T.C.
TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ISTEMCI TARAFLI ŞİFRELEME VE DAĞITIK DEPOLAMA İLE KİŞİSEL VERİ GÜVENLİĞİNİN ARTTIRILMASINA YÖNELİK BULUT TABANLI UYGULAMANIN GELİŞTİRİLMESİ

KEMAL HAKAN

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Tez Danışmanı: Dr. Öğr. Üyesi Tarık YERLİKAYA

EDİRNE-2019

KEMAL HAKAN'in hazırladığı "**İSTEMCİ TARAFLI ŞİFRELEME VE DAĞITIK DEPOLAMA İLE KİŞİSEL VERİ GÜVENLİĞİNİN ARTTIRILMASINA YÖNELİK BULUT TABANLI UYGULAMANIN GELİŞTİRİLMESİ**" başlıklı bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından Bilgisayar Mühendisliği Anabilim Dalında bir **Yüksek Lisans Tezi** olarak kabul edilmiştir.

Jüri Üyeleri (Ünvan, Ad, Soyad): İmza

Dr.Öğr.Üyesi.Tarık YERLİKAYA(Danışman) .....
Dr.Öğr.Üyesi Derya ARDA .....
Doç.Dr.Ercan BULUŞ .....

Tez Savunma Tarihi: 02.01.2020

Bu tezin Yüksek Lisans Tezi olarak gerekli şartları sağladığını onaylarım.

Dr.Öğr.Üyesi.Tarık YERLİKAYA
Tez Danışmanı
.....

İmza

Trakya Üniversitesi Fen Bilimleri Enstitüsü Onayı

Prof.Dr.Murat YURTCAN
Fen Bilimleri Enstitüsü Müdürü

T.Ü.FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ YÜKSEK LİSANS PROGRAMI

DOĞRULUK BEYANI

Trakya Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında, tüm verilerin bilimsel ve akademik kurallar çerçevesinde elde edildiğini, kullanılan verilerde tahrifat yapılmadığını, tezin akademik ve etik kurallara uygun olarak yazıldığını, kullanılan tüm literatür bilgilerinin bilimsel normlara uygun bir şekilde kaynak gösterilerek ilgili tezde yer aldığını ve bu tezin tamamı ya da herhangi bir bölümünün daha önceden Trakya Üniversitesi ya da farklı bir üniversitede tez çalışması olarak sunulmadığını beyan ederim.

02/01/2020

Kemal HAKAN

İmza



Yüksek Lisans Tezi

Istemci Taraflı Şifreleme ve Dağıtık Depolama ile Kişisel Veri Güvenliğinin Arttırılmasına Yönelik Bulut Tabanlı Uygulamanın Geliştirilmesi

T.Ü Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

Günümüzde merkeziyetsiz kripto para transferleri birçok alanda merkeziyetsiz işlemlerin yapılabilirliğini ön plana çıkarmış ve bu tezin ortaya çıkarılmasıda ilham kaynağı olmuştur. Bilgi güvenliğini hedefleyerek yapılmış olan bu tez, kurum ve kuruluşlarca saklanan dijital belgelerin güvenli bir şekilde depolanmasını ve makine öğrenmesi ile sınıflandırılmasını kapsamaktadır. Bu çalışmada geliştirilen yazılım kullanılarak, öncelikle bir sistem yöneticisi tarafından güvenle saklanması gereken belgeler işaretlenir daha sonra ise güvenli olmalı olarak işaretlenen veriler için belirlediğimiz güvenlik yöntemleri uygulanır. Bu yöntemlere göre güvenli işaretlenen belgeler indirilmek ya da silinmek istediğinde kullanıcıya ait mobil cihazdan 2FA ile doğrulama yapılması beklenecektir. Makine öğrenmesi ile belgeler otomatik sınıflandırılarak kullanıcı hatalarının önüne geçilecek ve performans arttırımı yapılacaktır. Belgeler IPFS ile bloklar halinde depolama alanlarında tutulacaktır. Böylece belgeler gizlilik, bütünlük ve güvenlik ilkelerine bağlı olarak korunması sağlanacaktır.

Yıl : 2019

Sayfa Sayısı : 80

Anahtar Kelimeler: Makine Öğrenmesi, Kişisel Verilerin Korunması, IPFS, Güvenli veri depolama

Trakya University Institute of Natural Sciences
Developing Cloud Based Application for Enhancing Personal Data Security with Client-Side Encryption and Distributed Storage
Computer Engineering Department

ABSTRACT

The recent popularity of decentralized crypto money transfers have brought forward the feasibility of decentralized transactions and has been an inspiration for my study. This study, which was conducted with the aim of information security, deals with the safe storage of digital documents by institutions/organizations and classification of them by machine-learning algorithms.

Using the software developed in this study, firstly the documents that should be kept safely are marked by a system administrator and then the security methods that we have determined are applied to the data that are marked as “keep safe”. Because of the proposed security measures, downloading or deleting data will require 2FA verification from the mobile device of the user. Through machine learning, the documents will be classified automatically, which will lead to fewer user errors being made and performance improvement will be achieved. Using IPFS, documents will be stored in a distributed and fragmented fashion. As a result, documents will be protected in accordance with privacy, integrity and security principles.

Year : 2019
Number of Pages : 80
Keywords : Machine Learning, Personel Data Protection, IPFS, Secure Data Storage

TEŞEKKÜR

Tez çalışmamda bana destek olan, tecrübe ve desteklerini esirgemeyen ve yol gösteren sayın danışman hocam Dr. Öğr. Üyesi Tarık YERLİKAYA sonsuz teşekkürlerimi sunarm.

Aynı zamanda her türlü zorluğa benimle göğüs gelen, moral ve motivasyon kaynağım olan sevgili eşim İlknur HAKAN ve canım kızım Irmak HAKAN'a sonsuz sevgilerimle...

İÇİNDEKİLER

DOĞRULUK BEYANI iii
ÖZET iv
ABSTRACT v
TEŞEKKÜR vi
İÇİNDEKİLER vii
SİMGELER VE KISALTMALAR DİZİNİ xi
ŞEKİLLER DİZİNİ xii
ÇİZELGELER DİZİNİ xiv
BÖLÜM 1 1
GİRİŞ 1
BÖLÜM 2 2
KRİPTOLOJİ 2
2.1 Kriptografi 2
2.1.1 Gizlilik (Confidentiality) 3
2.1.2 Bütünlük (Integrity) 3
2.1.3 Kimlik Denetimi (Authentication) 4
2.1.4 İnkâr Edememe (Non-repudiation) 4
2.2 Kriptografik Hash Fonksiyonları 4
2.2.1 Hash işlevinin özellikleri 4
2.2.2 SHA 5
BÖLÜM 3 8
BLOK ZİNCİRİ 8

3.1 Eşler Arası İşlem 9
3.2 Blok Zincir İşlemi 9
3.2 Blok Zinciri Katmanları 10
3.2.1 Uygulama Katmanı 10
3.2.2 Yürütme Katmanı 11
3.2.3 Anlamsal Katman 11
3.2.4 Yayılma Katmanı 11
3.2.5 Mutabakat Katmanı 11
3.3 Blok Yapısı 12
3.4 Blok Başlığı 13
3.5 Merkle Ağacı (Merkle Tree) 13
BÖLÜM 4 15
GEZEGENLER ARASI DOSYA SİSTEMİ (IPFS) 15
4.1 IPFS Düğümleri 16
4.2 IPFS İçerik Adresleme 17
4.3 IPFS Nesneleri 17
4.4 IPFS Dosyaları 18
4.5 IPFS Dosya Versiyonlama 18
4.6 IPFS Çoklu Özet 18
4.7 Swarm 20
BÖLÜM 5 21
KİŞİSEL VERİLERİN KORUNMASI 21
5.1 Kanun Çıkmadan Önce Toplanan Verilerin Durumu 22
5.2 KVKK Kapsamında Şirketlerin Yükümlülükleri 22
5.3 Kişisel Verilerin Şifrelenmesi 22
5.4 Kişisel Verilerin Yedeklenmesi 22
5.5 Sızma Testleri 23
BÖLÜM 6 24
MAKİNE ÖĞRENMESİ 24

6.1 Makine Öğrenmesi Kavramları 24
6.1.1 Denetimli Öğrenme 24
6.1.2 Yarı Denetimli Öğrenme 25
6.1.3 Denetimsiz Makine Öğrenmesi 25
6.1.4 Takviyeli Öğrenme 25
6.1.5 Yoğun Öğrenme 25
6.2 Sınıflandırma Yöntemleri 25
6.3 Sınıflandırma Algoritmaları 26
6.3.1 Naive Bayes (Denetimli Öğrenme) 26
6.3.2 Karar Ağaçları (Denetimli Öğrenme) 26
6.3.3 Rastgele Ağaç (Denetimli Öğrenme) 27
6.3.4 Yakın Komşu (Denetimli Öğrenme) 27
6.3.5 K-Ortalama (Denetimsiz Öğrenme) 28
6.4 İkili Sınıflandırma Algoritmaları 29
6.4.1 Ortalama Algılayıcı (Avaraged Perceptron) 30
6.4.2 Sdca Lojistik Regresyon (Logistic Regression) 30
6.4.3 Yüksek Verimli Eğim Arttırıcı Karar Ağacı (LightGbm) 31
6.5 Otomatik Makine Öğrenmesi 31
BÖLÜM 7 32
İKİ FAKTÖRLÜ KORUMA(2FA) 32
7.1 Google Şifrematik 33
BÖLÜM 8 36
UYGULAMA 36
8.1.Uygulama Arayüzleri 36
8.1.1 Oturum Açma 36
8.1.2 Kullanıcı Kaydı 37
8.1.3 2FA Doğrulama ve Tanımlama 38

8.1.4 Kayıtlı Dosya Listeleri 39
8.1.4 Güvenli İndirme 40
8.1.5 Güvenli Silme 42
8.1.6 Dosya Yükleme 43
8.1.7 Eğitim Ekranı 44
8.1.8 Makine Öğrenmesi Sonuçları 45
8.1.9 Bölümlendirilmiş Dosya Blokları 46
8.2 Sistem Diyagramları ve Şemaları 47
8.2.1 Uml Diyagramı 47
8.2.2 Belge Yükleme Şeması 49
8.2.3 Belge İndirme Şeması 50
8.2.4 Belge Silme Şeması 51
8.2.5 Belge Yükleme Akış Diyagramı 52
8.2.6 Belge İndirme Akış Diyagramı 53
8.2.7 Belge Silme Akış Diyagramı 54
BÖLÜM 9 55
SONUÇLAR VE TARTIŞMA 55
KAYNAKLAR 63
ÖZGEÇMİŞ 66

SİMGELER VE KISALTMALAR DİZİNİ

CIDs:	Content Identifiers
DAG:	Directed Acyclic Graphs
DHT:	Distributed Hash Table
BTC:	Bitcoin
ETH:	Ethereum
POW:	Proof ofWork
DAP:	Decentralized Applications
ASIC:	Application Specific İntegrated Circuit
IPLD:	InterPlanatery Linked Data
P2P	Peer to Peer
2FA	Two-Factor Authentication
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negaive
TOTP	Time-base One-Time Passwords
HOTP	HMAC-based One Time Password
OTP	One-Time Passwords
HMAC	Hash-based Message Authentication Code
KVKK	Kişisel Verilerin Korunması Kanunu
AUC	Area Under Curve
AUCPR	Area Under Precision Recall Curve

ŞEKİLLER DİZİNİ

Şekil 2.1 Kriptolojinin Dalları....2
Şekil 2.2 Simetrik Şifreleme....3
Şekil 2.3 Asimetrik Şifreleme....3
Şekil 2.4 Hash Blok Yapısı....5
Şekil 2.5 Sıkıştırma Fonksiyonu Örneği....5
Şekil 2.6 SHA-1 Diyagramı....6
Şekil 2.7 SHA-2 Ailesi Diyagramı....7
Şekil 3.1 Eşler Arası İşlem....9
Şekil 3.2 Blok Zinciri İşlem Akışı....10
Şekil 3.3 Blok Zinciri Katmanları....10
Şekil 3.4 Blok Zinciri Yapısı....12
Şekil 3.5 Blok İç Yapıları....12
Şekil 3.6 Merkle Ağacı....13
Şekil 3.7 Merkle Ağacı Hesaplama Örneği....14
Şekil 4.1 IPFS Teknik Özellikleri....16
Şekil 4.2 İçerik Adresleme ile Dağıtık Hash Yapısı Örneği....17
Şekil 4.3 Hash Değerine Uzunluk ve Kod Eklenmesi....19
Şekil 6.1 Naive Bayes Denklemi....26
Şekil 6.2 K -NN sınıflandırması Örneği....27
Şekil 6.3 K-Means Kümeleme Örneği....28
Şekil 6.4 İki Nokta Arası Mesafe Formülü....28
Şekil 7.1 OTP Sözde Kod Örneği....35
Şekil 8.1 Oturum Ekranı....37
Şekil 8.2 Uygulama Kayıt Ekranı....38
Şekil 8.3 2FA Tanımlama Ekranı....39
Şekil 8.4 Benim Dosyalarım Ekranı....40
Şekil 8.5 Güvenli İndirme Ekranı....41

