T.C
OSTİM TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YAZILIM MÜHENDİSLİĞİ ANA BİLİM DALI
YÜKSEK LİSANS PROGRAMI

ENCODE EDİLMİŞ BİR METNİN ENCODE TÜRÜNÜN DERİN ÖĞRENME YÖNTEMLERİYLE TAHMİNİ

YÜKSEK LİSANS TEZİ

HAZIRLAYAN
YAĞMUR IŞIK

TEZ DANIŞMANI
DR. ÖĞR. ÜYESİ OSMAN AKIN

ANKARA-2025

TEZ KABUL VE ONAY

Yağmur IŞIK tarafından hazırlanan "Encode Edilmiş Bir Metnin Encode Türünün Derin Öğrenme Yöntemleri ile Tahmini" başlıklı bu çalışma, 04/07/2025 tarihinde yapılan savunma sınavı sonucunda başarılı bulunarak jürimiz tarafından Yüksek Lisans/Doktora Tezi olarak kabul edilmiştir.

Kabul Tarihi: 04/07/2025

Jüri Üyesi:	**Dr. Öğr. Üyesi Fatih Sağlam** Ufuk Üniversitesi	_____
Jüri Üyesi:	**Dr. Öğr. Üyesi Ramazan Kocaoğlu** Ostim Teknik Üniversitesi	_____
Tez Danışmanı:	**Dr. Öğr. Üyesi Osman AKIN** Ostim Teknik Üniversitesi	_____

ONAY

Jüri tarafından kabul edilen bu çalışmanın Yüksek Lisans/Doktora Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

…../…../20….

_____

Prof. Dr. Halil Rıdvan ÖZ
Enstitü Müdürü

BİLDİRİM

Enstitü tarafından onaylanan Yüksek Lisans/Doktora tezimin tamamını veya herhangi bir kısmını basılı veya dijital biçimde arşivleme ve aşağıda belirtilen koşullar dahilinde erişime açma iznini Ostim Teknik Üniversitesine verdiğimi bildiririm. Bu izinle, Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak ve gelecekteki çalışmalar (makale, kitap, lisans, patent vb.) için tezimin tamamının veya bir bölümünün kullanım hakları yalnızca bana ait olacaktır.

Tezimin bütünüyle kendi çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Telif hakkı bulunan ve sahiplerinden yazılı izinle kullanılması zorunlu olan kaynakları, yazılı izin alarak kullandığımı ve istenildiğinde izinlerin suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayımlanan "Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge" kapsamında, tezim, aşağıda belirtilen koşullar haricince, YÖK Ulusal Tez Merkezi ve Ostim Teknik Üniversitesi Açık Erişim Sisteminde erişime açılır.

☐ Enstitü / Fakülte Yönetim Kurulu kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren 2 yıl ertelenmiştir.[1]

☐ Enstitü / Fakülte Yönetim Kurulunun gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren … ay en fazla 6 ay ertelenmiştir.[2]

☐ Tezimle ilgili gizlilik kararı verilmiştir.[3][4]

Tarih

İmza

[1] MADDE 6(1) Lisansüstü tezle ilgili patent başvurusu yapılması veya patent alma sürecinin devam etmesi durumunda, tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu iki yıl süre ile tezin erişime açılmasının ertelenmesine karar verebilir.

[2] MADDE 6(2) Yeni teknik, materyal ve metotların kullanıldığı, henüz makaleye dönüşmemiş veya patent gibi yöntemlerle korunmamış ve internetten paylaşılması durumunda 3. şahıslara veya kurumlara haksız kazanç imkanı oluşturabilecek bilgi ve bulguları içeren tezler hakkında tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulunun gerekçeli kararı ile altı ayı aşmamak üzere tezin erişime açılması engellenebilir.

[3] MADDE 7(1) Ulusal çıkarları veya güvenliği ilgilendiren, emniyet, istihbarat, savunma ve güvenlik, sağlık vb. konulara ilişkin lisansüstü tezlerle ilgili gizlilik kararı, tezin yapıldığı kurum tarafından verilir. Kurum ve kuruluşlarla yapılan işbirliği protokolü çerçevesinde hazırlanan lisansüstü tezlere ilişkin gizlilik kararı ise, ilgili kurum ve kuruluşun önerisi ile enstitü veya fakültenin uygun görüşü üzerine üniversite yönetim kurulu tarafından verilir. Gizlilik kararı verilen tezler Yükseköğretim Kuruluna bildirilir.

[4] MADDE 7(2) Gizlilik kararı verilen tezler gizlilik süresince enstitü veya fakülte tarafından gizlilik kuralları çerçevesinde muhafaza edilir, gizlilik kararının kaldırılması halinde Tez Otomasyon Sistemine yüklenir.

OSTİM TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ YÜKSEK LİSANS TEZİ ORİJİNALLİK RAPORU

Tezin Başlığı: Encode Edilmiş Bir Metnin Encode Türünün Derin Öğrenme Yöntemleriyle Tahmini

Öğrencinin Adı, Soyadı: Yağmur IŞIK

Tez Danışmanın Unvanı/Adı, Soyadı: Dr. Öğr. Üyesi Osman AKIN

Anabilim Dalı: Yazılım Mühendisliği

Programı: Yüksek Lisans

Tarih: 21 / 07 / 2025

Yukarıda başlığı belirtilen Yüksek Lisans/Doktora tez çalışmama ait Giriş, Ana Bölümler ve Sonuç kısmından oluşan ve toplam 69 sayfadan ibaret olan kısmı, 21 / 07 / 2025 tarihinde tez danışmanım/şahsım tarafından Turnitin intihal tespit programında incelenmiştir. Orijinallik raporunda aşağıda ifade edilen filtrelemeler uygulanmıştır.

Orijinallik raporuna göre, tezimin benzerlik oranı % 3'dür.

Uygulanan filtrelemeler:

1. Kaynakça (hariç)

2. Alıntılar (hariç)

3. Beş (5) kelimeden daha az örtüşme içeren metin kısımları (hariç)

Tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Öğrencinin İmzası:…….

ONAY
Tarih: 21 / 07 /2025
Danışmanın Unvanı, Adı, Soyadı, İmzası
Dr. Öğr. Üyesi Osman AKIN

ETİK BEYAN

Bu çalışmanın özgün bir çalışma olduğunu, çalışmanın hazırlık, veri toplama, analiz, bilgilerin sunumu ve diğer tüm aşamalarında bilimsel etik ve kurallara uygun davrandığımı, çalışmada bulunan tüm belge bilgileri akademik etik ve kurallar çerçevesinde elde ettiğimi, görsel, işitsel ve yazılı bütün bilgileri ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu, kullanmış olduğum verilerde herhangi bir tahrifat yapmadığımı, yararlandığım kaynaklara bilimsel normlara uygun olarak atıfta bulunduğumu, tezimin kaynak gösterdiğim durumlar dışında tarafımdan kaleme alındığını ve özgün olduğunu, Tez Danışmanım Osman Akın danışmanlığında ve tarafımdan üretildiğini ve OSTİM Teknik Üniversitesi Tez Yazım Kılavuzuna uygun olarak yazıldığını beyan ederim.

Yağmur IŞIK

04.07.2025

TEŞEKKÜR

Tez çalışmam süresince bilgi ve tecrübeleriyle bana rehberlik eden, desteğini ve sabrını hiçbir zaman esirgemeyen değerli danışmanım Dr. Osman Akın'a en içten teşekkürlerimi sunarım.

Bu süreçte her zaman yanımda olan, sevgileri ve anlayışlarıyla bana güç veren aileme ve manevi destekleriyle motivasyonumu artıran değerli arkadaşlarıma teşekkür ederim.

04.07.2025

Yağmur IŞIK

Yazar Adı ve Soyadı : Yağmur IŞIK

Üniversite : OSTİM Teknik Üniversitesi

Enstitü : Fen Bilimleri Enstitüsü

Program Adı : Yazılım Mühendisliği

Tezin Türü : Yüksek Lisans Tezi

Sayfa Sayısı : 69

Tarihi : 2025

ENCODE EDİLMİŞ BİR METNİN ENCODE TÜRÜNÜN DERİN ÖĞRENME MODELLERİYLE TAHMİNİ

Bu tez çalışması, kötü niyetli yazılımların sistemlere encoding mekanizmaları üzerinden sızmasını hedef alan saldırı türlerine karşı, encoding türlerini otomatik olarak tespit edebilen bir derin öğrenme modeli çözümü geliştirmeyi amaçlamaktadır. Çalışma kapsamında, Base64, Base32, Ascii85 ve Hexadecimal encode türlerinin yapısal örüntülerine dayanarak sınıflandırılması hedeflenmiş ve hem geleneksel makine öğrenmesi algoritmaları (Naive Bayes, KNN, Decision Tree) hem de derin öğrenme mimarileri (CNN, ResNet50) ile deneyler gerçekleştirilmiştir. Geleneksel yöntemler sınırlı başarı gösterirken, karakter düzeyinde CNN kullanımı ve encoding dizilerinin görselleştirilerek ResNet50 üzerinde sınıflandırılması, en yüksek doğruluğu sağlamıştır. Özellikle transfer öğrenme ile desteklenen modeller, daha az veriyle yüksek genelleme başarısı göstermiştir. Elde edilen sonuçlar, encoding yapılarını tespit eden derin öğrenme tabanlı yaklaşımların, zararlı yazılımların erken tespiti ve siber güvenlik altyapılarına yapay zekâ destekli karar mekanizmalarının entegrasyonu açısından önemli katkılar sunabileceğini göstermektedir.

Anahtar Sözcükler: Encode, encoding tespiti, derin öğrenme, metin sınıflandırma, siber güvenlik.

ABSTRACT

Thesis	: Yağmur IŞIK
University	: OSTİM Technical University
Institute	: Graduate School of Science and Engineering
Program's Name	: Software Engineering
Thesis Type:	: Master
Pages	: 69
Year	: 2025

PREDICTING THE ENCODE TYPE OF AN ENCODED TEXT WITH DEEP LEARNING MODELS

This thesis aims to develop a deep learning model solution that can automatically detect encoding types against attack types that target malicious software to infiltrate systems through encoding mechanisms. The study aims to classify Base64, Base32, Ascii85 and Hexadecimal encodings based on their structural patterns and experiments were conducted with both traditional machine learning algorithms (Naive Bayes, KNN, Decision Tree) and deep learning architectures (CNN, ResNet50). While traditional methods showed limited success, using CNN at the character level and visualizing the encoding sequences and classifying them on ResNet50 provided the highest accuracy. In particular, the models supported by transfer learning showed high generalization success with less data. The results show that deep learning-based approaches that detect encoding structures can make significant contributions to the early detection of malware and the integration of artificial intelligence-supported decision mechanisms into cyber security infrastructures.

Keywords: Encode, encoding detection, deep learning, text classification, cybersecurity.

TABLOLAR DİZİNİ

ŞEKİLLER DİZİNİ

SİMGELER VE KISALTMALAR

KNN	K-En Yakın Komşu
CNN	Evrişimsel Sinir Ağları
ResNet50	Artık Öğrenme
Hex	Hexadecimal

İÇİNDEKİLER

1. GİRİŞ

Dijital dünyada kötü niyetli yazılımların (malware) karmaşıklığı hızla artarak geleneksel imza tabanlı güvenlik sistemlerinin yetersiz kalmasına neden olmuştur. Özellikle zararlı içeriklerin çeşitli encoding yöntemleriyle gizlenerek iletilmesi, güvenlik filtrelerini aşmak için tercih edilen yaygın bir strateji haline gelmiştir. Encoding mekanizmaları, zararlı yük (payload) verisini maskeleyerek tespit edilmesini zorlaştırmaktadır. [1] Bu tehditlerin tespit edilmesini zorlaştıran önemli sebeplerden biri zararlı içeriklerin çeşitli encoding yöntemleri kullanılarak gizlenmesidir. Bu nedenle güvenlik sistemlerinin karşılaştığı zorlukları daha da derinleştirmektedir.

Bu tez çalışması, encoding türlerinin otomatik olarak tespit edilmesine odaklanarak, literatüre hem yöntemsel hem de uygulamalı açıdan iki temel katkı sunmaktadır. Birinci katkı, geleneksel makine öğrenmesi algoritmalarının (Naive Bayes, KNN, Karar Ağaçları) istatistiksel öznitelikler üzerinden gerçekleştirdiği sınıflandırmaların; encoding gibi yapısal örüntü içeren problemler karşısındaki sınırlı performansını deneysel olarak ortaya koymasıdır.

İkinci ve daha güçlü katkı ise, karakter düzeyinde uygulanan CNN tabanlı modellerin kısa ve orta uzunluktaki yapısal farklılıkları etkili biçimde yakalayabildiğini ve bu yapay sinir ağı mimarilerinin encoding tespiti gibi karmaşık sınıflandırma problemlerinde anlamlı sonuçlar üretebildiğini göstermesidir. Encoding dizilerinin RGB görsellerine dönüştürülerek ResNet50 gibi transfer öğrenme mimarileri üzerinde eğitilmesiyle, sınırlı veri ile dahi yüksek doğruluk ve güçlü genelleme kapasitesine sahip bir model elde edilmiştir. Bu yaklaşım, encoding türlerinin yalnızca metinsel analizle değil, aynı zamanda görsel temsiller üzerinden de yüksek doğrulukla sınıflandırılabileceğini göstermesi bakımından yenilikçi bir açılım sunmaktadır.

Çalışma kapsamında kullanılan ve sınıflandırılması hedeflenen encoding türleri Base64, Base32, Ascii85 ve Hex'tir. Bu amaçla hem geleneksel makine öğrenmesi algoritmaları hem de derin öğrenme mimarileri uygulanmıştır. Makine öğrenmesi sürecinde Naive Bayes, K-En Yakın Komşu (KNN) ve Karar Ağaçları gibi temel sınıflandırma algoritmaları kullanılmış, ancak bu yöntemlerin encoding türleri gibi yapısal örüntüler barındıran veri türlerinde sınırlı başarı sağladığı gözlemlenmiştir.

Derin öğrenme tarafında ise karakter dizileri görsel forma dönüştürülerek, transfer öğrenme yaklaşımıyla ResNet50 mimarisi eğitilmiştir.

Önerilen bu model ile %96 doğruluk, yüksek F1 skoru, Precision ve Recall değerleri elde edilerek encoding türlerinin etkili biçimde sınıflandırılabildiği ortaya konmuştur.

Önerilen model mevcut çalışmalardaki kısıtları aşmak üzere tasarlanmıştır. Hem klasik makine öğrenmesi algoritmaları hem de derin öğrenme mimarileri sistematik biçimde karşılaştırılmıştır. Encoding dizilerinin görsel temsili üzerinden transfer öğrenme yaklaşımıyla eğitilen ResNet50 modeli, kısa ve katmanlı kodlama yapılarında dahi yüksek doğruluk ve güçlü genelleme kapasitesi sergileyerek mevcut yöntemlere kıyasla anlamlı bir performans üstünlüğü ortaya koymuştur.

Tez kapsamında elde edilen bulgular, kötü niyetli yazılımların içeriklerini gizlemek amacıyla başvurdukları kodlama tekniklerine karşı, erken ve etkili bir tespit imkânı sunan yapay zekâ destekli savunma katmanlarının geliştirilmesine zemin oluşturmaktadır. Böylece, sadece mevcut tehditleri değil, aynı zamanda potansiyel saldırı varyantlarını da proaktif biçimde tespit edebilecek bir analiz modeli önerilmektedir.

2. ÖNCESİ LİTERATÜR TARAMASI

2.1. Encode Verilerinin Sınıflandırılmasına Yönelik Yaklaşım ve Yöntemler

Bu bölümde encode edilen verilerin sınıflandırılmasıyla ilgili literatürde yer alan yöntemler detaylı bir şekilde ele alınmakta. İlk olarak encode kavramı açıklanıyor ve sıkça kullanılan encode tiplerinin teknik özellikleriyle birlikte tanıtılıyor. Sonrasında geleneksel yöntemler kapsamında kullanılan kurallara dayalı ve desene dayalı eşleme yaklaşımları inceleniyor. Bu yöntemlerin sınırlılıkları tartışılıp yapay zekâ tabanlı sınıflandırma yaklaşımlarının önemi vurgulanıyor. Makine öğrenimi ve metin sınıflandırma tekniklerinin encode veriler üzerindeki uygulamalarına odaklanarak karakteristik verilerin özgünlükleri ve verilerin ön işleme stratejilerini ele alınacak.

2.2. Encode Kavramı ve Encode Türleri

Kodlama (encoding), verilerin dijital ortamda daha anlaşılır hale getirilebilmesini sağlamak için farklı bir formata dönüştürülmesi işlemidir. Genellikle kullanım alanı olarak verilerin iletilmesinde veya depolanmasında kullanılır. Kodlama işlem işlem sadece formata dönüşüm sağlar ve veriler üzerinde herhangi bir güvenlik sağlamaz. Temel kodlama tiplerinden bazıları arasında Base64, Onaltılık (Hex), URL Kodlama, ASCII, İkili ve Unicode tabanlı kodlamalar bulunmaktadır. Her bir kodlama yöntemi özel karakteristik özellikler gösterir. Örneğin, Base64 kodlaması sadece harfler, rakamlar ve '+' ,'/' gibi özel karakterlerden oluşur ve '=' karakterini padding için kullanır. Hex kodlaması ise yalnızca 0- 10 arası sayılar ve A-F harfleri içerir ve genellikle ikili verilerin okunabilir temsilinde kullanılır.

ASCII ve Unicode (UTF - 8 ve UTF - 16 dahil olmak üzere), karakter kodlamaları kategorisinde öne çıkan standartlar arasındadır. ASCII sadece 128 karakter temsil ederken; Unicode ise çoğu dilin sembollerini ve özel karakterlerini destekler. Bu kodlamalar genellikle metin verilerinin işlenmesinde önemli bir rol oynar.

2.3. Encode Verilerinin Tespiti: Geleneksel Yöntemler

Yıllar boyunca kodlanan verilerin yapısını analiz etmek genellikle manuel olarak yapılan analizler ve desen eşleme yöntemleriyle gerçekleştirilirdi. Bu süreçte sıkça kullanılan araçların başında düzenli ifadeler (regex) gelirdi. Her bir kodlama tipinin belirli karakter kümesine ve yapısal özelliklerine sahip olduğu için tanıma işlemine yardımcı olmak için desenler eşlenirdi.

Base64 kodlanmış diziler sadece belirli karakterlerden oluşur ve genellikle dört karakterlik bloklar halinde bulunurken; onaltılık (hexadecimal) verilerde ise her iki karakter bir baytı temsil eder. URL kodlanan veriler genellikle üç karakterden oluşur ve genellikle yüzde işareti (%) ile başlar (Örneğin %20).

Ancak bu tekniklerin bazı kısıtlamaları mevcuttur. Özellikle karma verilerde veya bir arada kullanılan birden fazla kodlama biçiminde, örneğin önce UTF-8 sonra Base64 gibi durumlarda, regex gibi sabit eşleme yöntemleri yetersiz olabilir. Ayrıca, regex tabanlı yaklaşımlar hata yapma eğilimindedir ve esnek olmayışlarından dolayı yanlış sonuçlar verebilir.

2.4. Yapay Zekâ Temelli Sınıflandırma Yaklaşımları

Yapay zekâ tabanlı sınıflandırma teknikleri günümüzde daha güçlü ve dinamik bir yaklaşımla geleneksel yöntemlerin yetersizliklerini ortadan kaldırmakta etkili olmaktadır. Özellikle makine öğrenme algoritmaları ve derin öğrenme modelleri encode edilen verilerin örgüsünü istatistiksel olarak analiz ederek sınıflandırma yapma konusunda yardımcı olabilirler. Bu tip uygulamalara dikkat edilmesi gereken ilk unsurlardan birinin verilerin ön işlenmesinin büyük öneme sahip olduğudur. Bu süreçte ilk adım encode edilen verilerin karakter düzeyinde veya belirli gruplar halinde tokenizasyon edilerek özelliklerin çıkarılmasıdır. Bu vektör temsillerinin elde edildikten sonra klasik algoritmalar olan Decision Tree Random Forest SVM gibi modellerle beraber Derin Öğrenme modellerinde de LSTM GRU CNN ve Transformer gibi modellerde de kullanılabilir.

Makine öğrenimi uygulamaları sade tarafından sınıflandırma işlevinin yanı sıra olasılıklara dayalı encode tipinin tahmin edildiği, anomalilerin tanımlandığı ve karmaşık decode kalıplarının çözdüğü görevlerde de kullanılabilir.

