte mesajı tarayabilmekte ve bütün piksel değerleri için p değeri hesaplanmaktadır. İlk başta p'nin değeri 1'e yakın olacak ve sonra mesajın sonuna gelindiğinde 0'a düşecektir. Böylelikle bu teknik ile hem mesajın varlığının olasılığı yüzdesel olarak çıkarılmakta hem de mesajın boyutu hesaplanabilmektedir. Bu teknik ile gizli bilginin hem varlığı hem de boyutu hakkında fikir edinilebilir. Ancak verinin içeriği konusunda bir bilgi alınamaz.

Eğer görüntüde mesaj taşıyıcı pikseller rasgele seçilirse bu test daha az etkili olmaktadır. Provos, bu tekniğin bir görüntünün daha küçük farklı alanlarına uygulanırsa, mesaj uzunluğu arttıkça p 'nin değeri gitgide azaldığını fark etmiştir. Fakat Provos bununla ilgili daha ileri istatistiksel analizleri sunmamıştır.

2.2.1.4. RQP analizi

RQP yöntemi Fridrich tarafından geliştirmiştir. Bu metod LSB gizlemesi tarafından yaratılan yakın renk çiftlerini analiz etmeye yöneliktir. Öncelikle seçilen resim için yakın renk çiftlerinin tüm renk çiftlerine oranı hesaplanır [17]. Daha sonra bu resim içerisine bir test mesajı gizlenerek oran yeniden hesaplanır. Bu iki oran arasındaki

fark büyük ise resmin içinde gizlenmiş bilgi yok demektir. Bu iki oranın birbirine yakın olması resmin içinde gizlenmiş bilgi olduğunu göstermektedir.

RQP, örtü verisindeki yakın renk çiftlerinin sayısı, piksellerin sayısının %30'undan küçük olduğu sürece gayet iyi sonuçlar vermektedir. RQP, gizli mesajın büyüklüğü hakkında sadece iyi bir tahmin sağlamaktadır. Eğer görüntüdeki yakın renk çiftlerinin sayısı piksellerin sayısının %50'sini geçerse, verilen sonuçlar giderek güvensiz olmaktadır. RQP'ın başka bir dezavantajı, gri seviyeli görüntülere uygulanamamasıdır [17].

2.2.1.5. RS analizi (ikili istatistik yöntemi)

Bu analiz, görüntülerde uzaysal korelâsyonlardan üretilen duyarlı ikili istatistiklerini kullanmaktadır. RS sıraçma, 24-bit renkli ve 8-bit gri seviye görüntülerde kullanılmaktadır. RS sıraçmasının temel mantığı, bir resmin piksellerinin üç bağımsız gruba ayrılmasına dayanır. Bu üç grup düzenli (Regular-R), tekil (Singular-S) ve kullanılmayan (Unused-U) olarak adlandırılmatadır [53]. Bu teknik, gizlemiş verilerin istatiksel analizi ile ilgilenen analizcilerin aksine, verilerin nasıl yerleştirildiği ile ilgilenmiştir. Gerçekten verilerin görüntü içerisine yerleşim şekli de analiz açısından birçok bilgiler barındırmaktadır. Sayısal kamera veya tarayıcı ile alınan yüksek çözünürlüklü görüntüler için, RS sıraçma güvenli bit-oranın her örnekleme için 0.005 bitten küçük olduğunu göstermektedir.

2.2.1.6. JPEG sıraçma

JPEG Sıraçma, JPEG formatındaki dosyalar üzerinde uygulanan sıraçma yöntemleridir. JPEG dosyalarındaki sıraçmada iki durum söz konusudur.

- Hem orijinal hem de içine bilgi saklanmış resmin elimizde olduğu durum (Bilinen sır saldırısı).
- Sadece bilgi saklanmış resmin elimizde olduğu durum (Seçilmiş sır saldırısı).

Bu iki durumu da kısaca anlatacak olursak:

Orijinal ve gizli bilgi barındıran resmin elimizde olduğu durumda, her iki resmin ilk blokları için nicelendirilmiş DCT matrisleri bulunur ve aralarındaki fark incelenir.

Sadece bilgi saklanmış resmin elimizde olduğu durumda ise hangi bloklarda şifrelenmiş metin olduğunu anlamak amacı ile JPEG Uygunluk Esasına Dayanan Sıraçma yapılır [53].

Eğer bir taşıyıcı-görüntü ilk olarak JPEG formatında saklanmışsa, JPEG sıkıştırma tarafından yaratılan yapının özellikleri mesaj gizlemeden dolayı silinmeyecektir, sadece biraz değişecektir.

Sırlı görüntüden 8x8 piksellik bloklardaki DCT katsayıların değerlerini analiz ederek JPEG ölçme tablosu elde edilebilmektedir. Bir görüntüdeki hangi blok JPEG sıkıştırma ile uygunluk göstermiyorsa, bundan gizlenen mesajın uzunluğu ve yerleşimi bulunabimektedir.

2.2.1.7. Evrensel tespit

Her hangi bir uzaysal sırörtme metoduna uygulanabilen JPEG uyumluluk sıraçması hariç, önceden önerilmiş bütün metotlar özel bir gömme algoritması veya varyasyonları için ısmarlama olarak yapılmıştır. Evrensel kör sıraçma orijinal ve sırlı görüntüler üzerinde eğitimden sonra gömme etki alanı dikkate alınmaksızın her hangi bir sırörtme metodunun tespiti için ayarlanabilir olması anlamında tespit ötesi bir metottur. Önemli olan, "ayırt edici" yeteneklere sahip uygun bir hassas istatistik nicelikleri kümesi (bir özellik vektörü) bulmaktır. Böylece yapay sinir ağları, kümelenme algoritmaları ve diğer yapay zekâ araçları doğru eşikleri bulmak ve toplanan deneysel verilerden tespit modelini inşa etmek için kullanılabilir. Farid, sırlı görüntünün dalgacık parçalanmasından türetilmiş yüksek seviyeli hassas istatistikler kümesi önermektedir [57]. Daha sonra, özellik vektörleri Fisher Doğrusal Ayırma analizini kullanılarak, biri sırlı görüntülere, diğeri orijinal görüntülere karşılık gelen iki doğrusal alt uzaya bölünür. Karar eşiği sahte pozitifler, gözden kaçırılmış tespitlerle takas etmek üzere ayarlanabilir. Bu yaklaşımın oldukça keyfi "geçici" tercihler içerdiği gerçeği göz önünde tutulursa, metodunun nasıl bu kadar iyi çalıştığı dikkate değer [58].

Farid'in yaklaşımı sırlı görüntünün, Vi(x,y) i ölçeğinde düşey, Hi(x,y) yatay, Di(x,y) ve diyagonal alt bantları ifade etmek üzere, n'inci seviye dalgacık parçalanması ile başlamaktadır. Daha sonra, bütün üç alt bant için bütün i=1,….,n-1 seviyeleri için ilk

dört momenti hesaplamaktadır. Bu toplam 12(n-1) istatistiksel nicelik verir. Daha sonra, en uygun doğrusal tahminci kullanarak dalgacık katsayısının fiili değeri ile en uygun doğrusal tahmin arasındaki tahmin hatası için uzaysal, yöneltme ve ölçek komşularından (toplam 7 komşu) aynı istatistiksel momentleri toplamaktadır.

Örneğin, düşey dalgacık katsayıları aşağıdaki doğrusal denklemi kullanarak tahmin edilmektedir:

Vi (x,y) = w1 Vi (x – 1 , y) + w2 Vi (x + 1 , y) + w3 Vi (x , y – 1) + w4 Vi (x , y + 1) + w5 Vi+1 (x/2 , y/2) + w6 Di (x , y) + w7 Di+1 (x/2 , y/2).

Karesi alınmış tahmin hatasını en aza indiren ağırlıklar, standart en küçük kareler metodu kullanılarak hesaplanır. Vi(x,y) için doğrusal tahmin $\check{V}_i(x,y)$ şeklinde ifade edilerek, tahminin hatası Ev(i)=log2(Vi)-log2($\check{V}_i$) olarak tanımlanır. Farid Ev(i)'nin ilk dört momentini özellik vektörünün başka bir parçası olarak hesaplamaktadır.

İşlemin bütünü, bütün n-1 ölçekler için yatay ve diyagonal alt bantlarla tekrarlanmaktadır. Böylece, özellik vektörünün uzunluğu 12(n-1)+4x3(n- 1)=24(n-1) olmuştur. Özellik vektörü orijinal görüntülerin ve sabit boyutlu mesaj gömülü sırlı görüntülerin geniş bir veri bankası için hesaplanmıştır. Özellik vektörlerini bir eşiğin ayırdığı iki kümede sınıflamak için Farid Fisher Doğrusal Ayırma (FLD) analizini kullanmaktadır [58]. Bu yaklaşımın avantajı FLD analizinin tek boyutta hızlı ve basit skalar eşik oluşturmayı sağlamasıdır.

Farid tarafından bildirilen sonuçlar [58], yaklaşımının yüksek başarımlı sonuçlar veren tutarlı bir yöntem olduğunu göstermektedir. Farid J-Steg, EZ Stego, ve OutGuess'in her iki sürümünü kullanarak küçük bir 256x256 piksel gri ölçek görüntü gömdüğü 1400x1000 yüksek kaliteli görüntülerin veri tabanını kullanmıştır. En iyi sonuçlar %1,8 sahte pozitiflerde %97,8 tespit güvenirliliği ile J-Steg için elde edilmiştir. EZ Stego %13,2 sahte pozitif ile %86,6 güvenirlilikle tespit edilmiştir. OutGuess için aynı sonuçlar %80,4 ve %19,4 (sürüm 0,1 için), ve sürüm 0,2 için %77,7 ve %23,8 olmuştur. Bu kısmı RS sıraçması gibi, özellikle özel bir sırörtülü metoda hedeflenmiş metotların her hangi bir evrensel kör sıraçma metodundan muhtemelen daha doğru ve güvenilir sonuçlar verecektir. Sonuç olarak, evrensel kör

yaklaşımlar esneklikleri ve yeni veya hiç bilinmeyen steganalitik metotlara kolayca ayarlanma yetenekleri nedeniyle yoğun şekilde kullanılmaktadır.

BÖLÜM 3. SIRÖRTME UYGULAMASI

Bu bölümde önceki bölümlerde anlatılanlara ve konumuzla ilgili olarak sıraçma çalışmalarına örnek olacak olan bir uygulama çalışması anlatılmıştır. Sıraçmada öncelikle olması gereken şey içersine veri gizlenmiş sayısal bir ortamdır. Bu veri gizlenmiş sayısal ortamı elde edebilmek için veri gizleme uygulamasının da olması gereklidir.

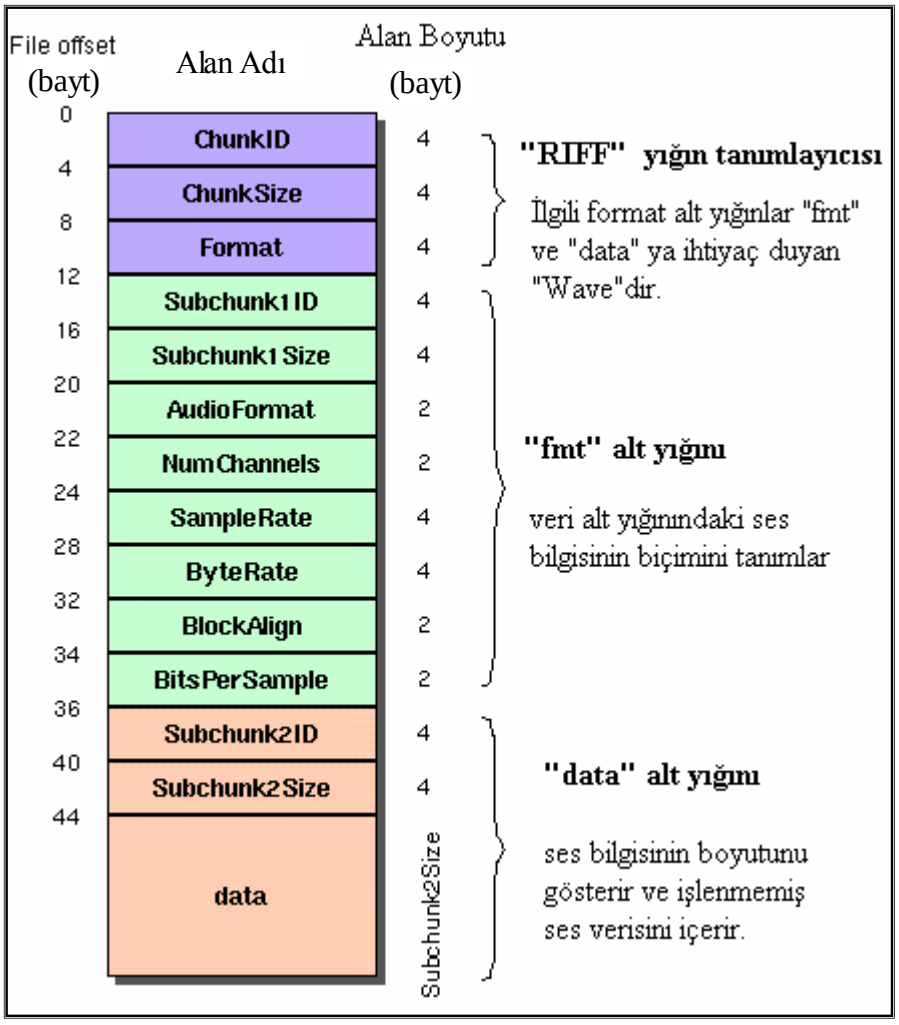
Daha önce de anlatıldığı gibi ses dosyalarına farklı yöntem ile veri gizlenebilinmekteidir. Bu yöntemler kısaca aşama kodlaması (phase coding), düşük bit kodlaması (Low Bit Encoding), yankı veri saklaması (echo data hiding) ve tayf yayılması(spread spectrum) olarak sınıflandırılmaktadır [35].

Micah Johnson ve ark. [59], ses sinyallerinin istatistikî düzenlemelerini alabilmek için bir program geliştirdiler ve sınıflandırma yapmak için doğrusal olmayan bir destek vektörü kullandılar. Ancak bu tekniğin düşük bit kodlaması tekniği ile gizlenmiş verileri bulabileceği konusu kesin değildir.

Steghide, mesajları JPEG, BMP görüntü dosyalarına ve WAV, AU ses dosyalarına yerleştirebilen internet üzerinde ücretsiz olarak bulunabilen bir araçtır [60].

3.1. WAV Dosya Yapısı

Doğada ses anolog sinyaller halinde bulunan ses bilgisayar teknolojileri ile analog ortamdan kurtularak sayısal ortamlara taşınmıştır. Bilgisayarın ses kartından alınan analog sesler, bilgisayar ortamında PCM (Pulse Code Modulation) yöntemi ile sayısallaştırılarak sayısal sinyallere dönüştürülmektedir.



Şekil 3.1. WAV dosya yapısı [49]

Ses kartından alınan ses bilgileri işlenmemiş haldedir ve ".wav" dosya tipindedir. Bu tip bir dosya yapısının ilkel versiyonu ise Microsoft'un ".riff" (Resource Interface File Format) uzantılı dosya yapısıdır. WAV dosyasının en büyük özelliği işlenmemiş ham veriler içermesi ve sırörtülü çalışmalara uygun olmasıdır. WAV ses dosyasının yapısı Şekil 3.1'de verilmektedir. Şekilde de göründüğü gibi bir WAV dosyası, diğer dosya türlerinde de olduğu gibi bir "RIFF" yığın tanımlayıcısı adı verilen başlık bölümü, bunu takip eden ve içerisinde iki adet alt veri parçası barındıran bir "WAVE" veri parçasından oluşmaktadır. Bu veri parçasının içindeki alt veri parçalarından birincisi "fmt" veri parçası olup, WAV dosyasının veri biçimiyle ilgili bilgileri içermektedir. Diğer alt veri parçası olan "data" (veri) ise asıl ses örnekleri verisini içermektedir [49].

WAV dosyasının veri bölümü ses örneklerinden (audio samples) oluşmaktadır. Ses örnekleri 8 veya 16-bit uzunluğunda olmaktadır. 8-bitlik ses örneklerinden oluşan WAV dosyalarına mono (tek kanal) 16-bitlik ses örneklerinden oluşan ses dosyalarına ise stereo (çift kanal) ses dosyaları denir. Stereo ses dosyalarında bilgi iki kanalla çoklanır ve daha kaliteli ses çıkışı sağlanır.

Başlık kısmı olan "RIFF" yığın tanımlayıcısının detay bilgileri Tablo 3.1'de verilmiştir.

Tablo 3.1. RIFF yığın tanımlayıcısı yapısı [49]

Boyut (Bayt)	**Ad**	**Açıklama**
4	ChunkID	ASCII biçimindeki "RIFF" yazısını içerir.
4	ChunkSize	Yığın boyutunu içerir.
4	Format	ASCII biçiminde "WAVE" bilgisini içerir.

ChunkID kısmında "RIFF" kelimesi yazılıdır. RIFF WAV dosyalarını da içeren genel dosya formatının ismidir [31]. ChunkSize alanındaki değer, kendisinden sonra gelen tüm bölümlerin toplam boyutunu bayt cinsinden belirtmektedir. Format kısmında "WAVE" kelimesi yazılıdır. Bu bilgiler bu dosyanın bir WAV dosyası olduğuna işaret etmektedirler.

Başlık kısmındaki bilgiler ses dosyalasının temel yapısı hakkında bilgi verdiği için bu kısımda veri saklama işlemi yapılmamakta, burada yapılan herhangi bir değişiklik doğrudan ses dosyasında yapısal olarak bozulmalara neden olmaktadır. Tablo 3.2'de "fmt" alt yığınının detay alanlarının açıklamaları verilmektedir.

Tablo 3.2. “fmt” alt yığını yapısı [49]

Boyut (Bayt)	Ad	Açıklama
4	Subchunk1ID	“fmt” yazısını içerir.
4	Subchunk1Size	PCM için 18’dır.
2	Audioformat	PCM için 1’dir. Sıkıştırma tiplerinin bazıları için 1’den faklı değerler mevcuttur.
2	NumChannels	Mono için 1, Stereo için 2’dir.
4	SampleRate	8000, 44100 gibi örnek oranlarıdır.
4	ByteRate	= SampleRate x NumChannels x BitsPerSample/8
2	BlockAlign	= NumChannels x BitsPerSample/8
2	BitsPerSample	8 bit için 8, 16 bit için 16.

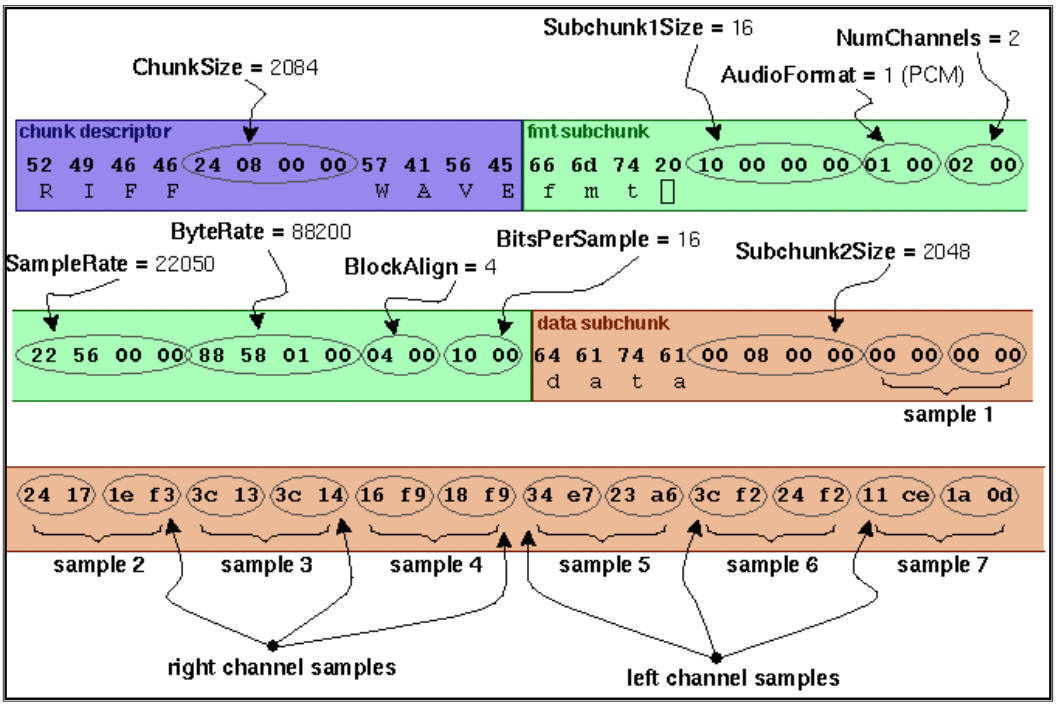
Subchunk1ID kısmında “fmt” yazılıdır ve format ile ilgili bilgiler başlamaktadır. Subchunk1Size değeri ise, kendisiyle Subchunk2ID alanı arasında kalan bölümlerin toplam boyutunun ne kadar bayt olduğunu göstermektedir. AudioFormat WAV dosyasının biçimini gösterir. PCM için bu değer 1 olmalıdır. Bunun dışındaki değerler sıkıştırma olduğu anlamına gelmektedir. Örneğin MP3 dosya yapısı sıkıştırılmış dosyadır ve bu alan MP3 dosyalar için 1’den farklıdır. NumChannels değeri kaç kanal olduğunu gösterir. Mono için 1, Stereo için 2 değerini alır. BitsPerSample verisi bir ses örneğinin kaç bitlik bir değere sahip olduğunu göstermektedir. WAV dosyaları için bu değerin 8 veya 16 olduğu yukarıda belirtilmişti.