Şekil 8.6 Güvenli Belge Silme 42
Şekil 8.7 Dosya Yükleme Ekranı 43
Şekil 8.8 Hafif Kutu Ekranı 44
Şekil 8.9 Öğrenme Durum Ekranı 45
Şekil 8.10 Bloklanmış Veriler 46
Şekil 8.11 Uml Diyagramı 48
Şekil 8.12 Belge Yükleme Şeması 49
Şekil 8.13 Belge İndirme Şeması 50
Şekil 8.14 Belge Silme Şeması 51
Şekil 8.15 Belge Yükleme Akış Diyagramı 52
Şekil 8.16 Belge İndirme Akış Diyagramı 53
Şekil 8.17 Belge Silme Akış Diyagramı 54
Şekil 9.1 Eğitim Test Sonucu 1 56
Şekil 9.2 Test Eğitim Sonucu 2 57
Şekil 9.3 Genel Eğitim Sonuçları 1 58
Şekil 9.4 Genel Eğitim Sonuçları 2 58

ÇİZELGELER DİZİNİ

Çizelge 2.1 SHA Algoritması Karşılaştırması 7
Çizelge 3.1 Blok Başlığı Özelliği ve Boyutları 13
Çizelge 4.1 Sha Ailesi Kodları 19
Çizelge 6.1 Sınıflandırma Modellerinde Yaygın Kullanılan Metrikler 29
Çizelge 6.2 Ortalama Algılayıcı Modül Parametleri 30
Çizelge 6.3 Sdca Lojistik Regresyon Modül Parametreleri 31
Çizelge 7.1 2FA'ın Yıl Bazlı Kullanım Artışı 33
Çizelge 9.1 Test Sonuçları 59
Çizelge 9.2 Genel Sonuçlar 61

BÖLÜM 1

GİRİŞ

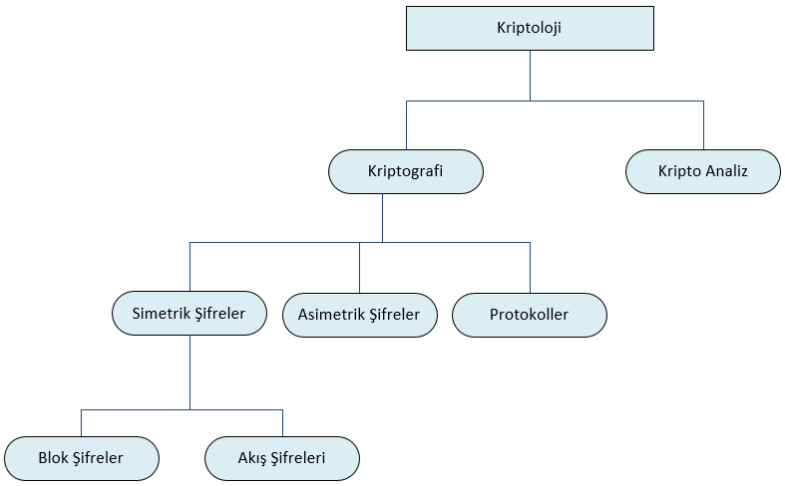
Günümüzde, teknolojinin gelişmesiyle birçok şey analog olmaktan çıktı ve dijital olmaya başladı. Böylece dijital olarak yapılan hemen hemen her işlemde güvenlik kavramının önemi arttı. Dijital ortamdan aktarılan verilerin korunması ve gizliliğinin sağlanmasına yönelik birçok çalışma yapıldı. Özellikle dijital verinin dağıtılması denilince akla ilk blok zinciri gelmektedir. Blok zinciri, Satoşi Nakamoto tarafından tasarlanmıştır ve teknik olarak, kriptoloji, oyun teorisi ve bilgisayar kavramlarının birleşiminden oluşmaktadır. Kullanım durumu ne olursa olsun işlemler kriptografi kullanılarak güvence altına alınmıştır. Blok zinciri 'in de fonksiyonelliği destekleyen temel bileşen şüphesiz merkle ağaçlarıdır. İkili ağacı ile aynı yapıdadır. Veri bütünlüğünün korunması ve doğrulanmasında hash tabanlı mimari kullanarak veriyi ölçeklendirir ve bunu yaparken kriptografik hash fonksiyonlarını kullanır. Blok zincirinde herbir blok, önceki bloğun içindeki verilerle birlikte hash işaretçisi ile birbirine bağlanır ve sonuç olarak çıkan son hash tüm durumu temsil eden bir parmak izidir. Bu durum en ufak bir bilgi değişikliğinin bile çok farklı bir parmak izi oluşturacağını göstermektedir.

Bu bilgilere bağlı olarak özellikle bilginin değişmezliği ve güvenliğinin önemli ölçüde değerli olduğu günümüz de birçok sektörde bilgi güvenliğine duyulan gereksinim her geçen gün artmaktadır. Bu tezde doküman veya belgelerin gizliliğinin ve bütünlüğünün korunması amacıyla IPFS, güvenliğinin sağlanması amacıyla 2FA ve kullanıcı hatalarını engelleme ve performans attırımı için adaptif öğrenme tekniklerini barındıran bir uygulama geliştirilmiştir.

BÖLÜM 2

KRİPTOLOJİ

Günümüzde, birçok ortamdan dijital olarak aktarılan bilgilerin güvenlik endişeleri her geçen gün artmaktadır. Buna bağlı olarak çeşitli güvenlik önlemleri alınmakta ve bu önlemler genellikle matematiksel tekniklerin içerdiği fonksiyonları barındırmaktadır. Kriptolojinin dalları Şekil 2.1.'de gösterilmiştir.

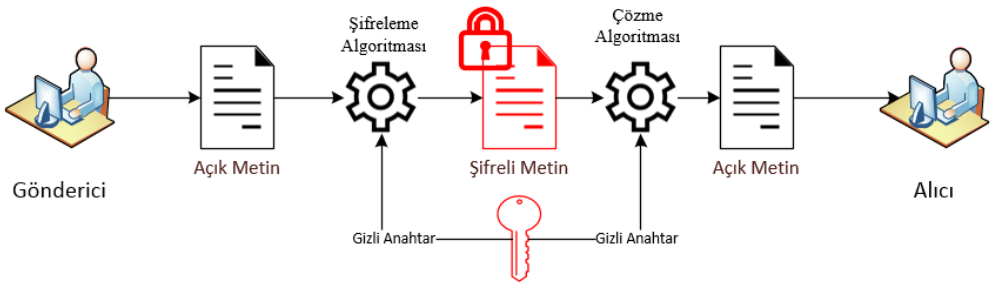


Şekil 2.1 Kriptolojinin Dalları

2.1 Kriptografi

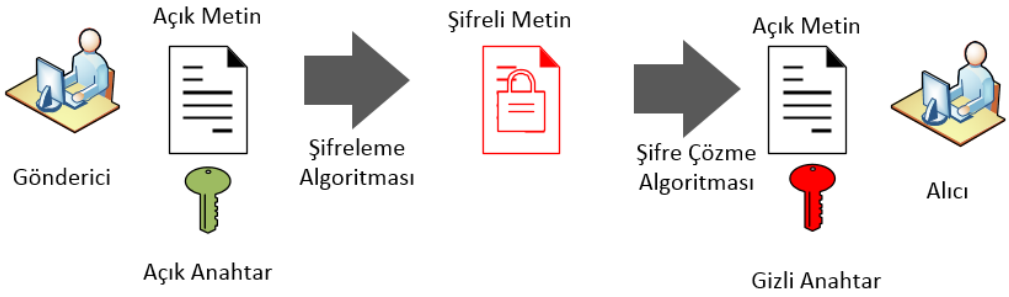
Kriptografi, bilginin paylaşılacak kişi dışındaki kişilerce anlaşılmaması için kullanılan teknikler bütünü olarak tanımlanabilir. Simetrik Algoritmalar, gönderici ve alıcının bir gizli anahtarı paylaşarak şifreleme ve çözme işlemini yaptığı algoritmalardır. 1976 yılına kadar yaygın olarak kullanılmıştır.

Şekil 2.2.'de Simetrik Şifreleme örneği gösterilmiştir.



Şekil 2.2 Simetrik Şifreleme

Asimetrik algoritmalar, kriptolojide 1976 yılında Hellman'ın önerisi üzerine "Açık Anahtarlı Sistemler" kavramıyla bir devrim geçirmiştir. Bu sistemde aynı anahtarın hem alıcı hem de gönderici tarafından bilinmeden de güvenli bir şekilde haberleşmenin yapılabileceği kanıtlanmıştır. Bu algoritmada hem gizli anahtar (Private Key) hem de açık anahtar (Public Key) bulunmaktadır. Şekil 2.3.'te asimetrik şifreleme örneği gösterilmiştir.



Şekil 2.3 Asimetrik Şifreleme

(Akleylek, Yıldırım, & Tok)

2.1.1 Gizlilik (Confidentiality)

Herhangi bir bilginin yetkisi olmayan kişilerce anlaşılmamasıdır.

2.1.2 Bütünlük (Integrity)

Veri bütünlüğünün istenmeyen veya yetkisiz kişilerce bozulmasına veya değiştirilmesine izin verilmeden içeriğinin korunmasıdır.

2.1.3 Kimlik Denetimi (Authentication)

Bilgi paylaşımı sırasında gönderici ve alıcı arasında bir kimlik doğrulama işleminin olmasını gerektirir.

2.1.4 İnkâr Edememe (Non-repudiation)

Gönderici ve alıcı veriyi gönderdiğinde veya aldığında işlemin geçerliliğini inkâr edememelidir.
(Akleylek, Yıldırım, & Tok)

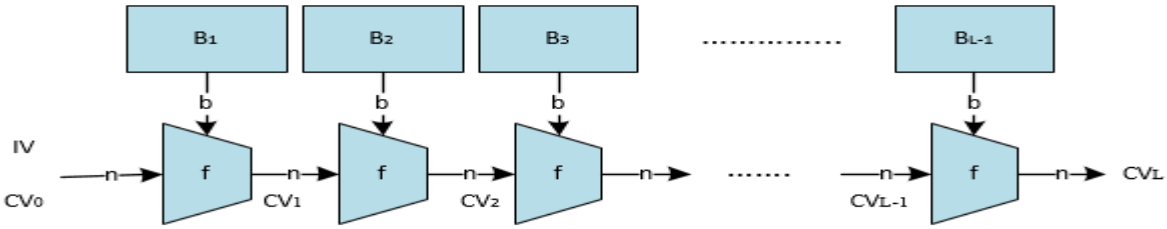
2.2 Kriptografik Hash Fonksiyonları

Kriptografik hash fonksiyonlardaki amaç girdi boyutundan bağımsız olarak sabit uzunlukta benzersiz bir çıktı oluşturmaktır. İki farklı girdinin çıktısı aynı olamaz bu bize hash fonksiyonlarının deterministik bir özelliğe sahip olduğunu gösterir. Üretilen son çıktılar karma (hash) ya da dijital parmak izi (dijital finderprint) olarak isimlendirilebilir.

2.2.1 Hash işlevinin özellikleri

1. Herhangi bir boyutta girdi alır
2. Sabit uzunlukta çıktı üretir
3. Hesaplaması kolaydır.
4. Herhangi bir (x, y) ikilisi için $x \neq y$ olmak üzere $H(x) = H(y)$ hesaplanması olanaksızdır.
5. Bir h değeri verildiğinde $H(x) = h$ fonksiyonu için x 'i bulmak olanaksızdır.

Eğer iki farklı girdi aynı çıktıyı üretir ise burada bir çakışma "Collision" vardır ve ilgili Hash fonksiyonu kırılmıştır. SHA-1 ve MD5 'de çakışma tespit edildiğinden günümüzde yeterince güçlü olarak kabul edilmemektedirler. Bir hash fonksiyonu tipik olarak dahili sıkıştırma fonksiyonuna ve sabit boyutlu blok girişlerine sahiptir. Örnek bir hash blok yapısı Şekil 2.4.'te gösterilmiştir.
(Watts, 2019)

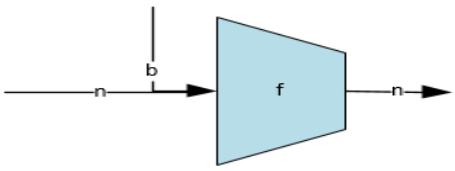


Sembol	Açıklama
B	Mesaj bloğu
b	Blok uzunluğu
n	Hash kodu uzunluğu
L(Length)	Uzunluk
IV(Initial Value)	Başlangıç değeri
CV(Compressed Value)	Sıkıştırılmış değer
$CV_L = H(M)$	Üretilen son hash değeri

Şekil 2.4 Hash Blok Yapısı

(Watts, 2019)

Zincir gibi birbirine bağlı blok şifreleri şeklinde sıralanırlar. Her bir sabit boyutlu bloğun değeri ve önceki bloğun değerine dayalı olarak hash değeri üretilir ve bu değer 1-bit değişiminde dahi çığ etkisi "Avalanche Effect" ve çıkış değerinde tahmin edilemez değişiklikler yaratır. Fonksiyon bazında açıklamak gerekirse sıkıştırma fonksiyonu "f" olmak üzere n ve b bitlerini alarak sabit uzunlukta n-bit çıkışını üretir. Burada n önceki blok ya da başlangıç değerinden gelen bit sayısıdır. b ise girişten gelen veri bitleridir. Şekil 2.5.'te sıkıştırma fonksiyonu örneği gösterilmiştir.