Yapay zekâ destekli sınıflandırma sistemlerinin başarısı genellikle verilerin temsil biçimine ve model mimarisine bağlı olup ön işleme süreçlerinin kalitesine dayanır. Özellikle kodlanmış verilerin -Base64 veya hexadecimal gibi- analizinde yapay zekâ modellerinin geleneksel yöntemlerden daha esnek ve güçlü olduğunu gözlemleyebiliriz. Çünkü bu veriler genellikle karakter diziler halinde kodlandığı için sıralı verilerin modellemesi önem taşır.

Literatürde karakter odaklı dizisel veri yapılarının sınıflandırılması konusunda Uzun-Kısa Süreli Belleğin (LSTM - Long Short-Term Memory) ve Kapılı Tekrarlayan Birimin (GRU - Gated Recurrent Unit) gibi Tekrarlayan Sinir Ağı tabanlı modellerin öne çıktığı göze çarpmaktadır. Hochreiter ve Schmidhuber (1997) tarafından geliştirilen LSTM özellikle uzun bağımlılıkları öğrenebilme yeteneği sayesinde kodlanan verilerdeki ardışık desenleri tespit etme konusunda yüksek performans sergilemektedir. [2]

Bu modellere ek olarak GRU'nun daha az parametre ile benzer performans sergiledikten dolayı işlem maliyetinin düşük olması tercih edilebilir hale gelebilmektedir. Bununla birlikte LSTM ve GRU modellerinin sıralı bilgi taşıma konusunda başarılı olsalar da paralelleştirme yetilerinin kısıtlı olmasından dolayı büyük verilerle eğitim süreleri uzayabilmektedir. Bu problem için geliştirilen Transformer mimarisi ise bağlam dikkati (self-attention) mekanizması sayesinde uzun dizilerdeki ilişkilerin paralel bir şekilde modellemeyi başarabilmektedir. [3] Transformer tabanlı modeller özellikle BERT, RoBERTa, GPT gibi önceden eğitilmiş versiyonlarıyla birlikte kodlanmış veri türlerinin sınıflandırılmasında etkili çözümler sunmaktadır. Özellikle dikkat mekanizması sayesinde karakter düzeyindeki ipuçlarından, kodlama türünü ayırt etmeye olanak tanıyan daha üst düzey temsiller öğrenilebilmektedir.

Gupta ve diğerleri (2023)'nin yaptığı bir araştırmada encode edilmiş verilerin sınıflandırılması için klasik makine öğrenme algoritmaları (örneğin SVM ve Random Forest) ile derin öğrenme modellerinin (LSTM ve Transformer gibi) karşılaştırıldığı belirtilmiştir. Raporda özellikle Transformer mimarisinin daha karmaşık desenleri tanımada başarılı olduğundan bahsedilmiştir. [4] Ayrıca araştırmada verilerin boyut indirgeme teknikleri kullanılarak görselleştirildikten sonra her encode tipinin belirli kümeleme davranışları sergilediği ve derin modellerin bu davranışları daha iyi ayıra bildiği sonucuna varılmış.

Son olarak yapay zekâ tabanlı sınıflandırma teknikleri sadece verinin hangi encode tipinde olduğunu tahmin etmekle kalmaz; Aynı zamanda bu kodlama biçimlerine özel olası anormallikleri tanımada ve analiz etmedeki başarısını da göstermektedir. Bu durum özellikle siber güvenlik verinin doğru kaynağı doğrulama ve otomatik verilerin işlenmesinde bu teknolojilerin pratik uygulanabilirliğini arttırmaktadır. Gelecekteki çalışmalarda bu modellerin çoklu encode yapılarını eş zamanlı olarak analiz edebilecek esnekliğe ulaşması ve az kaynak gereksinimine sahip aygıtlarda daha etkin bir şekilde çalışması sağlanmalıdır.

2.5. Encode Türü Belirleme Üzerine Yapılmış Önceki Çalışmalar

Kodlanmış verilerin doğru bir şekilde tanımlanması güvenlik alanlarında önemli bir öneme sahiptir. Dosya içeriği bazen bilinen türlerin bir kombinasyonu olabilir (örneğin PDF içine gömülmüş JavaScript). Dosya içeriği tipinin belirlenmesini (örneğin bir bayt dizisinin hangi dosya formatını temsil ettiğinin anlaşılması), tablo sütunlarının anlam verilerinin bulunmasını ve şifrelenmiş veya sıkıştırılmış verilerin tanınmasını ve şifreli ağ trafiğinin sınıflandırılmasını içeren problemlerle karşılaşabiliriz; bu problemler direkt olarak kodlanmış verilere dayanır ve bu veriler doğrudan içeriklerinden anlaşılamazlar. Son yıllarda bu alanlar için derin öğrenme temelli yaklaşımlar geliştirilmiştir. Bu kısımda, donanıma dayalı veri tipini tanımlamaya yönelik dört önde gelen çalışma olan Magika (2024), Sherlock (2019), EnCoD (2020) ve ET-BERT (2022) ele alınmıştır. Her bir modelin amacı ve çözdüğü sorun, kullanılan yöntemler, elde edilen sonuçlar ile uygulama alanları ve çalışmanın yayınlandığı kaynak aşağıda özetlenmiştir.

2.5.1. Magika (2024)

Magika adlı bir yapay zekâ destekli içerik tanıma aracı mevcut olup, bu teknoloji Google tarafından geliştirilmiştir. Temel amacı, bir bayttan yola çıkarak, o bayttaki verinin hangi tür veriler içerdiğini otomatik bir şekilde tespit edebilmektir (örneğin, Base64 ya da XML gibi). [5] Bu genellikle dosya uzantıları ya da sabit baytlarla yapılan tanımlamalara dayanarak çözülmeye çalışılmıştır; ancak bu teknikler kolayca yanıltılabilir ve özellikle kötü niyetli yazılımlar tarafından kötüye kullanılabilir. [5]

Bu Magika uygulaması, tam anlamıyla dosya içeriğini analiz ederek sorunu çözmek için derin öğrenme modelini benimser. Modelin eğitimi için 25 milyonun üzerinde dosya türünü kullanmayı amaçlayan bir metot izler. [5]

Eğitilmiş özel sinir ağı modelinin boyutu sadece 1 megabayt olup, tek bir işlemci üzerinde milisaniye mertebesinde hafif bir performans sergileyebilir. [5] Bu durum, Magika'nın herhangi bir uzantı veya sabit imza gerektirmeden dosyanın içeriğini incelemesine olanak tanır ve metin mi yoksa görüntü mü gibi kategorilere yüksek bir doğrulukla doğru bir şekilde sınıflandırabilir.

Magika'nın 100'den fazla içerik türü ve 1 milyondan fazla dosya üzerinde gerçekleştirdiği testlerde %99'luk bir F1 skoru elde ederek geleneksel yöntemlerin hepsini geride bıraktığı kanıtlanmıştır.

Özellikle metin tabanlı formatlarda (örneğin, program kaynak kodları veya yapılandırma dosyaları) gösterdiği başarı ile klasik araçların zorlandığı durumları bile doğru bir biçimde sınıflandırma yeteneğine sahiptir.

Magika'nın yüksek doğruluk ve verimliliği, gerçek dünya uygulamaları için cazip hale getiriyor; örneğin, Gmail'deki e-postaları tararken Magika'yı entegre etmiş ve VirusTotal zararlı yazılımları analiz ederken dosya türlerini belirlemek için Magika'nın kullanımını başlatmıştır [4]. Bu çalışma, Yanick Fratantonio ve ekibinin sunduğu "Magika: Yapay Zekâ Destekli İçerik Türü Tespiti" başlıklı makalede 2024'te yayımlanmıştır. [5] İlk sürümü açık kaynak olarak sunulan Magika, sonrasında 200'den fazla içerik türünü kapsayacak şekilde geliştirilmiştir.

2.5.2. Sherlock (2019)

Sherlock, yapılandırılmış tabloların sütunlarındaki verilerin anlam katmanlarını otomatik olarak tespit etmeye yönelik derin öğrenim tabanlı bir modeldir. Mesela, bir tablonun sütununda ülke adı, birey ismi, coğrafi koordinat ya da telefon numarası olup olmadığını belirlemek, veri bütünleştirme ve temizleme süreçlerinde kritik bir aşamadır. Bu sorun daha önce genellikle kelime eşleştirmeleri veya düzenli ifade kalıpları ile çözülmeye çalışılmıştır; ancak bu kelime/düzenli ifade temelli yöntemler, hatalı ya da eksik verilere karşı dirençli değildir ve tanıyabileceği tiplerin sayısı kısıtlıdır. [6]

Bu kısıtlamaların üstesinden gelmek için Hulsebos ve ekibi Sherlock modelini geliştirmiştir. [6] Sherlock, birden fazla veri görünümünü giriş olarak alan (çoklu-girdi) bir derin sinir ağı yapısı kullanır. Model, sütun verilerini dört ayrı özellik grubu çerçevesinde değerlendirir: (1) sütundaki değerlerin istatistiksel özellikleri, (2) karakter dağılımları, (3) sözcük temsilleri, (4) paragraf/belge vektörleri. [6]

Araştırmacılar, modeli VizNet veri setinden toplanan 686.765 sütun üzerinde eğitmiştir. Bu sütunlar, DBpedia ansiklopedik bilgilerinden elde edilen 78 farklı anlam tipiyle etiketlenmiştir. Sherlock modeli, bu geniş eğitim sonrasında yeni karşılaşılan sütunların tipini yüksek bir doğrulukla tahmin edebilmektedir. Deneysel sonuçlara göre model, 78 sınıf arasında 0.89 destek-ağırlıklı F1 skoru elde etmiştir ve bu performans, geleneksel makine öğrenimi tekniklerini, basit kelime/düzenli ifade yöntemlerini ve hatta insan kaynaklı kalabalık etiketleme yaklaşımlarını geride bırakmaktadır. [6]

Bu şekilde Sherlock, otomatik veri temizleme, şematik eşleme ve veri keşfi gibi veri bilimi görevlerinde değerli bir araç haline gelmiştir.

2.5.3. EnCoD (2020)

EnCoD (Encryption/Compression Distinguisher), dijital bilgilerin şifrelenip sıkıştırılmış olup olmadığını güvenilir bir şekilde ayırt etmek üzere tasarlanmış bir sınıflandırıcıdır. Güvenlik ve adli bilişim alanlarında, özellikle verilerin şifreli olup olmadığını tespit etmek hayati bir önem arz eder; örneğin, fidye yazılımlarını tanımak adına depolanan verinin aniden rastgele hale gelmesi (yüksek entropiye sahip olması) göz önünde bulundurulmalıdır. [7] Ancak geleneksel Shannon entropisi gibi istatistiksel ölçütler, modern sıkıştırma teknikleri karşısında yetersiz kalabilmektedir. Sıkıştırılmış verilere baktığımızda, tekrarları ortadan kaldırarak şifreli verilere benzer bir şekilde yüksek entropiye sahip olması nedeniyle, yalnızca entropi verilerine bakarak şifrelenmiş ve sıkıştırılmış veriler arasında ayırım yapmak zorlaşabilir. Üstelik, pratikte birçok sistem yalnızca verinin bir kesiti üzerinde analiz yapabildiğinden tam dosya formatına bağlı yaklaşımları uygulamak mümkün olmayabilir. [7]

EnCoD modelinin zorlukların üstesinden gelmek için derin öğrenme metotlarından yararlandığı bildirilmektedir. Fabio De Gasparive'nin ekibi tarafından önerilen EnCoDis adlı yapay sinir ağı sınıflandırıcısı, baytların istatistiksel özelliklerini girdi olarak kullanmaktadır. [7] Özellikle verilerin baytları arasındaki frekans dağılımları gibi unsurlar modele sunularak sıkıştırma ve şifreleme süreçlerinin eşsiz dağılımsal imzasının öğrenilmesi sağlanmaktadır. Deneylerde, EnCoD ismi verilen algoritma 16 çeşit dosya türünün ve 512 bayttan 8KB'a kadar değişen parça boyutlarının kullanıldığı bir standart veri kümesinde test edilmiştir. Sonuçlarımız, eski istatistiksel yöntemlerle karşılaştırıldığında EnCoD'un belirgin bir üstünlüğe sahip olduğunu göstermektedir:

512 baytlık küçük parçalarda bile yaklaşık %82 doğruluk elde edilirken, 8KB boyutundaki parçalar için bu oran %92 olarak tespit edilmiştir. [7] Hatta sıkıştırılmış ve şifrelenmiş veriyi ayırt etme yeteneği senaryolarında doğruluk oranının %94 olduğu ve bazı özel formatlar için ise %100'lük bir başarı sağlandığı rapor edilmiştir. Bu kabiliyet birçok alanda yararlıdır; örneğin, fidye yazılımlarını tespit etmek; diğer faydalı uygulamalar dijital adli analiz ve ağ trafiği incelemesi olarak öne çıkmaktadır.

2.5.4. ET-BERT (2022)

ET-BERT (şifreli trafik BERT), şifreli ağ trafiğini sınıflandırmak için geliştirilen derin bir öğrenme modelidir. Modern internet trafiğinin çoğu TLS gibi protokoller kullanılarak şifrelenir. Bu, bir ağ uygulamasının hangi uygulamaları içerdiğini anlamayı zorlaştırır. [8]

Geleneksel yaklaşımlar deneyimli istatistiksel özelliklere veya çok sayıda veriye dayanmaktadır. Bununla birlikte, bu yöntemler yeni şifreleme tekniklerine uyum sağlamak ve sınırlı etiket verileriyle genellemek için yeterli değildir. Xinjie Lin ve meslektaşları bu sorunu çözmek için trans tabanlı bir ET-BERT modeli önerdiler. [8]

ET-BERT, ağ trafiğini dil benzeri temsillere dönüştürerek eğitim öncesi (önkoşullar) gerçekleştirir. Maskeli patlama modellemesi (MBM) ve bağlamsal trafik ilişkileri ile aynı kaynak gömme tahminini (SBP) öğrenme görevleri. Daha sonra, birkaç tarihte farklı sınıflandırma görevleri (metal ayarlama) gerçekleştirilir. [8]

Deneysel İncelemesinde ET-BERT ", önceki yöntemi teslim ederek beş farklı görevde F1 puanları elde etti. VPN trafik algılama, mobil cihaz trafiği ve TLS 1.3 sınıflandırması zorlu alanlarda önemli bir başarı göstermiştir. [8]

2.6. Mevcut Yöntemlerin Değerlendirilmesi ve Kısıtları

Encode türü tespitiyle ilgili güncel literatürde yapay zekâ tabanlı modeller öne çıkmaktadır. Özellikle Magika, Sherlock, EnCoD ve ET-BERT gibi sistemler, farklı veri türlerini sınıflandırmak amacıyla geliştirilmiş ve encode benzeri yapıların ayrıştırılmasında dolaylı olarak kullanılabilir hale gelmiştir. Ancak bu modellerin çoklu encode yapılar ve güvenlik açısından kritik diziler üzerindeki etkinliği sınırlı kalmaktadır.

Magika, Google tarafından geliştirilen ve bayt düzeyinde içerik sınıflandırması yapan bir sinir ağı modelidir. Base64, XML gibi metin tabanlı içerikleri yüksek doğrulukla tanıyabilen bu sistem, özellikle dosya uzantılarına güvenmeden sınıflama yapabilmesiyle dikkat çekmektedir. [9] Ancak Magika'nın eğitim verisi genellikle düz ve tek katmanlı kodlama yapılarına odaklandığı için, iç içe geçmiş encode dizileri (örneğin önce UTF-8 sonra Base64) üzerinde çözümleme kapasitesi düşmektedir.

Sherlock modeli ise, yapılandırılmış veri sütunlarının semantik anlamlarını otomatik olarak çıkarmak üzere geliştirilmiş çoklu girişli derin öğrenme sistemidir. [10] Sütun verilerini karakteristik, istatistiksel ve sözcük düzeyinde temsil edebilse de anlamsız ya da sentetik karakter dizileri olan encode edilmiş içeriklerde, modelin dilsel bağlama dayalı doğası nedeniyle sınırlı performans gösterdiği gözlemlenmektedir.

EnCoD, özellikle sıkıştırılmış ve şifrelenmiş verileri birbirinden ayırmak üzere tasarlanmıştır. [11]

Bayt düzeyinde istatistiksel özelliklere dayalı olarak çalışan bu model, yüksek entropili verilerde başarılı sonuçlar sunarken, encode edilmiş fakat düşük entropili verilerde (örneğin ASCII veya Hex) ayrım gücü zayıflamaktadır. Çoklu encode zincirlerinde ilk katmandaki verinin sinyalleri zayıfladığı için modelin tahmin kabiliyeti düşmektedir.

ET-BERT ise, Transformer mimarisi üzerine inşa edilmiş ve şifreli ağ trafiği sınıflandırması için optimize edilmiş bir derin öğrenme sistemidir. [12] Bağlam dikkati (self-attention) mekanizması sayesinde uzun dizilerdeki örüntüleri modelleme konusunda güçlüdür. Ancak ET-BERT'in eğitildiği veri kümesi ağ trafiği örneklerinden oluştuğu için, kısa ve statik encode dizilerini analiz etmede genelleme problemi yaşayabilmektedir. Ayrıca katmanlı encode durumlarında ardışık bağlamların bozulması modelin performansını etkileyebilir.

Aslan ve Samet (2020) tarafından yapılan kapsamlı incelemede, bu tip yapay öğrenme temelli sistemlerin genel olarak yeni nesil kötü amaçlı yazılımların karmaşık yapıları karşısında sınırlı kaldığı belirtilmiştir. [13]

Yukarıda değerlendirilen yöntemler çoklu encode dizilerini yüksek doğrulukla sınıflandıracak biçimde tasarlanmamıştır. Encode edilmiş karakter dizileri, içerik anlamı taşımayan ve istatistiksel dağılımı bozan yapılar sunduğu için, klasik semantik tabanlı veya bayt istatistiği tabanlı modeller yetersiz kalmaktadır. Günümüzde kötü amaçlı yazılımlar, zararlı içeriği gizlemek amacıyla sıklıkla encode ya da encryption tekniklerini zincirleme şekilde kullanmaktadır. Bu yapıların doğru tanımlanamaması, zararlı yazılımların erken tespitini ve ayrıştırılmasını doğrudan zorlaştırmaktadır. [13] Özellikle şifreli veya encode edilmiş verilerin doğru şekilde analiz edilmesi, yalnızca içerik sınıflandırma açısından değil, aynı zamanda veri güvenliği ve siber tehditlerin erken teşhisi açısından da kritik rol oynamaktadır. Encode türlerinin doğru tanımlanması, bir verinin gerçekten şifreli mi yoksa sadece formatlanmış mı olduğunu ayırt etmeye yardımcı olur. Derin Öğrenme mimarileri karakter düzeyinde örüntüleri öğrenebilen ve sıralı bilgi taşıyabilen yapılardır. Bu doğrultuda encode edilmiş verilerin yapısal ipuçlarını yakalama ve çoklu encode zincirlerini modelleyebilme konusunda anlamlı sonuçlar çıkarabilmektedirler. Literatürde LSTM, GRU ve Transformer gibi dizisel öğrenme yeteneğine sahip modellerin, bu tarz verilerin sınıflandırılmasında klasik yöntemlerden daha yüksek başarı sağladığı gösterilmiştir. [13] Dolayısıyla, encode türlerinin doğru ve otomatik sınıflandırılması yalnızca veri etiketleme değil, aynı zamanda gizlenmiş zararlı içeriklerin açığa çıkarılması açısından önemli bir adımdır. Bu nedenle, derin öğrenme tabanlı yaklaşımlar hem güvenlik hem de doğruluk açısından gelecekteki sistemler için stratejik bir çözüm yoludur.

Bu tez çalışması, mevcut literatürdeki yapay öğrenme temelli çözümlerin encode yapılar üzerindeki sınırlı başarımını aşmak ve encode türlerinin yapay zekâ modelleri ile doğru biçimde sınıflandırılabileceğini ortaya koymak amacıyla gerçekleştirilmiştir. Literatürde yer alan Magika, Sherlock, EnCoD ve ET-BERT gibi sistemler, belirli veri türlerinde başarılı sonuçlar verseler de iç içe geçmiş (çok katmanlı) encode diziler veya kısa, yapısal anlamdan yoksun karakter dizileri gibi güvenlik açısından kritik veri yapıları üzerinde sınırlı performans göstermektedir. [11] Bu çalışma ile encoding dizileri karakter düzeyinde örüntü bazlı bir yaklaşımla ele alınmış, derin öğrenme mimarilerinin bu örüntüleri öğrenerek türleri birbirinden ayırt edebileceği gösterilmiştir. Böylece geleneksel yöntemlerin baş edemediği, semantik içerik taşımayan veya istatistiksel olarak ayrıştırılması zor olan encoding yapılarının sınıflandırılmasında yüksek doğruluk elde edilmiştir.