Yukarıda da bahsedildiği gibi ses dosyalarında veri “data” alt yığını güncel ses bilgilerini içermekte ve sırörtmeli veri gizleme yaklaşımlarının uygulandığı alanı temsil etmektedir. Tablo 3.3’de “data” alt yığınının detay alanlarının açıklamaları verilmektedir.

Tablo 3.3. “data” alt yığını yapısı[49]

Boyut (bayt)	Ad	Açıklama
4	SubChunk2ID	“data” yazısını içerir.
4	SubChunk2Size	= NumSamples x NumChannels x BitsPerSample/8 (Bu bilgi, data bölümünde bulunan veri boyutu bilgisidir.)
N	Data	Güncel ses bilgisi (44.bayttan itibaren).

Subchunk2ID kısmında “data” yazmaktadır. Subchunk2Size kısmında başlık kısmının haricindeki asıl verinin büyüklüğünü vermektedir. Bu kısım güncel ses bilgilerinin başladığını bildirmektedir. Bundan sonra gelen bilgiler güncel ses bilgileridir denilmektedir. Aşağıda örnek bir WAV dosya yapısı gösterilmektedir.



Şekil 3.2. WAV dosyası örneği[49]

3.2. Sırörtme işlemi ve detayları

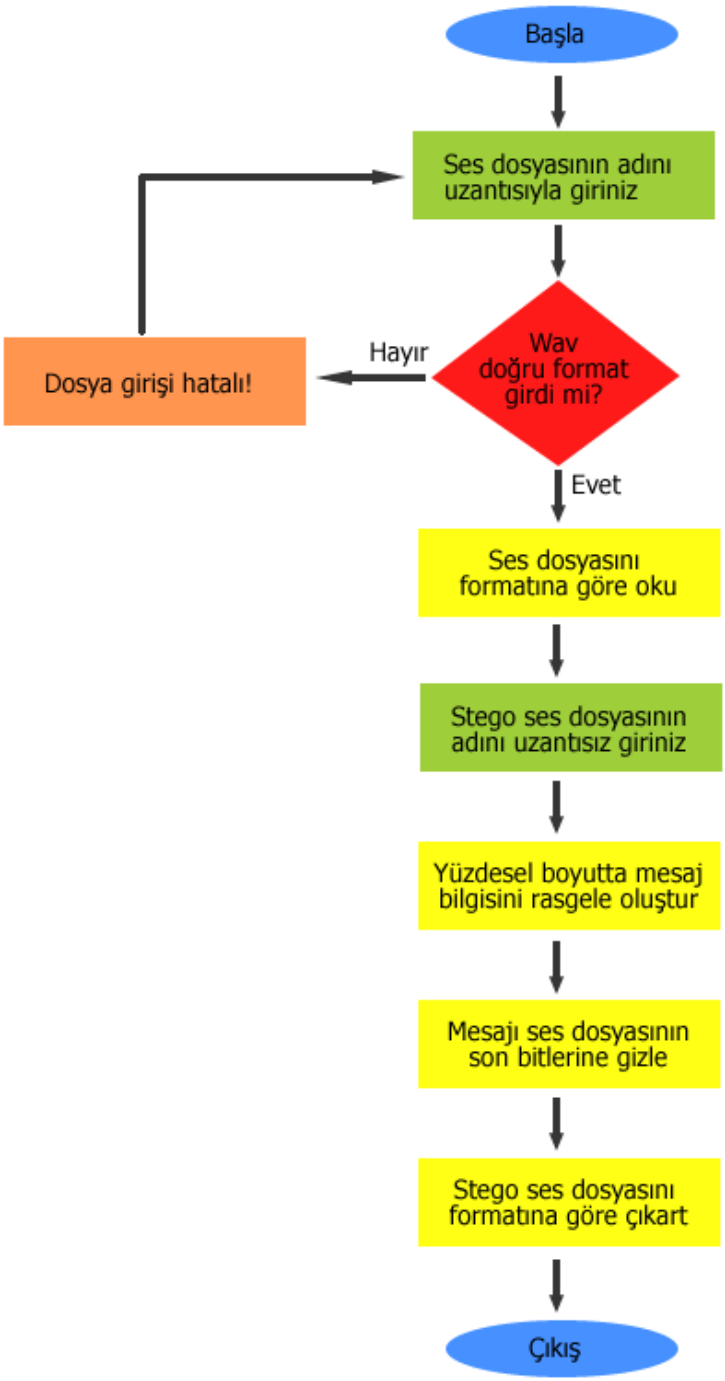
Asıl amacımız sıraçma olsa da sıraçma yapabilmek için öncelikle veri gizlenmiş dosyalara ihtiyacımız olacaktır. Bundan dolayıdır ki ilk olarak veri saklama algoritması ile işe başlamak gerekecektir. Zaten veri gizleme tekniklerini bilmeden gizli veri analiz işlemi pek söz konusu değildir.

Veri saklama yaklaşımında, veriler ses dosyalarının son bitlerine gizlenmektedir. Algoritma, ses örnekleri başına en fazla 1-bit veri gömmektedir. Stereo dosyalar anlatıldığı üzere 16 bit veriler içermektedir. Fakat algoritma 16 bitlik stereo ses dosyalarını 8 bitlik mono ses dosyalarına çevirmektedir. Bunun nedeni çalışmada kolaylık saklamak amaçlıdır. Dolayısıyla çalışılan ses örnekleri 8 bitliktir.

Veri gizleme algoritması her ses örneğine veri gömmemektedir. Bu analiz edilme olasılığını en az indirgemek için düşünülmüştür. Gizlenecek olan veri program aracılığıyla rastgele değerler içerecek şekilde otomatik olarak oluşturulmaktadır. Bunun nedeni sabit bir mesaj ile çalışmamak ve böylece daha güvenli bir sistem geliştirebilmektir. Yani olabilecek her durum için kararlı bir analiz yöntemi geliştirebilmek için bu şarttır. Böylece tasarlanan sistem doğruluğu sadece özel veriler için değil her türlü veri için test edilmiş olacaktır.

Ayrıca uygulama içinde verilecek bir parametre ile ses dosyasının yüzde kaçına veri gömüleceği ayarlanabilinmektedir. Böylece analiz işleminde veri gömülmüş ve gömülmemiş alanları tesbit etmek için ortam sağlanacaktır. Mesajlar, uygulama her çalıştırıldığında değişmektedir ve 0 ve 1 verileri rastgele oluşmaktadır. Böylelikle gizlenen mesajın içeriği de sürekli değişmekte ve bu veriler anlamsız olacağı için şifrelenmiş mesaj olarak kabul edilebilir.

Aşağıda veri saklama yaklaşımının algoritması akış diyagramı olarak verilmiştir.



Şekil 3.3. Veri saklama uygulaması akış şeması

Şekil 3.3'de veri saklama uygulaması akış şeması verilmiştir. Aşağıda yapılan uygulamanın kaynak kodlarına yer verilmekte ve yaptığı işlemler sırası ile anlatılmaktadır.

```
sonuc=false;
while sonuc == false

ortu_dosya_adi=lower(input('Örtü ses dosya adını uzantısıyla giriniz(Örn.
                                    ses.wav):','s'));
 uzunluk=size(ortu_dosya_adi);
 bulunanIndex=strfind(ortu_dosya_adi,'.wav');
 if (bulunanIndex == uzunluk(2)-3)
     sonuc=true;
 else
    sonuc=false;
    disp('Dosya girişi hatalı!');
 end
end
```

Şekil 3.4. Veri saklama uygulaması kod parçası 1

Şekil 3.4'de verilen kaynak kod, örtü ses dosyasının adını kullanıcıdan almak için yazılmıştır. Ses dosyasının adını yazarken uzantısıyla birlikte yazıldığına dikkat ediniz. Uzantısıyla birlikte yazıldığında yazılım girilen dosya adının WAV dosya adı mı yoksa au dosya adı mı olduğunu anlamaya çalışır. Strfind fonksiyonu ile geçerli olan ses dosyası uzantıları girilen ses dosya adında aranır ve hatalı giriş olup olmadığı kontrol eder. Eğer hatasız bir şekilde girildi ise işlemlerine devam eder. Eğer hatalı bir giriş yapılmış ise "Dosya girişi hatalı!" uyarısı verir ve tekrar başa döner. Burada girilen dosya adının büyük ya da küçük harf ile yazılması önemli değildir.

```
[ortu_verisi,cerceveHizi,kacBit]=wavread(ortu_dosya_adi);
stego_dosya_adi = input('Sırlı ses dosya adını uzantısız giriniz(Örn. SırlıSes):','s');
stego_dosya_adi = strcat(stego_dosya_adi,'.wav');
```

Şekil 3.5. Veri saklama uygulaması kod parçası 2

Şekil 3.5'de kullanıcı girişi ile alınan ses dosyasının verisi, çerçeve hızı ve stereo mu mono mu bilgisi okunur. Daha sonra kullanıcıdan oluşacak olan sırlı ses dosyasının bilgisi istenir ve dosya adı uzantısı eklenerek dosya adı oluşturulur.

```
ortu_verisi=round(2^(kacBit-1).*(ortu_verisi))+2^(kacBit-1);
[satir,sutun]=size(ortu_verisi);
```

Şekil 3.6. Veri saklama uygulaması kod parçası 3

Şekil 3.6'daki kodda örtü verisi, ses dosyası okunarak alındıktan sonra birkaç dönüşüm işleminden geçirilmektedir. Bunun nedeni, örtü verisi içerisindeki sayısal bilgiler çok küçük ondalık değerler içermektedir. Üstelik bu değerler arasında negatif değerler de bulunmaktadır. Yazılım bu değerleri belli bir değere normalize etmektedir. Böylelikle hem pozitif değerler ile çalışılmakta hem de istenen değerler elde edilebilmektedir. Daha sonra örtü verisinin satır sütun bilgileri alınarak ileride her bir ses örneği döngü ile incelenmek için kullanılacaktır.

```
TekOrtu = zeros(2^(kacBit-1),1); TekStego=zeros(2^(kacBit-1),1);
CiftOrtu= zeros(2^(kacBit-1),1); CiftStego=zeros(2^(kacBit-1),1);
for i=1:satir
   for j=1:sutun
      if rem(ortu_verisi(i,j),2)==0
         CiftOrtu((ortu_verisi(i,j)/2)+1)=CiftOrtu((ortu_verisi(i,j)/2)+1)+1;
      else
         TekOrtu(((ortu_verisi(i,j)-1)/2)+1)=TekOrtu(((ortu_verisi(i,j)-1)/2)+1)+1;
      end
   end
end

kategoriler= zeros(2^(kacBit-1),1);
for i=1:(2^(kacBit-1))
   kategoriler(i)=2*(i-1);
end
Fark=abs(TekOrtu-CiftOrtu);
baslik=strcat(ortu_dosya_adi,':','|Tek-Cift| Veri gizlenmeden önce');
figure(1),plot(kategoriler,Fark,':'); title(baslik);
xlabel('Kategoriler(i)'); ylabel('|Tek-Cift|');
```

Şekil 3.7. Veri saklama uygulaması kod parçası 4

Şekil 3.7'de örtü verisindeki her bir ses örneği 8-bit parçalar halinde incelenmekte ve bu sayısal ses örneklerinin değerlerine bakılmaktadır. Birbirinin aynısı olan çift

değerli ses örneklerinin sayısı CiftOrtu adlı dizide tutulmakta ve tek değerlere sahip değer çiftleri ise TekOrtu adlı dizide tutulmaktadır. Burada amaç, değer çiftlerinin frekanslarını bulmaktır. Son olarak bu değer çiftlerinin frekanslarının histogram grafiği çıkartılır.

```
sonuc=false;
while sonuc==false
        yuzdeKac=input('Ses dosyasının yüzde kaçına mesaj gizlensin? (1-100): ');
        if (yuzdeKac>0 && yuzdeKac<=100)  sonuc=true;
        else sonuc=false;
        end
end
kacSatir=floor((yuzdeKac/100)*satir);
gizliMesaj=rem(floor(rand(satir,sutun)*1000),2);
```

Şekil 3.8. Veri saklama uygulaması kod parçası 5

Şekil 3.8.'de dosya okunduktan ve çeşitli dönüşümlerden sonra normalize edilir. Ses dosyasına istenilen boyutta mesaj gizlenebilir. Bunun nedeni yüzde 10 veri gizlenmiş dosya ile yüzde 100 veri gizlenmiş dosyayı analiz ederek tesbitlerde bulunabilmektir. Böylece belli oranlarda veri gizlenmesi sağlamış olacak ve analiz sonuçları karşılaştıralabilecektir. Yukarıda yüzdesel değer istenmekte ve bu değerin 1 ile 100 arasında olup olmadığı kontrol edilmektedir. İstenen değer alınana kadar yüzdesel değer sorusu kullanıcıya sorulur. Böylece doğru değer girilmesi sağlanır.

Yüzdesel olarak veri girişi yapıldıktan sonra örtü ses dosyasının kullanıcının girdiği yüzdesel değerine karşılık gelen satır adedi hesaplanır. Daha sonra ses dosyasının boyutlarına göre rastgele mesaj oluşturulur. Matlab'daki rand komutu istediğiniz matris boyutunda rastgele ondalık sayı oluşturur. Örneğin 0.98751 Bu değerleri 1 ve 0 değerlerine dönüştürebilmek için 1000 ile çarpılır böylece değer 987.51 olur. Sonra floor komutu ile bunun tam kısmını alınır yani 987 sayısını elde ederiz. En son rem komutu ile 2 ile bölümünden kalanı, sayı tek ise 1 çift ise 0 değerini elde edilir. 9875 sayısı tek olduğu için 1 değeri elde edilir.

```
stego_verisi=floor(ortu_verisi/2)*2+gizliMesaj;
if kacSatir<satir
      stego_verisi=cat(1,stego_verisi(1:kacSatir,1:sutun),
                        ortu_verisi(kacSatir+1:satir,1:sutun));
end
```

Şekil 3.9. Veri saklama uygulaması kod parçası 6

Şekil 3.9'daki kodlar veri gizleme işleminin yapıldığı yerdir. Örtü verisinin son bitlerine gizli mesaj bilgisinin bitleri yazılmaktadır. Bu işlemde, örtü verisindeki değerler 2 ye bölünmektedir. Bu sadece tekil değerler için yapılan bir dönüşümdür. Örneğin 32577 değeri tekildir ve 2'ye bölünmesi sonucu 16288.5 değeri elde edilir. Sonra bu sayının tam kısmı alınarak 2 ile çarpılır yani 16288 * 2 = 32576 sayısı elde edilir böylece sayısal değer olarak bir azaltılmış olunur. Bu işlemin nedeni tekil sayıların ikilik karşılıklarının son biti 1, çift değerlerin son biti ise 0'dır. Son biti 1 olan bir sayıya 1 değeri ile toplama işlemi yapılırsa 10 sayısı elde edilir ve bu işlem neticesinde sadece son bit değil ikinci bit de değişmiş olur. Oysa ki sadece son bit üzerinde işlem yapmak istenmektedir. Son bit haricindeki değişimler ses kalitesinde fark edilebilir değişmelere neden olabilir. Çift değerlerin son biti ise 0'dır. 0 ile 1 sayısının toplamında 1 değeri elde edilir ve bu sadece son biti değiştirmiş olur. Yani çift değerlerde bir sıkıntı yoktur. Sorun tekil değerlerdedir ve bu yöntemi kullanarak bu sıkıntı atlatılır. İşlem sonucunda sayısal 1 azaltılır yani son bit 0'a kurulmuş olur ve gelen değer ile toplama konusunda bir sorun teşkil etmez. Bu yöntem maskeleme işlemi ile aynı yöntemdir. Bu komut ile sırlı dosyanın tüm bitlerine mesaj gizlenmektedir. Oysaki eğer girilen yüzdesel değer % 100 den az ise yani örtü dosyasının tüm bitlerine mesaj gizlenmeyecek ise o zaman sırlı dosya oluşturuken yüzde kaç ise o kadarına mesaj gizlemek gerekir. Bu durumda cat komutu ile tümüne gizli mesaj gizlenmiş sırlı verinin sadece yüzdesel karşılığa gelen satır değerine kadar olanı ile o satır değerinden sonrasını ise orijinal örtü verisinden alacak şekilde birleştiren işlem gerçeklenmektedir. Yani örtü ses dosyasına yüzdesel olarak mesaj gizleme işlemi yapılmıştır.

```
for i=1:satir
   for j=1:sutun
      if rem(stego_verisi(i,j),2)==0
         CiftStego((stego_verisi(i,j)/2)+1)=CiftStego((stego_verisi(i,j)/2)+1)+1;
      else
         TekStego(((stego_verisi(i,j)-1)/2)+1)=TekStego(((stego_verisi(i,j)-1)/2)+1)+1;
      end
   end
end

Fark=abs(TekStego-CiftStego);
baslik=strcat(stego_dosya_adi,':','|Tek-Cift| Veri gizlendikten sonra');
figure(2),plot(kategoriler,Fark,':');title(baslik);
xlabel('Kategoriler(i)'); ylabel('|Tek-Cift|');
```

Şekil 3.10. Veri saklama uygulaması kod parçası 7

Veri gizlendikten sonra oluşan sırlı dosyanın histogram grafiği çizilir. Şekil 3.10'da sayısal ses örneklerinin değerlerine bakılmaktadır. Birbirinin aynısı olan çift değerli ses örneklerinin sayısı CiftStego adlı dizide tutulmakta tek değerlere sahip değer çiftleri ise TekStego adlı dizide tutulmaktadır. Burada amaç değer çiftlerinin frekanslarını bulmaktır. Son olarak bu değer çiftlerinin frekanslarının histogram grafiği çıkartılır.

```
stego_verisi = stego_verisi - (2^(kacBit-1));
stego_verisi = stego_verisi ./ (2^(kacBit-1));
```

Şekil 3.11. Veri saklama uygulaması kod parçası 8

Sırlı veri oluştuktan sonra normalize işlemi yapıldığı için sırlı veriler normalize değerlere sahip verilerden oluşmaktadır. Dolayısıyla gerçek verilere dönmek için yazılmış olan Şekil 3.11'deki satırlarda normalize işleminin tersi yapılmaktadır. Böylece verilerin sayısal değerleri eski orijinal değerlerine dönmüş olur.

```
wavwrite(stego_verisi, cerceveHizi,kacBit,stego_dosya_adi);
```

Şekil 3.12. Veri saklama uygulaması kod parçası 9

Son olarak da oluşturulan sırlı dosyayı kaydetme işlemi yapılır. Şekil 3.12'de kullanıcının girmiş olduğu isim de ve orijinal örtü dosyasından alınan çerçeve hızı ve bit değerleri ile sırlı dosya kaydedilir.

Ses dosyasının stereo ve mono olması veri gizleme kapasitesi açısından farklılıklar içermektedir. Stereo ses dosyaları 16-bitlik ses örnekleri içermekte ve her 8-bitlik veriye bir bitten oluşan mesajlar gizlenebilmekte ve dolayısı ile her 16-bitlik örneğe 2 bit mesaj gizlenebilmektedir. Oysa mono ses dosyalarında her ses örneği 8-bit ile temsil edilir ve her ses örneği sadece 1-bit mesaj alır. Bu durumdan da anlaşılacağı üzere stereo ses dosyalarına mono ses dosyalarına oranla iki kat fazla bilgi saklanabilmektedir.