Şekil 2.5 Sıkıştırma Fonksiyonu Örneği

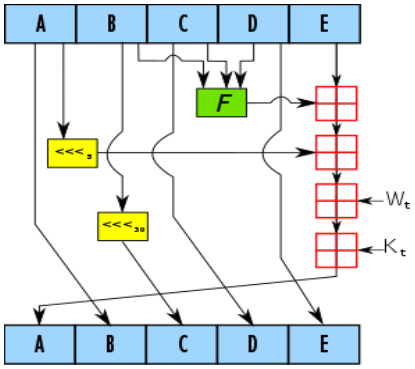
Biz bu tezde SHA algoritmalarından faydalanacağımız için bu kısımda tanımlamalar, kıyaslamalar ve performans değerlerinden bahsedilmiştir.

(Watts, 2019)

2.2.2 SHA

Bu algoritma 1993 yılında NIST (National Institute of Standarts and Techology) tarafından geliştirildi ve FIPS (Federal Information Processing Standart) 180 olarak yayınlandı ve ilk revizyon FIPS 180-1 SHA- 1 olarak 1995 yılında 160-bit hash değeri

üretecek şekilde tasarlanıp yayınlandı. SHA-1 sadece güvenlik için değil aynı zamanda veride herhangi bir değişiklik olup olmadığına emin olmak için kullanılmaktadır. SHA-1, 2^{64} -1 bitlik mesajdan 160'bitlik özet değer üretir. Dijital sertifikalar, kök sertifikalar ve dijital imza uygulamaları RSA ve SHA-1 üzerinde çalışmaktadır. FIPS 180-2 3. versiyonu 2002 yılında çıkarıldı. Bunlar SHA-256, SHA-384 ve SHA-512 versiyonlarıdır. SHA-1 için diyagram Şekil 2.6.'da gösterilmiştir.
(SHA-1, 2019)



Şekil 2.6 SHA-1 Diyagramı

(SHA-1, 2019)

Şıkıştırma işlevi içinde yinelenen A, B, C, D ve E 32- bit kelimeleridir F değişken olan doğrusal olmayan bir fonksiyondur. "n" sola bit dönüşünü belirtir ve her işlem için değişir. "Wt": t'nin genişletilmiş mesaj sözlüğüdür. "Kt": t nin yuvarlatılmış sabitidir. Bu işlemde mesaj 512 bitlik bloklar şeklinde ayrılır bit yetmediğinde son bit dizinine ekleme yapılarak 512 bite tamamlanır ve bu metin 16 bitlik 16 tane veri parçacığına ayrılır. Bu parçacıklarda normal dört işlem uygulandıktan sonra sola kaydırma işlemi uygulanır ve bu işlem 4 defa art arda uygulandıktan sonra 160 bitlik şifre dizisi oluşur. SHA-2, SHA-1'den önemli farklılıklar içermektedir ve basamakları 224,256,384 ve 512 bit olan 6 hash fonksiyonundan oluşmaktadır. SHA-256 çalışmamızda kullanıldığından bu kısımda açıklama yapılıp teknik detaylarından bahsedilecektir. SHA-256, 256 bit olup 32 byte 64 hexadecimal boyutundadır. Veriler hash değerine çevrildiğinde bu işlemin tersi geçekleşemez ve verideki en küçük değişiklik hash bilgisinde büyük değişikliğe sebep olabilmektedir.
(SHA-2, 2019)

Örneğin:

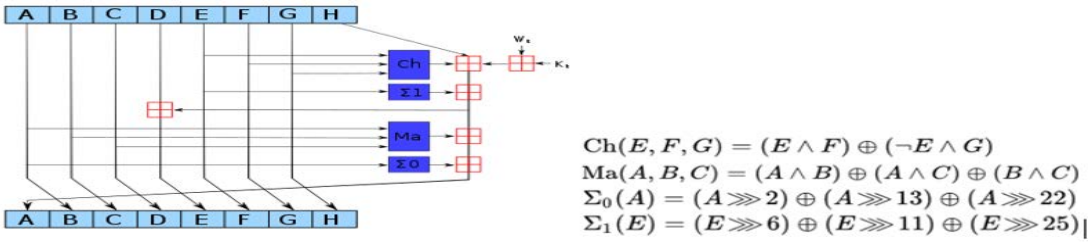
“hello” kelimesinin hash değeri:

185f8db32271fe25f561a6fc938b2e264306ec304eda518007d1764826381969 olan

“Hella” kelimasi için ise:

62197ce641a17c96cf39ed51e72e86fd486f3de7b0135f8b8f0f7364dc010e54

Şeklinde bir hash değeri üretilmektedir. “o” harfinin “a” ile değişmesi sonucu oluşan bu büyük farklılık SHA algoritmasının deterministik bir özelliğe sahip olduğunu göstermektedir. Şekil 2.7. de SHA-2 için diyagram gösterilmiştir.



Şekil 2.7 SHA-2 Ailesi Diyagramı

(SHA-2, 2019)

SHA-2 fonksiyonu, yaygın olarak güvenlik uygulamalarında ve protokollerinde kullanılmaktadır (TLS, SSL, PGP, SSH). Aynı zamanda bitcoin gibi para birimi işlerini doğrulamak ve para iletimlerini kanıtlamak için SHA-256 kullanılır. SHA-2‘nin çalışma hızındaki artıştan dolayı kripto temelli ispatlama çalışmalarında daha fazla kullanılmaya başlandı. Asic denilen kazım cihazlarının büyük bir kısmı bu algoritmayı kullanmaktadır. SHA karşılaştırmasını, özelliklerine göre karşılaştırdığımızda Çizelge 2.1.’deki gösterimi elde etmekteyiz.

Çizelge 2.1 SHA Algoritması Karşılaştırması

	SHA-1	**SHA-256**	**SHA-384**	**SHA-512**
Mesaj Özet Boyutu	160	256	384	512
Mesaj Boyutu	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$
Blok Boyutu	512	512	1024	124
Kelime Boyutu	32	32	64	64
Adım Sayısı	80	64	80	80
Güvenlik	80	128	192	256

(Comparison Of Cryptographic Hash Functions, 2019)

BÖLÜM 3

BLOK ZİNCİRİ

Çalışmamızda yer alacak olan hash(özet) kavramını önceki bölümlerde açıklamıştık. Şimdi çalışmamızda kullanacağımız hem merkle ağacının hem de hash ‘in etkin bir şekilde kullanıldığı blok zinciri teknolojisinden bahsedeceğiz. Blok zinciri, en kısa tanımıyla şifrelenmiş işlem takibi sağlayan bir veri tabanıdır. Internet teknolojileri ile etkileşimli diğer bir katman aynı zamanda kripto paraların arkasındaki temel teknolojidir.

1992’de tasarıma Merkle ağaçları eklendi ve birkaç belgenin bir bloğa toplanmasını sağlayarak sistem daha verimli hale getirildi. Fakat bu teknoloji kullanılmadı ve patenti 2004 yılında sona erdi. Bitcoin başlangıcından 4 yıl önce.,2004 ‘te bilgisayar bilimcisi ve kriptocu aktivist Hal Finney RPoW adında bir sistem geliştirdi. Bu sistem değiştirilemez ve geri ödenemez Hashcash temelli proof of work jetonu alma üzerine çalışıyordu karşılığında RSA-imzalı jeton yaratılıyordu ve bu jeton kişiden kişiye gönderilebiliyordu.

Rpow, çift harcama sorununu, kayıtlı tokenların mülkiyetini kayıtlı bir sunucuda kayıtlı hale getirerek çözdü. Bu sunucunun tasarımı, doğruluğunu ve bütünlüğünün kullanıcılar tarafından dünyanın herhangi bir yerinden gerçek zamanlı olarak doğrulanmasına izin veriyordu ve bu eski bir tipti.

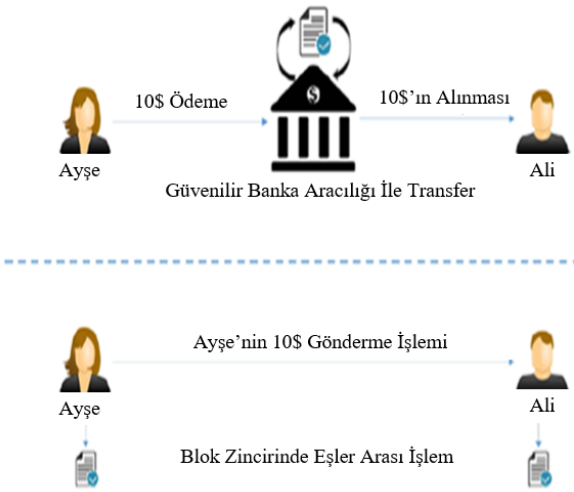
(Singhal, Dhmeja, & Panda, 2018)

2008 sonlarında, merkeziyetsiz eşler arası bir elektronik sistem tanıtan bir whitepaper takma isimli “Satoshi Namakato” tarafından yayınlandı. Bitcoinler, bireysel madenciler tarafından Proof of Work mekanizması kullanarak ödül elde etmek için çıkarılır ve daha sonra ağın merkeziyetsiz node’ları tarafından doğrulanır.

2009 ‘da ilk bitcoin bloğunun Satoshi Namakato tarafından kazılmasıyla bitcoin hayata geçmiş oldu. Bu ilk ödül 50 Bitcoin’di. İlk bitcoin transferi 12 Ocak 2009’da gerçekleşti. 2013’te Vitalik Buterin merkeziyetsiz programlar için dağıtılmış hesaplama platformu olan Ethereum’un kurulumuna başladı. Bu platform “Akıllı kontratlar” adında bir script fonksiyonu barındırıyordu. Akıllı kontratlar, Ethereum Blok zinciri üzerine yerleştirilmiş ve burada çalışan program yâda scriplerdir. Bu gelişim, blok zinciri mimarisi üzerinde finans başta olmak üzere, sağlık, sigortacılık, tedarik zincirleri gibi birçok alanda çalışma yapılmasına olanak sağlamıştır.

3.1 Eşler Arası İşlem

Blok zinciri sistemi veriyi iletirken banka, gümrük vb bir kuruma ihtiyaç duymaz. Veri iletişimi gönderici ile alıcı arasında birebir yapılır. Eşler arası işlem Şekil 3.1’de gösterilmiştir.



Şekil 3.1 Eşler Arası İşlem

(Singhal, Dhmeja, & Panda, 2018)

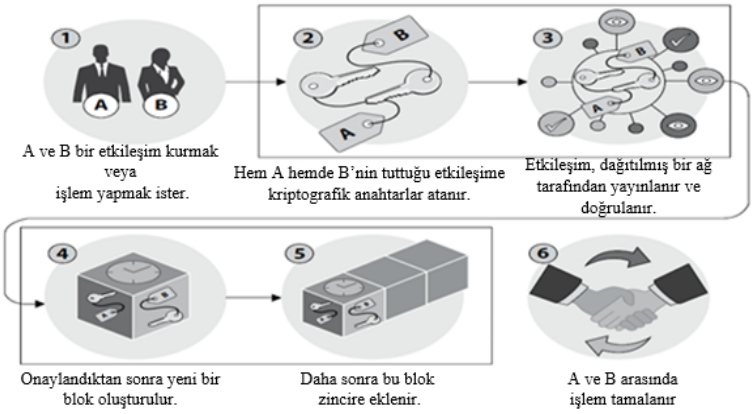
3.2 Blok Zincir İşlemi

Blok zinciri üzerinde geçerli bir işlem ya da etkileşim yapılmak istediğinde aşağıdaki adımlar ile gerçekleşmektedir:

1. A ve B arasında bir etkileşim ya da işlem yapılmak istenir
2. Şifreleme anahtarı hem A hem B’nin tuttuğu etkileşime atanır.
3. Önce doğrulama işlemi yapılır ve yeni blok yaratılır
4. Blok zincire eklenir

5. İşlem A ve B arasında tamamlanır.

Blok zinciri işlem akış örneği Şekil 3.2'te gösterilmiştir.

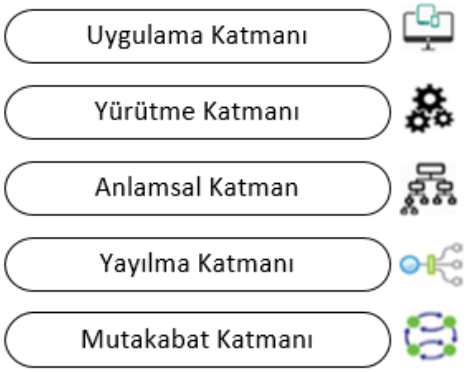


Şekil 3.2 Blok Zinciri İşlem Akışı

(Bambara & Allen, 2018)

3.2 Blok Zinciri Katmanları

Blok zinciri 5 katmandan oluşur bu katmanlar şekil 3.3'te gösterilmiştir.



Şekil 3.3 Blok Zinciri Katmanları

(Singhal, Dhmeja, & Panda, 2018)

3.2.1 Uygulama Katmanı

Kullanıcı ile etkileşimin sağlandığı ve istenilen fonksiyonların kodlandığı katmandır. Genellikle, istemci taraflı yazılım geliştirme gibi geleneksel teknoloji scriplerini, program yapılarını ve Api'leri içerir.

(Singhal, Dhmeja, & Panda, 2018)

3.2.2 Yürütme Katmanı

Yürütme Katmanı, bir blok zinciri ağındaki tüm düğümlerde yer alan uygulama katmanı tarafından verilen talimatların yerine getirildiği katmandır. Talimatlar basit veya Akıllı Sözleşme (Smart Contract) şeklinde birden çok komut kümesi olabilir. Tüm düğümlerde tutarsızlıkları önleyip aynı değer girişlerini ve daima aynı çıktıyı üreterek program ya da scriptlerde yürütülmesini zorunlu kılar. Sistem durumu, bir akıllı sözleşme işleme alındıktan sonra çağrılır.
(Singhal, Dhmeja, & Panda, 2018)

- Not: Akıllı sözleşme, yürütülebilir kod ve özel durumlara sahip bir hesap türüdür.