Çalışmanın diğer katkılarından biri farklı encoding türlerinin güvenilir biçimde sınıflandırılmasının, zararlı yazılımların tespiti sürecinde araya yerleştirilebilecek yapay zekâ destekli yeni bir savunma katmanının önünü açmasıdır. Günümüzde kötü amaçlı yazılımlar, zararlı kodlarını doğrudan iletmek yerine çeşitli encoding yöntemleriyle gizleyerek klasik filtreleme sistemlerini aşmayı hedeflemektedir. Encode türlerinin doğru ve otomatik biçimde ayırt edilmesi yalnızca veri türü sınıflandırması değil, aynı zamanda encode yapılarının içine yerleştirilmiş olabilecek kötücül kodları tespit etmede önemli bir ilk adım işlevindedir. Çalışmada elde edilen yüksek doğruluk oranları, encoding ayrımının derin öğrenme mimarileriyle yapılabileceğini göstermiş ve bu ayrımın, tehdit algılama sistemlerine entegre edilebilecek bir ön analiz katmanı olarak değerlendirilmesini mümkün kılmıştır. Böylelikle, siber güvenlik altyapıları için hem önleyici hem de destekleyici rol üstlenecek, encode merkezli bir tehdit tarama bileşeni önerilmiştir.

3. ÖNERİLEN YÖNTEM

3.1. Çalışmanın Amacı ve Problem Tanımı

Bu çalışmanın amacı encode edilmiş verilerin encode türünü yapay zekâ desteğiyle belirleyebilen bir sınıflandırma modeli oluşturmaktır. Bu model oluşturulurken kullanılan encode türleri Base64, Base32, Ascii85 ve Hex olarak belirlenmiştir. İçerik analizinin gerçekleştirilebilmesi ve olası güvenlik açıklarının belirlenebilmesine yardımcı olmak için öncelikle kullanılan kodlama yönteminin doğru bir şekilde belirlenmesi gerekmektedir. Encode türlerinin otomatik olarak belirlenebilmesi, veri güvenliği, tersine mühendislik ve adli bilişim alanlarında fayda sağlayabilecek bir ön adımdır. Literatürde encode modellerinin sınıflandırılmasıyla ilgili çok az doğrudan çalışma bulunsa da benzer alanlarında biçim tanıma, verinin tipini belirleme gibi çeşitli yaklaşımlar geliştirilmiştir. [14], [15]

Çalışmada encode edilmiş verilerin türünü otomatik olarak sınıflandırmaya yönelik bir model geliştirilmesi hedeflenmiştir. Bu amaç doğrultusunda hem makine öğrenmesi hem de derin öğrenme temelli yöntemler kullanılmış ve farklı algoritmalar karşılaştırmalı olarak değerlendirilmiştir. Makine öğrenmesi aşamasında Naive Bayes, K-En Yakın Komşu (KNN) ve Karar Ağacı (Decision Tree) algoritmaları uygulanarak temel sınıflandırma performansları değerlendirilmiştir. Ardından, daha yüksek doğruluk ve genel geçerlik elde edebilmek amacıyla derin öğrenme tabanlı CNN ve ResNet50 modelleri kullanılmıştır. Çalışma, Base64, Base32, Ascii85 ve Hex olmak üzere dört farklı encode türünün ayırt edilmesini hedefleyen çok sınıflı bir sınıflandırma problemine odaklanmaktadır.

Geliştirilen model, encode edilmiş verilerin biçimsel özelliklerinden yararlanarak hangi kodlama yönteminin kullanıldığını yüksek doğrulukla tahmin edebilmektedir. Bu çalışma, söz konusu verilerin türünü otomatik olarak belirleyerek veri güvenliği analizlerine katkı sağlamayı, adli bilişim süreçlerini desteklemeyi ve tersine mühendislik çalışmalarında başlangıç noktası oluşturmayı amaçlamaktadır. Ayrıca, sınıflandırma modeli literatürdeki boşluğu dolduracak biçimde özgün bir yaklaşım sunmakta ve bu alandaki ileri düzey çalışmalara altyapı sağlayabilecek potansiyeldedir.

3.2. Encode Nedir?

Bilgi güvenliği ve veri iletimi alanlarında, encoding (kodlama), verilerin belirli bir biçime dönüştürülmesi işlemidir. Bu dönüşüm, genellikle verilerin sistemler arasında güvenli, bütünlüklü ve hatasız bir şekilde taşınabilmesini sağlamak amacıyla gerçekleştirilir.

Kodlama işlemi sırasında verinin içeriği değiştirilmez; yalnızca temsil biçimi farklı bir formata çevrilir. Bu yönüyle encoding, encryption ya da hashleme gibi tekniklerden temel olarak ayrılır çünkü geri dönüştürülebilir bir işlemdir. [16] Kodlama işlemleri, genellikle insan tarafından okunamayan ikili verilerin (binary data) metin temelli protokollerde güvenli biçimde taşınabilmesini amaçlar. [17] Kodlama sürecinin temel amacı verilerin bütünlüğünü korumak ve iletim sırasında oluşabilecek hataları en az seviyeye indirgemektir. Özellikle verilerin farklı platformlar veya protokoller arasında alışverişinin yapıldığı durumlarda karakter uyumsuzluğunu önlemek ve verilerin kaybolmasını engellemek için oldukça önemli bir rol oynar. Ancak kodlama teknikleri sadece meşru amaçlar için kullanılmaz; kötü niyetli kişiler tarafından da kötüye kullanılabilirler. Bu gibi durumlarda kodlama yöntemleri verilerin analiz edilmesini engellemek için kullanılabilir veya güvenlik önlemlerini atlatmak amacıyla zararlı içeriklerin masum şekillerde iletilmesinde kullanılabilirler. Örneğin, zararlı yazılımlar veya saldırganlar, kodlama tekniklerini kullanarak zararlı yükleri analizden gizleyebilir ya da güvenlik duvarlarını aşabilir. [18] Bu bağlamda, kodlama türlerinin otomatik olarak tanımlanması, güvenlik uygulamaları açısından önem taşımaktadır.

3.3. Kullanılan Encode Türleri ve Özellikleri

Kodlama yöntemleri ikili verilerin metin tabanlı sistemlerde güvenli ve anlaşılır bir biçimde taşınmasını sağlamak amacıyla geliştirilmiştir. Kodlama süreçleri sadece verilerin aktarımını kolaylaştırmakla kalmaz; protokol uyumluluğuyla birlikte insanlar tarafından okunabilirlik, hata toleransı gibi çeşitli sistem gereksinimlerine göre optimize edilmiş yapılar sunarlar. [19] Bu araştırmada incelenen kodlama yöntemleri arasında Base64, Base32, Ascii85 ve On altılı (Hexadecimal-Hex) yöntemleri bulunuyor. Bu kodlama yöntemleri karakter setleri, veriyi şişirme oranları, okunabilirlik seviyeleri ve kullanım alanları açısından birbirinden farklıdır.

3.3.1. Base64

Base64, özellikle MIME tipi e-posta içerikleri ve XML/JSON tabanlı veri taşıma protokollerinde sıkça kullanılan bir kodlama yöntemidir.

Bu yöntemde, her üç baytlık veri bloğu dört karakterlik bir ASCII temsil ile ifade edilerek %33 oranında veri büyümesine neden olur. Kodlama, büyük ve küçük harfler (A–Z, a–z), rakamlar (0–9), artı (+) ve eğik çizgi (/) olmak üzere toplam 64 karakterden oluşan sabit bir karakter kümesine dayanır. [20]

Uygulama alanlarının genişliği ve desteğinin evrensel olması, Base64'ü en yaygın kullanılan encoding türlerinden biri haline getirmiştir. Base64 kodlamasının nasıl göründüğüne dair örnek Şekil 3.1'de gösterilmiştir.

```
Base64,a2VybmVsIHNjYW5Db250cm9sQ29tb+/
```

Şekil 3.1 Base64 Örnek Karakter Kümesi

3.3.2. Base32

Base32, yalnızca büyük harfler (A–Z) ile sınırlı bir karakter kümesi ve rakamların bir alt kümesi (2–7) kullanılarak kodlama yapar. Bu şekilde insanlar tarafından daha kolay okunabilir olur ve optik tarayıcılar ile sesli iletişim sistemleri gibi hata hassasiyetinin önemli olduğu ortamlarda güvenilir sonuçlar elde edilir. Base32, özellikle zaman tabanlı tek seferlik parola algoritmaları (TOTP), QR kod sistemleri ve bazı DNS çözümlerinde kullanılmaktadır. [21] Ancak veri şişirme oranı Base64'e kıyasla daha yüksektir. Base32 kodlamasının nasıl göründüğüne dair örnek Şekil 3.3.2'de gösterilmiştir.

```
Base32,HRHFKTCMHY6C2PR4JZKUYTB6LU======
```

Şekil 3.2 Base32 Örnek Karakter Kümesi

3.3.3. Ascii85

Ascii85 adı verilen bir kodlama yöntemi, verilerin sıkışmasını daha etkin hale getirmek için tasarlanmış bir yöntemdir. İki bitlik ikili verilerin beş karakterle ifade edilmesini sağlayarak verinin genişlemesini azaltır. Bu özelliği nedeniyle, özellikle Adobe'nin PostScript ve PDF formatlarında varsayılan kodlama yöntemi olarak yer almaktadır. [22] 85 karakterlik bir ASCII kümesi kullanan Ascii85, veri yoğunluğu açısından avantajlı olmakla birlikte karakter çeşitliliği nedeniyle ayrıştırılması Base64'e göre daha karmaşıktır. Ascii85 kodlamasının nasıl göründüğüne dair örnek Şekil 3.3'de gösterilmiştir.

```
Ascii85,"Ci<flFD5Z2A0?4CD0'>63[/K[6$?gWB4Z*LA9MO)
```

Şekil 3.3 Ascii85 Örnek Karakter Kümesi

3.3.4. Hexadecimal (Hex)

Hex kodlama yöntemi, ikili verileri doğrudan onaltılık sayı sistemi (0–9, A–F) ile temsil eder.

Her bir bayt için iki karakterlik bir gösterim gerektirdiğinden, veri hacmi anlamında en düşük verimliliğe sahiptir. Buna karşın sade yapısı ve yüksek şeffaflığı sayesinde hata ayıklama, bellek analizi ve adli bilişim gibi düşük seviyeli teknik analiz uygulamalarında tercih edilmektedir. [23]

Hexadecimal kodlama veri gizleme içermediği ve dönüşümü kolay olduğu için birçok güvenlik yazılımda ve adli araçta varsayılan analiz formatı olarak kabul edilmektedir. Hexadecimal kodlamasının nasıl göründüğüne dair örnek Şekil 3.4'de gösterilmiştir.

```
Hex,737973746565D737461747320C2A02048414C435F5
```

Şekil 3.4 Hex Örnek Karakter Kümesi

3.4. Geleneksel Makine Öğrenmesi Algoritmaları

Kodlama türlerinin sınıflandırılmasında, yorumlanabilirlik ve uygulanabilirlik açısından avantaj sağlayan üç geleneksel makine öğrenmesi algoritmaları kullanılmıştır. Bu kapsamda özellikle Naive Bayes, K-En Yakın Komşu (K-Nearest Neighbors, KNN) ve Karar Ağacı (Decision Tree) algoritmaları tercih edilmiştir. Söz konusu yöntemler, karakteristik öznitelikler üzerinden kodlama biçimlerini ayrıştırma yetenekleri sayesinde, sınıflandırma sürecinin yol haritasını belirleyen yapı taşları olmuştur.

Naive Bayes algoritması, özellikle sınıf koşullu olasılıkları hızlı hesaplayabilmesi ve metin tabanlı özniteliklerde yüksek başarı oranları göstermesi nedeniyle seçilmiştir. KNN algoritması, örüntü benzerliklerini mesafe ölçütleriyle değerlendirmesi sayesinde encoding türleri gibi yüzeysel yapısal benzerlik içeren veriler üzerinde etkili sonuçlar üretebilmektedir.

Karar Ağacı algoritması ise, karar kurallarını sezgisel olarak görselleştirme ve yorumlama imkânı sunması nedeniyle tercih edilmiştir. Bu üç algoritmanın birlikte değerlendirilmesi, model karşılaştırması açısından da analitik bir avantaj sağlamaktadır.

3.4.1. Naive Bayes

Naive Bayes algoritması, Bayes teoremini temel alan ve sınıflandırma problemlerinde yaygın olarak kullanılan olasılık tabanlı bir öğrenme yöntemidir. Temel varsayımı, özniteliklerin birbirinden koşulsuz olarak bağımsız olduğudur.

Bu varsayım pratik verilerde her zaman geçerli olmasa da modelin basitliği ve düşük hesaplama gereksinimi, özellikle metin madenciliği ve içerik analizi gibi yüksek boyutlu veri setlerinde etkili sonuçlar elde edilmesini mümkün kılmaktadır. [24] Bu algoritma bir gözlem kümesinin belirli bir sınıfa ait olma olasılığını, o sınıfa ait olma koşuluyla gözlemlerin gerçekleşme olasılığı ve sınıfın önsel (prior) olasılığı üzerinden tanımlar. Naive Bayes teoremi, Eşitlik (1)'de gösterildiği gibi ifade edilir. [25]

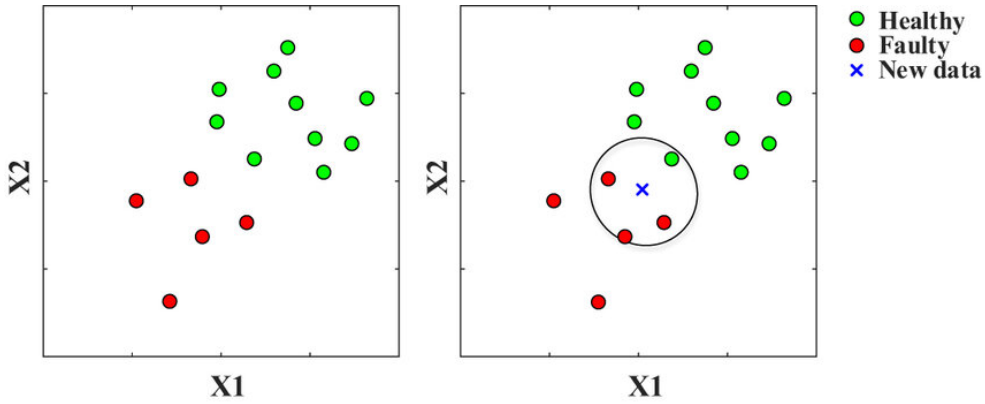
$$P(C_k \mid x) = (P(x \mid C_k) \cdot P(C_k)) / P(x) \quad (1)$$

Geleneksel makine öğrenmesi algoritmalarından biri olan Naive Bayes sınıflandırıcısı, encode edilmiş verilerin türlerini tespit etmek amacıyla uygulanmıştır.

Model eğitimi öncesinde, eksik değer içeren kayıtlar temizlenmiş ve veri rastgele bir şekilde %80 eğitim, %20 test olarak ayrılmıştır.

Uygulama sürecinde scikit-learn kütüphanesindeki GaussianNB sınıfı kullanılmış ve model varsayılan parametrelerle eğitilmiştir. Gaussian Naive Bayes modeli, özniteliklerin sürekli ve normal dağıldığı varsayımına dayandığından, bu dağılımın veri üzerinde kabul edilebilir olduğu varsayılmıştır. [26]

Algoritma, her sınıfta gözlemlenen niteliklerin olasılık dağılımını öğrenir ve sınıflandırma aşamasındaki bu dağılımlara göre en yüksek yeni bir gözlem olasılığını tahmin eder. Kodlama türlerinin belirlenmesinde, farklı encode biçimlerinin karakteristik frekans dağılımları ve sembol kullanım yapıları (örneğin, Base64'te "=" karakterinin sık görülmesi ya da Hex'te yalnızca 0–9 ve A–F karakterlerinin bulunması) gibi öznitelikler üzerinde etkili bir ayrıştırma yapılabilmektedir. Naive Bayes algoritmasının nasıl çalıştığına ait görsel Şekil 3.5'de verilmiştir. [27] Her sınıf için koşullu olasılık dağılımları hesaplanır. Gözlem noktasının ait olduğu sınıf, Bayes teoremi doğrultusunda en yüksek olasılığa göre belirlenir.



Şekil 3.5 Naive Bayes Algoritması

3.4.2. K-En Yakın Komşu (K-Nearest Neighbors – KNN)

K-En Yakın Komşu (K-Nearest Neighbors – KNN) algoritması, örnek temelli ve bellek tabanlı bir sınıflandırma yöntemidir. "Lazy learning" olarak sınıflandırılan bu algoritma, model eğitimi sırasında herhangi bir parametre öğrenimi yapmaz; bunun yerine karar mekanizmasını test aşamasında uygular. [28]

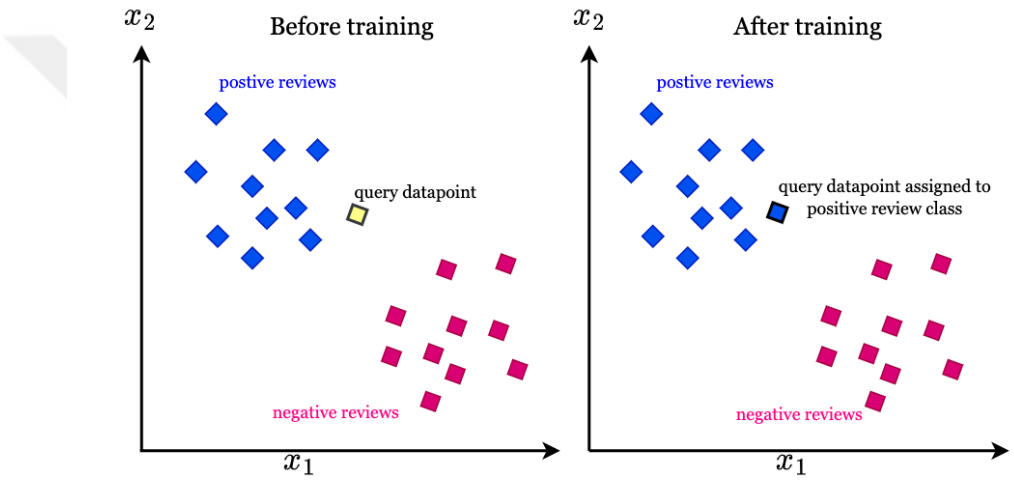
KNN algoritması belirli bir verinin sınıflandırılması için k kullanıcı tanımlı komşuları belirler ve bu komşuların sınıflandırma etiketlerine göre çoğunluk oylaması yaparak örneği sınıflandırır. KNN yalınca eğitim verilerinin yapısına dayanarak sınıflandırma kararı verir. En yaygın kullanılan uzaklık ölçüsü, Öklid (Euclidean) mesafesidir. Uzaklığın ifade edildiği Eşitlik (2)'de gösterildiği gibi ifade edilir. [29]

$$(i,j) = \sqrt{\sum_{k=1}^{p} \left(x_{ik} - x_{jk}\right)^2} \tag{2}$$

Encoding türlerinin sınıflandırılmasında KNN, veri örnekleri arasındaki yapısal benzerliklerden faydalanarak etkili sonuçlar verebilir. Örneğin, Base64 ya da Ascii85 ile kodlanmış diziler, karakter uzunlukları, sembol frekansları veya yapısal örüntüler açısından birbirine benzerlik gösterebilir. Bu benzerlikler, öznitelik uzayında yakın konumlanan veri örneklerinin aynı sınıfa ait olma eğilimi göstermesi sayesinde, KNN algoritması tarafından başarıyla ayrıştırılabilir.

Ayrıca, algoritmanın dağılım varsayımı gerektirmemesi ve doğrusal olmayan karar sınırlarını doğal biçimde öğrenebilmesi, kodlama türü gibi heterojen yapılı veriler üzerinde önemli bir avantaj sağlamaktadır.