3.3. Yapılan Testler ve Sonuçlar

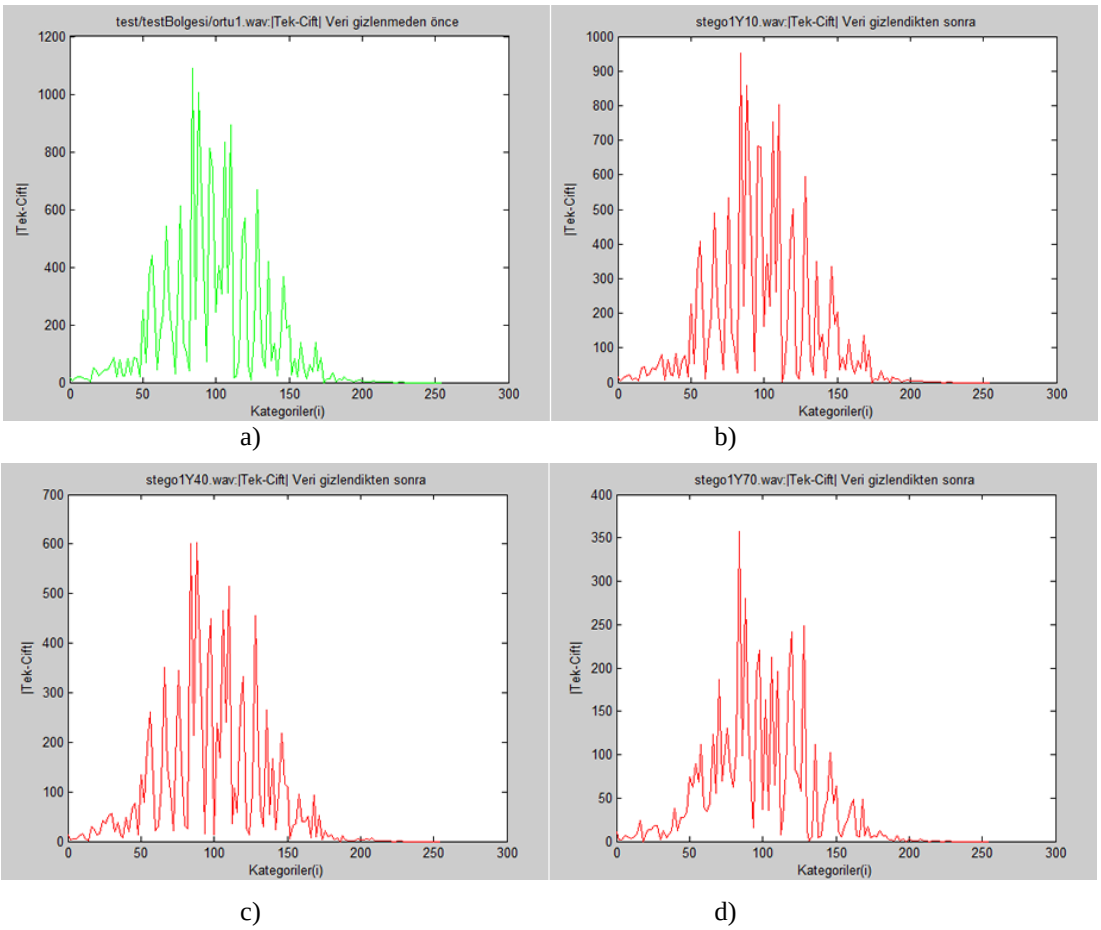
Uygulamanın çalışılabilirliğini test edebilmek için 5 farklı WAV dosyaya 4 farklı oranda gizli veri gömülmüştür. Örtü dosyaları Ortu1.wav…Ortu5.wav olarak isimlendirilmiştir. Her dosyaya %10, %40, %70, %100 oranında mesaj gömülmüş ve oluşan sırlı dosyalar belli bir formatta kaydedilmiştir. Örneğin Ortu1.wav dosyası % 40 veri gizlenmesi sonucu oluşan sırlı dosya Stego1Y40.wav olarak isimlendirişmiştir. Aynı şekilde Ortu3.wav dosyasına %70 oranında veri gömülürse oluşan sırlı dosya Stego3Y70.wav olarak isimlendirilir. Böylece dosya adından hem kaçıncı dosya hem de hangi oranda veri gömüldüğü anlaşılabilmektedir. Test edilen dosyaların boyutları birbirinden farklıdır. 10 sn ile 30 sn arası değişen sürelere sahiptirler. Bu sesler yabancı kaynaklı bir web sitesinden indirilmiş olup konuşma sesi olabildiği gibi hayvan sesleri ve müzik sesleri de bulunmaktadır ([61],[62]).

Bu dosyaları seçerken farklı ses bilgileri içermesine dikkat ettik. Seslerin farklı olması, testin doğru sonuçlar vermesi açısından önem taşımaktadır. Çünkü farklı sesler ile çalışarak her ortamda doğru sonuçlar elde edilip edilmediği test edilebilir. Ses dosyalarında konuşma seslerine de yer verildi. Konuşma sesleri aynı kişinin

konuşma sesi ise aynı tonlarda olmakta ve ses veri örnekleri yakın bilgiler içermektedir. Bu gibi dosyalarda sıraçma işlemi daha zor olmaktadır. Çünkü dosyadaki birbirini tekrar eden veri örnekleri bulunmakta ve değer çiftleri mantığına aykırı düşmektedir. Bu yüzden konuşma sesleri içeren dosyalar güvenlik açısından daha güvenli olduğu söylenebilir.

Sıraçma uygulamasında bulunan yapay sinir ağını eğitebilmek için öncelikle veri kümesine ihtiyaç vardır. Bu veri kümesini elde edebilmek için yukarıda anlatılan veri gizleme uygulaması kullanılarak Ek D'de verilen toplu veri gizleme uygulaması yazılmış ve bu uygulama ile seçilen dosyalara toplu olarak veri gömülmesi yapılmıştır. Yukarıda bahsedilen 5 örtü dosyasından farklı 10 adet örtü dosyası seçildi. Bunlar OrtuEgitim1.wav…OrtuEgitim10.wav şeklinde isimlendirildi. Her eğitim dosyası için iki farklı mesaj rastgele olarak oluşturulmuştur. Mesajların azami boyutları örtü ses dosyasının ses örneklerinin son bitlerinin sayısı kadar olabilir. Burada oluşacak olan mesajın matrissel olarak satır sayısı ses dosyasının satır sayısı ile aynı olacaktır. Aynı şekilde sütun sayıları da eşit olacaktır. Bu mesajların yüzde 1 ila 100 arası değerleri ses dosyasına gömülmüştür. Örneğin OrtuEgitim2.wav dosyasına 1.mesaj %67 oranında (ses dosyasının %67'sine veri gömülmüş anlamında) gömülmüş olsun. Bu durumda oluşacak sırlı dosya StegoEgitim2K1Y67.wav isminde olacaktır. Kaydetme biçimi [DosyaAdi][MesajNo][YüzdeDeğer].wav şeklindedir. Bu şekilde 10 farklı örtü dosyasına 2'şer mesajın yüzde 1 ila 100 arası gömülmesi sonucu 2000 farklı sırlı dosya oluşur. Bu dosyalar ile ilgili eğitim ve analiz kısımları Bölüm 4'de anlatılmaktadır.

Yukarıda bahsedilen 5 dosyaya %10, %40, %70, %100 oranında veri gömülmesi sonucunda oluşan histogram grafiklere yer verilecektir. Aşağıda Ortu1.wav dosyasına %10 veri gömülmesi sonucu oluşan histogram grafikler verilmektedir.



Şekil 3.13. Ortu1.wav dosyasına veri gizlenmesi

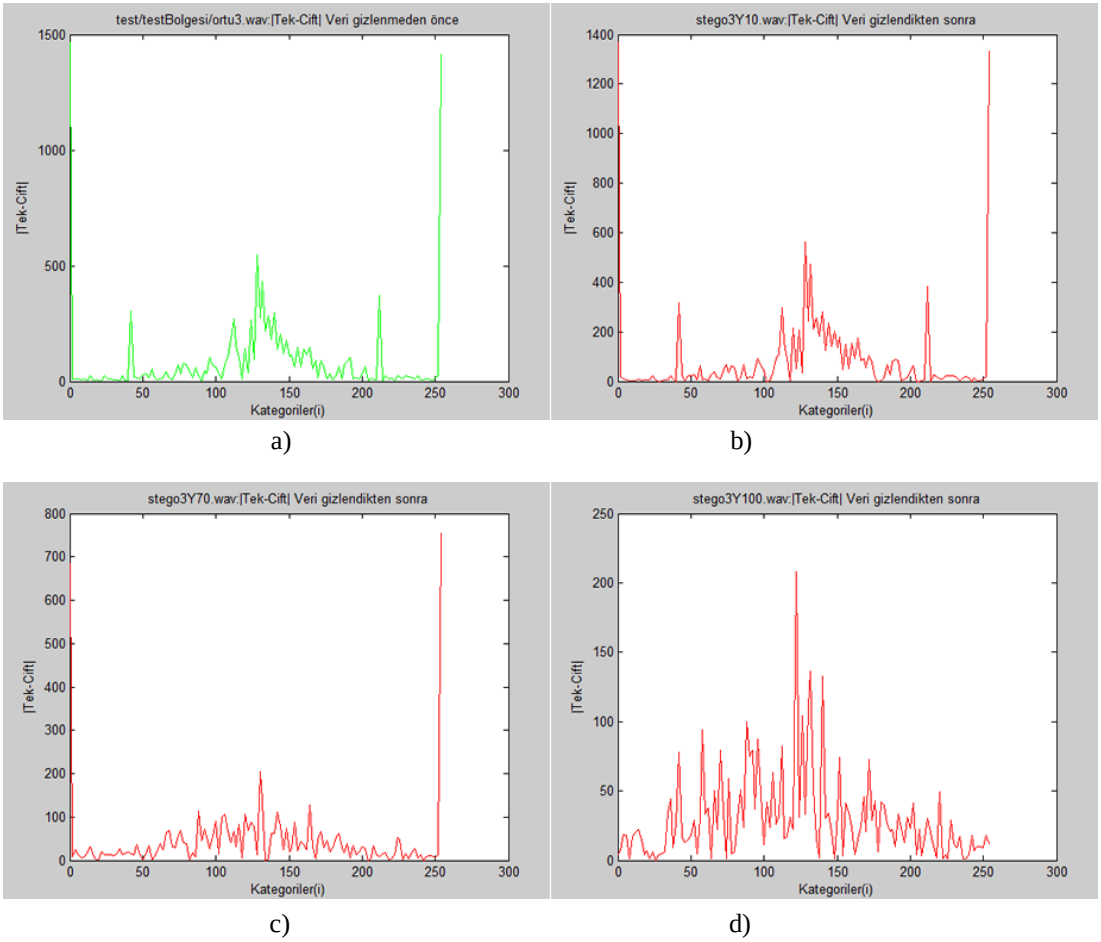
a)- Orijinal dosyanın histogram analizi

b)- %10 veri gizlenmiş sırlı dosyanın histogram analizi

c)- %40 veri gizlenmiş sırlı dosyanın histogram analizi

d)- %70 veri gizlenmiş sırlı dosyanın histogram analizi

Şekil 3.13'de Ortu1.wav dosyasına tasarlanan sırörtme uygulaması ile farklı oranlarda veri gömülmüş ve elde edilen histogram grafiklerinden de anlaşılacağı üzere veri gömülme oranı yükseldikçe şekilsel olarak bir farklılık görünmese de aslında bazı farklılıklar göze çarpmaktadır. Veri gömülme oranı arttıkça y eksenindeki veriler yani tekil değer çiftlerinin frekansları ile çift değer çiftlerinin frekansları farkı azalmaktadır. Bunun nedeni veri gömülme oranı arttıkça çift ve tek (2k, 2k+1) değer çiftlerinin frekansları arasındaki fark kapanmaktadır. Fakat kategori sayısında bir değişme olmamaktadır. Bu da şunu gösteriyor ki son bitlere ne kadar çok veri gömülürse değer çiftlerinin frekansları birbirine o kadar yaklaşmaktadır. Bu, zaten daha öncede anlatıldığı gibi Ki-kare saldırısının temelini oluşturan fikri doğrulamaktadır.



Şekil 3.14. Ortu3.wav dosyasına veri gizlenmesi

a)- Orijinal dosyanın histogram analizi

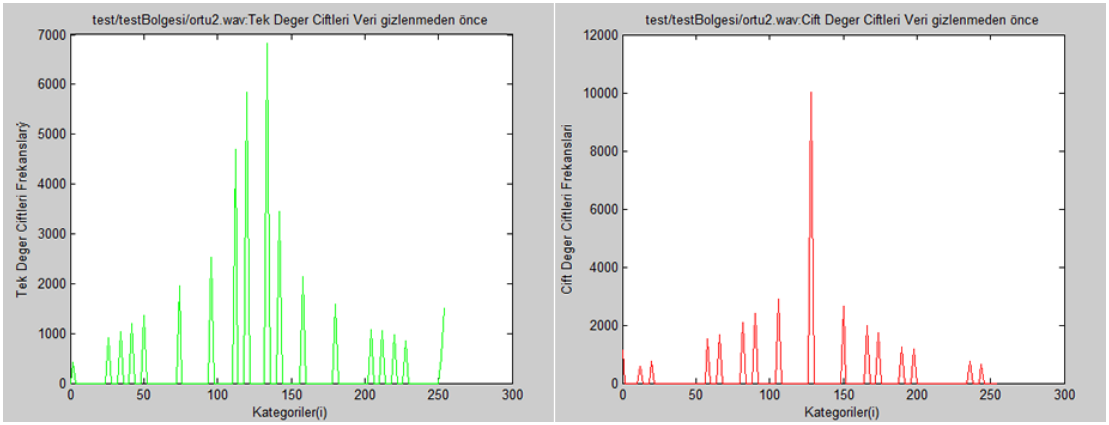
b)- %10 veri gizlenmiş sırlı dosyanın histogram analizi

c)- %70 veri gizlenmiş sırlı dosyanın histogram analizi

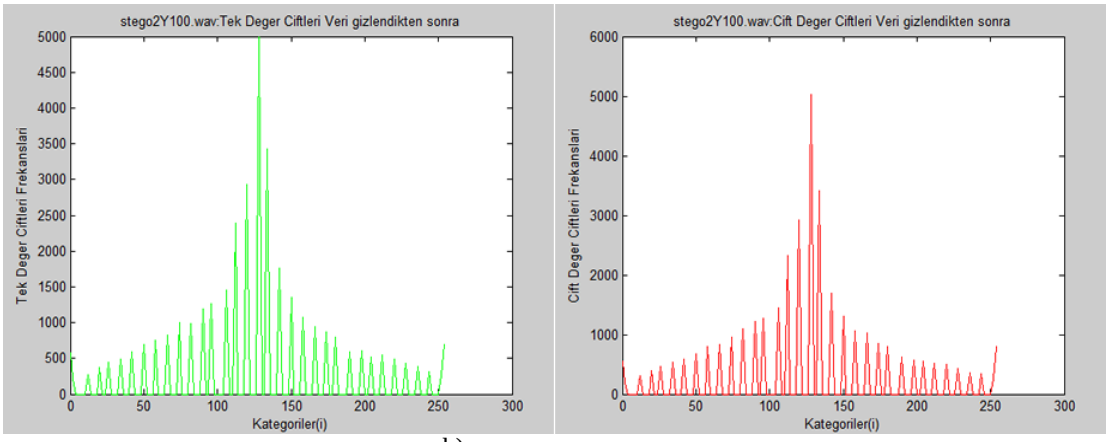
d)- %100 veri gizlenmiş sırlı dosyanın histogram analizi

Şekil 3.14'de Ortu3.wav dosyasına tasarlanan sırörtme uygulaması ile farklı oranlarda veri gömülmüş histogram grafiklerinde şekilsel olarak yine farklılık yoktur. Sadece son grafikte Şekil 3.14.d. için şekilsel bir farklılık varmış gibi gözükse de aslında normalize işlemden dolayı şeklin biraz büyütülmüş hali gözükmektedir. Yine gizlenen veri oranı arttıkça değer çiftleri arasındaki fark kapanarak frekans farklarını küçültmüştür. Şekil 3.14.b. ve Şekil 3.14.c.'de 250.kategoride görülen yüksek tepe değeri Şekil 3.14.d.'de kaybolmaktadır. Bunun nedeni dosya sonundaki Tek veya Çift değerlikli değer çiftleri arasındaki fark

oldukça fazla iken dosyanın sonuna kadar veri gömülmesi ile bu fark ortadan kalkmaktadır.



a)



b)

Şekil 3.15. Ortu2.wav dosyasına veri gizlenmesi

a)- Orijinal dosyanın Tek ve Çift değerlikli değer çiftlerinin analizi

b)- %100 veri gizlenmiş sırlı dosya Tek ve Çift değerlikli değer çift. analizi

Şekil 3.15'de verilen Ortu2.wav dosyasına tasarlanan sırörtme uygulamasının grafik çizimi değer çiftlerinin frekanslarını gösterecek şekilde değiştirilerek çizilen grafiklerde de göründüğü gibi orijinal dosyada tek ve çift değerlikli değer çiftleri frekansları birbirinden oldukça farklı iken %100 veri gömülü sırlı dosyada bu değerler birbirine çok yakın hatta neredeyse aynıdır. Sonuç olarak şu gözlenmiştir ki dosyanın son bitlerine veri gömülmesi ile değer çiftlerinin frekansları birbirine yaklaşmaktadır. Ne kadar fazla veri gömülürse o ölçüde frekanslar birbirine yaklaşmakta ve Şekil 3.15.b.'de verilen grafiği doğrulamaktadır.

BÖLÜM 4. Kİ-KARE YÖNTEMİ İLE SIRAÇMA UYGULAMASI

Bu bölümde daha önceki bölümde detayları verilen uygulama ile oluşturulan ses dosyaları geliştirilen yapay zekâ yöntemini kullanan sıraçma uygulaması ile analiz edilecektir. Sıraçma uygulaması, ses dosyası içinde gizli verinin varlığının ve ses dosyasının hangi oranda gizli veri olduğunun yanıtını arayacaktır.

Daha önce bahsedildiği gibi Internet teknolojilerinin gelişen Internet ağı ile daha da gelişmesi farklı iletişim kanallarının oluşmasına neden olmuştur. Önceleri sadece mesajlar ile iletişim kurulabilirdi. Bu yöntemi kullanan programlara örnek olarak sohbet programları gibi uygulamalardır. Internet ağlarının gelişmesi ve daha hızlı bilgi transferinin sağlanması ses ile iletişime olanak verdi. Bazı uygulamar ile Internet üzerinden telefon görüşmesi yapılmaya başlandı. Tabi bununla da sınırlı kalmayan teknoloji geliştikce artık veri aktarım hızı daha da artarak sesli videoların transferine kadar ilerledi. Artık insanlar çeşitli programlar aracılığıyla Internet üzerinden görüntülü konuşma yapabiliyorlar. Bu kadar ilerleyen Internet teknolojileri bu gibi ses ve video iletişimine olanak sağlamakla birlikte aslında gizli iletişim kanallarının da oluşmasına sebep olmuşlardır. Ses dosya transferleri ile insanlar üçüncü kişilerin fark edemeyeceği şekilde güvenli bir iletişim sağlayabilmektedirler. Sırörtme yöntemleri ile ses dosyalarına her türlü gizli bilgi saklanarak gizli iletişim sağlanabiliniyor.

Sırörtme yöntemlerinin gelişmesi ile sıraçma bilimi ortaya çıkmış ve yeni yeni gelişmeye başlamıştır. Ses dosyalarına yapılan bu tür sırörtme yöntemlerine karşı olarak geliştirilen uygulamada yapay zekâ yöntemlerinden yararlanılmıştır. Sıraçma çalışması yapan uygulamadan çıkarılan sonuçları yapay sinir ağında değerlendirerek daha yakın ve doğru sonuçlar elde edilmiştir. Aşağıda kullanılan yapay zekâ yöntemi açıklanmıştır. Yapay zekâ yöntemleri her alanda kullanılabilir ve çoğu uygulamada gerçeğe yakın sonuçlar üretmektedir.

Saldırı yöntemi olarak kullanılan Ki-kare yöntemi, Westfeld ve Pfitzmann bir görüntüdeki en az önemli bitlerin tam olarak rastgele olmadığını düşünmeleri ile ortaya çıkmıştır. Onlara göre her bir değer çiftindeki iki pikselin her birinin sıklığının değer çiftinin ortalamasından uzağa düşme eğiliminde olduğunu düşünmektedirler. Yani veri gizlenmiş bir görüntü içerisinde 2n piksel değerinin sıklığının 2n + 1 piksel değerinin sıklığına yakın olması olası bir durum değildir. Yukarıda anlattığımız veri gizleme uygulaması ile veri gömülmesi nedeniyle 2n ve 2n + 1 değer çiftlerinin frekansları birbirine eşit ya da eşit sayılır. İşte bu durum Ki-kare saldırısının temellerini oluşturmaktadır. Buradaki frekans eşitliğine göre ses dosyasına veri gömülme ihtimali hesaplanmaktadır.

Ki-kare saldırısı çeşitli gömme algoritmalarına uyarlanabilir olarak tasarımlanmıştır, ancak temel kavram gömme algoritmasına bakılmaksızın aynıdır. Sıraçma algoritması, ses dosyasına Ki-kare saldırısı gerçekleştirip veri gömme olasılığını çıktı olarak vermektedir. Algoritma dosyanın % 1'den % 100'e kadar test etmekte her bir yüzdesel kısım için olasılıkları hesaplamaktadır. Uygulama analiz işlemini veri gizleme tekniğinde olduğu gibi baştan sona doğru yapmaktadır. Sırörtme algoritmasının saldırgan açısından bilinmediği genel olarak varsayılmalıdır.