3.2.3 Anlamsal Katman

Bu katman işlem ve blokların bir düzene sahip olmasından ötürü mantık katmanı olarak işlev görür. Blokların birbilerine nasıl bağlanacağını tanımlar. Bir işlem'in geçerli olup olmadığını Yürütme katmanı aracılığı ile doğrulayan talimatlar kümesine sahiptir. Örneğin işlem bitcoin transferi ise ister meşru bir harcama isterse Double-spend attack olsun bu katman doğrulamayla yetkilidir.

3.2.4 Yayılma Katmanı

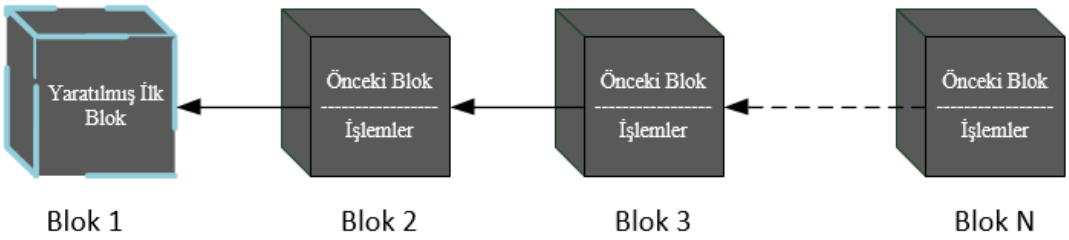
Yayılma Katmanı, eşler arası iletişim katmanıdır diğer düğümlerin mevcut durumlarıyla ilgili olarak birbiriyle konuşup senkronize olmasından sorumlu katmandır.
(Singhal, Dhmeja, & Panda, 2018)

3.2.5 Mutabakat Katmanı

Mutabakat katmanının amacı öncelikli olarak defterin tutarlı kalmasını sağlayarak tüm düğümleri kabul etmektir. Blok zincirinin güvenliği ve emniyeti bu katmanda sağlanır. Bitcoin ve ethereum Proof Of Work (PoW) yani rastgele seçilen bir düğümün önereceği bir bloğun sistemle mutabık kalma mekanizmasına dayalıdır. Blok önerildiğinde tüm meşru işlemlerle geçerli bir blok olup olmadığını kontrol eder ve PoW bulmacasını düzgün bir şekilde çözer. Bu blok kendi bloğuna eklenir.
(Singhal, Dhmeja, & Panda, 2018)

3.3 Blok Yapısı

Blok zinciri ağındaki her düğüm isim ve işlem bilgisinin olduğu özdeş kopyalara sahiptir. Her bloğun başlığı kendinden önceki bloğun referansı özet (Hash) bilgisi içerir. Böylece önceki bloğun işlem bilgisi değiştirilemez. Bloğun diğer bir kısmı içerik kısmı miktarları, doğrulanmış işlem listesini ve daha fazlasını içerir. Blok yapısı Şekil 3.4.'te gösterilmiştir.



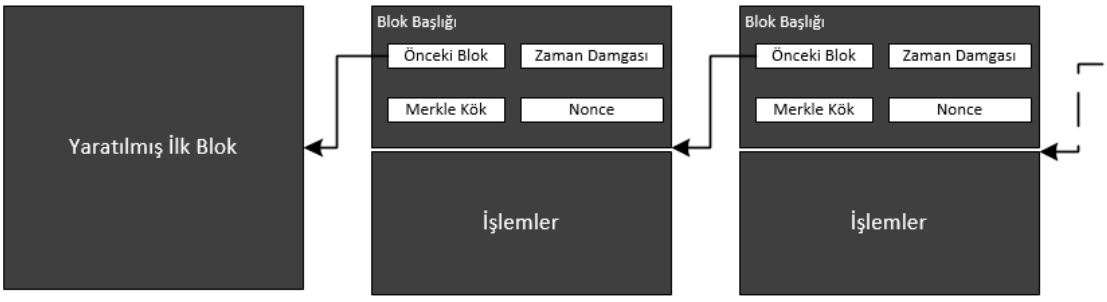
Şekil 3.4 Blok Zinciri Yapısı

(Singhal, Dhmeja, & Panda, 2018)

Blok başlığı 3 adet temel bilgiyi içermektedir. İlki bir önceki bloğun oluşturduğu hashlenmiş şifresidir. İkinci metadata seti, zorluk, zaman damgası ve proof of work katsayısı ile ilgilidir. Üçüncüsü ise merkle ağacı yapısına ait bilgilerdir. Genesis bloğu, 2009 yılında oluşturulan ilk blok yapısının adıdır. Şimdiye kadar oluşturulmuş blok zincirlerinde ki tüm blokların üretildiği bloktur. Genesis blokunun hashlemiş imzası aşağıdaki gibidir:

000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

Blok iç yapısı Şekil 3.5.'te gösterilmiştir.



Şekil 3.5 Blok İç Yapıları

(Singhal, Dhmeja, & Panda, 2018)

3.4 Blok Başlığı

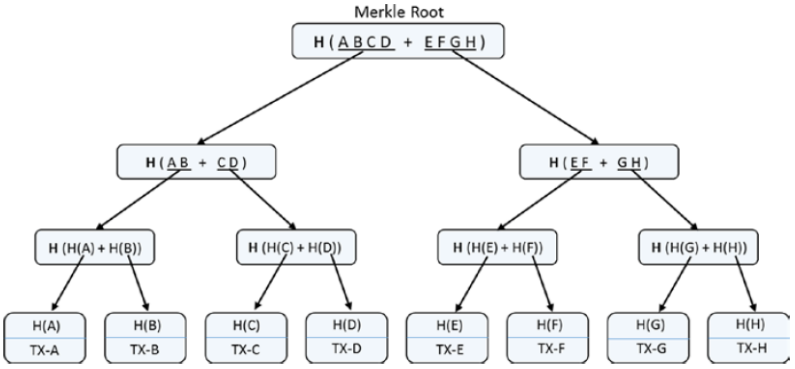
Bir bloğun parmak izi olarak, blok başlığında yer alan kriptografik şifredir. Sha-256 algoritması kullanılarak oluşturulmuştur. Blok başlığında yer alan boyutları Çizelge 3.1' de gösterilmiştir.

Çizelge 3.1 Blok Başlığı Özelliği ve Boyutları

Boyut	Özellik	Açıklama
4 bytes	Version	Versiyon
32 bytes	Previous Block Hash	Zincirde ki bir önceki node'un hashlenmiş şifre
32 bytes	Merkle Root	Merkle ağacının hashlenmiş yapısı
4 bytes	Timestamp	Oluşturulma zamanı unix timestamp cinsinden
4 bytes	Difficulty Target	Proof of work algoritmasının zorluk derecesi
4 bytes	Nonce	P-o-w için kullanılan sayaç

3.5 Merkle Ağacı (Merkle Tree)

Merkle ağacı bir bloktaki verinin parça parça teslim edilmesine olanak tanımaktadır. Bir düğüm blok başlığını bir kaynaktan ağacın kendiyle ilgili küçük bir bölümü başka bir kaynaktan indirebilir ve bilginin doğruluğundan emin olunabilir. Bunun işe yaramasının nedeni, özet(hash) değerlerin yukarı doğru yayılmasıdır. Kötü niyetli bir kullanıcı sahte bir işlemle Merkle ağacının yaprak değerinde takas etme girişiminde bulunursa, bu değişiklik yukarıdaki düğümde bir değişikliğe neden olur böylece bloğun hash değerinin tamamen farklı bir blok olarak kaydedilmesine ve geçersiz bir çalışma kanıtı olarak kaydedilmesine sebep olur. Örnek merkle ağacı Şekil 3.6. 'da gösterilmiştir.



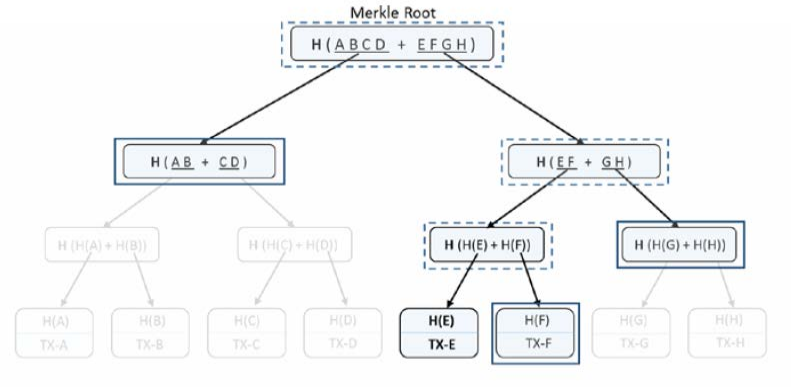
Şekil 3.6 Merkle Ağacı

(Singhal, Dhmeja, & Panda, 2018)

Özellikleri:

- Merkle Ağaçları, ikili şifreleme ağacının işaretçisi vardır.
- Blokların sahip olduğu işlemleri doğrulamanın en etkili yoludur.
- Doğrulama Log((n)) zamanında yapılır

Merkle ağacı hesaplama örneği Şekil 3.7' de gösterilmiştir.



Şekil 3.7 Merkle Ağacı Hesaplama Örneği

(Singhal, Dhmeja, & Panda, 2018)

- Eğer bir işlem Merkle ağacına ait ise tüm ağacı veya düğümleri bilmek gerekmez.
- Daha az bellek ve disk alanı kullanılır.

Bu işlemde merkle ağacı kökünde yer alan hash bilgisinin doğruluğunu control etmek için düz çizgili blok değerlerini bilmek yeterlidir. Sadece noktalı blok hash'ler hesaplanacaktır. 8 Adet işlem yani n=8 için $\log 2^8 = 3$ doğrulama yeterli olacaktır

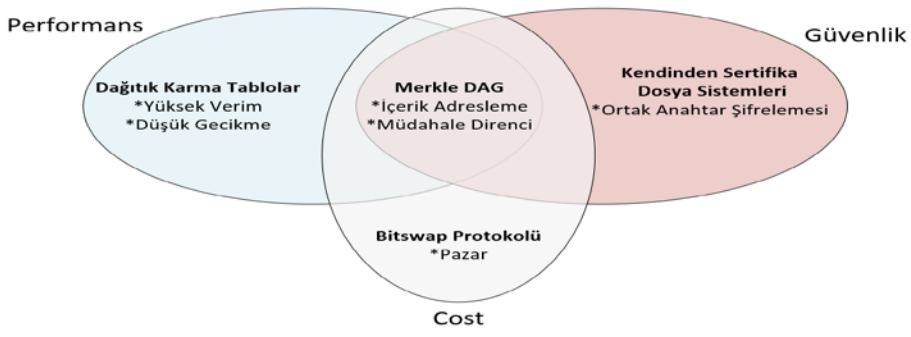
BÖLÜM 4

GEZEGENLER ARASI DOSYA SİSTEMİ (IPFS)

IPFS (Interplanatery File System), protocol Labs tarafından Ocak 2015'de ortaya çıkarılan peer-to-peer iletişim ile veri depolama ve paylaşım imkânı sağlayan dağıtık dosya sistemidir. Her saklanan dosya için işlemi yapan kullanıcıya benzer bir hash verilir. Bu işlem sonucu veri güvenliği korunurken, depolama ve hesaplama maliyeti azalır. Doğrulama işlemi "Merkle DAG" ile yapılır ayrıca düğümlar arasında veri aktarımını "BitSwap" ile sağlanır. En büyük özelliklerinden biri dosya versiyonlamadır.
(Benet, 2014)

IPFS, BitTorrent ve Git gibi p2p sistemlerden evrimleşmiş bir sistemdir ve büyük miktarda verileri versiyonlamak ve dağıtmak için yeni bir platform sunar. Tarafsız bir taşıma katmanıdır ve TCP, UDT, TOR hatta Bluetooth ile iletişim kurabildiği anlamına gelir.
(Bambara & Allen, 2018)

IPFS, temelde DHT ve DAG veri yapılarını kullanmaktadır. DHT, (key, value) çiftlerini tutar. Katılan herbir düğüm verilen anahtar ile ilgili değeri verimli bir şekilde getirir. Bu eşleştirmeyi anahtardan değere kadar sürdürme işlemini yeni katılımcı düğümler eklendiğinde en asgari düzeyde bozulmaya neden olacak şekilde dağıtmıştır. IPFS Teknik özellikleri olarak Şekil 4.1'de gösterilmiştir.



Şekil 4.1 IPFS Teknik Özellikleri

4.1 IPFS Düğümleri

Düğümler birbirlerine bağlanıp dosya, döküman, video vb nesnelerin transfer işlemlerini gerçekleştirirler. IPFS protokolü birbirinden farklı fonksiyonlardan soru alt özelliklere ayrılmıştır.