Ancak algoritmanın en belirgin dezavantajı, test aşamasında tüm eğitim verisiyle mesafe hesaplaması yapma zorunluluğudur. Bu durum, büyük veri kümelerinde yüksek hesaplama maliyetine yol açabilir. Bu nedenle, veri seti boyutunun sınırlı olduğu durumlarda KNN daha uygulanabilir bir yöntem olarak öne çıkmaktadır. [28], [30] KNN algoritmasının nasıl çalıştığına dair görsel Şekil 3.6'da verilmiştir. [31]



Şekil 3.6 KNN Algoritması

3.4.3. Karar Ağacı (Decision Tree)

Karar ağaçları (Decision Trees), sınıflandırma ve regresyon problemlerinde sıkça kullanılan, sezgisel ve yorumlanabilirliği yüksek bir makine öğrenmesi algoritmasıdır. Ağaç yapısı, bir kök düğümden (root node) başlayarak veri kümesini belirli öznitelik değerlerine göre dallara ayırır ve nihayetinde yaprak düğümler (leaf nodes) sınıf etiketlerini temsil eder. Bu yapı, sınıflandırma sürecini adım adım izlenebilir hale getirir ve bu özelliğiyle özellikle veri yorumlamanın kritik olduğu uygulamalarda tercih edilir. [32] Karar ağaçlarının yapılandırılmasında temel amaç, her dalmada veri setini mümkün olduğunca homojen alt gruplara ayırmaktır. Bu nedenle, dallanma kararları bilgi kazancı (Information Gain), Gini indeksi (Gini Index) veya Entropi gibi ayrım ölçütlerine dayalı olarak verilir.

En yaygın kullanılan ölçütlerden biri bilgi kazancıdır. Bilgi kazancı, bir özniteliğin veri kümesindeki belirsizliği ne kadar azalttığını ölçer. Eşitlik (3)'de gösterildiği gibi tanımlanır. [33]

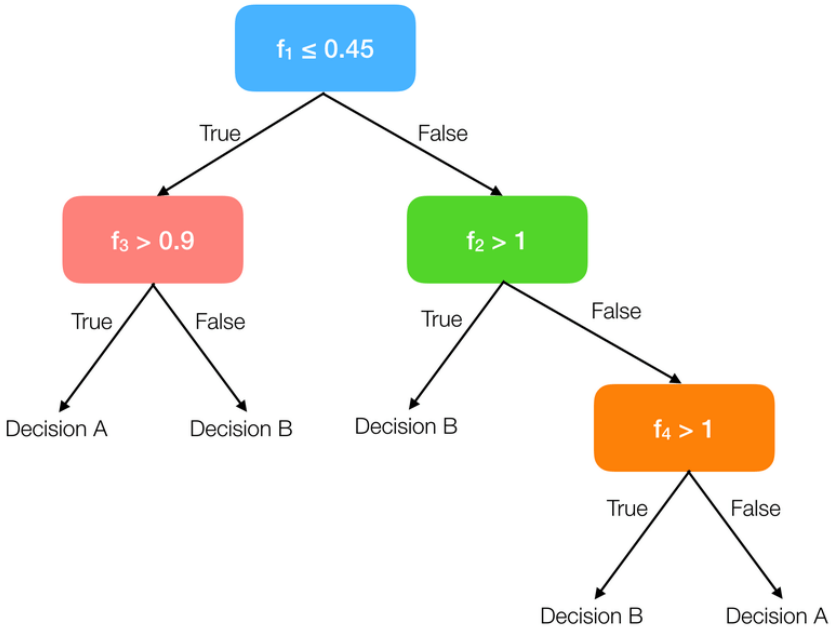
$$\text{Information Gain}(\mathrm{D}, \mathrm{A}) = \text{Entropy}(\mathrm{D}) - \sum_{v \in Values(A)} \frac{|D_\mathrm{v}|}{|D|} \cdot Entropy(D_\mathrm{v}) \tag{3}$$

Bu formülde D, karar verilecek veri kümesini, A, göz önündeki özniteliği, D_n, öznitelik A'nın v değerine sahip alt kümesini ve Entropy(D), kümenin bilgi içerik miktarını temsil eder. Entropi, Shannon'ın bilgi teorisine dayanır. Eşitlik (4)'deki şekilde hesaplanır. [33]

$$\text{Entropy}(\mathrm{D}) = -\sum_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{p_i} \log_2(\mathrm{p_i}) \tag{4}$$

Karar ağaçları, encoding türlerinin sınıflandırılması bağlamında karakter dizilerinden türetilmiş özniteliklerin ayrıştırılmasında etkili bir rol üstlenebilir. Örneğin, belirli bir karakterin varlığı, karakter sayısının belli bir eşik değerinin altında ya da üstünde olması gibi ayrımlar, karar ağacı yapısı içinde doğrudan kurala dönüştürülebilir. Özellikle simgesel ve ayrık veri türlerinde karar ağaçlarının açıklanabilirlik düzeyi yüksek olduğundan, elde edilen sınıflandırma sonuçlarının gerekçeleri de kolaylıkla analiz edilebilir.

Algoritmanın temel avantajları arasında hızlı eğitim süreci, eksik veriyle çalışma yeteneği ve sınıf etiketlerini üretici (generative) değil doğrudan karar verici (discriminative) biçimde ele alması yer alır. Bununla birlikte, karar ağaçları eğilimli veri setlerinde aşırı öğrenmeye (overfitting) açık olabilir. Bu tür durumlarda budama (pruning) teknikleri veya topluluk yöntemleri (örneğin Random Forest) ile modelin genellenebilirliği artırılabilir. [32], [33] Karar ağacı algoritmasının nasıl çalıştığına dair görsel Şekil 3.7'de verilmiştir. [34]



Şekil 3.7 Karar Ağacı Algoritması

3.5. Derin Öğrenme Tabanlı Modelleme Yöntemleri

Geleneksel makine öğrenmesi yöntemleri sınıflandırma görevlerinde yüksek başarı oranları sunsa da başarıları büyük ölçüde öznitelik mühendisliğinin kalitesine bağlıdır. Manuel olarak seçilen özniteliklerin veriyi yeterince temsil edememesi durumunda, modelin performansı yaklaşık olarak sınırlı kalabilir. Bu noktada, derin öğrenme yaklaşımları özellikle yüksek boyutlu, yapısal olmayan ve örüntü temelli veriler üzerinde güçlü bir alternatif olarak öne çıkar. Derin öğrenme, çok katmanlı sinir ağları aracılığıyla veriden doğrudan yüksek seviyeli temsiller ve ayırt edici öznitelikler öğrenerek, manuel müdahale gereksinimini en aza indirir. [35], [36]

Metin, görüntü, ses ve sinyal gibi veri türlerinin otomatik öznitelik çıkarımıyla başarılı analizleri, derin öğrenme modellerinin temsili ve öğrenme kabiliyeti sayesinde mümkün olmaktadır. [35] Encode edilmiş veriler de içerdiği karakteristik yapılar, sembolik dağılımlar ve istatistiksel örüntüler nedeniyle bu türden yapısal olmayan veri kümelerine dahildir.

Bu bağlamda, yalnızca makine öğrenmesi yöntemlerinin doğruluk ve hız avantajları değil, derin öğrenme modellerinin genellenebilirlik ve doğrusal olmayan ayrım gücü gibi üstünlükleri de sınıflandırma performansını önemli ölçüde artırabilir. [36], [37] Bu nedenle kodlanmış verilerin yapay sinir ağları tarafından temsil edilebilmesi ve sınıflandırılabilmesi amacıyla iki farklı derin öğrenme mimarisi tercih edilmiştir:

Evrişimsel Sinir Ağları (CNN) ve Residual Network 50 (ResNet50). Aşağıda bu mimarilerin yapılarına, kullanım gerekçelerine ve kodlama türlerinin ayrıştırılmasına katkılarına dair detaylı açıklamalar yer almaktadır.

3.5.1. Evrişimsel Sinir Ağları (Convoluional Networks – CNN)

Evrişimsel Sinir Ağları (Convolutional Neural Networks – CNN), derin öğrenme mimarileri arasında yer alan ve özellikle görüntü, metin ve zaman serisi gibi örüntü içeren veri yapılarında başarıyla uygulanan yapay sinir ağı modelleridir. CNN mimarileri, verilerdeki yerel bağıntıları (local patterns) yakalamak amacıyla geliştirilen evrişim (convolution) işlemleri üzerine kuruludur. Bu ağlar, giriş verileri üzerinde evrişim çekirdekleri (kernel/filter) kullanarak öznitelik haritaları oluşturur ve bu sayede hiyerarşik temsiller öğrenir. [38] CNN yapıları geleneksel yapay sinir ağlarından ayrılarak verinin yapısını koruyarak boyutlarına göre özellikler çıkarmaya çalışır. Genellikle bir CNN modelinde; evrişim katmanı, aktivasyon fonksiyonu, havuzlama katmanı ve tam bağlı katmanlar bulunur.

Evrişim katmanları veriden öznitelik haritaları çıkarırken, havuzlama katmanları boyut azaltma ve gürültü azaltma işlevini üstlenir. Sonunda tam bağlı katmanlar elde edilen temsillerin sınıflandırma görevine uygun hale getirilmesi sağlanır.

Evrişimsel Sinir Ağları (Convolutional Neural Networks – CNN), çok katmanlı yapay sinir ağları arasında yer alan ve özellikle uzamsal düzen içeren verilerde yaygın olarak kullanılan bir derin öğrenme mimarisidir. Bu yapılar, verilerdeki yerel bağıntıları yakalayarak, hiyerarşik temsiller öğrenmeye olanak tanır. CNN, klasik tam bağlı (fully connected) ağlardan farklı olarak, giriş verisine uygulanan evrişim işlemi ile yerel özellik çıkarımı yapar.

CNN mimarisinin temel bileşeni evrişim katmanıdır. Bu katmanda, giriş verisi X üzerine küçük boyutlu filtreler (çekirdekler) uygulanarak öznitelik haritaları (feature maps) elde edilir. Bir giriş tensörü ile bir filtre arasındaki evrişim işlemi, Eşitlik (5)'de gösterildiği gibi tanımlanır. [39]

$$S(i,j) = (X * K)(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X(i+m, j+n) \cdot K(m,n) \quad (5)$$

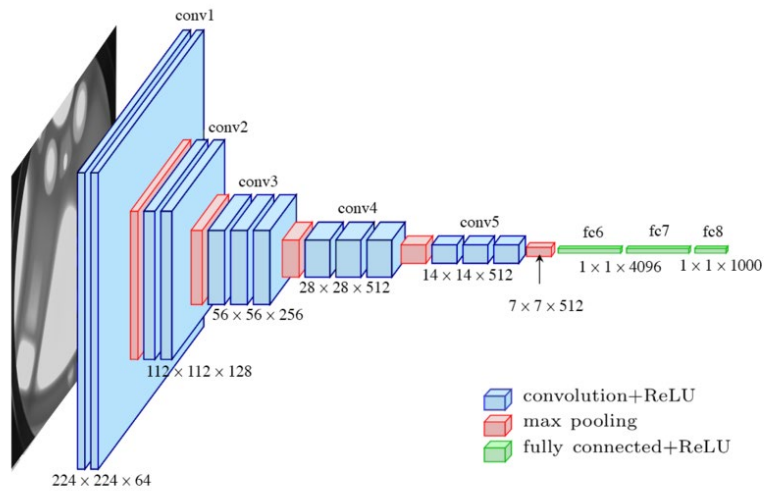
CNN yapılarında yaygın olarak kullanılan aktivasyon fonksiyonu ReLU (Rectified Linear Unit), doğrusal olmayanlık katmak için tercih edilir ve aşağıdaki Eşitlik (6)'da gösterildiği gibi tanımlanır. [40]

$$f(x) = \max(0, x) \tag{6}$$

Aktivasyon işlemi sonrasında, genellikle havuzlama (pooling) işlemi uygulanır. Havuzlama, verinin boyutunu küçültmek, gürültüyü azaltmak ve modelin genelleme kapasitesini artırmak amacıyla yapılır. En yaygın yöntem olan max-pooling işlemi, belirli bir pencere (örneğin 2×2) içindeki en büyük değeri seçerek, uzamsal çözünürlüğü azaltır. Evrişim ve havuzlama katmanlarından elde edilen öznitelik haritaları, sınıflandırma işlemi için tam bağlı katmanlara aktarılır. Son sınıflandırma katmanında, softmax aktivasyon fonksiyonu uygulanarak olasılık dağılımı üretilir. Çok sınıflı sınıflandırma problemleri için yaygın olarak kullanılan softmax fonksiyonu, Eşitlik (7)'de gösterildiği gibi tanımlanır. [41]

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, \ldots, K \tag{7}$$

Burada z_n, i-inci sınıfa karşılık gelen skor değerini, K ise toplam sınıf sayısını ifade eder. [41] CNN mimarisinin temel bileşenleri olan evrişim katmanları, aktivasyon fonksiyonları, havuzlama işlemleri ve tam bağlantılı katmanlar arasındaki bilgi akışı Şekil 3.8'de görsel olarak özetlenmiştir.



Şekil 3.8 CNN Modeli [42]

CNN mimarisi sayesinde, metin örnekleri içerisindeki kısa ve yerel örüntülerin otomatik olarak çıkarılması ve sınıflandırma sürecine dâhil edilmesi hedeflenmiştir.

Uygulama, encoding yapılarının karakteristik formasyonlarını (örneğin, Base64'te sıkça tekrar eden '=', '/', '+' gibi karakterler) modelin otomatik öğrenmesine olanak sağlamıştır. Veri seti, Encoded_Text alanındaki metinlerden oluşmakta ve bu metinler Tokenizer sınıfı yardımıyla karakter düzeyinde tokenize edilmiştir. Oluşturulan diziler, en uzun metin baz alınarak pad_sequences fonksiyonu ile sabit uzunluğa getirilmiştir. Böylece giriş vektörleri hem dizi haline dönüştürülmüş hem de CNN yapısına uygun hale getirilmiştir. Etiketler LabelEncoder kullanılarak sayısallaştırılmış ve çok sınıflı sınıflandırmaya uygun hale getirmek amacıyla to_categorical yöntemiyle encode edilmiştir.

Model mimarisi Keras Functional API kullanılarak oluşturulmuştur. Giriş katmanını takiben bir Embedding katmanı yer almakta ve karakter dizilerini vektör temsiline dönüştürmektedir. Bu katmanın çıktısı, 3 farklı kernel boyutunda (3, 5, 7) yapılandırılmış üç ayrı Conv1D katmanına aynı anda verilmiştir. Bu katmanlar metindeki kısa örüntüleri farklı uzunluklarda yakalamaya yönelik tasarlanmıştır. Her Conv1D katmanının çıktısı, GlobalMaxPooling1D işlemiyle boyut indirgenerek özetlenmiştir. Ardından bu üç vektör birleştirilmiş ve Dropout ile aşırı öğrenme (overfitting) riski azaltılmıştır.

Tam bağlantılı (Dense) katmanlarla öğrenme derinleştirilmiş ve çıkış katmanında softmax aktivasyonu kullanılarak sınıf olasılıkları tahmin edilmiştir.

Model categorical_crossentropy kayıp fonksiyonu ve adam optimizasyon algoritması ile derlenmiştir. Eğitim süreci toplam 10 epoch, 128'lik batch size ile gerçekleştirilmiş ve doğrulama verisi olarak eğitim verisinin %10'u ayrılmıştır. Model eğitimi boyunca validation_accuracy değerleri istikrarlı artış göstermiştir.

Encode edilmiş verilerin sınıflandırılmasında CNN mimarisi; karakter dizileri içerisindeki örüntüleri, sembol dağılımlarını ve yapısal farkları öğrenebilen güçlü bir model olarak kullanılmıştır. Kodlanmış diziler sayısal forma dönüştürülerek sabit boyutlu tensörlere çevrilmiş, bu tensörler CNN'e giriş olarak verilmiştir. Modelin evrişim filtreleri sayesinde, kodlama türlerine özgü sembolik ve yapısal kalıpların otomatik olarak öğrenilmesi sağlanmıştır.

Özellikle Base64, Hex ve Ascii85 gibi encoding türleri, karakter düzeyinde sembolik yapılardan oluştuğundan, bu dizilerdeki karakter gruplarının sıklığı, dizilim örüntüsü ve bağlamsal geçişleri CNN tarafından etkili biçimde temsil edilebilir.

Özellikle karakter düzeyindeki n-gram benzeri yapıların öğrenilmesi açısından, CNN mimarileri, geleneksel tam bağlı sinir ağlarına göre çok daha başarılıdır. [43] CNN'in bir diğer önemli avantajı, el ile öznitelik çıkarımı yapılmadan doğrudan veri üzerinden ayrıştırıcı temsiller öğrenebilmesidir. [37] Encode türleri gibi istatistiksel ve yapısal örüntülere dayalı sınıflandırma problemlerinde, manuel öznitelik belirleme süreci hem zaman alıcıdır hem de model başarımını sınırlayabilir.

CNN, verinin bu tür örüntüsel yapısını kendiliğinden öğrenebildiğinden geleneksel makine öğrenmesi algoritmalarına göre daha esnek ve genellenebilir bir çözüm sunar. CNN, giriş verisinde pozisyon bilgisine duyarlı çalışarak, aynı sembollerin farklı konumlardaki anlam farklılıklarını da göz önünde bulundurabilir.

Örneğin, Base64 kodlamasında padding karakteri olan "=" işaretinin dizinin sonunda bulunması anlamlı bir örüntüdür. Benzer şekilde, Hex kodlaması yalnızca belirli sembolleri içerirken, Ascii85 daha geniş bir karakter aralığına sahiptir. Bu örüntüler yalnızca sembollerin varlığıyla değil, aynı zamanda konumsal ilişkileriyle de ayırt edilebilir hale gelir. CNN'in çok katmanlı yapısı bu tür düşük seviyeli örüntüleri üst seviyeli temsillere dönüştürerek, farklı kodlama türlerinin doğru sınıflandırılmasına olanak tanır. [37] Görsel tanıma alanında yaygın olarak kullanılmasına rağmen, CNN mimarilerinin metin veya dizisel veriler üzerinde de başarılı performans gösterdiği çeşitli çalışmalarda kanıtlanmıştır. Özellikle karakter tabanlı CNN uygulamaları, doğal dil işleme, spam tespiti ve biyoinformatik gibi alanlarda etkili biçimde uygulanmıştır. [44], [45] Encode edilmiş veriler gibi sembolik ve yapısal özellikler taşıyan karakter dizilerinin de CNN ile işlenmeye uygun olduğu kabul edilmektedir.

Sonuç olarak, CNN mimarisi; sembol dizileri içinde yer alan yerel ve küresel örüntüleri öğrenme yeteneği, öznitelik mühendisliği ihtiyacını ortadan kaldırması ve yapısal farklılıkları çok katmanlı olarak modelleyebilmesiyle, encode türlerinin ayrıştırılması amacıyla uygun ve güçlü bir seçenek olarak tercih edilmiştir.

3.5.2. Artık Öğrenme (Residual Network 50 – ResNet50)

Derin yapay sinir ağları, çok katmanlı yapıları sayesinde yüksek seviyeli soyut temsiller öğrenebilme kapasitesine sahiptir. Ancak katman sayısı arttıkça modelin performansı her zaman doğrusal biçimde artmaz. Belirli bir derinlikten sonra gradyan sönümlenmesi (vanishing gradient) ve aşırı uyum (overfitting) gibi sorunlar gözlemlenebilir.

Bu probleme çözüm olarak geliştirilen Residual Network (ResNet) mimarisi, ağın öğrenmesini kolaylaştırmak amacıyla "artık öğrenme (residual learning)" adı verilen bir yöntemi benimsemiştir. ResNet mimarisinde, öğrenilmesi zor olan doğrudan temsiller yerine, daha kolay optimize edilebilen fark temelli temsiller öğrenilir. [46]

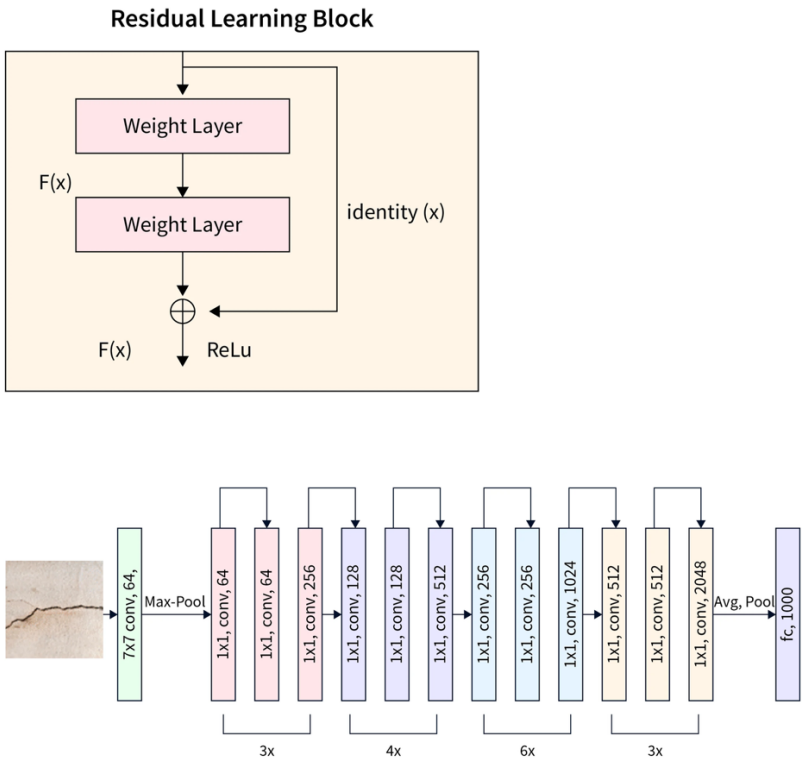
ResNet mimarisinin temelinde yer alan residual bloklar, standart katmanlara ek olarak doğrudan bağlantılar (skip connections) içerir. Bu yapılar sayesinde, önceki katmandan gelen giriş verisi, birkaç katman sonra yeniden ağın içine doğrudan aktarılır. Doğrudan geçişler sayesinde, gradyanlar ağ boyunca daha kolay geri yayılabilir; bu da çok derin ağlarda bile eğitimin kararlı biçimde gerçekleşmesine olanak tanır.