Cift(n) = (2n) sıklığı ve Tek(n)=(2n + 1) sıklığı, $0 \leq n \leq 127$ olacak şekilde $Cift^{128}$ ve Tek^{128} iki vektör olsun. İlk olarak Cift ve Tek'deki her bir eleman 0'a ayarlanır. Daha sonra algoritma ses dosyasındaki örnekleri sayar ve Cift ya da Tek'deki karşılık gelen elemanı artırır. Teorik olarak 2n ve 2n + 1'in ses örneklerinin değerlerinin beklenen sıklığı Zn = (Cift(n) + Tek(n)) / 2'dir. Şimdi n adet kategori olduğunu yani n değer çifti olduğunu farz ediniz. 8 bitlik gri görüntüler olması durumunda, 128 kategori (256 / 2) vardır. Genelliğini kaybetmeksizin, n kategorisinde ölçülen meydana gelme sıklığı Cift(n) olacak şekilde değer çiftlerinin çift değerleri üzerinde odaklanacağız. (Bir örtü görüntüsü ve sırlı görüntü içerisindeki 2n ve 2n + 1 piksel değerlerinin sıklığının toplamlarının aynı olduğuna yani bir örtü görüntü içerisindeki Cift(n) + Tek(n)'nın ve karşılık gelen sırlı görüntü içerisindeki Cift(n) + Tek(n)'e eşit olduğuna dikkat ediniz. Daha sonra Westfeld ve Pfitzmann minimum sıklık koşulu ortaya koymuştur, dolayısıyla eğer $0 \leq n \leq 127$ Aralığı için Cift(n) + Tek(n) ≤ 4 ise, Cift(n) = Tek(n) = Z(n) = 0 ve n = n - 1'dir. Diğer bir deyişle 2n ve 2n+1'in birleşik sıklığı 4'ten az ya da eşitse, 2n ve 2n + 1'in bireysel sıklık sayımları 0'a ayarlanır ve n

kategorisi sayısı 1 azaltılır. Daha sonra n – 1 serbestlik derecesiyle Ki-kare istatistiği hesaplanır.

$$X_{n-1}^2 = \sum_{i=1}^{127} \frac{(Cift(i) - Z(i))^2}{Z(i)} \quad , Z(i) = \frac{Cift(i) - Tek(i)}{2} \tag{4.1}$$

Bir sırlı görüntü için X_i'nin Z_i 'ye yakın olması gerektiğinden dolayı X_{n-1}^2'in görece küçük olması ve X_i'nin Z_i'den uzak olması gerektiğinden dolayı X_{n-1}^2'in görece büyük olması beklenir. İşlemin son adımı, yoğunluk işlevinin üst sınır olarak X_{n-1}^2 ile integralini alarak gömme olasılığını, p, hesaplamaktır.

$$P = 1 - \frac{1}{2^{\frac{n-1}{2}}\Gamma(\frac{n-1}{2})} \int_0^{x_{n-1}^2} e^{-\frac{u}{2}}\, u^{\frac{n-1}{2}-1} du \tag{4.2}$$

Bu gömme olasılığı Eşitlik (4.1)'deki bütün i'ler için X_i = Z_i koşulu altında X_{n-1}^2 olasılığıdır. 1 - p yoğunluk fonksiyonu, X_{n-1}^2 sonsuza yaklaşırken 1'e yakınsar, dolayısıyla X_{n-1}^2 sonsuza yaklaşırken p sıfıra yaklaşır. Dolayısıyla büyük X_{n-1}^2 değeri için gömme olasılığı 0'a yakındır. Bununla birlikte, X_{n-1}^2 n-1'a göre küçükken, 1 – p sıfırdır ve dolayısıyla p bire yakındır. Böylece görece küçük X_{n-1}^2 değeri için gömme olasılığı 1'e yakındır. Ek olarak, Westfeld ve Pfitzmann eğer piksellerin %100'ünden daha azı gömülü bilgi içeriyorsa, daha yüksek yüzde de pikseller test edildiğinde gömme olasılığı sert bir şekilde düşecektir.

4.1. Kullanılan yapay zekâ yöntemi

PNN (Probabilistic Neural Networks) yani olasılıksal sinirsel ağlar tekniği tez çalışmasında kullanılmıştır. Bu yöntem varolan bilgiler ile yeni bilgiler elde edebilme ya da bilgiyi bu bilgiler ışığında tahmin edebilme gibi işlemler için ayrıca sınıflandırma işlemleri için kullanılmaktadır. Her yapay sinir ağında olduğu gibi bu ağlarda da öncelikle eğitim sonrasında test aşaması vardır. Eğitimde öncelikle girdiler verilir ve bu girdilerin neye karşılık olduğunu belirten çıktıları verirsiniz. Bu şekilde ağ eğitim aşamasında geçer ve öğrenir. Daha sonra istediğiniz bir bilgiyi elindeki veriler ışığında cevaplar ve hangi sınıfta olduğunu bulmaya çalışır.

Ağ uygulaması MATLAB uygulamasında gerçeklenmiştir. Şekil 4.1.'de PNN ağının MATLAB'da kullanımı ile ilgili örnek bir kod bulunmaktadır.

```
P  = [1 2 3 4 5 6 7];
Tc = [1 2 3 2 2 3 1];
T = ind2vec(Tc);
net = newpnn(P,T,0.5);
Y = sim(net,P);
Yc = vec2ind(Y);
```

Şekil 4.1. Pnn ağının Matlab'da kullanımı

Burada P adlı değişkene giriş verileri yüklenir. Burada her bir sütundaki eleman bir giriş olur. Yani örnekte 7 adet giriş verisi bulunmaktadır. PNN ağının özelliği her giriş için bir nöron (sinir) hücresi oluşturmasıdır. Bu yöntemde nöron sayısı fazladır ve bu nedenle hızlı çalışmaktadır. Genelde yapay sinir ağlarında fazla nöron kullanılmak istenmez fakat istatiksel işlem yapan bu ağ da fazla sinir hücresi kullanılması zorunludur. Fazla nöron kullanımı, sonuçların daha iyi olmasını sağlar.

Tc değişkeni çıkışları temsil eder ve girişte kullanılan her sütüna karşılık bu matristede bir sütun olmalıdır. Yani giriş sayısı kadar çıkış da olmalıdır. Bunun nedeni her girilen giriş verisine karşılık bir çıkış değer gösterilmekte ve böylece ağa giriş ve çıkış arasındaki ilişki öğretilmektedir.

Akla şöyle bir soru gelmiş olabilir: "Her giriş verisi tek değerlerden oluşmuyor ise ne olacak?" Bu durumda her giriş değeriniz birden fazla ise bu giriş değerlerini satır olarak belirtebiliriz. Örnek olarak aşağıdaki kod konuyu daha iyi anlaşılır hale getirir.

```
P = [ 1   7   2
      3   9   4
      5   11  6];
```

Şekil 4.2. P Matris gösterimi

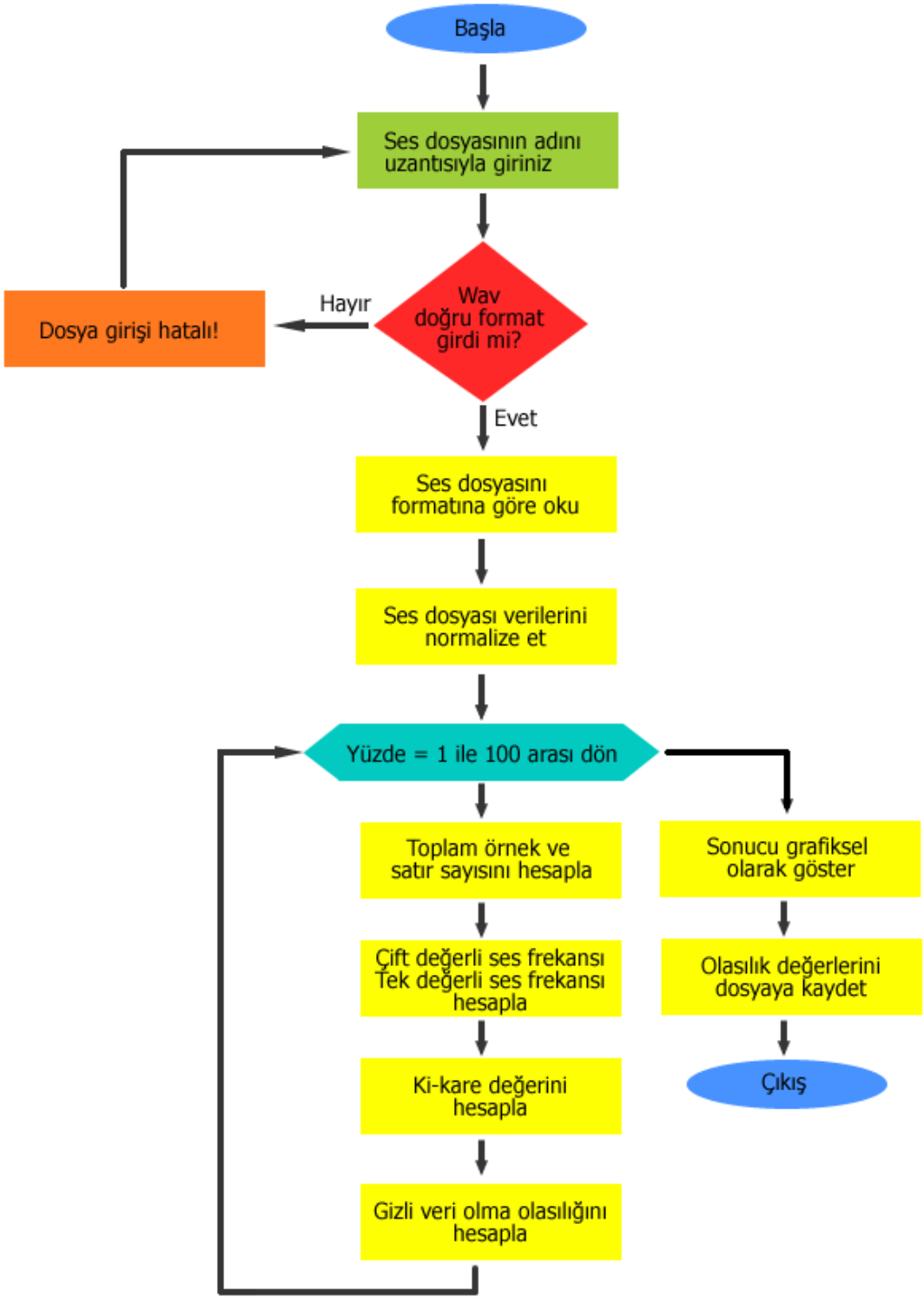
Şekil 4.2'deki örnekte 3 tane giriş vardır. Her giriş 3 değer içermektedir. Buradan da anlaşılacağı üzere her sütün bir girişi ifade etmektedir. 1, 3, 5 verisi bir giriş olarak kabul edilmektedir.

Ayrıca, her girişe karşılık farklı bir çıkış değeri olmak zorunda değildir. Zaten bu durum sınıflandırmanın doğasına aykırıdır. Böyle bir durum olsa, sınıflandırma diye bir kavram olmaz. Çünkü her giriş bir çıkıştan ibaret olacak ve sınıflandırılacak bir değer olmayacaktı. Burada birden fazla giriş aynı çıkışa karşılık gelebileceğini gösterilmiştir. Yani 1. sutundaki giriş ile 7.sütundaki giriş değeri aynı çıkışa yani 1 değerine karşılık geldiği gözlenmektedir.

Daha sonra ind2vec fonksiyonu ile matrisler PNN ağının istemiş olduğu vektör yapısına dönüştürülmektedir. Newpnn fonksiyonu ile ağ oluşturulur. Ağa giriş değerleri olan P ve çıkış değerlerinin vektörel dönüşümü olan T değişkenleri verilir. 0.5 olarak belirtilen değer ise dağılım (spread) faktörüdür. Bu değer verilmez ise default 0.1 olarak kabul edilir. Dağılım faktörü giriş değerlerinin dağılımı ile ilgili bir değerdir. Bu değer giriş değerlerinin dağılımına göre ayarlanabilir. PNN ağının en iyi sonucu bulabilmesi için bu değerinin en optimum düzeyde seçilmesi gerekir. Bu optimum düzey deneme yanılma yöntemi ile bulunmaktadır.

Ağ oluştuktan sonra sim fonksiyonu ile ağı eğitilir. Eğitim süresi giriş ve çıkış değerlerinizin sayısı ile orantılıdır. Yalnız bu ağ her giriş için bir nöron hücresi oluşturduğundan oldukça hızlıdır. Ne kadar çok bol örnek ile eğitilirse sonuçlar bir o kadar iyi olur. Eğitim sonrasında sonuç Y değişkenine aktarılır ve son olarak vec2ind komutu ile vektörel çıkış değeri matris yapısına dönüştürülür.

4.2. Sıraçma İşlemi ve Detayları



Şekil 4.3. Sıraçma uygulaması akış şeması.

Sıraçma uygulamamız düşük bit kodlaması yöntemi ile saklanan verilere duyarlı Ki-kare saldırısı yapan bir analiz uygulamasıdır. Düşük bit yöntemi ses örneklerinin

son bitleri ile gizlenecek verinin son bitlerini değiştirme işlemi yani LSB yöntemidir. Sesin analog ortamlara girmeksizin, tamamen sayısal ortamlarda transferi durumunda kullanılabilir.

Sıraçma çalışmaları gittikçe artmaktadır. Fakat bu çalışmalar genellikle resim dosyaları üzerinde yoğunlaşmıştır. Ses dosyaları üzerinde yapılan çalışmaların sayısı oldukça azdır. Sıraçma uygulamamız ile ses dosyaları içerisinde düşük bit kodlaması yöntemi ile gizlenmiş verinin varlığı araştırılmakta ve gizli veri varsa ses dosyasının hangi oranda veri olduğu bilgisi bulunmaya çalışılmıştır.

Gizli verinin varlığını araştıran sıraçma yöntemimiz aynı zamanda dosyanın hangi oranda veri gizlendiği hakkında bilgiler vermektedir. Ancak sayısal açıdan gizli veri miktarını oransal olarak vermemektedir. Bu nedenle bir yapay zekâ çalışması yapılmış ve sayısal bir değer verilmeye çalışılmıştır. Bunun için daha önce de anlattığımız PNN yöntemi kullanılmıştır.

Şekil 4.3'de sıraçma uygulamasının akış diyagramı verilmektedir. Aşağıda Şekil 4.4'de sıraçma uygulamasının kaynak kodlarına yer verilmekte ve yaptığı işlemler sırası ile anlatılmaktadır.

```
calismaYeri='test/testBolgesi/';
sonuc=false;
while sonuc == false
   stego_dosya_adi = lower(input('Sıraçma dosya adı(örnegin stego.wav)', 's'));
   uzunluk=size(stego_dosya_adi);
   bulunanIndex=strfind(stego_dosya_adi,'.wav');
   if (bulunanIndex == uzunluk(2)-3)
      sonuc=true;
      dosya_tipi='wav';
   else
      sonuc=false;
      disp('Dosya girişi hatalı!');
   end
end
```

Şekil 4.4. Sıraçma uygulaması kod parçası 1

Uygulamayı test ederken kolaylık sağlaması açısından test ettiğimiz dosyaları testBolgesi adlı klasörde tuttuk. Böylece yapılan birçok test sonucu oluşan dosya kalabalığından kurtulabiliriz. Uygulama analiz etmek istediği dosyayı testBolgesi adlı klasörde aramaktadır. Şekil 4.4'deki kod parçasında analiz edilecek WAV dosya adı kullanıcıdan istenir. Kullanıcı doğru bir giriş yapmazsa örneğin "ses.wavr" gibi bir giriş yapar ise "Dosya girişi hatalı!" uyarısını verecek ve tekrardan kullanıcıdan dosya ismi girmesini isteyecektir.

```
stego_dosya_adi=strcat(calismaYeri, stego_dosya_adi);
[stego_verisi, cerceveHizi, kacBit]  = wavread(stego_dosya_adi);
stego_verisi=round(2^(kacBit-1).*(stego_verisi))+2^(kacBit-1);
[satir, sutun] = size(stego_verisi);
```

Şekil 4.5. Sıraçma uygulaması kod parçası 2

Şekil 4.5'de kullanıcıdan alınan ses dosyası çalışma klasöründen çalışacak şekilde düzenlenir ve ses dosyasının verisi, çerçeve hızı ve bitlik ölçüleri okunur. Sır verisi, ses dosyası okunarak alındıktan sonra birkaç dönüşüm işleminden geçirilmektedir. Bunun nedeni sır verisi içerisindeki sayısal bilgiler çok küçük ondalık değerler içermektedir. Üstelik bu değerler arasında negatif değerler de bulunmaktadır. Yazılım bu değerleri belli bir değere normalize etmektedir. Böylelikle hem pozitif değerler ile çalışılmakta hem de istenen değerler elde edilebilmektedir. Daha sonra sır verisinin satır sütun bilgileri alınarak ileride her bir ses örneği döngü ile incelenmek için kullanılacaktır.

```
yuzde = zeros(100,1);
olasilik= zeros(100,1);
kategori_say = zeros(100,1);
```

Şekil 4.6. Sıraçma uygulaması kod parçası 3

Şekil 4.6'de sıraçma işlemi için kullanılacak olan 100 elemanlı dizi değişkenleri tanımlanmakta ve başlangıç olarak zeros fonksiyonu ile tüm elemanlara 0 değeri yüklenmektedir.

```
for y=1:100
   n = 2^(kacBit-1); % n tane kategori var
   yuzde(y) = yuzde(y) + y;
   toplam_ornek = floor((y/100)*satir*sutun);
   kacSatir = floor(toplam_ornek/sutun);
```

Şekil 4.7. Sıraçma uygulaması kod parçası 4

Şekil 4.7'deki kodda sıraçma uygulamasının algoritmasına göre dosya yüzdelik dilimler halinde incelenir. %1'den başlanarak %100'e kadar tüm dosya incelenmektedir. Sıra ile önce ses dosyasının %1'inde gizli verinin varlığı olasılıksal olarak incelenir ve 0 ile 1 arasında bir değer hesaplanır. Burada hesaplanan olasılık değeri dosyanın %1'inde yani baş kısmında gizli verinin varlığının olasılığıdır. %1'inde gizli veri var ise 1'e yakın bir değer tersi ise 0'a yakın bir değer hesaplanır. Bu şekilde %2'sinden %100'üne kadar araştırma devam eder. Son olarak genel tablodan dosyada ne kadar veri olduğu saptanmaya çalışılır.

WAV dosyaları daha öncede anlatıldığı gibi stereo ve mono olmak üzere iki tipte olmaktadır. Mono ses dosyalarında her örnek 8 bitlik sayısal değer ile temsil edilir. Mono dosyalarda en fazla 2^{8-1}=128 adet kategori oluşmaktadır. Stereo dosyalarda ise en fazla 2^{16-1}=32768 adet kategori oluşabilmektedir. Böylece oluşabilecek maksimum kategori sayıları hesaplanmış olunur. toplam_piksel değişkeni o an hangi yüzdede çalışılıyor ise o yüzdelik değere karşılık gelen toplam ses örneklerinin sayısını hesap eder. Hesaplanan toplam ses örneklerinin matrissel formatta ne kadar satıra karşılık geldiğini hesaplanarak değişkene kaydedilir.

```
Tek=zeros(2^(kacBit-1),1);
Cift=zeros(2^(kacBit-1),1);
kiKare=zeros(2^(kacBit-1),1);
```

Şekil 4.8. Sıraçma uygulaması kod parçası 5

Şekil 4.8'da Ki-kare saldırısı için kullanılacak olan değer çiftlerinin frekanslarını ve Ki-kare sonuçlarını tutacak dizi değişkenleri başlangıç değerleri 0 olacak şekilde

tanımlanıyor. Burada Tek tek sayılı ses frekansını Cift ise çift sayılı ses frekansını hesaplar ve kiKare değişkeni Ki-kare değerlerini tutar.

```
for i=1:kacSatir
      for j=1:sutun
         if rem(stego_verisi(i,j),2)!=0
            Tek((stego_verisi(i,j)/2)+1)=Tek((stego_verisi(i,j)/2)+1)+1;
         else
            Cift(((stego_verisi(i,j)-1)/2)+1)=Cift(((stego_verisi(i,j)-1)/2)+1)+1;
         end
       end
end
```

Şekil 4.9. Sıraçma uygulaması kod parçası 6

Şekil 4.9'da örtü verisindeki her bir ses örneği 8 bit parçalar halinde incelenmekte ve bu sayısal ses örneklerinin değerlerine bakılmaktadır birbirinin aynısı olan çift değerli ses örneklerinin sayısı Cift adlı dizide tutulmakta birbirinin aynısı olan tek değer çiftleri ise Tek adlı dizide tutulmaktadır. Burada amaç Ki-kare yönteminde kullanılacak olan değer çiftlerinin frekanslarını bulmaktır.

```
      Z = (Tek + Cift)/2;
      for i=1:(2^(kacBit-1))
            if (Tek(i)+Cift(i)) < 5
               Tek(i) = 0;
               Cift(i) = 0;
                n = n - 1;
            end
       end
```

Şekil 4.10. Sıraçma uygulaması kod parçası 7

Şekil 4.10'deki kod parçasında tüm kategoriler incelenerek Ki-kare yönteminde bahsedilen tekil değer çiftlerinin ve çift değer çiftlerinin frekanslarının toplamı 4 den küçük veya eşit ise o zaman o frekans değerleri yok sayılır yani sıfırlanır ve kategori sayısı bir azaltılır. Böylece Ki-kare yönteminde bahsedilen durum gerçeklenmiş olur.

```
        kiKare = (Cift-Z).^2;
        for i=1:(2^(kacBit-1))
            if Z(i)==0
                kiKare(i) = 0;
            else
                kiKare(i) = kiKare(i)./Z(i);
            end
        end
```

Şekil 4.11. Sıraçma uygulaması kod parçası 8

Şekil 4.11'de Bölüm 4'ün başında verilen Eşitlik (4.1) uygulanır. Eşitlikte Z değeri 0 olursa 0'a bölme hatası vereceğinden işlem yapılmaz ve kiKare değişkenine 0 değeri atanıyor eğer 0'dan farklı ise o zaman bölme işlemi gerçeklenir.

```
        C=sum(kiKare);
        Pcheck=1-gammainc(C/2,(n-1)/2);
        olasilik(y) = olasilik(y) + Pcheck;
        kategori_say(y) = kategori_say(y) + n;
end
```

Şekil 4.12. Sıraçma uygulaması kod parçası 9

Eşitlik (4.2)'deki toplam fonksiyonu gerçeklenerek istatiksel Ki-kare olasılığı hesaplanıyor. Bulunan olasilik değeri olasilik dizisinin ilgili indisine kaydedilir ve son olarak kategori sayısı da ilgili indise kaydedilir.

```
fig_title = [stego_dosya_adi,'Veri Gömme Olasılığı Yüzdesi'];
kategori_say=cat(1,[0],kategori_say(1:100));
yuzde = cat(1, [0],yuzde(1:100));
olasilik = cat(1, [0], olasilik(1:100));
sonuc = cat(2, yuzde, olasilik, kategori_say);
display(sonuc);
figure (1), plot(yuzde, olasilik,'*:');
title(fig_title); xlabel('Yüzde Testi');
ylabel('Veri Gömme Olasılığı');
stego_dosya_adi=strrep(stego_dosya_adi, '.wav', '.mat');
save(stego_dosya_adi,'olasilik');
```

Şekil 4.13. Sıraçma uygulaması kod parçası 10

Hesaplanan değerler ile oluşan durumu görüntülemek için Şekil 4.14'deki verilen kod parçasında histogram grafiği basılır. Yüzdesel değerler 1'den başladığı için grafiği 0'dan başlatabilmek için oluşan dizi değerlerinin başına 0 elemanı eklenir. Yüzdesel değere karşılık gelen olasılık hesaplaması grafiksel olarak gösterilmektedir.