1. Ağ (Network): farklı network'lerin altında çalışır ve kişiler arasındaki bağlantıyı yönetir.
2. Kimlik (Identities): Düğüm kimliği oluşturur ve doğrulanmasını yönetir.
3. Yönlendirme (Routing): Özel eşlerin ve nesnelerin yerini bulmak için bilgileri korur. Hem yerel hem de uzak sorgulara yanıt verir.
4. Takas (Exchange): Etkili blok dağıtımını düzenleyen yeni bir blok değişim protokolü (BitSwap). Pazar olarak modellenmiştir ve haftalık veri çoğaltmaya yardımcı olur.
5. Dosyalar (Files): Versiyonlanmış dosya sistemi hiyerarşisi vardır.
6. İsimlendirme (Naming): Kendi kendini onaylayan bir değişken isim sistemi
7. Nesneler (Objects): İçerik bazlı adresleme ile değişmez nesnelerin bir Merkle DAG'ı ve veri yapılarını isteğe bağlı temsil etmek için kullanılır

(Benet, 2014)

Açık anahtarlı (Public Key) şifreleme düğüm adresleme sistemine yerleştirilmiştir ve içerik adresleme içeriğini kullanır. Düğüm (Node) ve içerik adresleme (Content Adresses) IPNS olarak isimlendirilen merkeziyetsiz isim sisteminde depolanır. P2P ağındaki düğümlerin herbiri gizli anahtar (Private Key) tutarlar ve açık anahtarları serbest bırakırlar ve bu key ile veriler gönderilmeden önce şifrelenirler ve karşı taraf gizli anahtar ile şifreyi çözüp veriye ulaşır. Birçok güvenlik sistemi bu adresleme sistemine entegre olmuştur.

(Bambara & Allen, 2018)

4.2 IPFS İçerik Adresleme

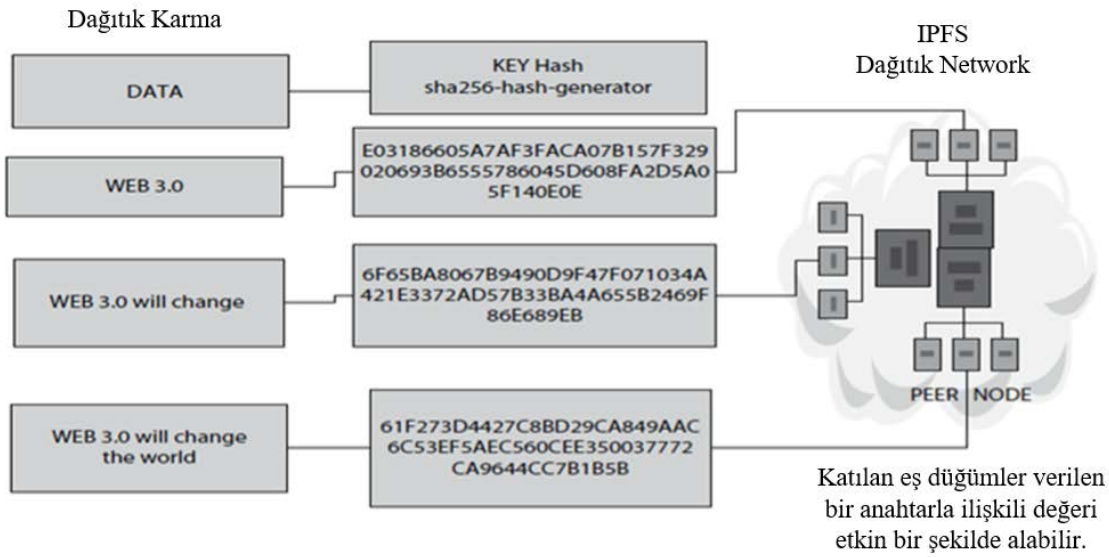
İçerik adresleri sisteme aktarılan verinin bir parçasının özetinin(hash) çıkarılmasıyla üretilir. Eğer sisteme eklenmiş bir verinin aynısı eklenmeye çalışırsa zaten sistemde aynı link çözümlemesi olacağı için IPNS bu kopyanın oluşturulmasını güvenlik sebebiyle engeller.

4.3 IPFS Nesneleri

Ipfs veri yapısında 2 alan vardır bunlar:

- Veri (Data): 256kb ‘ta eşit ya da daha az olan boyutta ikili veri
- Bağlantılar (Links): Bağlantı yapılarının bir dizisi

Bu veri yapılarının gösterildiği, içerik adresleme ile dağıtık hash yapısı örneği Şekil 4.2’de gösterilmiştir.



Şekil 4.2 İçerik Adresleme ile Dağıtık Hash Yapısı Örneği

(Bambara & Allen, 2018)

Her bir link’in 3 tane veri alanı vardır:

1. Ad (Name): Link’in adı
2. Hash: Link edilmiş IPFS nesnelerinin hash’i
3. Boyut (Size): Link edilmiş IPFS nesnelerinin kümulatif boyutu

IPFS nesneleri **Base58** kodlu hash olarak ifade edilmektedir ve tüm hash bilgileri “**Qm**” ile başlamaktadır bunun sebebi ise çoklu özet (multihash) olmasından ileri

gelmektedir. Çoklu Özet kavramı 4.6 Ipfs Çoklu Özet kısmında detaylı açıklanacaktır.

4.4 IPFS Dosyaları

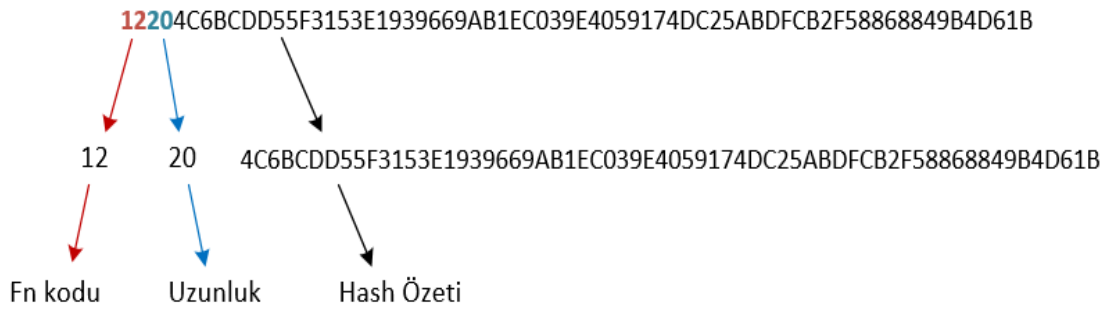
IPFS ‘de 256kb ‘ta eşit ya da daha küçük boyuttaki dosyalar küçük boyutlu dosyalar olarak nitelendirilmektedir ve link içermezler. Bu sebeple link dizisi boş olur. Dosya isimlerinin farklı içeriklerin aynı olması durumunda aynı hash üretilir isimlerinin farklı olması durumunda farklı hash üretilmez. 256kb dan büyük olan dosyalar büyük boyutlu dosyalardır ve bu dosyaların parçaları için birden fazla link oluşturulur.
(Bambara & Allen, 2018)

4.5 IPFS Dosya Versiyonlama

IPFS, dosya versiyonlama yapısına değinmek gerekirse git tarafından sürümlenmiş dosya sistemlerine izin vermek için kullanılan veri yapılarını temsil edebilir. Örnek olarak bir dökümanımızın revizyona ihtiyacı olduğunu varsayalım bu durumda yeni bir dosya sürüm v2 ek olarak oluşturulur bu her sürüm için tekrarlanır ve sonu yoktur. Her oluşturulan nesne bir önceki nesneye link’ler aracılığı ile bağlanır. Hepsi de tüm zinciri referans alan aynı verilere ait bağlı bir sürüm zinciri oluşturur.
(Bambara & Allen, 2018)

4.6 IPFS Çoklu Özet

IPFS tarafından üretilen tüm çoklu özetler “**Qm**” ile başlamaktadır. Peki bu işlem nasıl gerçekleşiyor? Çizelge 4.1.’de gösterildiği üzere her bir sha algoritmasının bir kodu vardır biz örnek olarak sha2-256’yı baz alacağız Çizelge 4.1.’de Sha2-256 için kod’un **0x12** olduğu görülmekte ve hex cinsinden uzunluğu **20** (veya 32 byte) dır.
” merhaba” kelimesi için oluşacak sha2-256 hash sonucu:
4C6BCDD55F3153E1939669AB1EC039E4059174DC25ABDFCB2F58868849B4D61B’ dir.
Sha2-256 kodu ve uzunluğu eklenir Şekil 4.3 ‘de örnek gösterilmiştir.



Şekil 4.3 Hash Değerine Uzunluk ve Kod Eklenmesi

Uzunuk ve kod eklendikten sonra Base58 dönüşümü yapılır ve aşağıdaki sonuç elde edilir:

QmTUy36jgKQSh8RqkKbEicB4DA4MDFks39BvB1n7vSER14

Böylece benzer görünümlü harfler ihmal edilmiştir. 0 (sıfır), O (büyük harf O), I (büyük harf i) ve 1 (Küçük harf L) ve alfa nümerik olmayan karakterler + (artı ve / (eğik çizgi) ihmal edilmiştir. Burada amaç insan tarafından biraz daha okunabilir hale gelmesidir.

Çizelge 4.1 Sha Ailesi Kodları

Kod	Fonksiyon
0x11	Sha1
0x12	Sha2-256
0x13	Sha2-512
0x14	Sha3-512
0x15	Sha3-384
0x16	Sha3-256
0x17	Sha3-224

4.7 Swarm

Swarm, Ethereum web3 yığının yerel bir temel katman hizmeti olan dağıtılmış bir depolama platformu ve içerik dağıtım hizmetidir. Swarm ve Ipfs yeni nesil internet için verimli merkezi olmayan depolama katmanları için kapsamlı çözümler sunar. Her ikisinde de hedefler ve teknoloji oldukça benzerdir. Her ikisi de Web 3.0 vizyonu için gerekli tüm dağıtılmış belge depolama özelliklerine sahip depolama katmanları olarak hizmet ederler.

Teknik özelliklerine değinecek olursak:

- 0 kesinti
- Sansüre dayanıklılık
- Düşük gecikme
- İçeriği ön belleğe alarak verilim ve otomatik ölçeklendirme
- İçerikte kalıcı versiyonlanmış arşiv

Swarm farklı iletişim katmanları ve eş yönetim protokolleri kullanır. Ethereum blok zinciri ile entegrasyonları vardır. Akıllı sözleşmeler ve yarı sabit (Semi-stable) eş havuzundan faydalırken Filecoin'i madenciliği geri kazanılabilirliğin bir kanıtını olarak kullanıyor. Düşük gecikmeli gerçek zamanlı etkileşimli web uygulamalarına hizmet etmenin yanı sıra, özellikle ağdaki düğümlerin bant genişliklerini ve depolama alanlarını bir araya getirmelerini sağlar. Swarm'ın planlanan özellikleri bütünlük koruması, Url tabanlı adresleme, rastgele erişim, Ethereum Ad hizmeti aracılığıyla etki alanı adı çözümesi ve şifreleme desteğidir.
(Bambara & Allen, 2018)

BÖLÜM 5

KİŞİSEL VERİLERİN KORUNMASI

Dünya'da internetin yaygınlaşmasıyla küresel bir ağ oluşmuştur ve kişisel verilere kolayca ulaşılabilmektedir. Bu veriler ülkeler veya kıtalar arasında hızlı bir şekilde paylaşılabilmektedir. Kişisel verilerin başkalarının eline geçmesi bu risk türlerinden biridir. Bu verilerin korunması için kişilere ve devlete düşen görevler vardır. Ülkemizde (KVKK) Kişisel verilerin korunması kanunu 24.03.2016 tarihli resmî gazetede yayımlanarak yürürlüğe girdi. Kanunun 3/d maddesinde "Kişisel veri, kimliği belirli veya belirlenebilir gerçek kişiye ilişkin her türlü bilgiyi ifade eder" olarak tanımlanmıştır. Bu kapsamda ad, soyad, eposta, telefon, adres, medeni hali, banka hesapları, telefon rehberi, sosyal medya hesapları, görüntü ve ses kayıtları "kişisel veri" olarak tanımlanabilir.
(Çetin, 2014)

Birçok işletme tarafından depolanan müşteri kimliği, dokümanlar ve belgeler, işlem dökümleri gibi verilerin korunması ciddi önem arz etmektedir. Bu konudaki önem kimlik hırsızlığı, kimlik bilgilerinin 3. Kişilerce verilerin kötüye kullanılmasını durdurmak olarak değerlendirilebilir. Yukarıda bahsi geçen bilgilere istinaden banka ve kredi kartı bilgileri de hassas durumları içerdiğinden tüm bu bilgilerin 6698 sayılı KVKK uyarınca korunması ve işletmelerin belirli ilkelere uyması esastır.

KVKK'un kişisel verilerin korunmasına yönelik yer alan temel ilkeleri:

- Hukuka ve dürüstlük ilkelerine uymak
- Doğru ve güncel olmak
- Belirli, açık ve meşru amaçlar için işlenmek
- Toplandıkları ve işlendikleri amaç ile bağlantılı, sınırlı ve ölçülü olmak

- İşlendikleri amaç için gerekli olan süre kadar muhafaza edilmek.