ResNet50, bu mimarinin 50 katmanlı bir versiyonudur ve özellikle görüntü sınıflandırma görevlerinde yaygın olarak kullanılmaktadır. [46], [47]

Bu bağlantılar sayesinde giriş, birkaç katman sonrasına doğrudan iletilerek, öğrenilmesi gereken fonksiyon doğrudan hedef değer yerine fark temelli olarak modellenir. Bu yaklaşım, modelin derinleştikçe performans kaybı yaşamasını engeller ve daha kararlı gradyan akışı sağlar. Bir residual bloğun temel matematiksel temsili Eşitlik (8)'de gösterilmiştir. [48]

$$y = \mathcal{F}(x, \{W_i\}) + x \tag{8}$$

Şekil 3.9'da ResNet-50 mimarisi iki yönlü olarak görselleştirilmiştir. Üstte yer alan "Residual Learning Block" diyagramı, ağı derinleştirirken bilgi kaybını önlemek amacıyla kullanılan artık bağlantıların (skip connections) nasıl çalıştığını göstermektedir. Alt bölümde, modelin tam mimarisi; giriş evrişim katmanı, dört farklı aşamada gruplanmış 1×1 ve 3×3 evrişimli blokları, global ortalama havuzlama ve tam bağlantılı katman sıralamasıyla sunulmuştur. [49]



Şekil 3.9 ResNet50 Modeli [49]

ResNet50 özellikle farklı encoding türlerinin taşıdığı sembolik ve yapısal örüntüleri derin katmanlar aracılığıyla ayırt edebilme yeteneği nedeniyle tercih edilmiştir. Base64, Base32, Hex ve Ascii85 gibi kodlama türleri, belirli karakter kümeleriyle sınırlı olmalarına rağmen, bu karakterlerin konumsal ilişkileri, uzunluk düzenleri ve sembol geçiş yapıları açısından farklılık göstermektedir. ResNet50, bu farklılıkları yalnızca yerel filtrelerle değil, aynı zamanda artık öğrenme blokları sayesinde çok seviyeli temsiller aracılığıyla modelleyebilmektedir. ResNet50'nin residual blokları, bu farkları katmanlar boyunca koruyarak ve derinleştirerek temsil etme yeteneği sunar. [50] Encoding türlerinin sınıflandırılması gibi örüntü tanıma temelli problemler, yüzeysel sınırlı özelliklerin ötesine geçerek verinin soyut düzeydeki yapısını analiz etmeyi gerektirir. ResNet50'nin 50 katmanlı mimarisi, çok seviyeli temsil öğrenimi gerçekleştirme kabiliyetiyle bu ihtiyacı karşılamakta; residual bağlantılar sayesinde öğrenme sürecinde bilgi kaybını önleyerek ayrım gücü yüksek bir modelleme sunmaktadır. [51]

ResNet50 modelinin uygulanmasındaki yapıda, veri setindeki Encoded_Text alanındaki her bir örnek, ilk 300 karakteri alınarak bir görsel haline dönüştürülmüştür.

Görsel oluşturma sürecinde, her metin beyaz arka plan üzerine siyah metin olarak yazılmış ve 224×224 piksel boyutunda gri tonlamalı resimler elde edilmiştir. Bu işlem sonucunda, her sınıfa ait veriler ilgili dizinlere ayrılarak flow_from_directory yöntemine uygun bir dosya yapısı oluşturulmuştur. Veriler, ImageDataGenerator sınıfı ile normalize edilmiş (rescale=1./255) ve %80 eğitim, %20 doğrulama olmak üzere ayrılmıştır.

Model mimarisi, imagenet ağırlıkları ile önceden eğitilmiş ResNet50 temel ağı kullanılarak inşa edilmiştir. include_top=False parametresi ile son sınıflandırma katmanı çıkarılmış ve yalnızca özellik çıkarıcı katmanlar kullanılmıştır. Bu katmanların çıktısı, bir GlobalAveragePooling2D katmanına aktarılmış; ardından 128 nöronlu, ReLU aktivasyonlu bir Dense katman eklenmiş; en sonda ise sınıf sayısı kadar nöron içeren softmax aktivasyonlu çıkış katmanı yapılandırılmıştır. Modelin eğitimi sırasında temel ResNet50 katmanları dondurularak sadece üst katmanlar eğitilmiş; böylece transfer öğrenme uygulanmıştır.

Model adam optimizasyon algoritması ve categorical_crossentropy kayıp fonksiyonu ile derlenmiş, eğitim boyunca EarlyStopping stratejisi kullanılarak aşırı öğrenme engellenmiştir. Eğitim süreci 10 epoch boyunca yürütülmüş ve doğrulama verileri üzerinde modelin başarımı izlenmiştir

ResNet50 mimarisi; encode edilmiş verilerin yapısal örüntülerini yüksek doğrulukla öğrenebilmesi, derin katmanlı temsiller oluşturabilmesi ve residual yapısı sayesinde sınıflar arası farkları koruyabilmesi nedeniyle, bu çalışmada tercih edilmiştir. Ayrıca modelin transfer öğrenmeye açık yapısı, sınırlı veri ile gerçekleştirilen deneylerde dahi etkili sonuçlar alınmasına olanak tanımıştır. [52]

Bu tez çalışmasında ResNet50 mimarisi, encoding türlerinin sınıflandırılmasına yönelik olarak görselleştirilmiş karakter dizileri üzerinden uygulanmıştır. Encode edilmiş metin dizileri, öncelikle sabit boyutlu görüntülere dönüştürülerek, her bir karakterin piksel temelli temsili elde edilmiştir. Böylece klasik metin verilerinde karşılaşılan anlamsal yoksunluk problemi aşılmış; encoding türlerinin taşıdığı yapısal örüntüler, evrişimsel filtreler aracılığıyla modellenebilir hale getirilmiştir. ResNet50 bu süreçte yalnızca düşük seviyeli sembolik farklılıkları değil, aynı zamanda derin katmanlar boyunca taşınan konumsal ve biçimsel örüntü farklarını da başarıyla öğrenmiştir. Özellikle Base64 ile Hex gibi yüzeysel olarak benzer görünen encoding türleri arasındaki ayrımlar, residual bloklar sayesinde ağ boyunca korunmuş ve ayrıştırma başarımı artırılmıştır.

Bu yaklaşım, literatürde karşılaşılan iki temel soruna çözüm sunmuştur. Birincisi, dilsel ya da istatistiksel bağlam taşımayan, anlamdan yoksun karakter dizilerinin analiz edilebilirliğini sağlamak. İkincisi ise, çok katmanlı ya da iç içe geçmiş encoding zincirlerinde türlerin ayırt edilebilmesini mümkün kılmak olmuştur. Transfer öğrenme stratejisiyle önceden eğitilmiş ağırlıkların kullanılması, sınırlı veri koşullarında dahi güçlü genelleme yeteneği sağlamış; modelin eğitim süresini azaltarak overfitting riskini düşürmüştür. Sonuç olarak ResNet50, encoding türlerinin derin temsillerle sınıflandırılması konusunda bu çalışmanın önerdiği çözümün merkezinde yer almış ve yüksek doğruluk oranlarıyla yöntemin etkinliğini deneysel olarak kanıtlamıştır.

4. DENEYLER

4.1. Ortam Kurulumu ve Teknik Altyapı

Bu çalışmada gerçekleştirilen deneysel uygulamalar Python 3.10 programlama dili kullanılarak geliştirilmiştir. Kodlama ve modelleme süreçleri, Jupyter Notebook ortamında yürütülmüştür. Derin öğrenme modellerinin hesaplama açısından yoğun işlemler gerektirmesi nedeniyle, bu süreçlerde Google Colab platformu tercih edilmiştir. Colab'ın sağladığı GPU desteği sayesinde, özellikle CNN ve ResNet50 gibi derin öğrenme mimarilerinin eğitim süreleri azaltılmıştır.

Veri işleme, görselleştirme, modelleme ve değerlendirme adımlarında çeşitli Python kütüphanelerinden yararlanılmıştır. Veri yükleme, dönüştürme ve analiz işlemleri için pandas, numpy, random ve collections kütüphaneleri kullanılmıştır. Elde edilen sonuçların grafiksel olarak yorumlanması amacıyla matplotlib, seaborn, tqdm ve PIL kütüphaneleri tercih edilmiştir. Model eğitimi ve test süreçlerinde ise, geleneksel makine öğrenmesi algoritmaları için scikit-learn kullanılmıştır. Derin öğrenme mimarileri için tensorflow.keras kütüphanesi kullanılmıştır. Tüm kütüphaneler pip ve conda aracılığıyla kurularak çalışma süresince bütünleşmiş biçimde çalıştırılmıştır.

4.2. Veri Setinin İçeriği ve Yapısı

Çalışmada kullanılan veri seti, Windows ve macOS işletim sistemlerinden alınan sistem loglarından oluşmuştur. Veri seti, bu tez kapsamında özgün olarak hazırlanmış, herhangi bir hazır veri kaynağından temin edilmemiştir. Bu veri setinin oluşturulmasındaki temel amaç, sistem loglarının doğası gereği uzun ve düzensiz bir yapıya sahip olması nedeniyle, makine öğrenmesi ve derin öğrenme algoritmalarının bu veriler üzerinden anlamlı örüntüler çıkarmakta zorlanabileceği varsayımını test etmektir. Kodlama türlerinin sınıflandırılması problemi bu bağlamda, model performansını değerlendirmek için anlamlı bir test alanı sunmaktadır. Sistem logları; işlem tanımları, zaman damgaları, sistem çağrıları ve hata mesajları gibi çeşitli bileşenler içerir. Bu log verileri Base64, Base32, ASCII85 ve Hex olmak üzere dört farklı encoding algoritmasıyla dönüştürülerek her biri ayrı bir sınıfa karşılık gelecek şekilde etiketlenmiştir.

Veri seti iki temel sütundan oluşmaktadır:

Encode Type: Uygulanan encoding algoritmasını belirtir (Base64, Base32, ASCII85, Hex).

Encoded Text: İlgili sistem logunun belirli bir encoding yöntemiyle dönüştürülmüş hali.

Her sınıf veri setinde dengeli bir şekilde temsil edilmiştir. Ayrıca, encode edilmiş metinler üzerinde yapay bozulma işlemi uygulanmamış, bu sayede modellerin yalnızca encoding yöntemine özgü yapısal farklara odaklanarak öğrenmesi sağlanmıştır. Bu yönüyle çalışma, encoding türlerinin temel karakteristiklerine dayanarak ayırt edilebilirliğini değerlendirmeyi amaçlamaktadır.

Veri seti toplamda 140.384 örnek içermektedir ve her bir encoding türü için 35.096 adet veri örneğiyle dengeli bir sınıf dağılımı sağlanmıştır. Encoded_Text sütunundaki metinlerin uzunlukları encoding algoritmasına bağlı olarak değişiklik göstermektedir. Ortalama metin uzunluğu yaklaşık 204 karakter olarak hesaplanmıştır. Karakter uzunluğu derin öğrenme tabanlı yaklaşımlar açısından giriş boyutunun yönetilebilir seviyede kalmasını sağlamaktadır. Veri setinde 4 eksik kayıt ve 4 yinelemeli kayıt tespit edilmiş olup, bu kayıtlar ön işleme aşamasında temizlenerek analiz dışında bırakılmıştır. Geniş ve dengeli yapısı sayesinde bu veri seti, encoding türlerinin ayırt edilmesine yönelik sınıflandırma problemlerinde etkili bir değerlendirme zemini sunmaktadır.

Encoded_Text sütununda yer alan kodlanmış metinler girdi olarak kullanılmış; bu verilerden çeşitli istatistiksel öznitelikler çıkarılarak model girişine dönüştürülmüştür. Özellikle her metnin uzunluğu, sayısal karakter oranı ve alfabetik karakter oranı gibi üç temel özellik belirlenmiştir. Bu öznitelikler, encoding türleri arasında anlamlı yapısal farklılıklar gösterebildiği için sınıflandırma performansına katkı sağlamıştır.

4.3. Geleneksel Makine Öğrenmesi Modellerinin Uygulanması

4.3.1. Naive Bayes Modeli ile Sınıflandırma Deneyleri

Modelin sınıf bazlı performansı Precision, Recall ve F1-skor gibi ölçütlerle değerlendirilmiş; bu ölçütlerdeki dengesizlik, sınıflar arasında başarı farklılıklarını açıkça ortaya koymuştur. Bazı sınıflarda F1- skor değerinin 0.80'in üzerine çıktığı görülürken, bazı sınıflarda bu değerin 0.60 seviyelerine kadar gerilediği gözlemlenmiştir. Bu farklılık, verinin sınıf dağılımına, örüntü karmaşıklığına ve özniteliklerin temsil gücüne bağlanabilir. Modelin gerçek dünya koşullarına daha yakın şekilde değerlendirilebilmesi amacıyla, encode edilmiş veriler üzerinde çeşitli karakter bozulmaları uygulanarak sentetik bir “Bozuk” sınıfı oluşturulmuştur.

Bu sınıf, rastgele karakter silme, değiştirme, ekleme ve yer değiştirme işlemleriyle orijinal verilerin yapısını bozarak üretilmiştir. Böylece, modelin yalnızca doğru kodlanmış verileri sınıflandırma becerisi değil, aynı zamanda yapısal bütünlüğü bozulmuş veya geçersiz verileri tanıma yetkinliği de analiz edilmiştir.

Modelin genel performansı Makro Ortalama (Macro Average) ve Ağırlıklı Ortalama (Weighted Average) değerleriyle de değerlendirilmiştir. Makro ortalama, tüm sınıfların başarı metriklerinin eşit ağırlıkla ortalamasını alırken; ağırlıklı ortalama, her sınıfın veri kümesindeki temsil oranına göre hesaplanır. Bu bağlamda, makro ortalamanın daha düşük çıkması bazı sınıflarda modelin zayıf performans gösterdiğini; ağırlıklı ortalamanın daha yüksek olması ise modelin daha çok örneğe sahip baskın sınıflarda daha başarılı olduğunu göstermektedir. Bu fark, sınıf dengesizliğinin model üzerindeki etkisini ortaya koymaktadır.

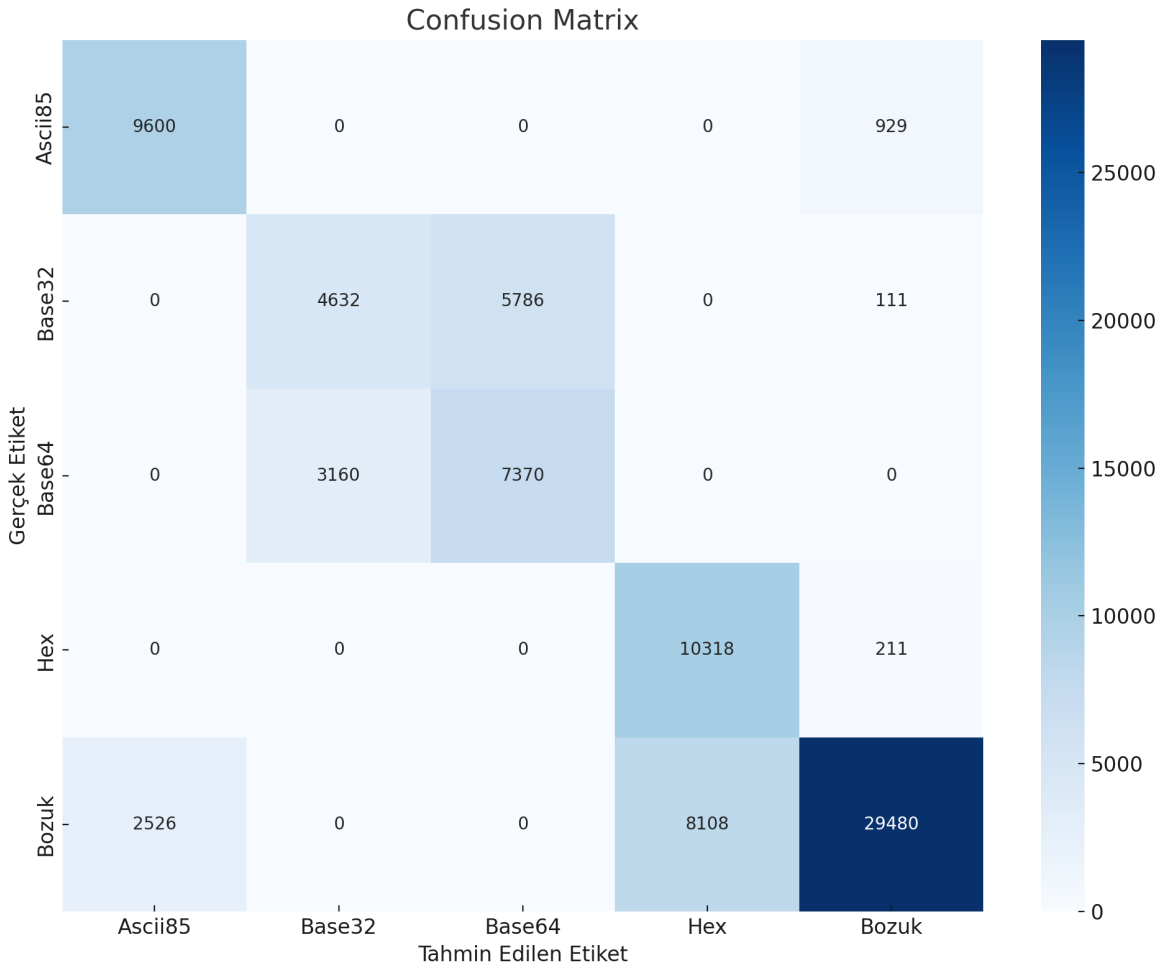
Naive Bayes modeli üzerinde yaklaşık %72 doğruluk (accuracy) değeri elde etmiştir. Bu sonuç, modelin genel olarak encoding türlerini ayırt etme konusunda makul bir başarı sağlasa da her sınıf için aynı düzeyde başarı elde edilememiştir. Özellikle Base64 ve Hex gibi yapısal açıdan benzer karakter dağılımlarına sahip encoding türleri arasında sınıflandırma hataları gözlemlenmiştir. Bu durum, modelin sınıf ayırıcı öznitelikleri yeterince derinlemesine işleyememesiyle ilişkilendirilebilir. Naive Bayes modeline ait sınıflandırma deney raporu Tablo 4.1'de gösterilmiştir.

Tablo 4.1 Naive Bayes Modeli Sınıflandırma Raporu

Sınıflandırma Raporu	**Precision**	**Recall**	**F1-Skor**	**Support**
Ascii85	0.60	0.91	0.72	10529
Base32	0.56	0.44	0.50	10529
Base64	0.94	0.70	0.80	10529
Hex	0.62	0.98	0.76	10529
Bozuk(Anomaliler)	0.94	0.70	0.80	42114
Doğruluk Oranı			0.72	84228
Makro Ortalama	0.66	0.73	0.68	84228
Ağırlıklı Ortalama	0.76	0.72	0.72	84228

Naive Bayes modeli, hızlı eğitim süreci, düşük hesaplama maliyeti ve yorumu kolay sonuçları sayesinde çalışmanın ilk aşamasında uygulanabilir bir temel sağlamıştır. Ancak, modelin özellikle karmaşık yapılı sınıflar arası ayırt edici performansının sınırlı kaldığı görülmüştür. Özellikle encoding türleri arasında ortak karakter yapısı bulunan sınıflarda, modelin belirleyiciliği düşmektedir. Bu da Naive Bayes algoritmasının "özellikler bağımsızdır" varsayımının bu problemde tam olarak karşılanmamıştır.