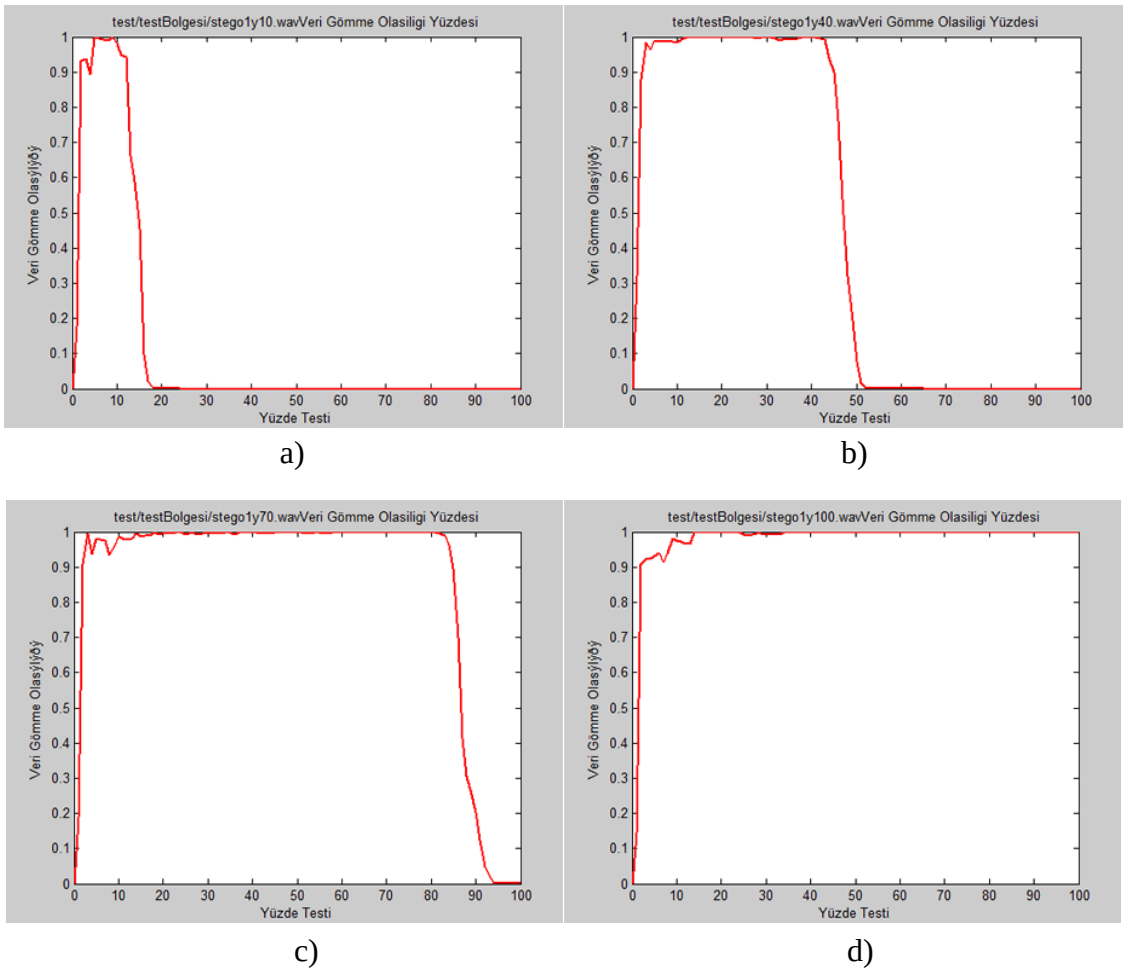
4.3. Yapılan Testler ve Sonuçlar

Sıraçma uygulamasının çalışılabilirliğini test edebilmek için 5 farklı WAV dosyaya 4 farklı yüzdesel oranla veri gizlenmişti. Örtü dosyaları Ortu1.wav…Ortu5.wav olarak isimlendirilmişti. Her dosyaya yüzdesel değerler olarak ise %10, %40, %70, %100 oranında mesaj gömülmüş ve oluşan sırlı dosyalar belli bir formatta kaydedilmişti.

Sıraçma uygulamasında bulunan yapay sinir ağını eğitebilmek için toplu veri gizleme uygulaması yazılmış ve bu uygulama ile seçilen dosyalara toplu olarak veri gömülmesi yapılmıştı. Yukarıda bahsedilen 5 örtü dosyasından farklı 10 adet örtü dosyası seçildi. Bunlar OrtuEgitim1.wav…OrtuEgitim10.wav şeklinde isimlendirildi. Her eğitim dosyası için iki farklı mesaj rastgele olarak oluşturulmuştu. Bu mesajların yüzde 1 ila 100 arası değerleri ses dosyasına gömülmüştü. Bu şekilde 10 farklı örtü dosyasına ikişer mesajın yüzde 1 ila 100 arası gömülmesi sonucu 2000 farklı sırlı dosya oluşmuştu.

Daha sonra oluşan 2000 adet sırlı dosyayı sıraçmadan geçirilerek sonuçları kaydedecek bir yazılıma ihtiyaç duyuldu ve Ek-E de verilen kaynak kodu yazıldı. Bu kod 2000 adet sırlı dosyaya sıra ile erişerek yukarıda anlatılan yöntemle sıraçma yapmış ve sonuçlar [stego_dosya_adi][.mat] formatında kaydetmiştir. Böylece her bir dosya için sıraçma yapılarak veri gömülme olasılığı hesaplanmış ve MAT dosyalarına kaydedilmiştir. Her MAT dosyası 100 değer içermektedir. Çünkü sıraçma uygulamamızda analiz yapılan dosyanın %1'i ile %100'ü arası test edilmiş ve her bir değer kaydedilerek 100 adet olasılık değeri hesaplanmıştır. Böylece 2000 adet olasılık sonuçları içeren MAT dosyası elde edilir. Her dosyada 100 veri olduğundan toplamda 200.000 olasılık değeri yapay sinir ağını eğitmek için kullanılmıştır. Oluşan 2000 MAT dosyasını eğitime sokacak bir yazılıma daha ihtiyaç duyulmuş ve Ek-F'de verilen kaynak kodu yazılmıştır. Bu kod, 2000 dosyaya tek tek erişerek MAT dosyalarını yüklemekte ve içerdiği 100 adet olasılık

değerini 1 giriş değeri olacak şekilde yukarıda 4.1.'de anlatılan P giriş dizisine eklemektedir. Tabi giriş değerine karşılık gelen çıkış değerini de Tc çıkış dizisine eklemektedir. Burada amaç, örtü dosyasına aynı mesajın %1-100'ü arasında farklı oranlarda gizleyerek, her gizleme sonucunda oluşan dosyalar ve gizleme oranları yardımıyla yapay sinir ağı eğitmektir. Bu şekilde her dosya için iki mesaj gizlenmiştir. Eğitim sonrası daha önceden eğitimde kullanılmamış bir dosyaya belli bir oranda veri gizlenerek yapay sinir ağından bu dosyaya yüzde kaç veri gizlendiğini tahmin etmesi istenecektir. Yukarıda bahsedilen 5 dosyaya %10, %40, %70, %100 oranında veri gömülmesi sonucunda oluşan sırlı dosyalara yaptığımız sıraçma saldırılarının sonuçlarına yer verilecektir.



Şekil 4.14. Stego1 grubu dosyaları için sıraçma saldırı sonuçları

a)- %10'nuna veri gizlenmiş (Stego1Y10) b)- %40'ina veri gizlenmiş (Stego1Y40)

c)- %70'ine veri gizlenmiş (Stego1Y70) d)- %100'üne veri gizlenmiş (Stego1Y100)

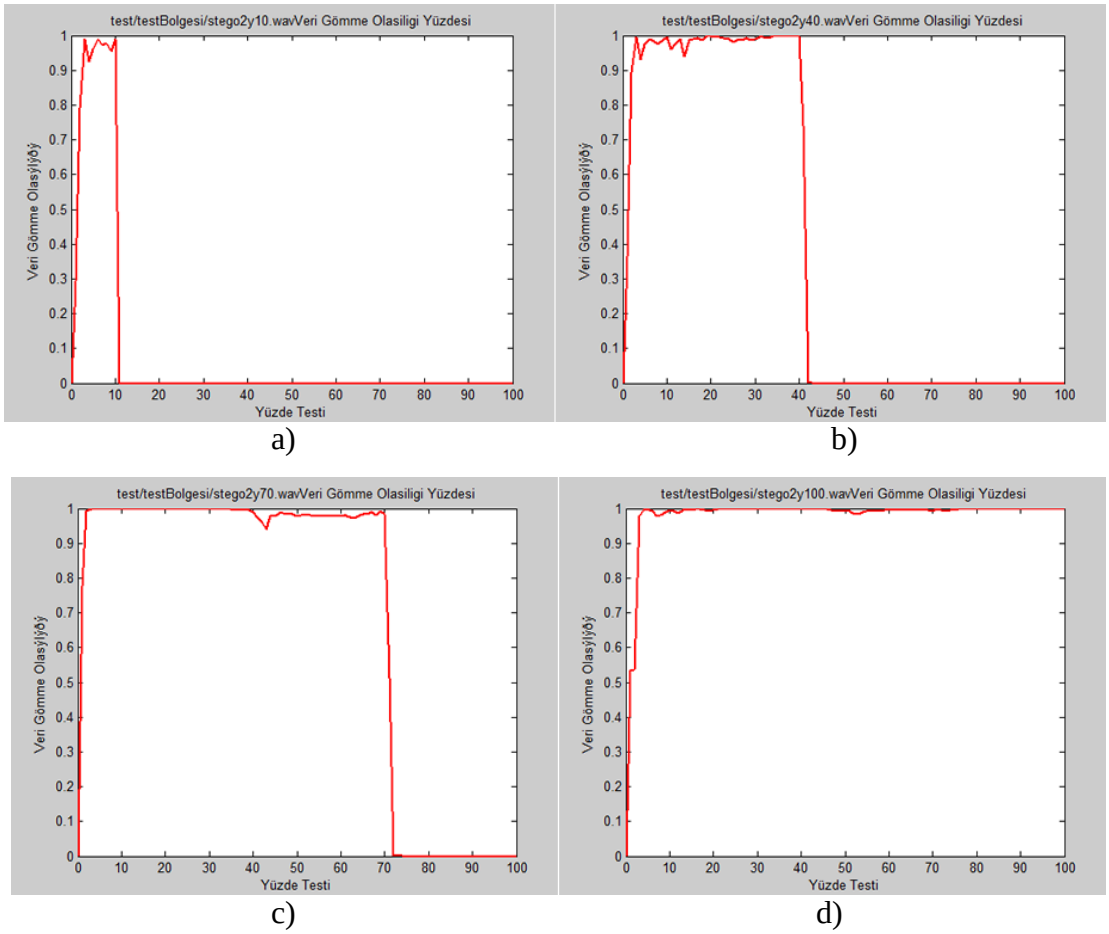
Şekil 4.14'deki tasarlanan sıraçma uygulamasının oluşturduğu histogram analizlerinde iyi bir sonuç gözlenmektedir. Ortu1.wav dosyasının Şekil 4.14.a. durumunda x ekseni tam %10'da iken ani bir düşme gözlenmiş %19'dan sonra olasılık değeri sıfır seviyelerine düştüğü gözlenmiştir. %40'lık veri gömülmesi sonucu Şekil 4.14.b. grafiğinde görüldüğü gibi %43 de iken ani bir düşme gözlenmiştir. %70'lik veri gömülmesi sonucu %82 'iye kadar düşme gözlenmemiş ve sapma gözlenmiştir. %100'lük veri gömülmesinde ise tam doğru bir şekilde ölçüm tesbit edilmiştir.

Yapılan bu dört sıraçma sonucunda olasılık değerlerinin kaydedildiği MAT dosyaları ile PNN yapay sinir ağında test edildiğinde Stego1Y70.mat dosyası oluşmaktadır. Stego1 dosyaları PNN ağı ile test edildiğinde Tablo 4.1'deki sonuçları vermiştir.

Tablo 4.1. Stego1 gurubu dosyaları için PNN ağı olasılık sonuçları

	Yüzdesel Oranlar (%)			
Gizlenen	10	40	70	100
Bulunan	12	44	78	100

Yukarıdaki sonuçlardan da anlaşılacağı üzere sıraçma saldırısı iyi sonuç veriyor ise PNN yapay sinir ağı de o ölçüde başarılı sonuçlar üretmektedir. PNN ağı ile bir dosyanın hangi oranda veri olduğunu söyleyebilmekteyiz. PNN yapay sinir ağı ne kadar çok fazla veri ile eğitilirse o ölçüde daha çok öğrenme gerçekleşmiş olur. Giriş değerleri arttıkça olabilecek durumlara karşı verilen cevaplar da o kadar iyi olacaktır. Bu şuna benzetilebilir, bir problem çözümünde ne kadar çok örnek mevcut ise o sorunu çözmek o kadar kolay olur ama o konu hakkında fazla bir örnek yok ise o zaman o sorunun çözümü daha zorlaşmaktadır.



Şekil 4.15. Stego2 grubu dosyaları için sıraçma saldırı sonuçları

a)- %10'nuna veri gizlenmiş (Stego2Y10) b)- %40'ine veri gizlenmiş (Stego2Y40)

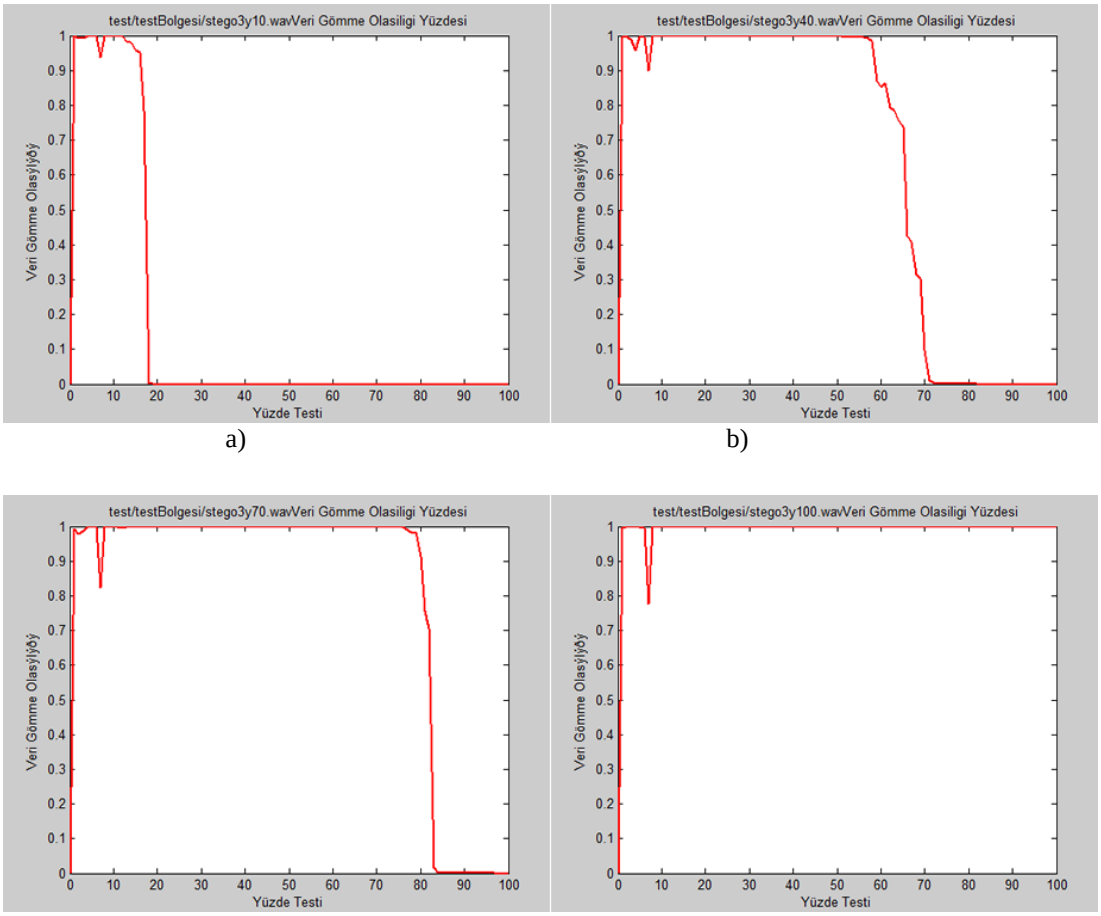
c)- %70'ine veri gizlenmiş (Stego2Y70) d)- %100'üne veri gizlenmiş (Stego2Y100)

Ortu2.wav dosyasından elde edilen sonuçlar Ortu1.wav dosyasından elde edilen sonuçlara göre daha iyidir. Ortu2.wav dosyasının şekil 4.15.a. durumunda x ekseni tam %10'da iken ani bir düşme gözlemiş %11'dan sonra olasılık değeri sıfır seviyelerindedir. %40'lık veri gömülmesi sonucu şekil 4.15.b. grafiğinde görüldüğü gibi %41 de iken ani bir düşme gözlenmiştir. %70'lik veri gömülmesi sonucu %71'de düşme gözlenmiş %100'lük veri gömülmesinde ise tam doğru bir şekilde ölçüm tesbit edilmiştir.

Tablo 4.2. Stego2 gurubu dosyaları için PNN ağı olasılık sonuçları

	Yüzdesel Oranlar (%)			
Gizlenen	10	40	70	100
Bulunan	10	41	71	100

PNN ağı tablodaki değerlere göre ortalama bir değer üretmektedir denilebilir.



a) b)

c) d)

Şekil 4.16. Stego3 grubu dosyaları için sıraçma saldırı sonuçları

a)- %10'nuna veri gizlenmiş (Stego3Y10) b)- %40'ine veri gizlenmiş (Stego3Y40)

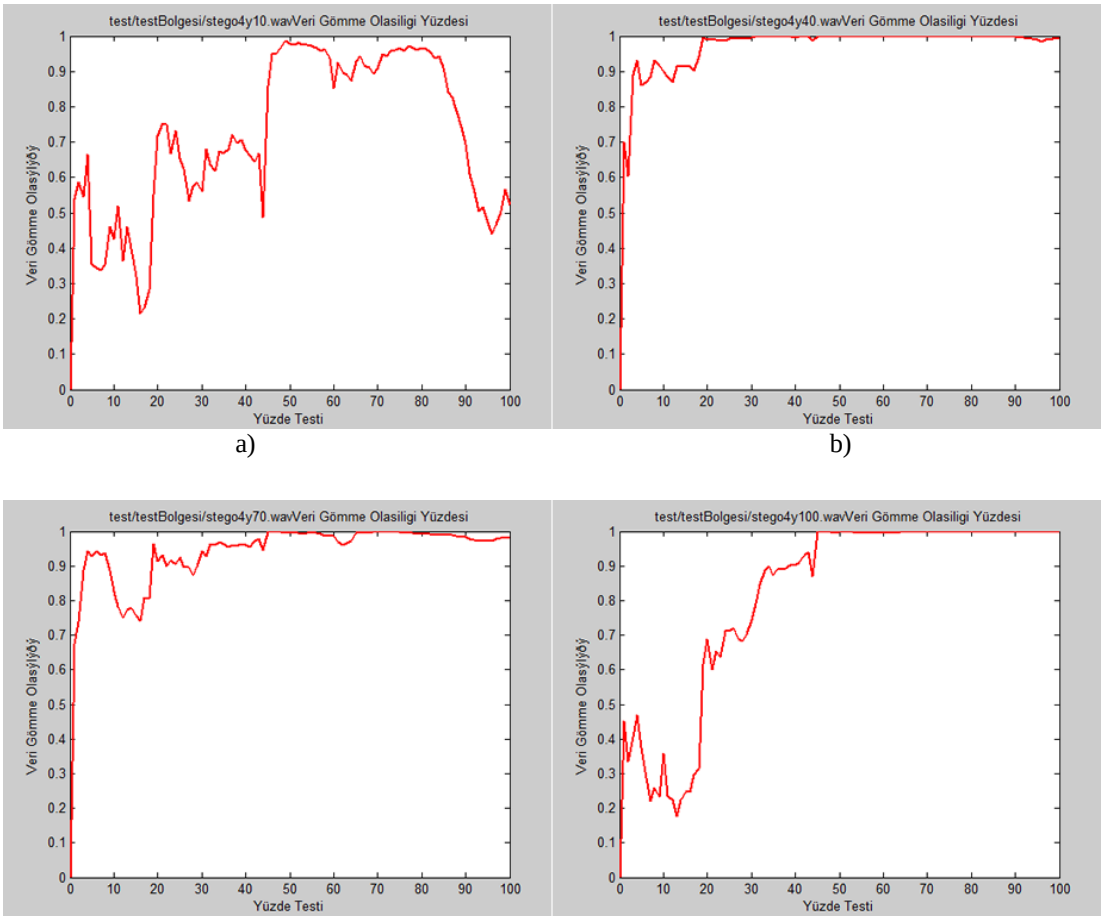
c)- %70'ine veri gizlenmiş (Stego3Y70) d)- %100'üne veri gizlenmiş (Stego3Y100)

Ortu3.wav dosyasından elde edilen sonuçlar diğerlerine göre biraz daha fazla sapma mevcuttur. Ortu3.wav dosyasının şekil 4.16.a. durumunda x ekseni %9'da iken ani bir düşme gözlemiş %19'dan sonra olasılık değeri sıfırlanmıştır. Bu ani düşme dosyaya %10 oranında veri gizlendiğinden tam o anda değer çiftlerinin frekanslarında farklılık olmasındandır. %40'lık veri gömülmesi sonucu şekil 4.16.b. grafiğinde görüldüğü gibi sapma olmuş ve %58 olana kadar düşme gözlenmemiştir.