Bu kapsamlarda verinin şifrelenip, sınıflandırılması, maskelenmesi, değiştirilmesini ve kaybını önlemek için önlemler değerlendirilmelidir.
(BeyazNet, 2019)

5.1 Kanun Çıkmadan Önce Toplanan Verilerin Durumu

6698 sayılı KVKK'nın tarihinden önce depolanan veriler kanun tarihinin yayım tarihinden itibaren 2 yıl içinde kanun hükümlerine uygun hale getirilmelidir. Hükümlere aykırı olduğu belirlenen kişisel veriler yok edilir veya anonim hale getirilir.
(Dağsalı, 2019)

5.2 KVKK Kapsamında Şirketlerin Yükümlülükleri

Şirketler kişisel veri kayıtlarda tutarak müşterinin veri gizliliğini korumalı, herhangi bir güvenlik ihlali olduğunda ise müşteriye bilgi vermelidir. Kurum ve kuruluşlar bu bilgilere dayanarak kişi bilgilerinin 3. Kişilere ulaşmaması için bilgi güvenliğine yönelik çalışmalar yürütmeleri gerekmektedir.
(BeyazNet, 2019)

5.3 Kişisel Verilerin Şifrelenmesi

Her türlü bilgi ve belgenin güvenliğinin temelini şifreleme oluşturmaktadır. Kötü niyetli kişilerin veriye ulaşmasını, yetkisiz erişimler ile bilgiyi elde edilememesi en temel yöntemidir. Önemli ve değerli bilgi ve belgelerin kurumlar ve kuruluşlar tarafından güvenli bir şekilde muhafaza edilmesi gerekmektedir.
(BeyazNet, 2019)

5.4 Kişisel Verilerin Yedeklenmesi

Verilerin herhangi bir olumsuz durumdan etkilenmemesi için güvenli alanda bir veya birden fazla kopyasının oluşturulması işlemine kısaca yedekleme diyebiliriz. Herhangi bir risk durumunda silinme, değişme vb orijinal veriye geri dönüş için sistem üzerinde imkân sağlar. Yedekleme sürecinde, bulut teknolojilerinden, Nas (Network-Attached Storage) sistemleri, SAN (Storage Area Network) vb depolama çözümlerinden yararlanılabilir. Bunların yanı sıra günümüzde dağıtık (Distributed) yedekleme alanları ortaya çıkmıştır. Bu yapılarda veri dağıtık bir yapıda ve çeşitli

kriptolojik işlemlerden geçirilerek güvenli bir şekilde tutulmaktadır.
(BeyazNet, 2019)

5.5 Sızma Testleri

Bilgi sistemlerinin güvenlik açıkları nedeniyle bilgi kayıplarına hangi türden bir zafiyet ile zarar göreceğine ilişkin yapılan testlerdir. Böylece zafiyet test edildikten sonra gerekli önlemlerin alınması için büyük önem taşımaktadır. Çeşitli zafiyet raporları çıkartılarak periyodik olarak alınacak önlemler ile güvenliğin arttırılmasında büyük katkı sağlanabilmektedir. Bu çeşit önlemler ile tehdit ve riskleri en aza indirecek önemler alınarak siber güvenlik sağlanmalıdır.
(BeyazNet, 2019)

BÖLÜM 6

MAKİNE ÖĞRENMESİ

Çalışmamızda belgelerin ve dokümanların kişisel veri içerip içermediği bilgisini makine öğrenmesi ile gerçekleştireceğiz bu bölümde makine öğrenmesinin tanımı, kavramları, algoritmalar, çalışma prensibi ve uygulanacak yöntem üzerinde genel bilgiler verilecektir. Bu bilgiler ışığında kısaca makine öğrenimi nedir sorusu için bilgisayarlardan herhangi bir giriş verisi ya da veri tabanları gibi veri türlerine dayalı öğrenimi olanaklı kılan algoritmaların tasarım ve geliştirme sürecini konu edinen bir bilim dalıdır diyebiliriz. Makine öğrenmesinin temel prensibi, giriş verisini alarak algoritmalar oluşturmak ve sonuçları yeni veriler ortaya çıktıkça güncellerken bir çıktıyı tahmin etmek için istatiksel analiz kullanmaktır. Başlıca kavramları, denetimli öğrenme, denetimsiz öğrenme, yarı denetimli öğrenme, takviyeli öğrenme ve yoğun öğrenme olarak tanımlayabiliriz.

6.1 Makine Öğrenmesi Kavramları

6.1.1 Denetimli Öğrenme

Bu öğrenme yönteminde, giriş ve çıkışları içeren bir veri kümesinden yararlanılır bir algoritma kullanılarak makine öğretilir. Bu öğretim sürecinde veri içerisindeki tanımlamaları ve değerleri algoritma öğrenir ve tahminlerde bulunur bu tahminlerdeki hatalar geliştirici tarafından düzenlenir. Bu işlem algoritmada doğruluk düzeyi yüksek seviyeye gelinceye kadar devam edilir bu işlemi yaparken aşağıdaki yöntemlerden faydalanır:

Tahmin: Mevcut değerleri çeşitli parametreler ile gelecekle ilgili sonuç çıkarma sürecidir.

Sınıflandırma: Eğitim aşamasından sonra yeni değerlerin hangi kategoride yer alması gerektiğini belirtir.

Regresyon: iki ya da daha çok değişken arasındaki ilişkiyi ölçmek için kullanılan analiz metodudur.
(Regresyon Analizi, 2019)

6.1.2 Yarı Denetimli Öğrenme

Denetimli öğrenmeden farklı olarak hem etiketlenmemiş büyük ölçekli veriyi hem de etiketlenmiş küçük miktarda veriyi kullanır. Amaç, etiketlenmemiş verileri kullanarak makine öğrenmesi algoritmaları ile etiketlemeyi öğretmektir.

6.1.3 Denetimsiz Makine Öğrenmesi

Bu öğrenme işleminde içerisinde sınıf bulundurmayan verilerden faydalanılarak yapıları ve kalıpları tanımlamak için veriyi inceler. Bu işlem sonunda korelasyon ilişkilerini ortaya çıkarır. Böylece veriler gruplanarak daha düzenli görünmesi sağlanır.

6.1.4 Takviyeli Öğrenme

Makine öğrenmesi sonucunun daha iyi doğruluk düzeyinde sonuç vermesi amacıyla geliştirici tarafından doğru ya da yanlış olarak işaretlenerek sistemin iyileştirilmesidir.

6.1.5 Yoğun Öğrenme

Karmaşık olan çoklu işlem katmanlarından ortaya çıkarılmış veriler ile soyutlamalar kullanılarak oluşturulan model girişlerine bağlı bir veya birden çok algoritmalarla geliştirilmiş makine öğrenmesidir.

6.2 Sınıflandırma Yöntemleri

- İkili Sınıflandırma (Binary Classification)

İkili sınıflandırma algoritmaları 2 olası sonuç ile sınıflandırma yapmaktadır. (Pozitif-Negatif)

- Çoklu Sınıf Sınıflandırma (MultiClass Classification)

İkili sınıflandırma sürecinden farklı olarak, tahmin için puan eşiği seçilmesine gerek yoktur. Öngörülen cevap, tahmin edilen en yüksek puana sahip sınıftır.

- Çoklu Etiket Sınıflandırma (Multi Label Classification)

Herhangi bir verinin bir veya birden fazla sınıfa dahil olduğunu belirleyen sınıflandırmadır.
(Binary Classification, 2019)

6.3 Sınıflandırma Algoritmaları

6.3.1 Naive Bayes (Denetimli Öğrenme)

Basit bir ifadeyle, bir Naive Bayes sınıflandırıcısı, bir sınıftaki belirli bir özelliğin varlığının diğer herhangi bir özelliğin varlığı ile ilgisiz olduğunu varsayar. Naive Bayes modelinin oluşturulması kolaydır ve özellikle çok büyük veri kümeleri için kullanışlıdır. Sadeliği ile birlikte, Naive Bayes'in oldukça karmaşık sınıflandırma yöntemlerinden bile daha iyi performans gösterdiği bilinmektedir. Naïve Bayes denklemi Şekil 6.1'de gösterilmiştir.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

Şekil 6.1 Naive Bayes Denklemi

P(A|B); B olayı gerçekleştiği durumda A olayının meydana gelme olasılığıdır.
P(B|A); A olayının gerçekleştiği durumda B olayının meydana gelme olasılığıdır.
P(A) ve P(B); A ve B olaylarının önsel olasılıklarıdır.
(Naive Bayes Sınıflandırıcı, 2019)

6.3.2 Karar Ağaçları (Denetimli Öğrenme)

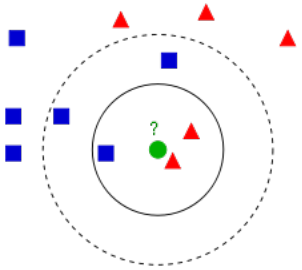
Karar ağaçları, karar verilerini eğitim verilerinden uyarmaya yönelik denetimli bir makine öğrenmesi tekniğidir. Bir öğe hakkındaki gözlemlerden hedef değer hakkındaki sonuçları gösteren bir tahmin modelidir. Karar ağacı algoritmaları öncelikle eğitim verileri, test verileri, sezgisel değerlendirme fonksiyonu ve durma kriteri işlevlerinden oluşur. Olası her sonuç dallanma yöntemini kullanan bir ağaç yapısına benzer, ağaçtaki her düğüm bir değişken üzerindeki testi temsil eder ve sonuçlar dal olarak temsil edilir.

6.3.3 Rastgele Ağaç (Denetimli Öğrenme)

Bu algoritma, regresyon, sınıfladırma ve eğitim aşamasında karar ağaçları gibi daha iyi sonuç için algoritmaları birleştirerek öğrenme işlemini gerçekleştirmektedir. Ağacın en üst düğümüne bir girdi girilir ve alt düğümlere doğru belirli değişkenlere bağlı olarak daha küçük kümelere bölünür.

6.3.4 Yakın Komşu (Denetimli Öğrenme)

Bu algoritma sınıflandırma yaparken çıkarılan özelliklerden sınıflandırılması beklenen verinin daha önceki verilerden k tanesine yakınlığına bakmasıdır. K değerine göre yeni bir veri sınıflandırılmak istendiğinde, önceki durumda sınıflandırılmış elemanlardan en yakın K tanesi seçilir. Bu elemanların dahil olduğu sınıfa yeni eleman dahil edilir. Bir örnek ile açıklayacak olursak Şekil 6.2 da K-NN sınıflandırma örneği gösterilmiştir.

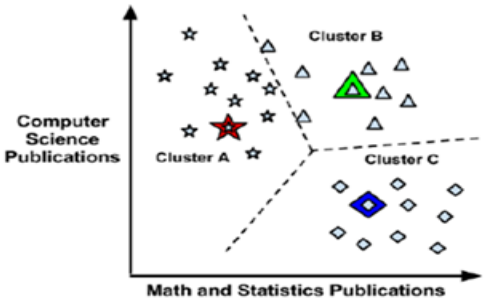


Şekil 6.2 K -NN sınıflandırması Örneği

Test numunesi (yeşil nokta) mavi karelere veya kırmızı üçgenlere göre sınıflandırılmalıdır. Eğer k = 3 ise (düz çizgi daire) kırmızı üçgenlere atanır. Çünkü iç daire içinde 2 üçgen ve sadece 1 kare vardır. Eğer k = 5 ise (kesikli çizgi çemberi) mavi karelere atanır (3 kareye karşı dış çember içindeki 2 üçgen).
(K-nearest Neighbors Algorithm, 2019)

6.3.5 K-Ortalama (Denetimsiz Öğrenme)

K-Means algoritması en sık kullanılan bir kümeleme yöntemidir ve birçok karmaşık kümeleme tekniklerine sahiptir. Bu algoritma kümeleme merkezleri için özellik alanında k noktaları seçerek başlangıç yapar. Bu merkezler kalan örnekleri destekler ve bu noktalar rastgele eğitim seti örneklerinden seçilir. Burada amaç her bir küme içindeki farkı minimize edip kümeler arasındaki farkı maksimum yapmaktır. Örnek olarak "Computer Science" ve "Math and Statistics" yayınlarını ele alalım k=3 seçildiğinde 3 adet küme oluşacaktır. Bu k noktaları yıldız, dörtgen ve üçgen olarak Şekil 6.3.' te gösterilmektedir.



Şekil 6.3 K-Means Kümeleme Örneği

Bu bilgiler dahilinde n özellik sayısı olmak üzere x ve y noktalarını hesaplanmak için Şekil 6.4'deki formülü kullanabiliriz.

$$\mathrm{dist}(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$$

Şekil 6.4 İki Nokta Arası Mesafe Formülü

Yukarıda verdiğimiz mesafe fonksiyonunu kullanarak her bir örnek ve küme merkezleri arasındaki mesafe R yi bulabiliriz.

1. 5 adet "Computer Science" ve 3 adet "Math and Statistics" yayını
2. 3 adet "Computer Science" ve 7 adet "Math and Statistics" yayını

2 adet özellik bilgisi yukarıdaki bilgilere göre mesafe fonksiyonunda tanımladığımızda;

R = Sqrt $((5-3)^{2} + (3-7)^{2})$

R= 4.2426 olarak elde ederiz.

(Lantz, 2013)

6.4 İkili Sınıflandırma Algoritmaları

Çalışmamız önemli ve önemsiz olarak bilgilerin ayrılması yönünde olacağından eğitim için ikili sınıflandırma algoritmalarını kullanacağız. Dinamik oluşturacağımız veri seti ve otomatik seçilmiş algoritma sayesinde sistemin kararlı çalışması sağlanacaktır. Sistemde 2 adet veri seti oluşturulacak ve algoritmaların verecekleri ortak karar ile belgeler önemli ve önemsiz olarak ayrılacaktır. Eğer 2 algoritmada doğru(true) değeri dönerse önemli evrak olarak kayıt edilecek eğer yanlış(false) dönerse önemsiz olarak kayıt edilecektir. Biri doğru diğeri yanlış olur ise kullanıcı kararına bırakılacaktır. Makine öğrenmesi fonksiyonları true/false değerine göre aşağıdaki değerlerden oluşmaktadır:

- Doğru Pozitif TP (True Pozitif): Gerçekleşen olumlu eylemin doğru olduğu durum
- Doğru Negatif TN (True Negatif): Gerçekleşen olumsuz eylemin doğru olduğu durum
- Yanlış Pozitif FP (False Pozitif): Gerçekleşen olumlu eylemin yanlış olduğu durum
- Yanlış Negatif FN (False Negatif): Gerçekleşen olumsuz eylemin yanlış olduğu durum

Belirtilen kategorilerin metrik değerleri ile birlikte Çizelge 6.1'de gösterilmiştir.