Modelin sınıf bazlı performansını incelemek amacıyla, tahmin edilen etiketler ile gerçek etiketlerin karşılaştırıldığı bir Karmaşıklık Matrisi (Confusion Matrix) oluşturulmuştur. Görsel olarak Şekil4.1'de sunulan matriste satırlar gerçek sınıfları, sütunlar ise modelin tahmin ettiği sınıfları temsil etmektedir. Ana köşegende (diagonal) yer alan yüksek değerler modelin doğru sınıflandırma oranlarını, köşegen dışı (off-diagonal) değerler ise hatalı sınıflandırmaları ifade etmektedir.



Şekil 4.1 Naive Bayes Modeli Karmaşıklık Matrisi

4.3.2. K-En Yakın Komşu (K-Nearest Neighbors – KNN) ile Sınıflandırma Deneyleri

KNN modeli, örnekler arasındaki benzerlik temelli mesafe ölçümüne dayandığı için, özellikle yapısal farklılıklar barındıran metin verilerinde etkili olabileceği öngörülmüştür. Bu öznitelikler, encoding türlerinin karakteristik dağılımlarını yansıtma potansiyeli taşıdığı için doğrudan model girdi kümesini oluşturmuştur.

KNN modelinin uygulanmasında scikit-learn kütüphanesi kullanılmış ve en başarılı sonuçlar k = 5 komşu değeriyle elde edilmiştir. Veriler, model eğitimi öncesinde StandardScaler ile ölçeklendirilmiş ve öznitelikler arasındaki büyüklük farklarının sınıflandırma üzerindeki etkisi minimize edilmiştir. Ayrıca, veriyi görselleştirmek ve daha anlaşılır hale getirmek amacıyla PCA (Principal Component Analysis) ile boyut indirgeme işlemi uygulanmış, bu sayede veri dağılımı görsel olarak da incelenmiştir.

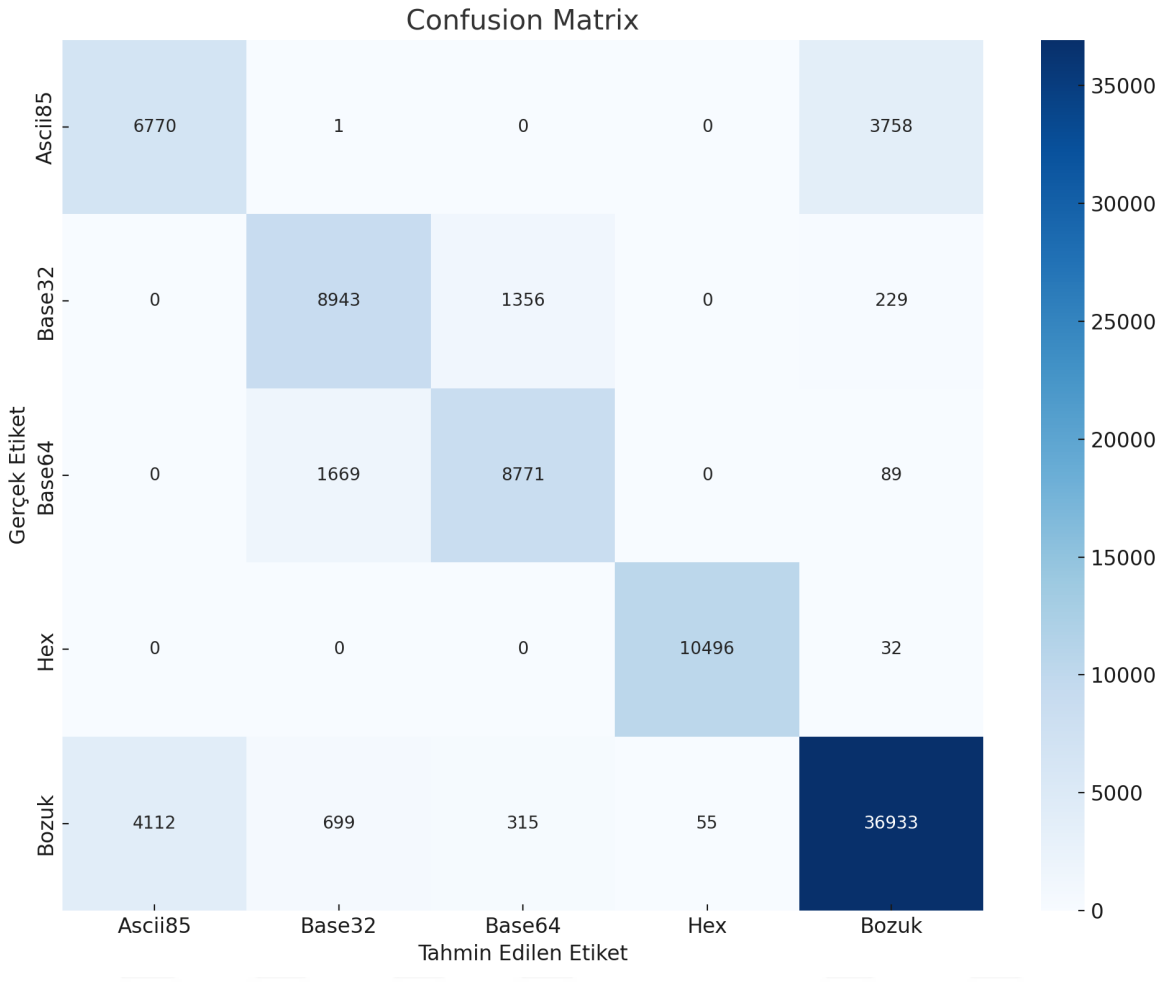
Elde edilen sonuçlara göre, KNN modeli üzerinde yaklaşık %85 doğruluk oranına ulaşmıştır. Sınıflandırma raporunda yer alan F1- skor değerleri ise sınıflar arasında farklılık göstermekle birlikte çoğunlukla %75 ila %85 aralığında seyretmiştir. Özellikle Base64 encoding türünde daha yüksek başarı oranları gözlemlenmiş; buna karşın yapısal benzerlik gösteren sınıflar arasında sınırlı düzeyde karışıklık meydana gelmiştir. Makro ortalama F1 skoru 0.83 olarak hesaplanmıştır. Bu değer, her sınıfın eşit ağırlıkta değerlendirildiği durumda modelin ortalama başarısını temsil etmektedir. Bu açıdan bakıldığında, modelin tüm sınıflarda genel olarak tutarlı bir performans gösterdiği söylenebilir. Öte yandan, ağırlıklı ortalama (weighted average) F1 skoru 0.85 olup, bu metrik her sınıfın veri kümesindeki temsil oranına göre ağırlıklandırılmış bir ortalama sunmaktadır. KNN modeline ait sınıflandırma deneyi raporu Tablo 4.2'de gösterilmiştir.

Tablo 4.2 KNN Modeli Sınıflandırma Raporu

Sınıflandırma Raporu	Precision	Recall	F1-Skor	Support
Ascii85	0.62	0.64	0.63	10529
Base32	0.79	0.85	0.82	10529
Base64	0.84	0.83	0.84	10529
Hex	0.99	1.00	1.00	10529
Bozuk(Anomaliler)	0.90	0.88	0.89	42114
Doğruluk Oranı			0.85	84228
Makro Ortalama	0.83	0.84	0.83	84228
Ağırlıklı Ortalama	0.86	0.86	0.85	84228

Modelin sınıf bazlı başarımını analiz eden Karmaşıklık Matrisinin (Confusion Matrix) ana köşegenindeki yüksek değerler, ilgili sınıfların doğru sınıflandırıldığını; köşegen dışı hücrelerdeki sayılar ise diğer sınıflarla karıştırılma oranlarını göstermektedir. Analiz sonucunda, özellikle "Hex" ve "Bozuk" sınıflarında yüksek doğru tahmin oranları gözlemlenmiştir. Bu durum, bu sınıfların karakteristik özelliklerinin KNN algoritması tarafından başarılı bir şekilde ayrıştırılabildiğini göstermektedir.

Buna karşın, "Base32" ve "Base64" gibi yapısal olarak birbirine yakın encoding türlerinin daha sık karıştırıldığı ve bu nedenle bu sınıflar arasındaki hata oranlarının arttığı dikkat çekmektedir. Bu gözlem, KNN'nin benzer özelliklere sahip sınıflar arasında mesafe temelli ayrım yapmakta zorlandığını ortaya koymaktadır. Karmaşıklık Matrisi KNN modelinin hangi sınıflarda yüksek doğruluk sağladığını, hangi sınıflarda hata yaptığını sistematik biçimde ortaya koyarak modelin performansını sınıf bazlı değerlendirme açısından önemli katkı sunmuştur. KNN Karmaşıklık Matrisi (Confusion Matrix) Şekil 4.2'de sunulmuştur.



Şekil 4.2 KNN Modeli Karmaşıklık Matrisi

Genel olarak değerlendirildiğinde, KNN algoritması düşük model karmaşıklığına rağmen tatmin edici sınıflandırma performansı sergilemiştir. Veri setinin genişliği ve boyutu arttıkça hesaplama süresinin artabileceği dikkate alınarak, bu algoritmanın büyük ölçekli veri kümeleri için uygunluğu sınırlı olabilir. Yine de basit uygulama yapısı ve dengeli performans düzeyi sayesinde, KNN modeli bu çalışmanın deneysel fazında değerli bir kıyas unsuru olarak işlev görmüştür.

4.3.3. Karar Ağacı (Decision Tree) ile Sınıflandırma Deneyleri

Kullanılan geleneksel makine öğrenmesi algoritmalarından biri de Karar Ağacı (Decision Tree) modelidir. Karar ağaçları, veriyi özelliklerine göre dallandırarak karar kuralları oluşturan ve sınıflandırma işlemini bu kurallar aracılığıyla gerçekleştiren sezgisel ve açıklanabilir modellerdir. Bu yönüyle, özellikle verideki yapısal ayrımları görselleştirmek ve yorumlamak açısından avantaj sunmaktadır.

Encoding türlerini temsil eden metinlerin sınıflandırılması amacıyla Karar Ağacı (Decision Tree) algoritması uygulanmış ve performansı diğer modellerle karşılaştırılmıştır.

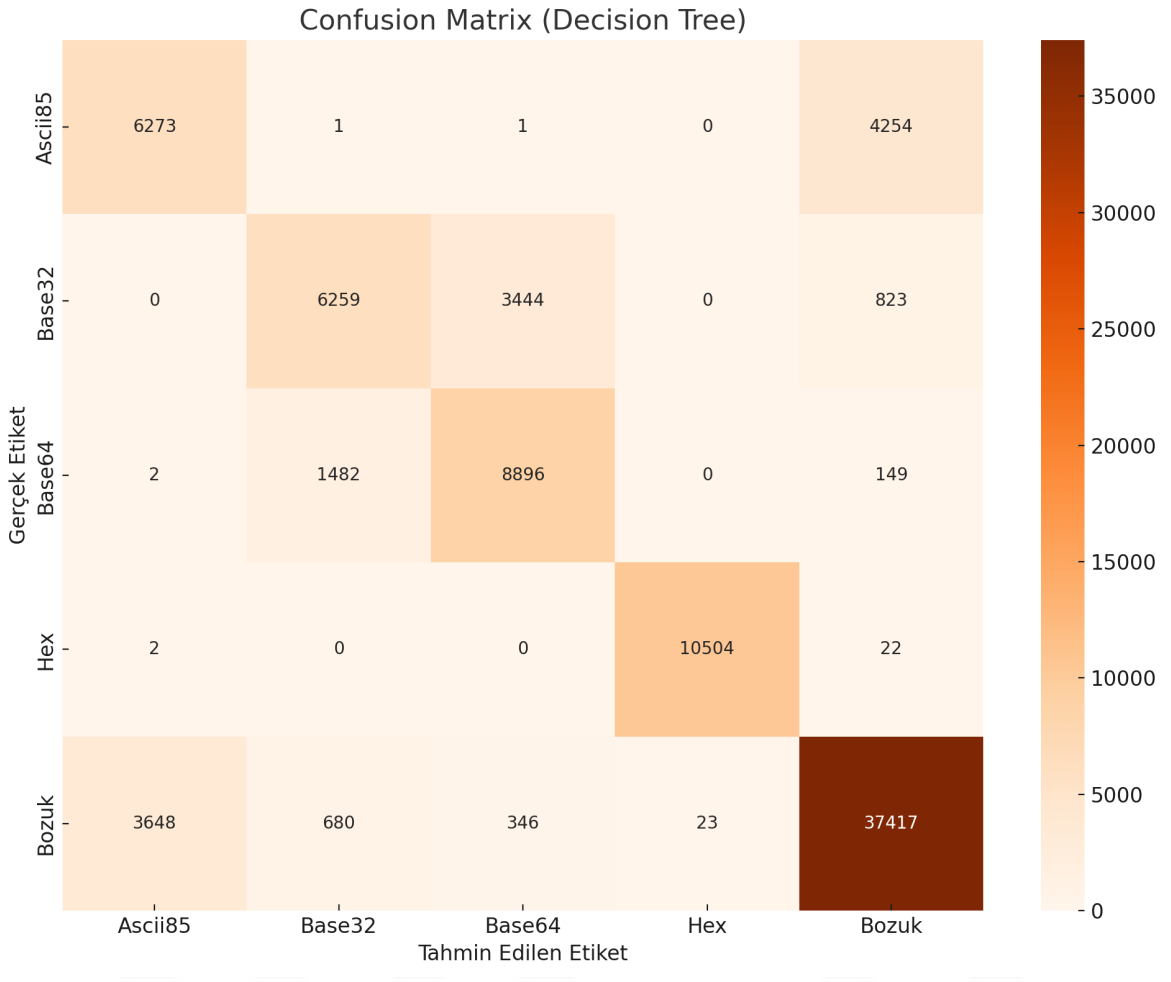
Karar ağacı modeli, scikit-learn kütüphanesindeki DecisionTreeClassifier sınıfı kullanılarak oluşturulmuştur. Modelin derinliği ve dallanma kriterleri otomatik olarak belirlenmiş, veri setinin yapısına göre en uygun dallanma biçimi oluşturulmuştur. Model eğitimi öncesinde veriler ölçeklendirme işlemine tabi tutulmamıştır çünkü karar ağaçları, özniteliklerin mutlak büyüklüğünden çok ayrıştırıcı değerlerine odaklanmaktadır. Modelin derinliği max_depth=10 parametresi ile sınırlandırılmıştır. Bu sayede, modelin aşırı öğrenmeye eğilimi azaltılarak daha dengeli ve genellenebilir bir yapı elde edilmesi amaçlanmıştır. Elde edilen Karar Ağacı modeli %82 doğruluk oranına ulaşmıştır. Modelin doğruluk oranı ve sınıf bazlı başarı metrikleri incelendiğinde, özellikle Hex ve Bozuk sınıflarında yüksek doğruluk ve F1 skoru değerlerine ulaşıldığı gözlemlenmiştir. Bu durum, bu sınıfların karakteristik özniteliklerle belirgin biçimde ayrıştırılabildiğini göstermektedir. Ancak, Ascii85 ve Base32 gibi yapısal açıdan benzer örüntülere sahip sınıflarda modelin daha fazla hata yaptığı; bu sınıflar arasında yanlış sınıflandırma oranının arttığı dikkat çekmektedir.

Karar Ağacı Modeli üzerinde yapılan analizlerde, modelin sınıf bazlı başarılarında dengesizlikler olduğu ve bu dengesizliklerin destek sayıları (support) ile doğrudan ilişkili olduğu da anlaşılmıştır. Büyük örneklemli sınıflarda (örneğin Bozuk), modelin daha tutarlı tahminler ürettiği, ancak daha az sayıda örnek içeren sınıflarda (örneğin Ascii85) karar sınırlarının yeterince ayrışamadığı gözlemlenmiştir. Karar Ağacı algoritması, sınıflar arası ayrımın net olduğu durumlarda etkili bir sınıflandırma performansı sunarken; karmaşık veya benzer yapılı sınıflarda daha derin modellerin veya farklı algoritmaların gerekliliğini ortaya koymuştur. Karar Ağacı (Decision Tree) modeline ait sınıflandırma raporu Tablo 4.3'de gösterilmiştir.

Tablo 4.3 Karar Ağacı Modeli Sınıflandırma Raporu

Sınıflandırma Raporu	Precision	Recall	F1-Skor	Support
Ascii85	0.63	0.60	0.61	10529
Base32	0.74	0.59	0.66	10529
Base64	0.70	0.84	0.77	10529
Hex	1.00	1.00	1.00	10529
Bozuk(Anomaliler)	0.88	0.89	0.88	42114
Doğruluk Oranı			0.82	84228
Makro Ortalama	0.79	0.78	0.78	84228
Ağırlıklı Ortalama	0.82	0.82	0.82	84228

Genel olarak değerlendirildiğinde, Karar Ağacı algoritması açıklanabilirlik açısından önemli avantajlar sunmuş ve girdi verisindeki yapısal farklılıkları iyi yansıtarak anlamlı sınıflandırma sonuçları üretmiştir. Ancak modelin doğası gereği aşırı öğrenmeye (overfitting) yatkın olduğu ve karmaşık yapılı verilerde genelleme kabiliyetinin sınırlı kalabileceği göz önünde bulundurulmalıdır. Bu nedenle, derin yapay sinir ağları gibi daha güçlü öğrenme kapasitelerine sahip modellerle karşılaştırıldığında, Karar Ağacı algoritmasının sınıflandırma performansı belirli sınırlara sahiptir. Karar Ağacı (Decision Tree) modeline ait Karmaşıklık Matrisi (Confusion Matrix) Şekil 4.3'de sunulmuştur.



Şekil 4.3 Karar Ağacı Modeli Karmaşıklık Matrisi

4.4. Derin Öğrenme Tabanlı Modellerin Uygulanması

4.4.1. Evrişimsel Sinir Ağları (CNN) Sınıflandırma Deneyleri

Encoding türlerinin sınıflandırılması amacıyla, karakter düzeyinde çalışan bir Evrişimsel Sinir Ağları (Convolutional Neural Networks - CNN) mimarisi uygulanmıştır. CNN mimarileri, özellikle görsel verilerle çalışmak üzere geliştirilmiş olsa da karakter düzeyinde sembolik verilerin özelliklerini öğrenme konusunda da başarılı olmuştur. Encoding türleri, belirli karakter kümeleri ve sembol dizilimlerinden oluştuğu için, bu yapılar arasında hem lokal hem de pozisyonel farklılıkların öğrenilmesi önem arz etmektedir. Evrişimsel katmanlar sayesinde model, karakter dizileri içindeki ardışık yapıları filtreleyerek her bir encoding türüne özgü örüntüleri otomatik olarak çıkarabilmektedir.

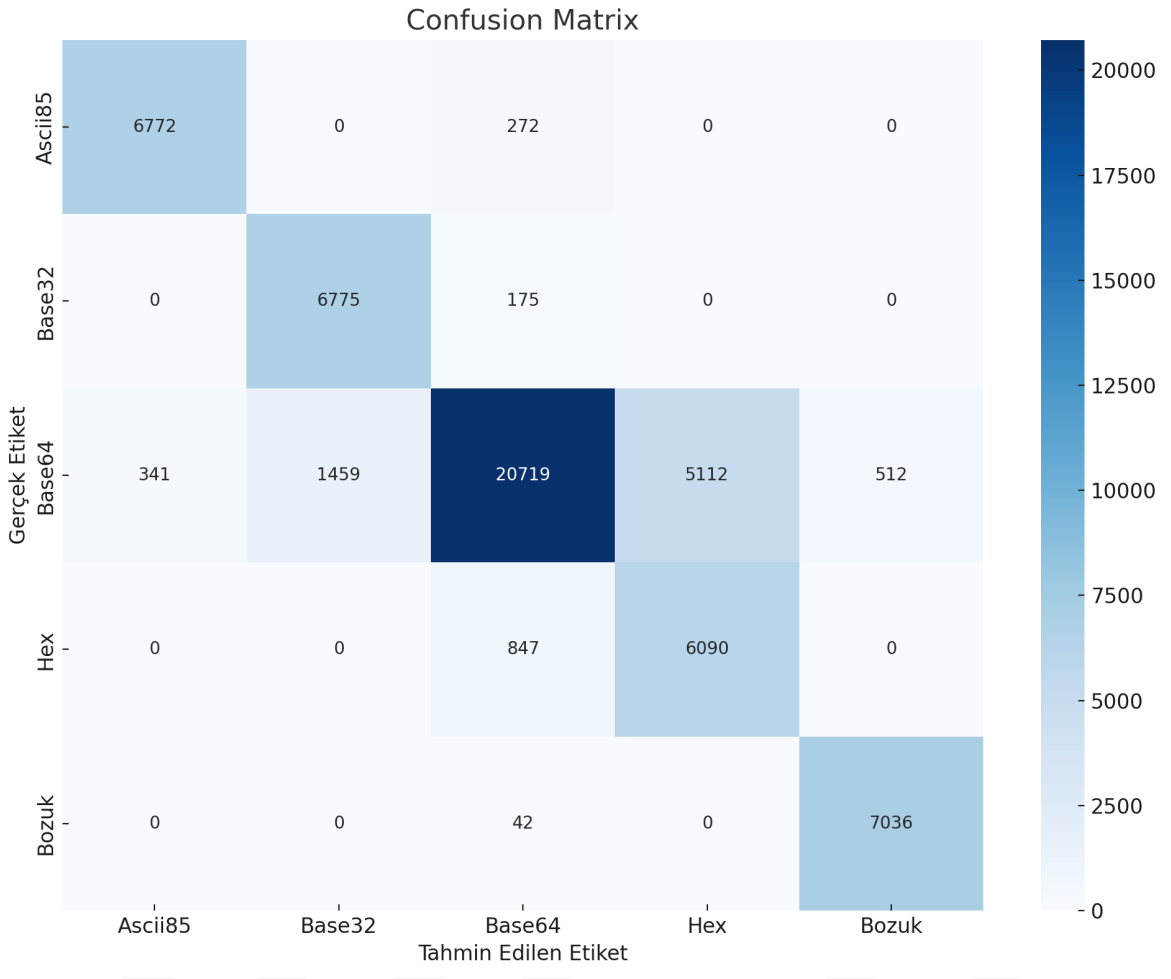
Karakter dizilerinden elde edilen vektörel temsiller üzerinde CNN uygulanması, özellikle istatistiksel ya da dilsel bağlamdan yoksun verilerde etkili bir öğrenme sağlayarak, sınıflandırma doğruluğunu artırmaktadır.