Tablo 4.3. Stego3 gurubu dosyaları için PNN ağı olasılık sonuçları

	Yüzdesel Oranlar (%)			
Gizlenen	10	40	70	100
Bulunan	10	49	78	100

Şekil 4.16.a'da x ekseni %9'da , Şekil 4.16.b'de %9'da, Şekil 4.16.c'de %9'da ve Şekil 4.16.d' de de %9'daki ani düşme Ortu3.wav dosyasına farklı oranlar veri gömülse bile dosyanın tam o noktasına veri gizlenmesi sonucu oluşan değer çiftleri frekansları bu durumu ortaya çıkarmaktadır. Her dört durumda da aynı yerde düşme olmasının nedeni dosyanın o bölgesine farklı veriler gizlense dahi değer çiftlerindeki değişim aynı olmuştur. Bu yüzden dört grafiktede aynı yerde düşme gözlenmiştir.



a) b)

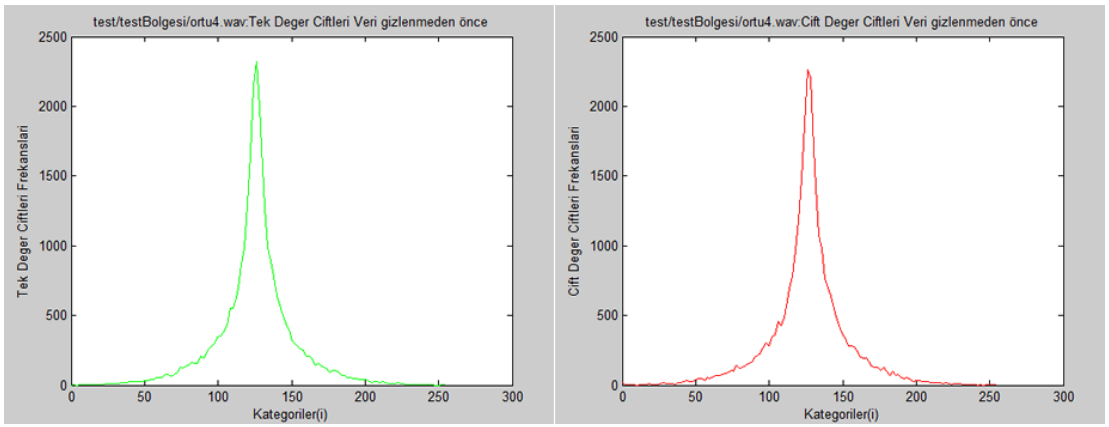
c) d)

Şekil 4.17. Stego4 grubu dosyaları için sıraçma saldırı sonuçları

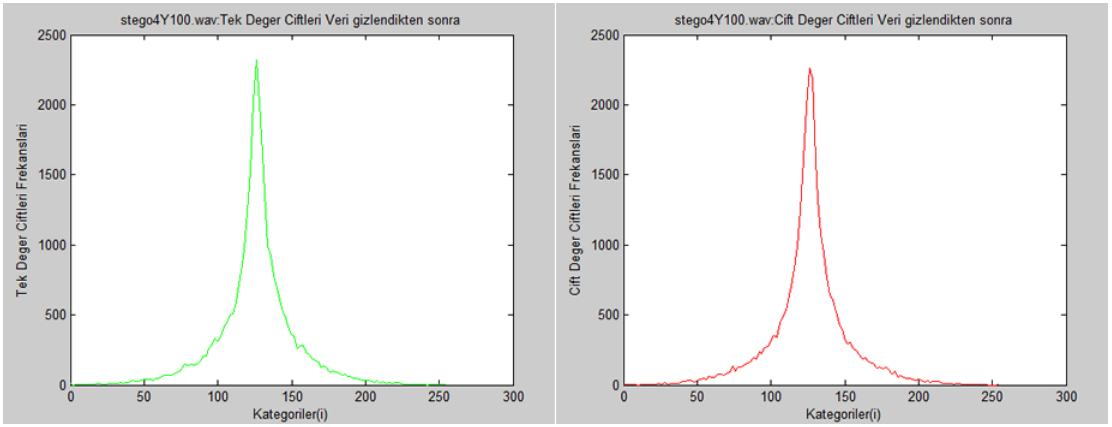
a)- %10'nuna veri gizlenmiş (Stego4Y10) b)- %40'ine veri gizlenmiş (Stego4Y40)

c)- %70'ine veri gizlenmiş (Stego4Y70) d)- %100'üne veri gizlenmiş (Stego4Y100)

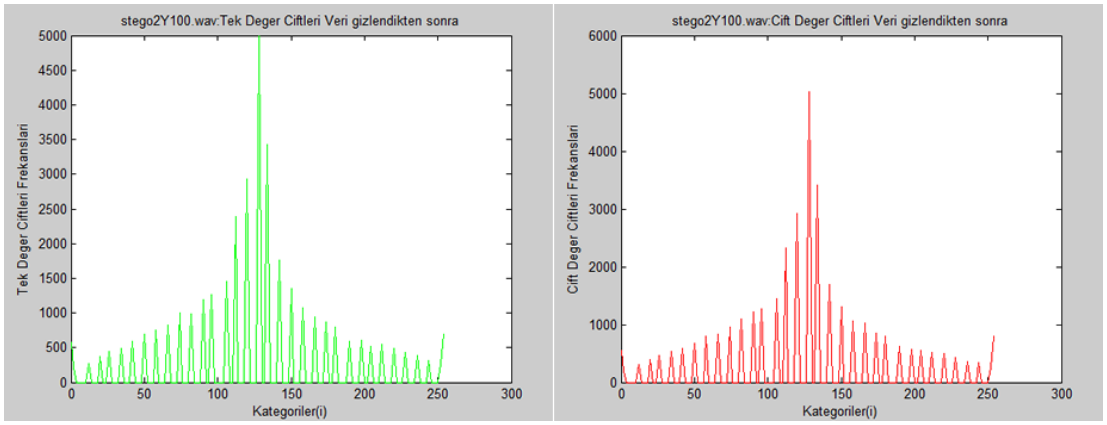
Şekil 4.17'deki grafiklernde görüldüğü gibi Ortu4.wav dosyasından elde edilen sonuçlar beklenenden çok farklı çıkmaktadır. %10 veri gömülmüş dosyanın sıraçmasında oluşan grafik normal değerden çok farklı çıkmaktadır. Keza diğer durumlarda normal değerlerden çok farklı çıktığı gözükmektedir. Bu durumda orijinal dosyanın sıraçmasında dahi çıkan grafik olması gerekenden çok farklı değerler içermektedir. Bu durum incelendiğinde dosya içerisindeki veriler döngüsel değerler içeriyorsa yani orijinal dosyanın ses örnekleri birbirini tekrarlayan değerler içeriyorsa o zaman Ki-kare yönteminin temelini oluşturan değer çiftleri oluşmuyor ve böylece sıraçma sonuçları olması gereken değerlerden çok farklı çıkıyor. Aşağıda Ortu4.wav dosyasına veri gömülmesi sonucu oluşan değer çiftlerinin histogram analizi verilmiştir. Durum bu grafiklerden daha iyi anlaşılacaktır.



a)



b)



c)

Şekil 4.18. Ortu4.wav (A) ve Ortu2.wav (B) dosyalarına veri gizlenmesi

a)- A orijinal dosyasının Tek ve Çift değerlikli değer çiftlerinin analizi

b)- A'ya %100 veri gizlenmiş,Tek ve Çift değerlikli değer çift. analizi

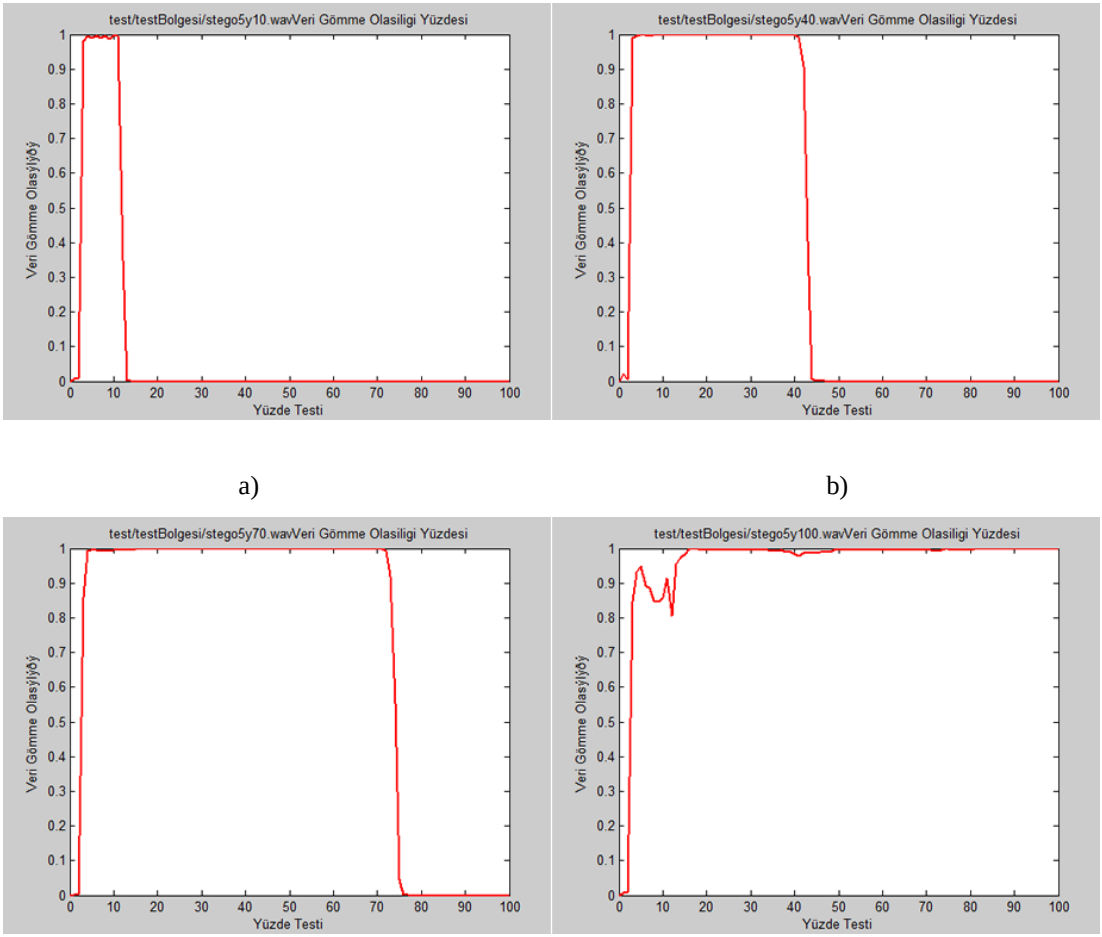
c)- B'ye %100 veri gizlenmiş, Tek ve Çift değerlikli değer çift. analizi

Şekil 4.18'da Ortu4.wav dosyasının orijinal halinin tek ve çift değerlikli değer çiftlerinin histogram analizinde görüldüğü gibi kategorilere göre değer çiftlerinin frekansları artarak gitmekte ve tepe noktasına ulaştıktan sonra aynı şekilde azalmaktadır. Çift değerlikli değer çiftlerinin histogram analizinde de aynı durum söz konusudur. Ortu4.wav dosyasına %100 veri gizlendiğinde dahi değer çiftlerinin frekanslarında herhangi bir değişme olmamıştır. Bunun nedeni yukarıda da bahsedildiği gibi ses dosyasını oluşturan ses örneklerinin değerleri döngüsel bir şekilde birbirini tekrar etmekte ve Şekil 4.18.a'daki tepe değerliğine sahip histogram analizini oluşturmaktadır. Veri gömülmesi sonucunda da değer çiftlerinde bu döngüsel durumdan dolayı bir değişiklik olmamaktadır. Normalde Şekil 4.18.c'de gösterilen Ortu2.wav dosyasına %100 oranında veri gömülmesi sonucu oluşan tek ve çift değerlikli değer çiftlerinin histogram analizinde görüldüğü gibi değer çiftlerinin frekansları belirli aralıklarla artıp azalmaktadır. Bu dosyada herhangi bir döngüsel durum söz konusu değildir. Yani değerler birbirini tekrarlayacak şekilde döngüsel bir yapıda kendisini tekrar etmemektedir. Şekil 4.18.c'de değer çiftlerinin frekansları aralıklarla yükselerek her adımda bir sıfıra düşmektedir. Oysa şekil 4.18.a ve Şekil 4.18.b'de görüldüğü gibi değer çiftlerinin frekansları gittikçe artmakta ve tepe değerine ulaştıktan sonra azalmaktadır. Oluşan histogram analizindeki farklılık da dosyaların yapısal olarak birbirinden ne kadar farklı olduğunu göstermiştir. Ortu4.wav dosyası bir konuşma sesidir ve yapısı incelendiğinde bu duruma sebeb olan şey aslında aynı tonda birbirine yakın seslerde hatta aynı tonun arka arkaya tekrarlanmasından dolayı olduğu anlaşılmaktadır. Örnek olarak birkaç kelime aynı sırada arka arkaya söylenerek ses kaydı yapıldığında da benzer sonuçlar elde edilmektedir.

Bu da yöntemin istisnai bir durumudur. Ki-kare testi bu durumlarda doğru sonuçlar elde edemez. Stego4 gurubu için PNN ağının sonuçları Tablo 4.4'de verilmiştir.

Tablo 4.4. Stego4 gurubu dosyaları için PNN ağı olasılık sonuçları

	Yüzdesel Oranlar (%)			
Gizlenen	10	40	70	100
Bulunan	85	94	94	100



a) b)

c) d)

Şekil 4.19. Stego5 grubu dosyaları için sıraçma saldırı sonuçları

a)- %10'nuna veri gizlenmiş (Stego5Y10) b)- %40'ine veri gizlenmiş (Stego5Y40)

c)- %70'ine veri gizlenmiş (Stego5Y70) d)- %100'üne veri gizlenmiş (Stego5Y100)

Şekil 4.19'de görüldüğü gibi Ortu5.wav dosyasından olumlu sonuçlar elde edilmiştir. Ortu5.wav dosyasına %10 veri gömülü dosyanın sıraçmasında x ekseni tam 11'de iken ani bir düşme gözlemiş 13'den sonra olasılık değeri sıfır seviyelerindedir.

Tablo 4.5. Stego5 gurubu dosyaları için PNN ağı olasılık sonuçları

	Yüzdesel Oranlar (%)			
Gizlenen	10	40	70	100
Bulunan	10	40	71	100

Yapılan testlerden çıkan sounçlar görüldüğü gibi Ki-kare yöntemi bazı istisnai durumlar hariç başarılı sonuçlar vermektedir. Bu durumlar haricinde ses dosyasında gizli veri mevcudiyeti ve miktarı sorularına yanıt alınabilir.
Aşağıda ses dosyalarına %0, %10, %40, %70 ve %100 veri gömülmesi sonucu PNN yapay sinir ağının vermiş olduğu cevaplar verilmiştir.

Tablo 4.6. Ki-kare saldırısı ile PNN ağının bulduğu olasılık sonuçları

Dosyalar		Veri Gizleme Oranları (%)				
		0	10	40	70	100
Ortu1.wav	PNN Ağı Bulma Oranları (%)	0	11	44	78	100
Ortu2.wav		0	10	41	71	100
Ortu3.wav		0	10	49	78	100
Ortu4.wav		45	85	94	94	100
Ortu5.wav		0	10	40	71	100

Tablo 4.6'daki Ki-kare testi ses dosyalarındaki gizli verileri tesbit edebilir fakat daha önce de belirtildiği gibi istisnai durumlar içeren dosyalar için doğru sonuçlar üretemeyebilir. Ayrıca gürültülü ses dosyaları içinde gürültülü olan kısımları için olumlu sonuç döndürebilir fakat çalışılırlığı test edilmiş bir saldırı yöntemidir. Örneğin Ortu3.wav dosyası tam net sesler içeren bir dosya değildir. Seste algılanabilir gürültüler vardır fakat Ki-kare bu dosyayı düzgün bir şekilde analiz edebilmiştir. Sadece yukarıdaki kısımlarda bahsettiğimiz istisnai durum içeren Ortu4.wav dosyası için sonuçlar düzgün çıkmamıştır. Bu durumda Ki-kare saldırısının çalışma mantığına uymayan durumlar oluşması nedeniyle düzgün analiz sonuçlar elde edilememiştir. Tablo 4.6'daki %100 oranlarına bakıldığında tam doğru sonuçlar elde edildiği görülmektedir. Bunun nedeni %100 oranında veri gömülmesi sonucu Şekil 4.19.d' de görüldüğü gibi olasılık değeri x ekseni 15'e kadar %90'lı değerlerde 15'ten sonra ise %99 - %100 arasında değişmektedir. Bu diğer tüm örneklerde aynıdır. Bu da PNN ağının tam doğru cevap vermesini sağlar.

Yukarıdaki sonuçlar doğrultusunda bir hata analizi yapacak olursak aşağıda kullanılan formül ile test edilmiş ve sonuçlar Tablo 4.7 'de verilmiştir.

$$\text{Tahmin Edilen Yüzde} <> 0 \text{ ise,} \quad \text{Hata Yüzdesi 1} = \left|100 - \frac{\text{Gerçek Yüzde x 100}}{\text{Tahmin Edilen Yüzde}}\right|$$

Tablo 4.7. Ki-kare saldırısı ile PNN ağının bulduğu olasılık sonuçları hata analizi 1

Dosyalar		Veri Gizleme Oranları (%)				
		0	10	40	70	100
Ortu1.wav	Hata Analizi 1 Oranları (%)	0	9	9,09	10,2	0
Ortu2.wav		0	0	2,4	1,4	0
Ortu3.wav		0	0	18,3	10,2	0
Ortu4.wav		100	88,2	57,4	25,5	0
Ortu5.wav		0	0	0	1,4	0

Tablo 4.7'deki hata analiz tablosundan çıkan sonuçlara göre istisnai durum haricindeki tüm oranlarda iyi düzeyde başarı elde edilmiştir. %100 oranında veri gömüldüğünde PNN ağı %0 hatayla sonucu bulmuştur. En az başarı %40 veri gömülmesi sonucu bulunan sonuçlarda olmuştur.

Sonuçlar daha önce yapılmış benzer çalışma [63] ile kıyaslayabilmek için ortak hata analiz formülü kullanılmıştır. Dosyalara veri gizleme yüzdeleri her iki çalışmada da aynıdır. Aşağıdaki hata analiz formülüne göre PNN yapay sinir ağı sonuçlarımız verilmiştir.

$$\text{Hata Yüzdesi 2} = |\text{ Tahmin Edilen Yüzde} - \text{Gerçek Yüzde }|$$

Tablo 4.8. Ki-kare saldırısı ile PNN ağının bulduğu olasılık sonuçları hata analizi 2

Dosyalar		Veri Gizleme Oranları (%)				
		0	10	40	70	100
Ortu1.wav	Hata Analizi 2 Oranları (%)	0	1	4	8	0
Ortu2.wav		0	0	1	1	0
Ortu3.wav		0	0	9	8	0
Ortu4.wav		45	75	54	24	0
Ortu5.wav		0	0	0	1	0

Tablo 4.8.'den de anlaşılacağı üzere sadece istisnai durumu olan Ortu4.wav dosyası için sonuçlar hatalı çıkmış diğer durumlarda sonuçlar oldukça başarılıdır. Bu başarıda kuşkusuz PNN yapay sinir ağının rolü de vardır. Yapılan eğitim sayesinde oldukça doğru sonuçlar elde edilmektedir.