Çizelge 6.1 Sınıflandırma Modellerinde Yaygın Kullanılan Metrikler

Değer	Formül	Açıklama
Doğruluk	$\frac{TP+TN}{TP+TN+FP+FN}$	Makine öğrenmesinin doğru olarak yaptığı tahminlerin tüm tahminlere oranıdır.
Hassasiyet	$\frac{TP}{TP+FP}$	Doğru tahmin edilmiş pozitif gözlemlerin, öngörülen toplam gözlemlere göre oranıdır.
Duyarlılık	$\frac{TP}{TP+FN}$	Doğru tahmin edilmiş pozitif gözlemlerin gerçek sınıflardaki tüm gözlemlere oranıdır.
F1 Skoru	$\frac{2TP}{2TP+FP+FN}$	Hassasiyet ve Duyarlılığın ağırlıklı ortalamasıdır.

6.4.1 Ortalama Algılayıcı (Avaraged Perceptron)

Ortalama algılayıcı algoritması eğitim seti üzerinde aşırı uyma (Overfitting) problemini azaltmak amacıyla geliştrilmiştir. Denetimli bir öğrenme yöntemidir ve etiket sütunu içeren bir veri kümesi gerektirir. Yapay sinir ağının çok basit bir versiyonudur. Bu yaklaşımda girdiler doğrusal bir işleve dayalı olarak birkaç olası çıktıya sınıflandırılır ve ardından özellik vektöründen türetilen bir dizi ağırlıkla birleştirilir bu sebeple "perceptron" adı verilmektedir. Çizelge 6.2. 'de modül parametreleri gösterilmiştir.
(Collins,2002)

Çizelge 6.2 Ortalama Algılayıcı Modül Parametleri

Parametre	Oran	Tip	Varsayılan	Açıklama
Öğrenme Oranı	>double. Epsilon	Float	1.0	Stochastic Gradient Descent optimizer için Başlangıç öğrenme oranı
Maksimum iterasyon sayısı	>=1	İnteger	10	Eğitim seti üzerinde uygulanan Stochastic Gradient Descent iterations sayısı
Rastgele sayı	Hepsi	Integer		Model tarafından kullanılan rastgele sayı. Varsayılan olarak boştur.
Bilinmeyen kategori seviyeleri	Hepsi	Bollean	True	True ise her kategorik sütun için ek bir seviye oluşturur. Eğitim setinde uygun olamayan test veri setilerinin tüm seviyeleri bu ek sebiye ile eşleştirilir

6.4.2 Sdca Lojistik Regresyon (Logistic Regression)

Stokastik çift koordinat çıkış (Stochastic dual coordinate ascent) yöntemini kullanan bir ikili lojistik regresyon modelidir. Eğitilmiş model kalibre edilirek doğrusal fonksiyonun çıkış değerini bir plattCalibrator'a besleyerek olasılık üretir. Çizelge 6.3'te modülün çıkış parametreleri gösterilmiştir.

Çizelge 6.3 Sdca Lojistik Regresyon Modül Parametreleri

Çıkış Kolon İsmi	Kolon Tipi	Açıklama
Score	Single	Model tarafından hesaplanan bağlantısız puan
PredictedLabel	Boolean	Puan'ın işaretine dayanan tahmin etiketidir. Bir negatif puan "false" ile eşleşir, bir pozitif puan ise "true" ile eşleşir.
Probability	Single	Etiket olarak doğru olma puanı kalibre edilere hesaplanan olasılık [0,1] aralığındadır.

6.4.3 Yüksek Verimli Eğim Arttırıcı Karar Ağacı (LightGbm)

Popüler bir makine öğrenme algotimasıdır. Ağaç tabanlı algoritmaları kulanan bir framework'tür. Aynı zamanda paralel bazı deneyler, LightGBM'in belirli ayarlarda eğitim için birden fazla makine kullanarak doğrusal bir hız kazanabileceğini göstermektedir.

Avatajları:

- Daha hızlı eğitim ve daha yüksek verimlilik
- Düşük bellek kullanımı
- Daha iyi doğruluk
- Büyük ölçekli verileri kullanabilme

6.5 Otomatik Makine Öğrenmesi

Çalışmamızın en önemli noktalarından biri otomatik makine öğrenmesidir (AutoML). Peki otomatik ML 'nin faydası nedir? Bu soruyu kısaca cevaplayacak olursak gerçek hayattaki problemlere uçtan uca makine öğrenme sürecini uygulayarak otomatize eder diyebiliriz. Geleneksel makine öğrenmesi yönteminin ötesinde uygulama zorluğuna yapay zekâ bazlı bir çözüm olarak çıkmıştır. Makine öğreniminin baştan sona otomatikleştirilmesi, daha basit çözümler üretilmesi, hızlı oluşturulması ve elle modellenen makine öğrenmesinden daha iyi performans sağlaması yönünden daha avantajlıdır.

BÖLÜM 7

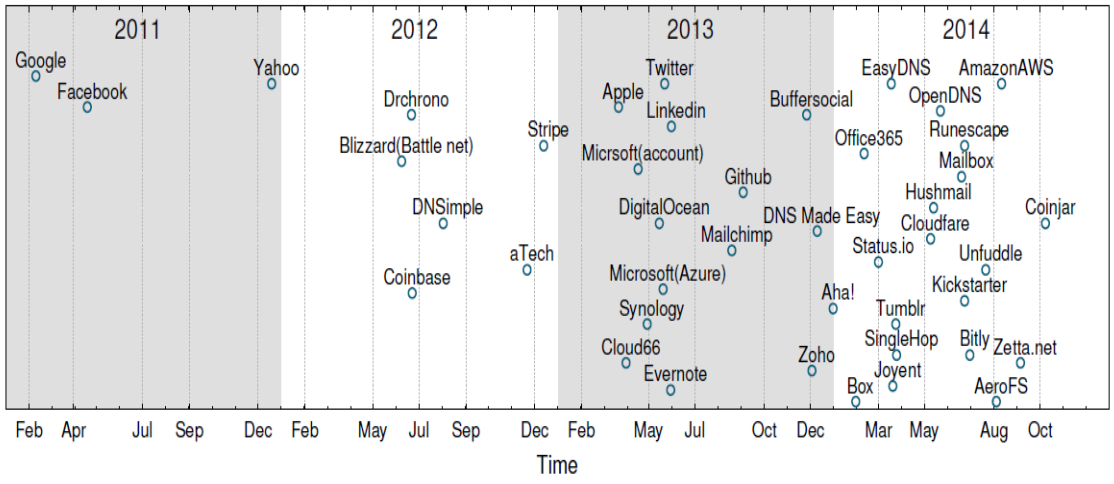
İKİ FAKTÖRLÜ KORUMA(2FA)

Uygulamamızın istemci taraflı şifreleme tekniğinde iki faktörlü koruma (Two-Factor Authentication) kullanacağız. 2FA hem kullanıcı bilgilerini hem de kullanıcının kaynaklarını daha iyi koruyabilmelerini sağlamak, kendilerini doğrulamak için iki farklı kimlik doğrulama sistemidir. Saldırganların bir kişinin cihazlarına veya çevrimiçi hesaplarına erişmesini zorlaştırarak kimlik doğrulama işlemine ek bir güvenlik katmanı ekler çünkü kurbanın şifresini yalnızca bilmek, kimlik doğrulama denetimini geçmek için geçerli değildir. Çoğu saldırı uzak internet bağlantılarından kaynaklanmaktadır bu nedenle 2FA bu saldırıları daha az tehdit edici hale getirmektedir. Bir saldırganın bir kullanıcı hesabıyla ilişkili ikinci kimlik doğrulama faktörünü elde etmesi de muhtemel değildir.

Biz uygulamamızda Google Authenticator'ı kullanacağız. Google Authenticator'ı destekleyen web siteleri ya da web tabanlı hizmetlere ulaşırken her kullanıcı adı ve şifre kullanmaktadır. 2FA açıksa kullanıcıdan 6 basamaklı bir sayı girmesi istenir. Google Authenticator zamana ve kullanıcı bilgisine göre her 30 saniyede bir 6 basamaklı şifre üretir. Kullanıcı bu şifreyi sisteme girerek doğrulama işlemini tamamlar ve doğru cihaza sahip olduğunu kanıtlar. Detaylı olarak 7.1 Google Authenticator kısmında belirtilecektir.

İki faktörlü kimlik doğrulama, çevrimiçi bankacılıkta yaygın olarak kullanılır ve internet servis sağlayıcılar tarafından giderek daha fazla Kabul edilmektedir. Zamana bağlı olarak web siteleri ve web servisleri tarafından 2FA 'ın kullanımı Çizelge 7.1.'de gösterilmiştir.

Çizelge 7.1 2FA'ın Yıl Bazlı Kullanım Artışı



(Petsas, 2015)

7.1 Google Şifrematik

Kullanıcıların kimliklerini doğrulama için iki faktörlü kimlik (2FA) temelli bir uygulamadır. Zaman tabanlı bir kerelik şifre (TOTP) ve HMAC tabanlı bir kerelik şifre (HOTP) kullanılır. Her 30 saniyede bir üretilen 6 haneli şifre ile desteklenen uygulamalara istemci taraflı şifre ile erişim sağlayamaya yarayan bir uygulamadır. TTOP ve HTOP kavramlarını açıklayacak olursak;

TTOP (Time-Based One Time Password): Paylaşılan bir gizli anahtardan ve o anki zamandan tek seferlik bir parola hesaplayan bir algoritmadır ve aşağıdaki bileşenlere sahiptir:

- Paylaşılan Sır (Shared Secret)
- Geçerli zamandan türetilmiş bir girdi
- Bir imzalama fonksiyonu

HTOP (HMAG-based One Time Password): Tek seferlik parola oluşturmak için HMAC algoritmasını kullanan bir algoritmadır.

HMAC (Hash-based Message Authentication Code) Algoritması, Hash tabanlı ileti kimlik doğrulama kodunu ifade eder ve bir değeri imzalamak için SHA1 kullanan algoritmadır. Orjinalliğin doğrulanmasında kullanılır ve yalnızca sırrı bilen insanlar aynı girdi için aynı çıktıyı üretir.

Google Authenticator parametreleri:

- SHA1 HMAC
- Base32 harici anahtar

- 80 Bit Gizli Anahtar
- 64 Bit zamana dayalı mesaj boyutu
- 30 Saniye periyodu
- 6 Digit kod uzunluğu

Bu parametrelerin kullanımını bir sözde kod (Pseudo Code) ile tanımlayacak olursak;
Paylaşılan sır (Secret) manuel olarak ya da QR kod ile telefona girilmelidir. Bu kodu bir değişken ile gösterecek olursak:

Orjinal_Gizli_Key = xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx, bu değer 32 karakter yani 256 bittir.

Tüm byte değerleri görüntülenebilir olmadığı için küçük harfler ve boşluklar kaldırılarak Base32 dönüşümü yapılır;

Gizli_Key = Base32_decode (TO_UpperCse (Remove_spaces (Orjinal_Gizli_Key)

Bir sonraki aşama olarak input değerinin geçerli zamandan türetilmesi işlemidir. Bu işlemde unix zamanı kullanılmaktadır.30 saniye zaman aralığında sabit kalacak bir değer elde etmek için unix zamanı 30 saniyeye bölünmektedir.

Giriş = Current_Unix_Time ()/30

Sonraki aşamada HMAC-SHA1 imzalama fonksiyonu kullanılmaktadır;

Hmac = HMAC-SHA1(Gizli_Key, Giriş)

Hmac değeri standart uzunlukta bir SHA1 değeridir (20-byte, 40 hex karakteri). 20 Byte SHA1 ‘yı 6 basamaklı bir sayıya çevirmek için SHA1 ‘nın 4 bitini ve bu endeksten sonraki 4 baytı kullanıp 32 bitlik işaretsiz tam sayıya dönüştüreceğiz.

Buyuk_Sayi=INT (Hmac [Last_Byte (Hmac): Last_Byte (Hmac)+4]) bu değer 2^{32}-1 bir sayı olabilir dolayısıyla 7 haneli bir sayıya bölerek kalanı 6 haneli bir sayı garanti edilir.

6Haneli_Sayi = Buyuk_Sayi %1,000,000 // 6 Haneli sayı üretilmiş olur.