Modelin eğitimi tamamlandıktan sonra doğruluk oranı %84 olarak hesaplanmıştır. Performans değerlendirmesi için, sklearn.metrics.classification_report fonksiyonu kullanılarak Precision, Recall ve F1-skor değerleri sınıf bazında hesaplanmıştır. Modelin özellikle Base64, Binary ve Hex gibi yapıların sınıflandırılmasında yüksek başarı göstermiştir. Evrişimsel Sinir Ağları (Convolutional Neural Networks – CNN) modeline ait sınıflandırma raporu Tablo 4.4'de gösterilmiştir.

Tablo 4.4 Evrişimli Sinir Ağları (CNN) Sınıflandırma Raporu

Sınıflandırma Raporu	**Precision**	**Recall**	**F1-Skor**	**Support**
Ascii85	0.54	0.88	0.67	6937
Base32	0.95	0.96	0.96	7044
Base64	0.82	0.97	0.89	6950
Hex	0.93	0.99	0.96	7078
Bozuk(Anomaliler)	0.94	0.74	0.83	28143
Doğruluk Oranı			0.84	56152
Makro Ortalama	0.84	0.91	0.86	56152
Ağırlıklı Ortalama	0.88	0.84	0.85	56152

Modelin sınıflandırma başarımı değerlendirildiğinde yüksek doğruluk oranları elde edildiği görülmektedir. Encode sınıflarında doğruluk oranları tatmin edici düzeydedir. Sınıflar arası ayrım büyük ölçüde başarılı bir şekilde gerçekleştirilmiş olup, yalnızca bazı sınıflar arasında sınırlı düzeyde karışıklıklar gözlenmiştir. Bu da modelin genel olarak sağlam bir sınıflandırma kabiliyetine sahip olduğunu ve temel sınıflar arasında öğrenme başarısını yüksek oranda sağladığını göstermektedir. Evrişimsel Sinir Ağları (Convolutional Neural Networks – CNN) modeline ait Karmaşıklık Matrisi Şekil 4.4'de verilmiştir.



Şekil 4.4 Evrişimsel Sinir Ağları (CNN) Modeli Karmaşıklık Matrisi

Sonuç olarak, karakter düzeyinde yapılandırılmış CNN mimarisi, encoding türleri arasında bulunan yapısal örüntüleri etkili biçimde öğrenebilmiş ve sınıflandırma görevinde yüksek başarı sağlamıştır.

Özellikle karakter dizilerinin vektör temsilleri üzerinde gerçekleştirilen çok ölçekli konvolüsyon işlemleri sayesinde hem kısa hem de orta uzunluktaki örüntüler model tarafından ayırt edici biçimde yakalanabilmiştir. Uygulanan ön işleme teknikleri ile mimari düzeyde yapılan optimizasyonlar, geleneksel makine öğrenmesi algoritmalarına kıyasla önemli ölçüde daha yüksek doğruluk ve genelleme kapasitesi elde edilmesini sağlamıştır. Bu sonuçlar, CNN tabanlı yaklaşımların, örüntü temelli metin sınıflandırma problemlerinde güçlü ve ölçeklenebilir bir alternatif sunduğunu ortaya koymaktadır.

4.4.2. Artık Öğrenme (Resiual Network50) Sınıflandırma Deneyleri

Derin öğrenme yaklaşımlarının encoding türlerini sınıflandırmadaki performansını değerlendirmek amacıyla bu çalışmada transfer öğrenme yöntemiyle yapılandırılmış bir ResNet50 mimarisi uygulanmıştır.

ResNet50, 50 katmandan oluşan, artık öğrenme (residual learning) mekanizması sayesinde derin yapılarla oluşabilecek kaybolan gradyan problemini minimize eden güçlü bir konvolüsyonel sinir ağı modelidir. Bu özellikleri nedeniyle, görsel olarak ifade edilen encoding yapılarının sınıflandırılmasında tercih edilmiştir.

ResNet50 modelinde edilen doğruluk oranı %96 düzeyindedir. Sınıflandırma raporu ve karışıklık matrisi (confusion matrix) analizleri, modelin özellikle Base64, Hex ve Binary gibi yapıların yüksek doğrulukla ayırt edilebildiğini ortaya koymuştur. Base32 sınıfında da yüksek precision %97 ve recall %99 değerleri sayesinde f1-skor %98 gibi güçlü bir seviyeye ulaşmıştır.

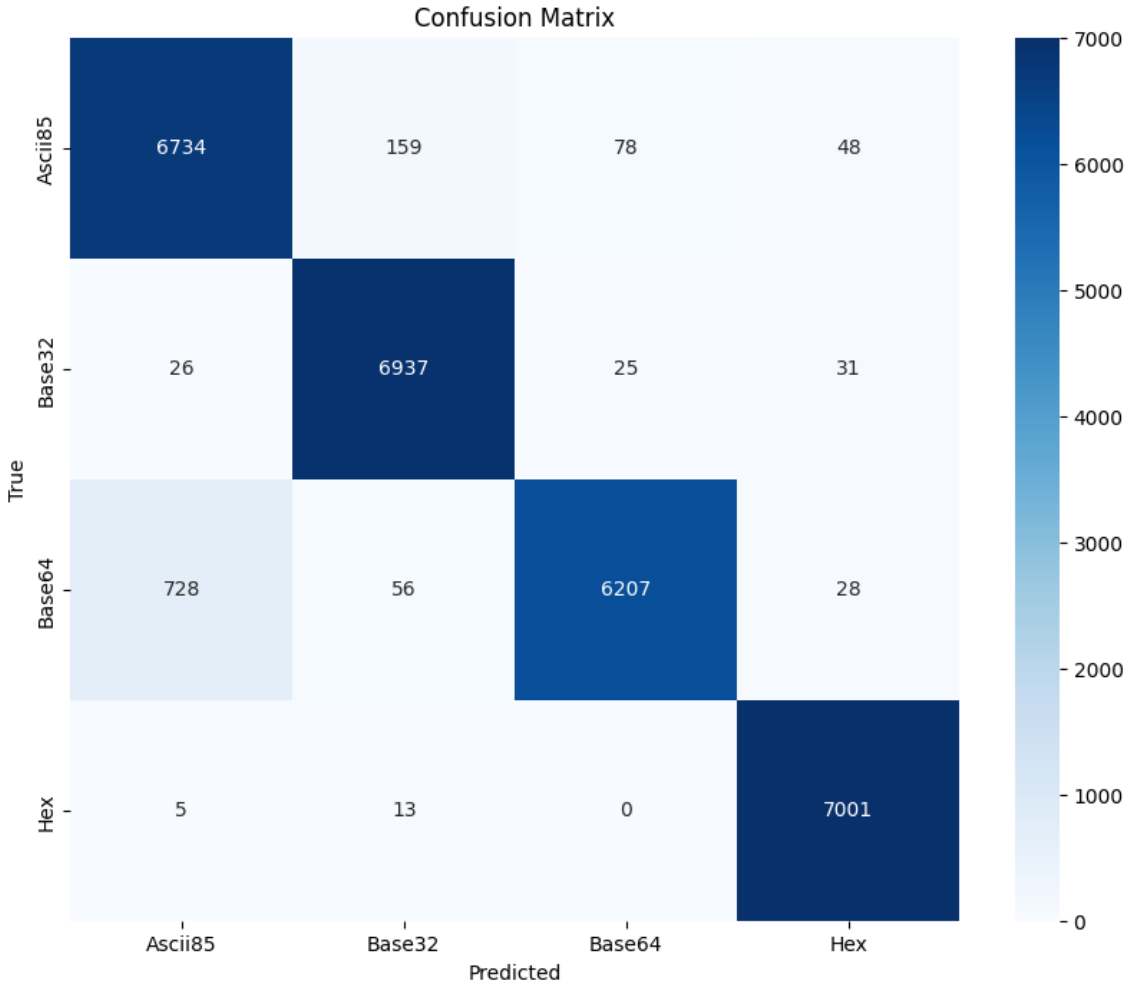
Ascii85 ve Base64 sınıflarında da genel başarı yüksek olmakla birlikte, bu sınıflarda nispeten daha düşük recall (özellikle Base64 için %88) değerleri dikkat çekmektedir. Bu durum, bazı Ascii85 ve Base64 örneklerinin diğer sınıflarla karıştığını göstermektedir. Ancak f1- skor değerlerinin her iki sınıf için de %93 düzeyinde olması, modelin bu sınıflarda da genel başarıyı koruduğunu göstermektedir. Makro ve ağırlıklı ortalama metriklerin tamamının %96 düzeyinde gerçekleşmiş olması, sınıflar arasında dengesizliğe rağmen modelin her sınıfa yüksek düzeyde dikkat gösterdiğini ifade etmektedir. Genel olarak, model; sınıflar arası ayrımı etkili bir biçimde gerçekleştirmekte olup, yüksek doğruluk ve f1- skor değerleriyle güçlü bir sınıflandırıcı olduğunu kanıtlamaktadır.

Artık Öğrenme (Resiual Network50 - ResNet50) modeline ait sınıflandırma raporu Tablo 4.5'de gösterilmiştir.

Tablo 4.5 Artık Öğrenme (ResNet50) Sınıflandırma Raporu

Sınıflandırma Raporu	Precision	Recall	F1-Skor	Support
Ascii85	0.90	0.96	0.93	7019
Base32	0.97	0.99	0.98	7019
Base64	0.99	0.88	0.93	7019
Hex	0.98	1.00	0.99	7019
Doğruluk Oranı			0.96	28076
Makro Ortalama	0.96	0.96	0.96	28076
Ağırlıklı Ortalama	0.96	0.96	0.96	28076

Modelin sınıflandırma performansı değerlendirildiğinde başarılı sonuçlar elde edildiği görülmektedir. Özellikle Base32 ve Hex sınıfları neredeyse hatasız bir şekilde sınıflandırılmıştır; Base32 için doğru tahmin sayısı 6937, Hex için ise 7001 gibi oldukça yüksek değerler elde edilmiştir. Ascii85 sınıfında da büyük oranda doğru sınıflandırma sağlanmış, yalnızca sınırlı sayıda Base32 ve Base64 sınıflarıyla karışıklık yaşanmıştır. Sınıflar arasında çapraz hataların sayıca az olması, modelin ayırt edici özellikleri doğru şekilde öğrenebildiğini ve yüksek genelleme kapasitesine sahip olduğunu göstermektedir. Bu doğrultuda, modelin hem doğruluk oranı hem de sınıf ayrım yeteneği açısından güçlü bir yapıya sahip olduğu söylenebilir. Artık Öğrenme (Resiual Network50 - ResNet50) modeline ait Karmaşıklık Matrisi Şekil 4.5'de gösterilmiştir.



Şekil 4.5 Artık Öğrenme (ResNet50) Modeli Karmaşıklık Matrisi

ResNet50 modeli, karakter dizilerinden görsel temsillere dönüştürülmüş encoding örneklerini sınıflandırma görevinde yüksek başarı göstermiştir. Özellikle modelin önceden eğitilmiş ağırlıklar üzerinden düşük seviyeli özellikleri etkili biçimde çıkarabilmesi, encoding türleri arasında bulunan görsel örüntüleri başarılı şekilde ayrıştırmasını sağlamıştır. Doğruluk oranları, ResNet50 mimarisinin genelleme kapasitesinin güçlü olduğunu ortaya koymaktadır. Bu bulgular, derin evrişimsel yapılarla desteklenen transfer öğrenme yaklaşımının, görselleştirilmiş metinsel veriler üzerinde güçlü bir sınıflandırma altyapısı sunduğunu ve encoding temelli analiz çalışmalarında uygulanabilir bir çözüm olarak değerlendirilebileceğini göstermektedir.

4.5. Modellerin Performans Karşılaştırılması

Yapılan deneyler sonucunda oluşturulan modellerde, temel sınıflandırma metrikleri olan doğruluk oranı, F1-skor, makro ortalama (precision) ve duyarlılık (recall) üzerinden değerlendirilmiştir. Tablo 4.6'da her bir modelin genel başarım düzeyleri bu metrikler açısından karşılaştırmalı olarak sunulmaktadır.

Tablo 4.6 Sınıflandırma Modellerinin Performans Ölçütleri

Sınıflandırma Modeli	Doğruluk Oranı	F1-Skor	Precision	Recall
ResNet50	0.96	0.96	0.96	0.96
CNN	0.84	0.86	0.84	0.91
Karar Ağacı	0.82	0.78	0.79	0.78
KNN	0.85	0.83	0.83	0.84
Naive Bayes	0.72	0.68	0.66	0.73

Tablo incelendiğinde, önerilen derin öğrenme mimarisi olan ResNet50 modelinin tüm metriklerde en yüksek başarıyı sağladığı görülmektedir. %96 doğruluk oranına ek olarak, %96 seviyesinde makro F1-puanı, precision ve recall değerleri elde edilmiştir. Bu sonuç, modelin tüm sınıflarda dengeli ve yüksek kaliteli sınıflandırma yapabildiğini, öğrenilen örüntüleri genelleme kapasitesinin güçlü olduğunu ortaya koymaktadır.

Evrişimli Sinir Ağı (CNN) mimarisi ise ResNet50'nin sonra yüksek oran sağlayan model olmuştur. Recall %0.91'lik değeri ile encoding türlerinin büyük kısmını doğru şekilde tespit edebilmiştir. Ancak residual bağlantılar gibi derin temsili kuvvetlendiren yapısal avantajları barındırmaması nedeniyle, ResNet50'ye kıyasla genel doğruluk oranında bir miktar geride kalmıştır.

Geleneksel yöntemlerden KNN ve Karar Ağacı modelleri, sırasıyla %85 ve %82 doğruluk oranlarına ulaşmıştır. Bu modeller, özellikle Hex ve Bozuk (Anomalili) sınıflarda yüksek doğruluk göstermekle birlikte, yapay sinir ağlarının örüntü yakalama kabiliyeti karşısında sınırlı kalmışlardır. Ayrıca, daha az dengeli sınıflarda başarı düşüşü gözlemlenmiştir.

En düşük başarıya sahip model olan Naive Bayes, %72 doğruluk ve %68 F1-puanı ile encoding sınıflandırma gibi yapısal örüntü yoğun görevlerde yetersiz kalmıştır. Özellikle düşük precision ve recall değerleri, modelin belirli sınıflarda doğru tahmin yapma konusunda zorlandığını göstermektedir.

Encoding türlerinin sınıflandırılmasında derin öğrenme temelli yaklaşımların, geleneksel makine öğrenmesi algoritmalarına kıyasla önemli ölçüde bir üstünlük sağladığını göstermektedir. Derin öğrenme modelleri, özellikle karakter dizileri içerisindeki sembolik, yapısal ve konumsal örüntüleri çok katmanlı temsiller aracılığıyla öğrenebilme yeteneğine sahip olmuştur. Bu sayede, düşük düzeyde anlam veya bağlam içeren encode edilmiş verilerde dahi, modelin sınıflar arası ayrımı doğru biçimde yapabilmesi mümkün olmaktadır. Çalışmadaki geleneksel modeller karmaşık örüntüleri yakalama konusunda sınırlı kalmaktadır. Özellikle sınıflar arasında görsel veya yapısal benzerlikler bulunduğunda, ResNet50 ve CNN gibi mimariler öğrenme sürecini derinleştirerek ayrım gücünü artırmakta ve daha dengeli sonuçlar üretmektedir.

Sonuç olarak, encoding yapılarının ayırt edilmesinde derin öğrenme modellerinin özellikle görselleştirilmiş veri temsilleriyle birlikte kullanıldığında klasik yöntemlerin ötesine geçen bir performans sergilediği söylenebilir. Bu çalışmada önerilen ResNet50 tabanlı yaklaşımın güvenilirlik, doğruluk ve genelleme açısından üstünlük sağladığı deneysel olarak kanıtlanmıştır.

4.6. ResNet50 Modelinin Literatürdeki Çalışmalarla Karşılaştırılması

Encoding türlerinin sınıflandırılmasına yönelik önerilen ResNet50 modeli ile literatürde yer alan Magika, Sherlock, EnCoD ve ET-BERT gibi güncel yöntemler doğruluk ve F1-skoru metrikleri açısından karşılaştırılmıştır. Aşağıda, elde edilen performans sonuçlarını özetleyen Tablo 4.7 yer almaktadır.

Tablo 4.7 Literatürdeki Çalışmalarla Karşılaştırılma

Sınıflandırma Modeli	Doğruluk Oranı	F1-Skor	Veri Seti
ResNet50	0.96	0.96	Base64, Base32, ASCII85, Hex encode edilmiş sistem log verileri
Magika	0.99	0.99	GitHub + VirusTotal kaynaklı test verisi
Sherlock	-	0.89	VizNet veri kümesi semantik veri türü
EnCoD	0.86	-	16 farklı dosya türü üzerinde; sıkıştırılmış vs. şifreli veri ayrımı
ET-BERT	≅0.99	≅0.99	ISCX-TOR, ISCX-VPN-Service, USTC-TFC veri kümeleri; şifreli trafik verileri

Magika modeli içerik türü sınıflandırması (örneğin XML mi, JSON mu?) yapacak şekilde tasarlanmıştır ve daha çok **tek katmanlı ve metin tabanlı içerikleri** tanımlama yetkinliğine sahiptir. Tez kapsamında geliştirilen ResNet50 tabanlı model ise, encoding yapılarının sınıflandırılmasına odaklanan özel bir problemi ele almaktadır. Özellikle sistem logları gibi düzensiz ve bağlamsız veri örneklerinden elde edilen içeriklerin Base64, Base32, ASCII85 ve Hex gibi encoding türlerine göre doğru biçimde ayrıştırılması hedeflenmiştir. Magika, doğrudan encoding türlerini değil, daha genel içerik kategorilerini ayırt etmek üzere eğitildiğinden, iç içe geçmiş encoding yapılar ya da kısa-şekilsiz diziler gibi karmaşık örneklerde dezavantajlı hale gelebilir. Bu yönüyle bakıldığında, önerilen sistem daha dar ancak **güvenlik açısından kritik** bir problem alanına özgü olarak geliştirilmiştir.

Magika'nın geniş kapsamlı genellemeci yapısına karşın, bu çalışma derinlemesine örüntü ayrıştırmayı önceleyen bir yaklaşımla literatüre katkı sağlamaktadır.

ET-BERT modeli ise, şifreli ağ trafiği verilerini sınıflandırmak amacıyla Transformer tabanlı bir mimariyle geliştirilerek ISCX-TOR, CSTNET-TLS ve benzeri ağ veri kümeleri üzerinde yüksek doğruluk elde etmiştir.

Ancak bu modelin temel amacı, ağ seviyesinde şifreli verilerin hangi uygulama veya servis ile ilişkili olduğunu belirlemeye yöneliktir. Buna karşın, bu tez kapsamında geliştirilen model, farklı encoding türlerini doğrudan sınıflandırarak özellikle zararlı yazılımların encoding katmanları arkasına gizlenmiş içeriklerini tespit etmeye yönelik çalışmaktadır.