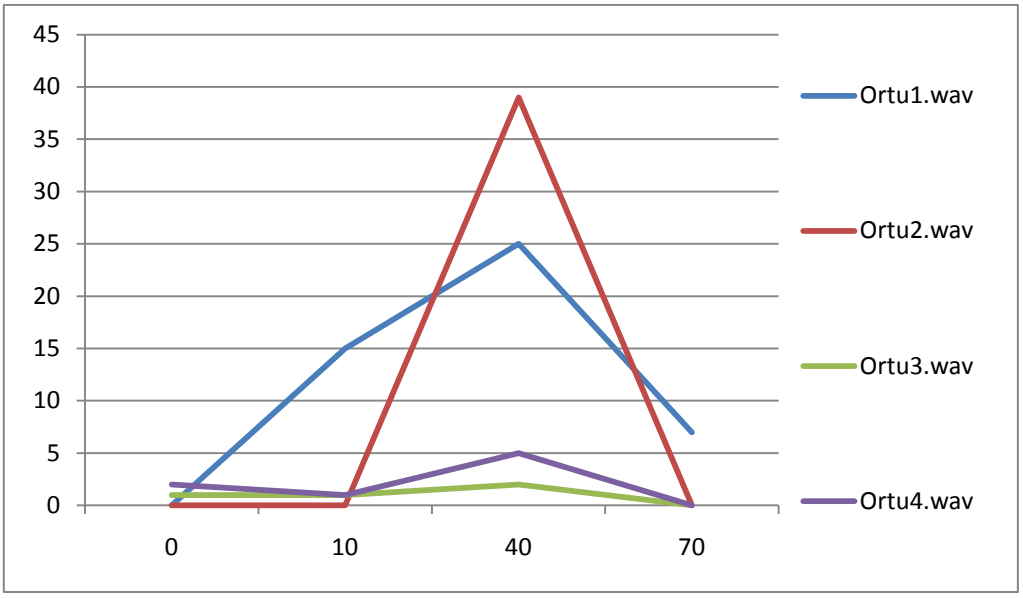
Yukarıdaki hata analizi2 sonuçlarına göre benzer çalışma [63]'nın sonuçlarının hata analizi sonuçları kıyaslanmıştır. Hata analiz formülleri aynıdır. Karşılaştırma için gerekli olan %50 oranında veri gizleme kıstası yukarıdaki anlatılan sonuçlarda olmadığı için aynı dosyalara %50 oranında veri gizlenmiş ve PNN yapay sinir ağında test edilerek sonuçları kıyaslanmıştır.

Tablo 4.9. Benzer çalışma[63] (A) ile PNN ağı (B) hata analizi 2 sonuçlarının karşılaştırılması

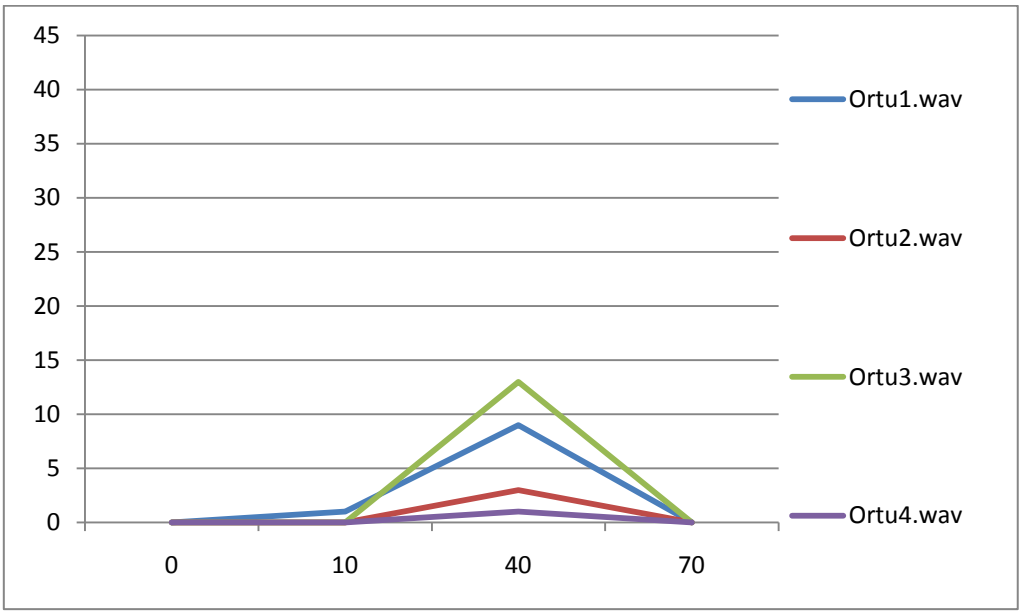
Dosyalar		Veri Gizleme Oranları (%)			
		0	10	40	70
Ortu1.wav	Hata Analizi 2 Oranları A / B (%)	0 / 0	15 / 1	25 / 9	7 / 0
Ortu2.wav		0 / 0	0 / 0	39 / 3	0 / 0
Ortu3.wav		1 / 0	1 / 0	2 / 13	0 / 0
Ortu4.wav		2 / 0	1 / 0	5 / 1	0 / 0

Tablo 4.9'daki sonuçlar incelendiğinde PNN yapay sinir ağının daha başarılı olduğu görülmektedir.

Ki-kare sıraçma saldırısı dosya yapısından doğrudan etkilenmekte ve bazı dosyalar da çok iyi sonuçlar verirken bazılarında ise yanlış sonuçlar verebilmektedir.



Şekil 4.20. Benzer çalışma [63] hata analizi 2 sonuçları



Şekil 4.21. PNN ağı (B) hata analizi 2 sonuçları

Şekil 4.20 ve Şekil 4.21'de verilen grafiklerde de görüldüğü gibi Pnn ağının hata analizi sonuçları benzer çalışmaya [63] göre daha iyi olduğu görülmektedir.

BÖLÜM 5. SONUÇLAR VE TARTIŞMA

Tez çalışmasında ses dosyalarına hem sırörtme yöntemiyle veri gizleme hem de sır açma teknikleri ile gizli veriyi bulma işlemlerini yapan bir uygulama yapılmıştır. Tez çalışması sırasında pek çok güçlükle karşılaşılmış, konunun yeni olmasından dolayı yeterli çalışma bulunamamıştır. Ülkemizde de sıraçma konusu yeni yeni araştırılmaya başlanmıştır. Özellikle konu hakkında Türkçe tez ve makale, birkaç örnek ile sınırlıdır. Literatürdeki çalışmalar daha çok resim dosyaları üzerinedir. Fakat bu ve bunun gibi tezlerin bundan sonra yapılacak olan tezlere örnek olacağı düşünülmektedir.

Geliştirilen uygulamada Ki-kare tekniği ile sıraçma yöntemi kullanılmaktadır. PNN yapay sinir ağı ile de Ki-kare testinin sonuçları iyileştirilmiştir. PNN yapay sinir ağı, Ki-karenin yaptığı ufak hataları tolere etmiş ve sırlı dosyaya ne kadar veri gizlendiğini sayısal olarak belirtilmiştir. PNN yapay sinir ağının başarısı kesinlikle doğrudan sıraçma tekniğinin başarısı ile ilgilidir ama daha öncede örneklenen durumlardan yola çıkarak analizde oluşan küçük hataları tolere ederek daha doğru sonuçlar elde etmiştir.

Sıraçma çalışmamızda kullanılan Ki-kare testi geliştirilebilir. Her zaman doğru sonuçlar üretmemektedir. Özellikle tekrarlı sesler içeren konuşma sesleri gibi durumlarda yanlış sonuçlar üretmektedir. Yine gürültülü ses dosyalarında da gizli verinin varlığına ilişkin sonuçlar verebilir. Değer çiftleri yöntemi her zaman geçerli bir yöntem değildir. Verilerin rastgele düzende gizlenmesi durumunda da Ki-kare doğru sonuçlar vermeyecektir.

Yapılan tez çalışmasında geliştirilebilir hususlar; PNN yapay sinir ağını seçilen dosyaların sayılarını arttırarak daha fazla veri ile eğitime tabi tutarak sonuçların karşılaştırılması ile ilgili bir çalışma yapılabilir. Bunun yanında PNN yapay sinir ağının giriş elemanlarındaki her 100 değerin sıralarının değişmesi PNN yapay sinir

ağının vereceği cevabı değiştirmemektedir. Bu durumdan faydanılarak Ki-kare yönteminin rastgele gömülmüş verileri analiz edememesi probleminin çözümlerini içeren bir çalışma yapılabilir.

KAYNAKLAR

[1] MURRAY, A.H., Burchfiled, R.W (eds.), "The Oxford English Dictionary: Being Corrected Re-issue", Oxford, England: Clarendon Press, (1933).

[2] CUMMINS, J., DISKIN, P., LAU, S., PARLETT, R., " Steganography and Digital Watermarking" (2004). http://www.cs.bham.ac.uk/~mdr/teaching /modules03/security/students/SS5/Steganography.pdf (Erişim:30.04.2010).

[3] OZDEMIR C., OZCERIT A., "A new steganography algorithm based on color histograms for data embedding into raw video streams", Sakarya University, Turkey, Journal of Elsevier,Computers & Security 28, 670 - 682 (2009).

[4] AMIN, M., SALLEH, M., IBRAHIM, S.,KATMIN, M.R., SHAMSUDDIN, M.Z.I., "Information hiding using steganography" Telecommunication Technology NCTT Proceedings.4th National Conference, Shah Alam, MALAYSIA(2003).

[5] KERCKHOFFS, A., Cryptographie Militaire, (1883), History of Steganografik and Cryptography, http://www.petitcolas.net /fabien/steganography/history.html, (Erişim: 02.04.2010).

[6] BRIQUET, C.M., Les Filigranes, Geneva , (1907), History of Steganografik and Cryptography, http://www.petitcolas.net/fabien/steganography /history.html, (Erişim: 02.04.2010).

[7] ADLI, A.; NAKAO, Z., "Three steganography algorithms for MIDI files", Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference , Guangzhou, China, 4 : 2401- 2404 (2005).

[8] XU, C., PING, X., ZHANG, T., Steganography in Compressed Video Stream, Proceedings of the First International Conference on Innovative Computing, IEEE (2008).

[9] KRATZER,C.,DITTMANN,J.,VOGEL,T.,HILLERT,R., Design and Evaluation of Steganography for Voice-over-IP, Advanced Multimedia and S. Lab. (AMSL) Otto-von-Guericke-Universitat, Magdeburg, Germany, (2006).

[10] CHANG, L., MOSKOWITZ, I., Critical Analysis of Security in Voice Hiding Techniques, Information Technology Division, MaLl Code 5540, Center for

High Assurance Computer Systems, Naval Research Laboratory, Washington, DC 20375 USA (2006).

[11] SAJATOVIC, M., PRINZ, J., KROEPJ, A., Increasıng The Safety Of The Atc Voıce Communıcatıons By Usıng In-Band Messagıng, Frequents Nachrich tenfechnik GmbH, IEEE, Vienna, Austria (2003).

[12] MATSUI, K., TANAKA, K., NAKAMURA, Y. , Digital Signature on Facsmile Document by Recursive MH Coding, International Symposium on Cryptography and Information Security (CIS89) (1989).

[13] HARTUNG, F., KUTTER, M., Multimedya Watermarking Techniques, Proceedings of the IEEE, Vol.87, No.7, pp 1079–1107 (1999).

[14] DELAIGLE, J. K., Protection of Intellectual Property of Images by Perceptual Watermarking, Doktora Tezi, Universitè Catholique de Louvain (2000).

[15] SAGIROGLU, S., TUNCKANAT, M., "A Secure Internet Communication Tool", Turkish Journal of Telecommunications, 1(1):40-46 (2002).

[16] CHENG, J., KOT, A.C., LIUAND, J., CAO, H., Steganalysis of Data Hiding in Binary Text Images, Proceedings of the IEEE, pp 4405–4408 (2005).

[17] FRIDRICH, J., DU, R., LONG, M., "Steganalysis of LSB encoding in color images," Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference , New York, USA , 3:1279-1282 (2000).

[18] FARID, H., "Detecting steganographic message in digital images". Technical Report TR2001-412, Dartmouth College, 1-9 (2001).

[19] Provos, N., Honeyman, P., "Detecting Steganographic content on the internet", Tech. Rep. CITI 01-1a, University of Michigan, 1-14 (2001).

[20] OZER H,. SANKUR B., "Ses İşaretleri için Algısal Kıyım Fonksiyonu", 12. Sinyal İşleme ve Uygulamaları Kurultayı, Kuşadası, Nisan (2004).

[21] FRIDRICH, J., "Minimizing the embedding impact in steganography", Proceeding of the 8th Workshop on Multimedia and Security, Geneva-Switzerland, 2-10 (2006).

[22] OZER, H., SANKUR, B., MEMON, N., AVCIBAS, Đ., "Detection Of Audio Covert Channels Using Statistical Footprints Of Hidden Messages.", Digital Signal Processing 16 (4): 389-401 (2006).

[23] AVCIBAS, I., "Audio Steganalysis With Content Independent Distortion Measures," IEEE Signal Processing Letters, 13(2): 92-95 (2006).

[24] RU et al., “Audio Steganalysis Based On ‘Negative Resonance Phenomenon’ Caused By Steganographic Tools.”, Journal of Zhejiang University Science A, 7(4):577-583 (2006).

[25] QINGZHONG Liu et al. "Detecting Information-Hiding in WAV Audios", New Mexico Tech (2006).

[26] QINGZHONG Liu et al. "Spectrum Steganalysis of WAV Audio Streams", New Mexico Tech, Lecture Notes in Computer Science (2009).

[27] Veri Gizle http://www.verigizle.com (Erişim: 06.03.2010).

[28] LAU, S., “An Analysis of Terrorist Groups’ Potential Use of Electronic Steganografy”, SANS Security Essentials GSEC Practical Assignment Version 1,3-18 (2003).

[29] CANBEK, G., SAGIROGLU, Ş., “Bilgi ve Bilgisayar Güvenliği Casus Yazılımlar ve Korunma Yöntemleri”, Grafiker, Ankara, 1-50 (2006).

[30] “Steganografi ve LSB” http://www.bilgisayarkavramlari.com/2009/06/05/steganografi-ve-lsb/ (Erişim: 06.03.2010).

[31] JAMIL, T., “Steganography: the art of hiding information in plain sight”, Potentials, IEEE, 18 (1): 10-12 (1999).

[32] http://www.marie-stuart.co.uk/ (Erişim: 07.03.2010).

[33] TRITHEMIUS, J., “Steganographia:hoe est ars per occultam scripturam animi sui voluntatem absentibus aperiendi certa”, http://www.esotericarchives.com/tritheim/stegano.htm (Erişim: 08.03.2010).

[34] KOLATA, G., “A Mystery Unraveled, Twice”, http://cryptome.unicast.org/cryptome022401/tri.crack.htm (1998). (Erişim: 08.03.2010).

[35] BENDER, W., GRUHL, D., MORIMOTO, N., and LU, A. “Techniques for data hiding”. IBM Syst. J., 35(3&4):313-336 (1996).

[36] LEE, Y.K., CHEN, L.H., "High capacity image steganographic model" Vision,Image and Signal Proc., IEE Proceedings-,147(3):288-294 (2000).

[37] NODA, H., SPAULDING, J., SHIRAZI, M.N., KAWAGUCHI, E., "Application of bitplane decomposition steganography to JPEG2000 encoded images" Signal Processing Letters, IEEE 9(12):410-413 (2002).

[38] TSENG, H.W., CHANG, C.C., "Steganography using JPEG-compressed images" Computer and Information Technology, CIT '04. The Fourth International Conference , Wuhan, China ,12- 17 (2004).

[39] BRISBANE, G., SAFAVI-NAINI, R., Ogunbona, P., "High-capacity steganography using a shared colour palette," Vision, Image and Signal Processing, IEE Proceedings, 152: 787- 792 (2005).

[40] NIIMI M., NODA, H., KAWAGUCHI, E., EASON, R.O., "High capacity and secure digital steganography to palette-based images," Image Processing Proceedings 2002 International Conference, Rochester, New York, USA, 2: 917-920 (2002).

[41] SHAHREZA, S., "Stealth steganography in SMS" Wireless and Optical Communications Networks, IFIP International Conference, Bangalore (2006).

[42] CVEJIC, N., SEPPANEN, T., "Increasing the capacity of LSB-based audio steganography," Multimedia Signal Processing, 2002 IEEE Workshop , St. Thomas, Virgin Islands, USA, 336- 338 (2002).

[43] GOPALAN, K., "Audio steganography using bit modification", 2003 InternationalConference on Multimedia and Expo, Baltimore, Maryland, 629-632 (2003).

[44] Spammimic - hide a message in spam, https://www.spammimic.com (Erişim: 05.03.2010).

[45] KESSLER, G.C., "An Overview of Steganography for the Computer Forensics Examiner", Forensic Science Communications , 6(3):1-29 (2004).

[46] SAGIROGLU, S., TUNCKANAT, M., 2002, Güvenli Internet Haberleşmesi için Bir Yazılım: TurkSteg, Olympos Security, http://www.teknoturk.org/docking/yazilar/tt000106-yazi.htm (Erişim: 04.03.2010).

[47] CHANDRAMOULI, R., MEMON, N.,"Analysis of LSB Based Image Steganography Techniques", Proceedings of the International Conference on Image Processing, Thessalonica, Ekim, Yunanistan, 1019-1022 (2001).

[48] CHOU, J., RAMCHANDRAN, K., ORTEGA, A., "High capacity audio data hiding for noisy channels," Information Technology: Coding and Computing, Proceedings. International Conference, Las Vegas,108-112 (2001).

[49] Microsoft WAVE soundfile format https://ccrma.stanford.edu/courses/422/projects/WaveFormat (Erişim: 04.03.2010) .

[50] SUI, X.G., LUO, H., "A new steganography method based on hypertext", Radio Science Conference, 2004. Proceedings. 2004 Asia-Pacific, 18-184, 24-27 (2004).

[51] EL-KHALIL, R., KEROMYTIS, A.D., "Hydan: Hiding Information in Program Binaries" http://www1.cs.columbia.edu/~angelos/Papers/hydan.pdf (Erişim: 01.03.2010).

[52] WESTFELD, A., PFITZMANN, A., “Attacks on Steganographic Systems”, Proceedings of the Third International Workshop Information Hiding, Dresden, Germany, 61-76 (2000).

[53] FRIDRICH, J., GOLJAN, M., "Practical steganalysis of digilal images state of the art", Proc. SPIE Photonics West, 4675: 1-13 (2002).

[54] http://tr.wikipedia.org/wiki/Gama_fonksiyonu (Erişim: 09.03.2010).

[55] PROVOS, N., “Defending Against Statistical Steganalysis”, 10th USENIX Securit Symposium, Washington, 323-335 (2001).

[56] SAHIN A. “Görüntü steganografide bulunan yeni metodlar ve bu metodların güvenirlilikleri”, Doktora Tezi, Trakya Üniversitesi (2007).

[57] FARID, H., "Detecting hidden messages using higher-order statistical models", International Conference on Image Processing. http://www.ists.dartmouth.edu/library/36.pdf (Erişim: 06.03.2010).

[58] FARID, H. "Detecting Steganographic Messages in Digital images." Technical Report TR2001-412, Dartmouth College, 1-9 (2001).

[59] JOHNSON, M., LYU,S., FARID, H., “Steganalysis in recorded speech", Security,Steganography, and Watermarking of Multimedia Contents VII, San Jose, CA, US, 664-672 (2005).

[60] HETZL, S., "Steghide," http://steghide.sourceforge.net (Erişim: 07.03.2010).

[61] http://members.fortunecity.com/curlysdisneymagic/disneywav4.html (Erişim: 09.03.2010).

[62] http://simplythebest.net/sounds/WAV (Erişim: 09.03.2010).

[63] HASSAN M. “Steganaliz Yaklaşımların Karşılaştırılması”, Yüksek Lisans Tezi, Gazi Üniversitesi (2008).

[64] WATKINS J., “Steganography - Messages Hidden in Bits”, Multimedia Systems Coursework, Department of Electronics and Computer Science, University of Southampton, 8-10 (2001).

[65] AKBAL T., OZCERIT A., “Ses Verilerine Sıkıştırılmış ve Şifrelenmiş Ham Verilerin Gömülmesi”, Yüksek Lisans Tezi, Sakarya Üniversitesi (2008).

[66] http://www.petitcolas.net/fabien/steganography/mp3stego
(Erişim: 08.03.2010).

[67] ÖZDEMIR C., OZCERIT A., “Hareketli Görüntü Uygulamaları İçin Sırörtme Yaklaşımı ile Veri Gömme Algoritması Tasarımı”, Doktora Lisans Tezi, Sakarya Üniversitesi (2008).