Transfer öğrenme tekniği ile güçlü genelleme yeteneği sergileyen sistem, encoding türlerinin yapısal örüntülerini doğru şekilde modelleyerek zararlı içeriklerin tespiti gibi uygulamalarda daha özelleşmiş bir katman olarak hizmet verebilir.

Yukarıda belirtilen modeller, her ne kadar kendi problem alanlarında güçlü derin öğrenme mimarilerine dayansa da doğrudan encoding türlerinin sınıflandırılmasına yönelik tasarlanmamışlardır. Magika, bayt düzeyinde içerik sınıflandırması yaparak XML, Base64 gibi formatları yüksek doğrulukla ayırt edebilmiştir. Eğitim verisinin çoğunlukla düz ve tek katmanlı yapıdan oluşması nedeniyle, çoklu (nested) encoding dizilerinin ayrıştırılmasında sınırlı performans göstermektedir. [53] Sherlock, semantik veri türlerini tanımlamak üzere tasarlanmış çoklu girişli bir sinir ağı modeli olup, anlam taşımayan ya da yapay karakter dizilerinden oluşan encode edilmiş içeriklerde dil bağlamına dayalı yapısı nedeniyle zayıf genelleme göstermektedir. [54]

EnCoD modeli, şifrelenmiş ve sıkıştırılmış dosya parçacıklarını birbirinden ayırma görevine odaklanmıştır. Ancak bayt düzeyinde çalışan bu sistem, encode edilmiş fakat düşük entropili verilerde (örneğin ASCII veya Hex) ayrım kabiliyeti göstermekte zorlanmakta; ayrıca çok katmanlı encode zincirlerinde verinin ilk katmanına ait örüntülerin zayıflamasıyla birlikte sınıflandırma başarımı düşmektedir. [55]

Benzer şekilde, ET-BERT, Transformer tabanlı bir mimariyle şifreli ağ trafiğini sınıflandırmakta etkili olsa da kısa ve statik karakter dizileriyle temsil edilen encoding yapıları için eğitilmemiştir. Encoding katmanlarının ardışık yapısının bozulması durumunda modelin performansı azalmaktadır. [56]

Bu bağlamda, önerilen ResNet50 tabanlı yöntem, encoding dizilerinin görsel örüntülerini modelleyerek yalnızca karakter düzeyinde değil, aynı zamanda yapısal ilişkileri çok katmanlı biçimde öğrenebilen bir sınıflandırma mimarisi sunmaktadır.

Görselleştirilen encoding dizileri üzerinden eğitilen model, residual öğrenme yeteneği sayesinde iç içe geçmiş (multi-layered) encoding yapılarında dahi kararlı ve yüksek doğruluklu sınıflandırmalar gerçekleştirebilmiştir. Böylece mevcut yöntemlerin sınırlı kaldığı encoding analizi problemlerine, güvenlik perspektifinden stratejik bir çözüm alanı sunmakta; özellikle zararlı yazılımların encoding katmanları ardına gizlenmiş içeriklerinin tespitinde yeni nesil yapay zekâ temelli karar destek sistemleri için bir temel oluşturmaktadır.

5. SONUÇLAR

Bu tez çalışması, kötü niyetli yazılımların sistemlere çeşitli encoding mekanizmaları üzerinden sızmasını hedef alan saldırı türlerine karşı yeni nesil bir savunma katmanı geliştirme amacı taşımaktadır. Özellikle malware temelli veri akışlarının tespiti sürecinde, sisteme ulaşan içeriklerin hangi tür encoding yöntemiyle kodlandığının otomatik olarak belirlenmesi, anomali saptama sistemleri açısından önemli bir güvenlik adımıdır. Encoding türlerinin doğru ve hızlı biçimde tespit edilmesi, zararlı içeriklerin sahte kodlama yapılarının arkasına saklanmasının önüne geçmekte kritik rol oynamaktadır. Çalışmada encoding türlerinin yapısal örüntülerine dayanarak sınıflandırılması hedeflenmiş ve bu sınıflandırma işlemi için farklı öğrenme yaklaşımları denenmiştir.

Geleneksel makine öğrenmesi algoritmaları olan Naive Bayes, KNN ve Decision Tree yöntemleri, belirli istatistiksel özniteliklere dayanarak sınıflandırma denemeleri gerçekleştirmiştir. Bu yöntemler düşük maliyetli ve yorumlanabilir olmalarına rağmen, encoding türleri gibi yapısal ve örüntüsel temsiller içeren problemler karşısında sınırlı başarı göstermiştir. Bu durum, sistem güvenliği gibi hata toleransının düşük olduğu alanlarda bu yöntemlerin tek başına yeterli olmadığını ortaya koymuştur.

Derin öğrenme temelli yaklaşımlar ise, encoding yapılarının altında yatan örüntüleri öğrenme ve genelleme konusunda çok daha yüksek başarı sergilemiştir. Karakter düzeyinde uygulanan CNN modeli, encoding türlerinin kısa ve orta ölçekli yapısal farklılıklarını başarıyla öğrenerek geleneksel yöntemlerin üzerinde doğruluk sağlamıştır. Daha da ötesinde, karakter dizilerinin görselleştirilerek evrişimsel sinir ağı mimarisi olan ResNet50 üzerinde eğitilmesiyle, encoding yapılarının resimsel örüntüleri üzerinden sınıflandırılması sağlanmış ve deneysel olarak en yüksek doğruluk bu yöntemle elde edilmiştir.

5.1. Ulaşılan Bulgular

Elde edilen bulgular, derin öğrenme mimarilerinin yalnızca metin verisini doğrudan işleyerek değil, aynı zamanda alternatif temsiller (görsel dönüşüm gibi) üzerinden de yüksek başarıya ulaşabildiğini göstermektedir. Özellikle transfer öğrenme yöntemiyle yapılandırılan modellerin, daha az sayıda örnekle dahi güçlü bir genelleme kapasitesine ulaşabildiği gözlemlenmiştir.

Bu durum, siber güvenlik gibi dinamik tehditlerin bulunduğu alanlarda öğrenme modellerinin hızlı uyum sağlayabilmesi açısından stratejik bir avantaj sunmaktadır.

Encoding türlerinin sınıflandırılmasına yönelik gerçekleştirilen bu tez çalışması, gelecekte siber güvenlik altyapılarına entegre edilebilecek yapay zekâ tabanlı bir karar destek katmanının tasarımına yönelik önemli deneysel ve kavramsal çıktılar sunmaktadır. Çalışma kapsamında elde edilen sonuçlar, encoding yapılarının otomatik olarak analiz edilmesiyle, sistemlere gelen veri akışının gerçek bir encode formatına sahip olup olmadığının belirlenmesine katkı sağlayabilecek bir modellemeyi işaret etmektedir. Bu tür bir analiz mekanizması, özellikle encoding yapılarını sahte biçimde oluşturarak geleneksel filtreleri aşmaya çalışan zararlı yazılımların erken tespiti açısından kritik öneme sahiptir.

Çalışma encoding temelli güvenlik analizlerinde derin öğrenme mimarilerinin nasıl etkin biçimde kullanılabileceğine dair uygulamalı bir yaklaşım ortaya koymakta ve bu sayede yapay zekâ tabanlı siber güvenlik katmanlarının geliştirilmesine yönelik bir zemin oluşturmaktadır. Gelişen yapay zekâ yöntemlerinin yalnızca veri madenciliği veya doğal dil işleme gibi klasik alanlarla sınırlı kalmadığı, aynı zamanda sistem güvenliği, tehdit algılama ve zararlı kod analizi gibi stratejik alanlarda da önleyici nitelikte çözümler üretebileceği bu çalışma ile desteklenmiştir. Özellikle encoding yapılarının ayırt edilmesine dayalı mikro düzeydeki analizlerin, makro ölçekteki sistem güvenliği kararlarına katkı sağlayacak şekilde ölçeklenebileceği ve bu sayede çok katmanlı güvenlik mimarilerinde yeni nesil yapay zekâ bileşenlerinin rolünün giderek artacağı öngörülmektedir.

5.2. Gelecekte Yapılabilecek Çalışmalar

Elde edilen bulgular, encoding temelli veri analizinin, siber güvenlik alanında yapay zekâ tabanlı karar destek sistemlerine entegre edilebilecek güçlü bir bileşen olduğunu ortaya koymuştur.

Çalışmada kullanılan encoding türleri sınırlı sayıda olup, daha nadir fakat potansiyel olarak tehdit unsuru taşıyan encoding yöntemlerinin de ilerleyen araştırmalara dahil edilmesi, modelin kapsamını genişletmek açısından faydalı olacaktır.

Encoding tespit modellerinin gerçek zamanlı veri akışlarını analiz edebilecek şekilde optimize edilmesi ve sistem performansını olumsuz etkilemeden güvenlik katmanlarına entegre edilebilmesi, gelecekteki uygulamalarda kritik rol oynayacaktır.

Geliştirilen derin öğrenme modellerin SIEM, IDS ve antivirüs gibi mevcut siber güvenlik sistemleriyle nasıl bütünleştirilebileceği üzerine yapılacak entegrasyon çalışmaları, encoding analizinin yalnızca akademik bir uygulama olmaktan çıkıp operasyonel güvenlik stratejilerinin parçası haline gelmesini sağlayacaktır.

Derin öğrenme modellerinin daha düşük kaynak tüketimiyle çalışabilmesi için donanım destekli optimizasyonlara gidilmesi de özellikle uç nokta cihazlarda kullanılabilirliğin artırılması açısından değerli görülebilir.

Tüm bu olasılıklar, encoding yapılarının otomatik analizi üzerine temellenen yapay zekâ çözümlerinin, siber tehditlere karşı daha esnek, hızlı ve öngörülü bir savunma mekanizmasının inşasına katkı sunabileceğini göstermektedir. Bu doğrultuda gerçekleştirilecek gelecek çalışmalar hem akademik derinliği artıracak hem de sahada uygulanabilir yeni nesil güvenlik çözümlerine öncülük edecektir.

KAYNAKLAR

[1] A. Tahir, «Neutralizing ransomware detection using base64 encoding,» *Electronics,* p. 1030, 2022.

[2] S. Hochreiter; J. Schmidhuber, «Long short-term memory,» *Neural Computation,* cilt 9, no. 8, p. 1735–1780, 1997.

[3] A. Vaswani, N. Shazeer, N. Parmar; J. Uszkoreit; L. Jones; A. N. Gomez; Ł. Kaiser; I. Polosukhin, «Attention Is All You Need,» %1 içinde *Advances in Neural Information Processing Systems*, Long Beach, CA, Aralık 2017.

[4] D. Gupta; A. Patel; S. Kumar, «Automated Encoding Type Classification in Structured Data using Deep Learning Approaches,» %1 içinde *[3] Gupta, D., Patel, A., & Kumar, S. (2023). Automated Encoding Type Classification in Structured Data using Deep Learning Approaches. Proceedings of the 2023 IEEE International Conference on Big Data.*, 2023.

[5] Fratantonio, Y.; Invernizzi, L.; Farah, L.; Thomas, K.; Zhang, M.; Albertini, A.; Bursztein, E., «Magika: AI-Powered Content-Type Detection,» arXiv, 2024.

[6] M. Hulsebos, K. Hu, M. Bakker, E. Zgraggen, A. Satyanarayan, T. Kraska, Ç. Demiralp ve C. Hidalgo, «Sherlock: A Deep Learning Approach to Semantic Data Type Detection,» %1 içinde *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*, Anchorage, AK, USA, 2019.

[7] F. De Gaspari; D. Hitaj; G. Pagnotta; L. De Carli; L. V. Mancini, «EnCoD: Distinguishing Compressed and Encrypted File Fragments,» %1 içinde *Network and System Security – NSS 2020, Lecture Notes in Computer Science, vol. 12570*, Network and System Security – NSS 2020, Lecture Notes in Computer Science, vol. 12570, 2020.

[8] X. Lin, G. Xiong, Z. G. Gou, J. Shi ve J. Yu, «ET-BERT: A Contextualized Datagram Representation with Pre-training Transformers for Encrypted Traffic Classification,» %1 içinde *Proceedings of the ACM Web Conference 2022 (WWW '22), Virtual Event*, Lyon, France, 2022.

[9] Y. Fratantonio; L. Invernizzi; L. Farah; K. Thomas; M. Zhang; A. Albertini, «Magika: AI-Powered Content-Type Detection,» arXiv, 2024.

[10] R. Hulsebos; M. V. Gutiérrez; J. Zwiener; T. Rekatsinas, «Sherlock: A Deep Learning Approach to Column Type Annotation,» %1 içinde *25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2019)*, Anchorage, 2019.

[11] F. De Gaspari; M. Pogliani, «EnCoD: A Deep Learning Based Method for Detecting Encrypted and Compressed Data,» %1 içinde *Network and System Security (NSS 2020)*, Melbourne.

[12] X. Lin; Y. Li, «ET-BERT: A Deep Learning Framework for Encrypted Traffic Classification,» %1 içinde *The ACM Web Conference 2022*, Lyon, 2022.

[13] Ö. Aslan; R. Samet, «A Comprehensive Review on Malware Detection Approaches,» *IEEE Access,* p. 6249–6271, 2020.

[14] Z. Zhong; H. Liu; Y. Chen; Y. Xiang, «DeepEncoding: Detecting and Decoding Encoded Data with Deep Learning,» *IEEE Access,* cilt 8, p. 189 745 – 189 758, 2020.

[15] D. T. Nguyen; T. D. Nguyen; H. H. Bui, «Data Format Classification Using Machine Learning Techniques,» %1 içinde *Proceedings of the 2021 International Conference on Data Mining and Big Data*, 2021.

[16] W. Stallings, Cryptography and Network Security: Principles and Practice, Pearson Education, 2016.

[17] N. Freed; N. Borenstein, «RFC 2045: Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies,» IETF, 1996.

[18] X. Ugarte-Pedrero; I. Santos; P. G. Bringas; G. Álvarez, «Just call them commands: Leveraging malware development artifacts for real-time detection,» *Computers & Security,* cilt 48, p. 83–94, 2015.

[19] «Binary-to-text encoding,» 2025. [Çevrimiçi]. Available: https://en.wikipedia.org/wiki/Binary-to-text_encoding. [Erişildi: 14 Haziran 2025].

[20] S. Josefsson, «RFC 4648: The Base16, Base32, and Base64 Data Encodings,» IETF, Ekim, 2006.

[21] D. M'Raihi; S. Machani; M. Pei; J. Rydell, «RFC 6238: TOTP: Time-Based One-Time Password Algorithm,» Internet Engineering Task Force (IETF), 2011.

[22] Adobe Systems Incorporated, PostScript Language Reference Manual, 2nd ed., Addison-Wesley Professional, 1990.

[23] B. Carrier, File System Forensic Analysis, Addison-Wesley Professional, 2005.

[24] A. McCallum ve K. Nigam, «A Comparison of Event Models for Naive Bayes Text Classification,» %1 içinde *AAAI-98 Workshop on Learning for Text Categorization*, Madison, WI, USA, 1998.

[25] K. P. Murphy, Machine Learning: A Probabilistic Perspective, Cambridge, MA, USA; London, UK: The MIT Press, 2012.

[26] H. Zhang, «The Optimality of Naive Bayes,» %1 içinde *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS-17)*, Miami, 2004.

[27] S. Chauhan; V. Tyagi; D. Kumar, «A Hybrid Deep Learning Model for Sentiment Classification,» %1 içinde *ResearchGate*, 2020.

[28] T. M. Cover; P. E. Hart, «Nearest Neighbor Pattern Classification,» *IEEE Transactions on Information Theory,* cilt 12, no. 1, p. 21–27, 1967.

[29] Taşcı, E.; Onan, A., «K-En Yakın Komşu Algoritması Parametrelerinin Sınıflandırma Performansı Üzerine Etkisinin İncelenmesi,» %1 içinde *Akademik Bilişim 2016*, Aydın, Türkiye, 2016.

[30] N. S. Altman, «An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression,» *The American Statistician ,* cilt 46, no. 3, p. 175–185, 1992.

[31] intuitivetutorial.com, «K-Nearest Neighbors Algorithm,» Intuitive Tutorials, 2023.

[32] J. R. Quinlan, «Induction of Decision Trees,» *Mavhine Learning,* cilt 1, no. 1, p. 81–106, 1986.

[33] L. Breiman; J. H. Friedman; R. A. Olshen; C. J. Stone, Classification and Regression Trees, Belmont, CA, USA: Wadsworth (Wadsworth & Brooks/Cole), 1984.

[34] Ioannis Mollas; Grigorios Tsoumakas; Nick Bassiliades, «LionForests: Local Interpretation of Random Forests through Path Selection,» arXiv, 2019.

[35] G. Zhong; L.-N. Wang; J. Dong, «An Overview on Data Representation Learning: From Traditional Feature Learning to Recent Deep Learning,» arXiv preprint, 2016.

[36] J. C. Olamendy, «Back to Basics: Feature Extraction with CNN,» Towards Data Science (Medium), 2023.

[37] A. Khan; A. Sohail; U. Zahoora; A. S. Qureshi, «A Survey of the Recent Architectures of Deep Convolutional Neural Networks,» *Artificial Intelligence Review,* p. 5455–5516, 2020.

[38] Y. LeCun; L. Bottou; Y. Bengio; P. Haffner, «Gradient-Based Learning Applied to Document Recognition,» *Proceedings of the IEEE ,* cilt 86, no. 11, p. 2278–2324, 1998.

[39] Y. LeCun; L. Bottou; Y. Bengio; P. Haffner, «Gradient-Based Learning Applied to Document Recognition,» *Proceedings of the IEEE,* cilt 86, no. 11, p. 2278–2324, 1998.

[40] V. Nair; G. E. Hinton, «Rectified Linear Units Improve Restricted Boltzmann Machines,» %1 içinde *27th International Conference on Machine Learning (ICML 2010)* , Haifa, Israel, 2010.

[41] C. M. Bishop, Pattern Recognition and Machine Learning, New York: Springer, 2006.

[42] CliffsNotes (Learneo, Inc.), «K-means Clustering,» 2024.

[43] I. Goodfellow; Y. Bengio; A. Courville, «Convolutional Networks,» %1 içinde *Deep Learning* , Cambridge, MA, USA, MIT Press, 2016.

[44] Y. Kim, «Convolutional Neural Networks for Sentence Classification,» %1 içinde *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, Doha, Katar, 2014.

[45] X. Zhang; J. Zhao; Y. LeCun, «Character-level Convolutional Networks for Text Classification,» %1 içinde *Advances in Neural Information Processing Systems*, Montreal, Kanada, 2015.

[46] K. He; X. Zhang; S. Ren; J. Sun, « Deep Residual Learning for Image Recognition,» %1 içinde *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, Las Vegas, NV, USA, 2016.

[47] C. Szegedy; S. Ioffe; V. Vanhoucke; A. Alemi, «Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,» %1 içinde *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017)*, San Francisco, CA, USA, 2017.

[48] K. He; X. Zhang; S. Ren; J. Sun, «Deep Residual Learning for Image Recognition,» %1 içinde *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, Las Vegas, 2016.

[49] Scaler, «Residual Networks (ResNet),» Scaler, 2025.

[50] K. He; X. Zhang; S. Ren; J. Sun, «Deep Residual Learning for Image Recognition,» %1 içinde *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, Las Vegas, 2016.

[51] A. Khan; A. Sohail;U. Zahoora; A. S. Qureshi, «A Survey of the Recent Architectures of Deep Convolutional Neural Networks,» *Artificial Intelligence Review,* cilt 53, no. 8, p. 5455–5516, 2020.

[52] Sinno Jialin Pan; Qiang Yang, «A Survey on Transfer Learning,» *IEEE Transactions on Knowledge and Data Engineering,* cilt 22, no. 10, p. 1345–1359, 2010.

[53] Y. Fratantonio; L. Invernizzi; L. Farah; K. Thomas; M. Zhang; A. Albertini, «Magika: AI-Powered Content-Type Detection,» arXiv, 2024.

[54] M. Hulsebos, K. Hu, M. Bakker, E. Zgraggen, A. Satyanarayan; T. Kraska; C. Demiralp; C. Hidalgo, «Sherlock: A Deep Learning Approach to Semantic Data Type Detection,» %1 içinde *25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2019)*, 2019.

[55] F. De Gaspari; D. Hitaj; Pagnotta, G.; Carli, L. De; Mancini, L. V., EnCoD: Distinguishing Compressed and Encrypted File Fragments, Sprinter, 2020.

[56] X. Lin; G. Xiong; G. Gou; Z. Li, J. Shi; J. Yu, «ET-BERT: A Contextualized Datagram Representation with Pre-training Transformers for Encrypted Traffic Classification,» %1 içinde *The ACM Web Conference 2022*, 2022.

[57] S. P. Mallick, «Understanding CNNs,» GitHub , 2025.