EKLER

EK-A Veri Gizleme Uygulaması Matlab Kodları

```
% Veri Gizleme Algoritması 25.04.2010 18:21 Pazar BAUM %
 % giriş ses dosyasını kullanıcıdan al (cover file)
calismaYeri='test/testBolgesi/';
sonuc=false;
while sonuc == false
    ortu_dosya_adi = lower(input('Örtü ses dosya adını uzantısıyla giriniz(Örn.
                                        ses.wav):','s'));
    uzunluk=size(ortu_dosya_adi);
    bulunanIndex=strfind(ortu_dosya_adi,'.wav');
     % strfind fonksiyonu .wav ararken,
     %  sonucu “.” noktanın bulunduğu pozisyonun değerini döndürür
    if (bulunanIndex == uzunluk(2)-3) % size fonksiyonundan 1X2 lik matris döndüğü
    % için uzunluk(2) kullanıldı , -3 'wav' kelimesinin uzunluğu çıkarıldı
       sonuc=true;
    else
       sonuc=false;
       disp('Dosya girişi hatalı!');
    end
end
ortu_dosya_adi=strcat(calismaYeri,ortu_dosya_adi);
% örtü verisini oku
[ortu_verisi,cerceveHizi,kacBit]  = wavread(ortu_dosya_adi);

% çıkış ses dosyasının adını kullanıcıdan al (sırlı dosya adı)
stego_dosya_adi = input('Stego ses dosya adını uzantısız giriniz(Örn. StegoSes):','s');
bulunanIndex=strfind(stego_dosya_adi,'.wav');
if (isempty(bulunanIndex)==true)
   stego_dosya_adi = strcat(stego_dosya_adi,'.wav');  % dosya uzantısını ekle
end
ortu_verisi=round(2^(kacBit-1).*(ortu_verisi))+2^(kacBit-1);  % verileri çok küçük 10
%üzeri -6 civarlarında ondalıl sayılardır. Onları tam sayı yapmak için kaçbitlikse o
%kadar çarparak 0 ve 1 %seviyesine getiriyoruz Sonra kaç bitlik ise okadar sayı
%ekliyoruz son olarak 16 %bitlikse 0 -> 32767 1 -> 32768 oluyor. çünkü bunları
%dizide kullanacağız içindeki %veri 32768 olursa dizi indeksleri çalışmıyor...
[satir,sutun]=size(ortu_verisi);
```

```
TekOrtu = zeros(2^(kacBit-1),1); TekStego=zeros(2^(kacBit-1),1);
CiftOrtu= zeros(2^(kacBit-1),1); CiftStego=zeros(2^(kacBit-1),1);
for i=1:satir
   for j=1:sutun
      if rem(ortu_verisi(i,j),2)==0
         CiftOrtu((ortu_verisi(i,j)/2)+1)=CiftOrtu((ortu_verisi(i,j)/2)+1)+1;
      else
         TekOrtu(((ortu_verisi(i,j)-1)/2)+1)=TekOrtu(((ortu_verisi(i,j)-1)/2)+1)+1;
      end
   end
end

 % Örtü Verisi Histogramı Çiziliyor...
kategoriler= zeros(2^(kacBit-1),1);
for i=1:(2^(kacBit-1))
   kategoriler(i)=2*(i-1);
end
Fark=abs(TekOrtu-CiftOrtu);
baslik=strcat(ortu_dosya_adi,':','|Tek-Cift| Veri gizlenmeden önce');
figure(1),plot(kategoriler,Fark,'LineWidth',1,'Color','green');title(baslik);
xlabel('Kategoriler(i)'); ylabel('|Tek-Cift|');
sonuc=false;
while sonuc==false
        yuzdeKac=input('Ses dosyasının yüzde kaçına mesaj gizlensin ? (1-100): ');
        if (yuzdeKac>0 && yuzdeKac<=100)  sonuc=true;
        else sonuc=false;
        end
end
% Gizli Metin oluşturuluyor...
kacSatir=floor((floor((yuzdeKac/100)*satir*sutun))/sutun;
gizliMesaj=rem(floor(rand(satir,sutun)*1000),2);
stego_verisi=floor(ortu_verisi/2)*2+gizliMesaj;
if kacSatir<satir  % yüzde 100 den kücük ise yuzdeKac o zaman diger geriye kalanı
%örtü verisinden tamamla

stego_verisi=cat(1,stego_verisi(1:kacSatir,1:sutun),ortu_verisi(kacSatir+1:satir,…
                        1:sutun));
end

for i=1:satir
   for j=1:sutun
      if rem(stego_verisi(i,j),2)==0
         CiftStego((stego_verisi(i,j)/2)+1)=CiftStego((stego_verisi(i,j)/2)+1)+1;
      else
         TekStego(((stego_verisi(i,j)-1)/2)+1)=TekStego(((stego_verisi(i,j)-1)/2)+1)+1;
      end
```

```
   end
end

 % Sırlı Veri Histogramı Çiziliyor...
Fark=abs(TekStego-CiftStego);
baslik=strcat(stego_dosya_adi,':','|Tek-Cift| Veri gizlendikten sonra');
figure(2),plot(kategoriler,Fark,'LineWidth',1,'Color','red');
title(baslik); xlabel('Kategoriler(i)'); ylabel('|Tek-Cift|');
stego_verisi = stego_verisi - (2^(kacBit-1));  % carpma bolme işlemlerini geri al
% gercek veriye dön
stego_verisi = stego_verisi ./ (2^(kacBit-1));

% stereo ses dosyası ise stereo olarak kaydet.
stego_dosya_adi=strcat(calismaYeri,stego_dosya_adi);
wavwrite(stego_verisi,cerceveHizi,kacBit,stego_dosya_adi);
```

EK-B Veri Gizleme Uygulaması Versiyon 2 Matlab Kodları

Bu kod, veri gömülmeden önceki ve sonraki değer çiftlerinin Tek değerlikli ve Çift olarak değerlikli olarak ayrı ayrı historam analizini çıkarmaktadır. Kod açısından diğerinden farkı yoktur. Ek-A'daki kod, bu değer çiftlerinin farklarının histogram analizini çıkarmaktadır. Sadece aradaki farkı histogram analizi farklılığıdır.

```
% Veri Gizleme Algoritması 30.04.2010 16:14 Cuma BAUM %
 % giriş ses dosyasını kullanıcıdan al (cover file)
calismaYeri='test/testBolgesi/';
sonuc=false;
while sonuc == false
    ortu_dosya_adi = lower(input('Örtü ses dosya adını uzantısıyla giriniz(Örn.
                                        ses.wav):','s'));
    uzunluk=size(ortu_dosya_adi);
    bulunanIndex=strfind(ortu_dosya_adi,'.wav');
    if (bulunanIndex == uzunluk(2)-3)
       sonuc=true;
    else
       sonuc=false;
       disp('Dosya girişi hatalı!');
    end
end
ortu_dosya_adi=strcat(calismaYeri,ortu_dosya_adi);
% örtü verisini oku
[ortu_verisi,cerceveHizi,kacBit]  = wavread(ortu_dosya_adi);

% çıkış ses dosyasının adını kullanıcıdan al (stego file)
stego_dosya_adi = input('Stego ses dosya adını uzantısız giriniz(Örn. SırlıSes):','s');
bulunanIndex=strfind(stego_dosya_adi,'.wav');
if (isempty(bulunanIndex)==true)
   stego_dosya_adi = strcat(stego_dosya_adi,'.wav');  % dosya uzantısını ekle
end

ortu_verisi=round(2^(kacBit-1).*(ortu_verisi))+2^(kacBit-1
[satir,sutun]=size(ortu_verisi);
TekOrtu = zeros(2^(kacBit-1),1); TekStego=zeros(2^(kacBit-1),1);
CiftOrtu= zeros(2^(kacBit-1),1); CiftStego=zeros(2^(kacBit-1),1);
for i=1:satir
```

```
   for j=1:sutun
      if rem(ortu_verisi(i,j),2)==0
         CiftOrtu((ortu_verisi(i,j)/2)+1)=CiftOrtu((ortu_verisi(i,j)/2)+1)+1;
      else
         TekOrtu(((ortu_verisi(i,j)-1)/2)+1)=TekOrtu(((ortu_verisi(i,j)-1)/2)+1)+1;
      end
   end
end

 % Örtü Verisi Histogramı Çiziliyor...
kategoriler= zeros(2^(kacBit-1),1);
for i=1:(2^(kacBit-1))
   kategoriler(i)=2*(i-1);
end
baslik=strcat(ortu_dosya_adi,':','Tek Deger Ciftleri Veri gizlenmeden önce');
figure(1),plot(kategoriler,TekOrtu,'LineWidth',1,'Color','green');title(baslik);
xlabel('Kategoriler(i)'); ylabel('Tek Deger Ciftleri Frekanslari');

baslik=strcat(ortu_dosya_adi,':','Cift Deger Ciftleri Veri gizlenmeden önce');
figure(2),plot(kategoriler,CiftOrtu,'LineWidth',1,'Color','red');title(baslik);
xlabel('Kategoriler(i)'); ylabel('Cift Deger Ciftleri Frekanslari');

sonuc=false;
while sonuc==false
        yuzdeKac=input('Ses dosyasının yüzde kaçına mesaj gizlensin ? (1-100): ');
        if (yuzdeKac>0 && yuzdeKac<=100)  sonuc=true;
        else sonuc=false;
        end
end
% Gizli Metin oluşturuluyor...
kacSatir=floor((floor((yuzdeKac/100)*satir*sutun))/sutun);
gizliMesaj=rem(floor(rand(satir,sutun)*1000),2);
stego_verisi=floor(ortu_verisi/2)*2+gizliMesaj;
if kacSatir<satir
    stego_verisi=cat(1,stego_verisi(1:kacSatir,1:sutun),ortu_verisi(kacSatir+1:satir,…
                            1:sutun));
end

for i=1:satir
   for j=1:sutun
      if rem(stego_verisi(i,j),2)==0
         CiftStego((stego_verisi(i,j)/2)+1)=CiftStego((stego_verisi(i,j)/2)+1)+1;
      else
         TekStego(((stego_verisi(i,j)-1)/2)+1)=TekStego(((stego_verisi(i,j)-1)/2)+1)+1;
      end
   end
```

```
end

 % Sırlı Veri Histogramı Çiziliyor...
baslik=strcat(stego_dosya_adi,':','Tek Deger Ciftleri Veri gizlendikten sonra');
figure(3),plot(kategoriler,TekStego,'LineWidth',1,'Color','green');title(baslik);
xlabel('Kategoriler(i)'); ylabel('Tek Deger Ciftleri Frekanslari');

baslik=strcat(stego_dosya_adi,':','Cift Deger Ciftleri Veri gizlendikten sonra');
figure(4),plot(kategoriler,CiftStego,'LineWidth',1,'Color','red');title(baslik);
xlabel('Kategoriler(i)'); ylabel('Cift Deger Ciftleri Frekanslari');
stego_verisi = stego_verisi - (2^(kacBit-1));
stego_verisi = stego_verisi ./ (2^(kacBit-1));

% stego dosyasını çıkart
stego_dosya_adi=strcat(calismaYeri,stego_dosya_adi);
wavwrite(stego_verisi,cerceveHizi,kacBit,stego_dosya_adi);
```

EK-C Sıraçma Uygulaması Matlab Kodları

```
% Sıraçma Ki Kare ile %
calismaYeri='test/testBolgesi/';
sonuc=false;
while sonuc == false
    stego_dosya_adi = lower(input('Sıraçma dosya adı(örnegin sırlı.wav)', 's'));
    uzunluk=size(stego_dosya_adi);
    bulunanIndex=strfind(stego_dosya_adi,'.wav');
       sonuc=true;
       dosya_tipi='wav';
    else
       sonuc=false;
       disp('Dosya girişi hatalı!');
    end
end
stego_dosya_adi=strcat(calismaYeri,stego_dosya_adi);
[stego_verisi,cerceveHizi,kacBit]  = wavread(stego_dosya_adi);

stego_verisi=round(2^(kacBit-1).*(stego_verisi))+2^(kacBit-1);
% diziyi döngü ile taramak için size alındı
[satir,sutun] = size(stego_verisi);
yuzde = zeros(100,1);
olasilik= zeros(100,1);
kategori_say = zeros(100,1);
for y=1:100
   n = 2^(kacBit-1); % n tane kategori var
   yuzde(y) = yuzde(y) + y;
   toplam_ornek = floor((y/100)*satir*sutun);
   kacSatir = floor(toplam_ornek/sutun);

   Tek=zeros(2^(kacBit-1),1);
   Cift=zeros(2^(kacBit-1),1);
   kiKare=zeros(2^(kacBit-1),1);

   for i=1:kacSatir
      for j=1:sutun
         if rem(stego_verisi(i,j),2)==0
            Cift((stego_verisi(i,j)/2)+1)=Cift((stego_verisi(i,j)/2)+1)+1;
         else
            Tek(((stego_verisi(i,j)-1)/2)+1)=Tek(((stego_verisi(i,j)-1)/2)+1)+1;
         end
         end
```

```
   end

   Z = (Tek + Cift)/2;
   for i=1:(2^(kacBit-1))
      if (Tek(i)+Cift(i)) < 5
         Tek(i) = 0;
         Cift(i) = 0;
         n = n - 1;
      end
   end

   kiKare = (Cift-Z).^2;
   for i=1:(2^(kacBit-1))
      if Z(i)==0
         kiKare(i) = 0;
      else
         kiKare(i) = kiKare(i)./Z(i);
      end
   end
   Toplam=sum(kiKare); %C burada Ki-kare istatistiğine işaret etmektedir.
   olasi=1-gammainc(Toplam/2,(n-1)/2); %gammainc fonksiyonun kullanarak
olasılığı hesaplarız.
   olasilik(y) = olasilik(y) + olasi;
   kategori_say(y) = kategori_say(y) + n;
end
fig_title = [stego_dosya_adi,'Veri Gömme Olasiligi Yüzdesi'];
kategori_say=cat(1,[0],kategori_say(1:100));
yuzde = cat(1, [0],yuzde(1:100));
olasilik = cat(1, [0], olasilik(1:100));
sonuc = cat(2, yuzde, olasilik, kategori_say);
figure (1), plot(yuzde,olasilik,'LineWidth',2,'Color','red');
title(fig_title);  xlabel('Yüzde Testi'); ylabel('Veri Gömme Olasılığı');
stego_dosya_adi=strrep(stego_dosya_adi, '.wav', '.mat');
save(stego_dosya_adi,'olasilik');
%  en son sonuçları kayıt etmektedir.
```

EK-D Toplu Veri Gizleme Uygulaması Matlab Kodları

Aşağıdaki kod 10 dosyaya (Ortu1.wav…Ortu10.wav) iki farklı mesajın %1'inden başlayarak %100'üne kadarını sırası ile gömmektedir. Oluşan sır dosyaları kaydetmektedir. Toplamda 2000 sır dosya oluşmaktadır. Bu dosyalar PNN yapay sinir ağının eğitimi için kullanılmıştır.

```
for dosyaId=1:10
   ortu_dosya_adi = strcat('test/ortu',int2str(dosyaId),'.wav');
   [ortu_verisi, cerceveHizi,kacBit]  = wavread(ortu_dosya_adi);
   ortu_verisi=round(2^(kacBit-1).*(ortu_verisi))+2^(kacBit-1);
   [satir,sutun]=size(ortu_verisi);
   for kez=1:2
      gizliMesaj=rem(floor(rand(satir,sutun)*1000),2);
      for yuzdeId=1:100
         stego_dosya_adi = strcat('test/stego/stego',int2str(dosyaId),'K',
                                  int2str(kez),'Y',int2str(yuzdeId),'.wav');
         yuzdeKac=yuzdeId;
         kacSatir=floor((floor((yuzdeKac/100)*satir*sutun))/sutun);
         stego_verisi=floor(ortu_verisi/2)*2+gizliMesaj;
         if kacSatir<satir
            % yuzdeKac yüzde 100 den kücük ise o zaman diger geriye kalanı örtü
            % verisinden tamamla
            stego_verisi=cat(1,stego_verisi(1:kacSatir,1:sutun),
                           ortu_verisi(kacSatir+1:satir,1:sutun));
         end
         stego_verisi = stego_verisi - (2^(kacBit-1));
         stego_verisi = stego_verisi ./ (2^(kacBit-1));
         wavwrite(stego_verisi,cerceveHizi,kacBit,stego_dosya_adi);
      end
   end
end
```

EK-E Toplu Sıraçma Uygulaması Matlab Kodları

Aşağıdaki kod, yukarıdaki kodun oluşturduğu 2000 sırlı dosyayı sırayla sıraçma yöntemine tabi tutmaktadır. Her dosya için dosyanın %1'inden başlayarak %100'üne kadar gizli veri varlığı hakkında olasılık değerleri hesaplamaktadır. Her dosya için 100 olasılık değeri hesaplar ve sonuçları [dosyaadi].[mat] şeklinde MAT dosyası olarak kaydeder. Toplamda 2000 tane MAT dosyası oluşur.

```
calismaYeri1='test/stego/';
calismaYeri2='test/egitimBolgesi/';
for dosyaId=1:10 % 10 dosya için dön
  for kez=1:2 % bir dosya için 2 kez 2 farklı mesaj ile % 1 ile 100 arası veri gömüldü
    for yuzdeId=1:100 % yüzde oranı
      stego_dosya_adi = strcat(calismaYeri1,'stego',int2str(dosyaId),'K',…
                                    int2str(kez),'Y',int2str(yuzdeId),'.wav');
      [stego_verisi,cerceveHizi,kacBit]  = wavread(stego_dosya_adi);
      stego_verisi=round(2^(kacBit-1).*(stego_verisi))+2^(kacBit-1);
      [satir,sutun] = size(stego_verisi);
      olasilik= zeros(100,1);
      for y=1:100
        n = 2^(kacBit-1); % n tane kategori var
        toplam_piksel = floor((y/100)*satir*sutun);
        kacSatir = floor(toplam_piksel/sutun);
        sonSatir = toplam_piksel - (kacSatir*sutun);
        Tek=zeros(2^(kacBit-1),1);
        Cift=zeros(2^(kacBit-1),1);
        kiKare=zeros(2^(kacBit-1),1);
        for i=1:kacSatir
          for j=1:sutun
            if rem(stego_verisi(i,j),2)==0
              Cift((stego_verisi(i,j)/2)+1)=Cift((stego_verisi(i,j)/2)+1)+1;
            else
              Tek(((stego_verisi(i,j)-1)/2)+1)=Tek(((stego_verisi(i,j)-1)/2)+1)+1;
            end
            end
        end
        Z = (Tek + Cift)/2;
        for i=1:(2^(kacBit-1))
          if (Tek(i)+Cift(i)) < 5
            Tek(i) = 0;
            Cift(i) = 0;
```

```
                n = n - 1;
            end
        end
        kiKare = (Cift-Z).^2;
        for i=1:(2^(kacBit-1))
            if Z(i)==0
                kiKare(i) = 0;
            else
                kiKare(i) = kiKare(i)./Z(i);
            end
        end
        toplam=sum(kiKare); %C burada Ki-kare istatistiğine işaret etmektedir.
        olasi=1-gammainc(toplam/2,(n-1)/2); %gammainc fonksiyonun kullanarak
olasılığı hesaplarız.
        olasilik(y) = olasilik(y) + olasi;
    end
    olasilik = cat(1, [0], olasilik(1:100));
save(strcat(calismaYeri2,'stego',int2str(dosyaId),'K',int2str(kez),'Y',int2str(yuzdeId)),'
olasilik');
  end
 end
end
```

EK-F PNN Yapay Sinir Ağı Eğitimi Matlab Kodları

Yukarıdaki kodun oluşturduğu 2000 tane MAT dosyasını tek tek okuyarak her dosya bir giriş olacak şekilde yapay sinir ağına verir ve her girişe karşılık bir çıkış değeri oluşturur. Bu şekilde alınan veriler ile PNN yapay sinir ağını eğitir.

```
P=[]; Tc=[];
for dosyaId=1:10 % 10 dosya için dön
   for kez=1:2 % bir dosya için 2 kez 2 farklı mesaj ile % 1 ile 100 arası veri gömülştü
      for yuzdeId=1:100 % yüzde oranı
        olasilik=strcat('test/egitimBolgesi/stego',int2str(dosyaId),'K',int2str(kez),'Y',…
                        int2str(yuzdeId));
        load(olasilik);
        if yuzdeId==1 % yüzde 0 için veri ekle
            P = cat(2,P,zeros(101,1)); % ilk seferde 0 değeri için 0 verileri ekleniyor
            Tc = cat(2,Tc,[101]);  % yüzde 0 -> 101 olarak gösterdik 0 hata veriyor
        end
        P = cat(2,P,olasilik); % her dosya bir giriş değeri olarak alınıyor
        Tc = cat(2,Tc,[yuzdeId]); % her giriş değerine karşılık bir çıkış değeri veriliyor
      end
   end
end
T=ind2vec(Tc);
net=newpnn(P,T);
Y = sim(net,P);
disp('Eğitim tamamlandı.');
```

EK-G PNN Yapay Sinir Ağı Testi Matlab Kodları

Son olarak yukarıdaki kod ile eğitilen PNN yapay sinir ağına hiç tanımadığı sırlı ses dosyasının içinde veri varmı yokmu, var ise yüzde ne kadarında veri var sorularına cevap verecek olan test kodları aşağıda verilmiştir.

```
P=olasilik;
Y = sim(net,P);
sonuc = vec2ind(Y);
if (sonuc==101)
    sonuc='Bu dosya orjinal dosyadır.';
else
    sonuc=strcat('% ',int2str(sonuc),' veri gizlenmiş.');
end;
disp(sonuc);
```

ÖZGEÇMİŞ

1982 yılında Ankara'da doğdu. 1999 yılında Ankara Balgat Teknik Lisesi'nden mezun oldu. Daha sonra 2001 yılında Ankara Üniversitesi Çankırı Meslek Yüksekokulu Bilgisayar Programcılığı bölümünden birincilikle mezun oldu. Aynı yıl Ankara'da özel bir yazılım şirketinde yazılım geliştirici olarak çalıştı. 2003 yılında Sakarya Üniversitesi Bilgisayar Sistemleri Öğretmeliği bölümünü kazanarak tekrar eğitimine devam etti. 2007 yılında bölümünden birincilikle mezun olduktan sonra 2 yıl Sakarya Üniversitesi Bilgi İşlem Daire Başkanlığında web yazılım uzmanı olarak görev yaptı. Daha sonra Sakarya Üniversitesi Bilgisayar Araştırma ve Uygulama Merkezine atanarak aynı görevine devam etti. Bu görevine halen devam etmektedir. 2008 yılında Sakarya Üniversitesi'nde Yüksek Lisans öğrenimi görmeye başladı. Evli ve Ayşe Nehir adında bir kızı vardır.