Durumu özetini Şekil 7.1'deki pseudo kod örneğinde gösterilmiştir.

```
function GoogleAuthenticatorCode(string secret)
    message := floor(current Unix time / 30)
    hash := HMAC-SHA1(secret, message)
    offset := last nibble of hash
    truncatedHash := hash[offset..offset+3]  //4 bytes starting at the offset
    Set the first bit of truncatedHash to zero  //remove the most significant bit
    code := truncatedHash mod 1000000
    pad code with 0 from the left until length of code is 6
    return code
```

Şekil 7.1 OTP Sözde Kod Örneği

(Google Authenticator, 2019)

BÖLÜM 8

UYGULAMA

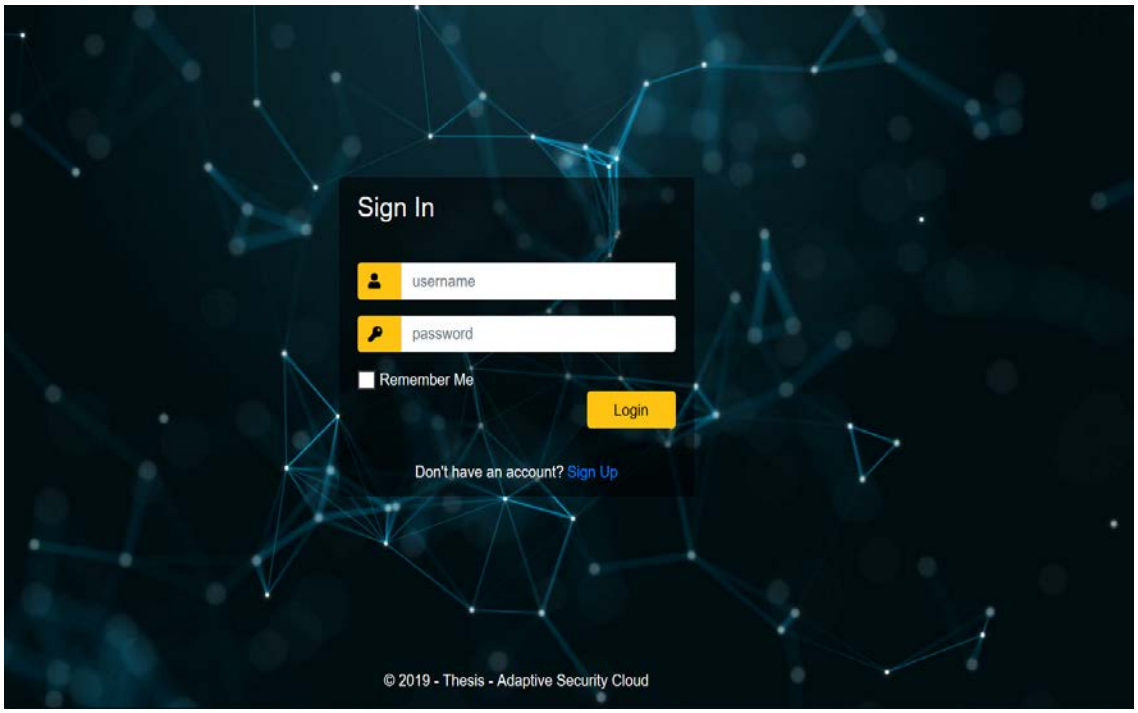
Çalışmamızda, birçok uygulama tekniği birleştirilmiştir. Bunlar istemci taraflı şifreleme, dağıtık depolama ve makine öğrenmesi teknikleridir. Öncelikli amacımız bilgi ve belgelerin içerdiği bilginin kişisel veri içerip içermediği durumuna göre hem depolama alanında hemde belge ve bilgiye erişim zamanında güvenlik yöntemlerinin uygulanmasıdır. Bu yöntemlerimiz uygulama ara yüzlerimiz ile detaylı olarak açıklanacaktır.

8.1.Uygulama Arayüzleri

8.1.1 Oturum Açma

Günümüzde bir sistemin sadece bilinen bir şifre korunması artık yetersiz kalmaktadır. Bu sebeple alternatif güvenlik tedbirleri birçok yönetim sisteminde devreye alınmaya başlamıştır. Bizde bu tezde 2FA istemci taraflı şifreleme ile ek güvenlik yöntemi uyguladık. Uygulamaya daha önce kayıt yapmış kullanıcıların giriş ekranı ile başladık. Bu ekrandan giriş yapmak isteyen kullanıcıların sadece mevcut şifrelerini doğru girmeleri login olmaları için yetersizdir. Bu adımda kullanıcı adı ve şifresini doğru girildikten sonra 2FA için Google Authenticator kodu istenecektir.

Şekil 8.1 ‘de Oturum Ekranı gösterilmiştir.



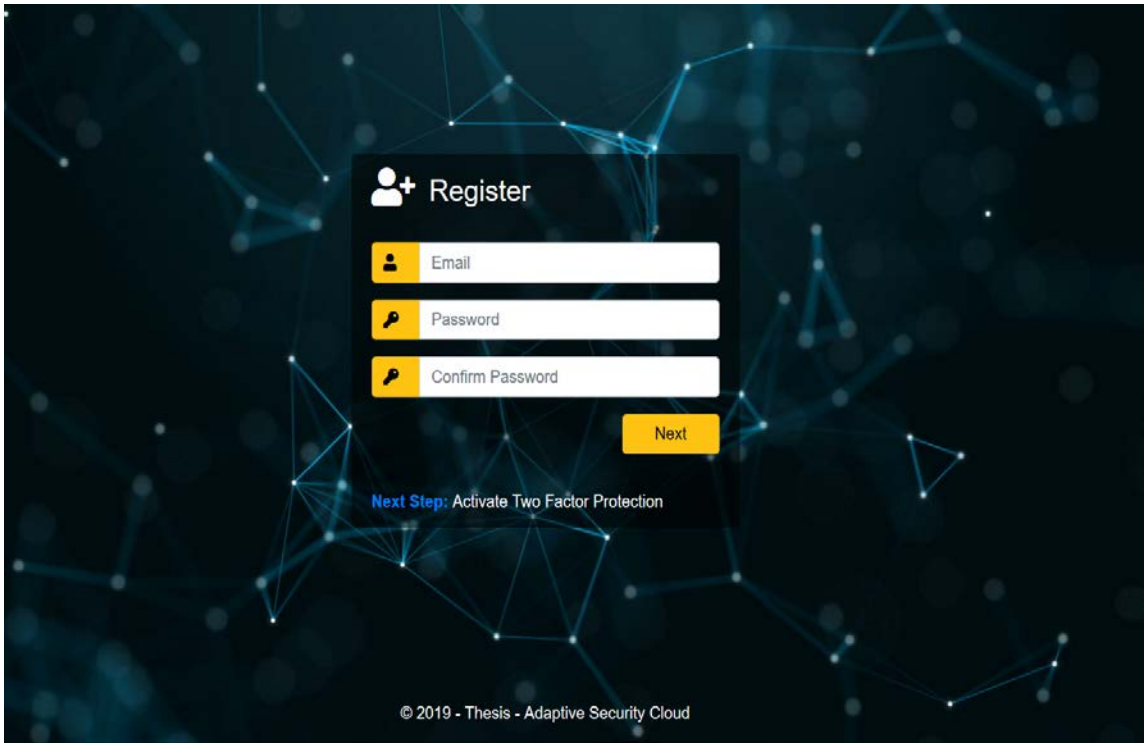
Şekil 8.1 Oturum Ekranı

8.1.2 Kullanıcı Kaydı

Birçok uygulama sistemi kullanıcı kayıt işlemlerinde temel bilgilerin yanı sıra güvenlik tedbirlerinin arttırılmasının yönelik çok farklı bilgiler de isteyebilmektedir. Bu bilgilerin başında kullanıcı adı ve şifre olmakla birlikte anne kızlık soyadı, cep telefonu, evcil hayvan adı, okul adı vb bilgilerde istenebilmektedir. Ancak, bu bilgiler her ne kadar kullanıcı tarafından güvenli olarak görülse de brute force gibi ataklar veya truva atı gibi virüsler ve buna benzer birçok saldırı tekniği ile şifrelerin ve kullanıcı bilgilerinin elde edilmesi günümüzde oldukça kolaylaşmıştır.

Bu sebeple birçok sistem, kullanıcıların şifrelerini belirli periyodlar ile değiştirmesini önermekte ya da zorunlu kılmaktadır. Fakat uygulanan bu yöntemlerde kullanıcılar her seferinde değiştirmek zorunda oldukları şifrelerin birçoğunu yine dijital ortamda not ederek kayıt altına almak durumunda kalmıştır. Bu durum güvenliği arttırmak yerine yeni güvenlik sorunlarını ortaya çıkarmıştır. Bu sebeple, şifreleri değiştirmeden zaman bazlı bir şifreleme tekniği ile belirli cihazlar üzerinde güvenli bir erişim sağlamak için 2FA uygulamamız eklenmiştir.

Uygulama üzerinde işlem yapabilmek için kullanıcının email adresi ve tanımlayacağı bir şifreyi sisteme girerek kayıt olması gerekmektedir. Mail adresi herkes için benzersiz bir kimlik niteliği taşımaktadır. Bu eşsiz mail kimliği ile yeni eşsiz bir anahtar üretilmektedir ve 2FA kodu için anahtar niteliği taşımaktadır.
Uygulama kayıt ekranı Şekil 8.2'te gösterilmiştir.



Şekil 8.2 Uygulama Kayıt Ekranı

8.1.3 2FA Doğrulama ve Tanımlama

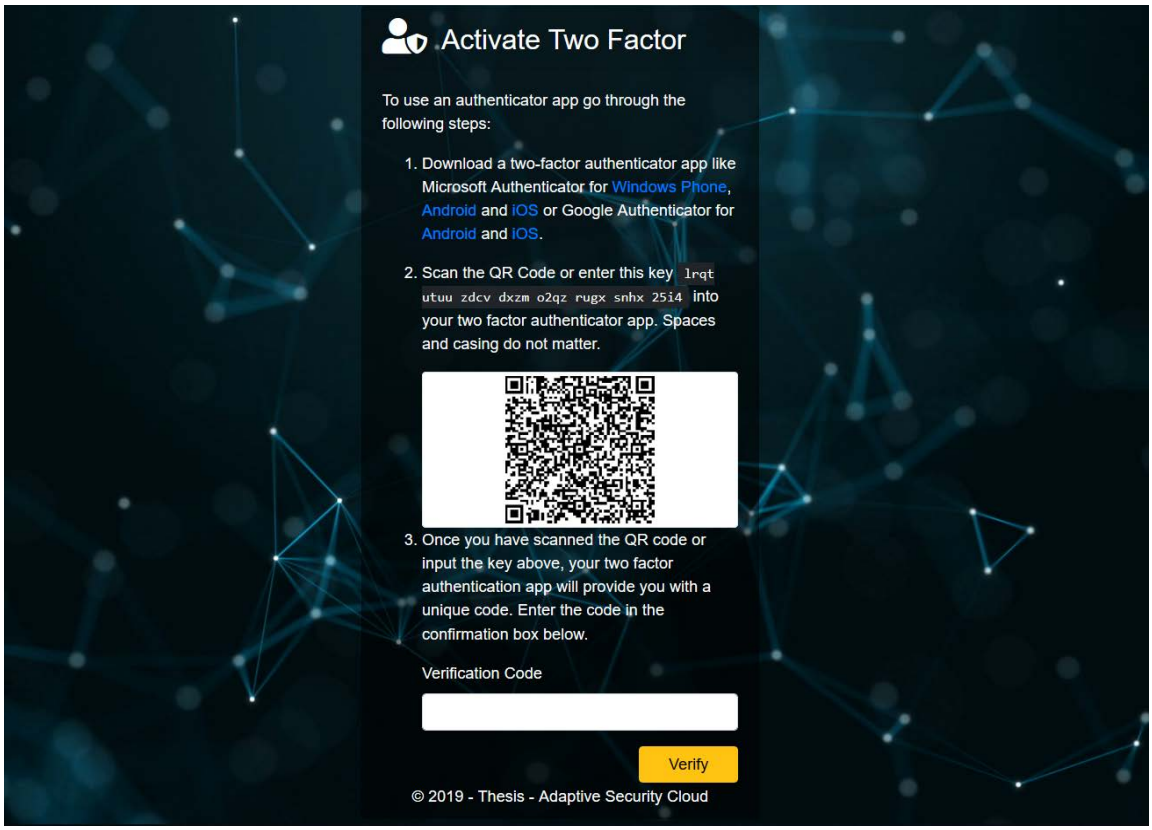
Kullanıcılar mail ve şifre girme adımını geçtikten sonra karşımıza gelen ekrandır. Bu ekranda kullanıcılara herhangi bir mobil cihazlarına yükleyecekleri 2FA uygulamasına oluşturulan anahtarın QR kodun taranması yöntemi ya da el yordamıyla tanımlanması istenir.

Mobil cihazlara girişi yapılan anahtar için doğrulama kodu (verification code) uygulama tarafından istenir. Girilen bu anahtar ile kullanıcı kimliği tanımlanır ve oturum açma işlemi başarılı bir şekilde gerçekleşir. Kullanıcılar bu işlemden sonra her oturum açma işleminde 2FA uygulamasının zamana bağlı olarak üreteceği 6 haneli kodu girerek güvenli girişlerini sağlayacaklardır.

Kullanıcıların mobil cihazındaki yerel saat ve sunucu saatinin uyuşmaması durumunda mobil cihazda üretilen kod geçersiz olacaktır. Bunun sebebi 2FA'nın zaman

bazlı çalışıyor olmasıdır. Bu zaman hem sunucu hem de istemci tarafında eşit olmalıdır. Farklı olması durumda girilen anahtar geçersiz olacağında oturum açma işlemi başarısızlıkla sonuçlanacaktır.

Kullanıcı "1rqt utuu zdev dxzm o2qz rugx snhx 25i4" anahtarını mobil cihazın kaybolması, çalınması ya da zarar görmesi gibi durumlarda tekrar 2FA'yı devreye sokması için saklaması, ya da yedeklemesi gerekmektedir. Şekil 8.3'te 2FA tanımlama ekranı gösterilmiştir.



Şekil 8.3 2FA Tanımlama Ekranı

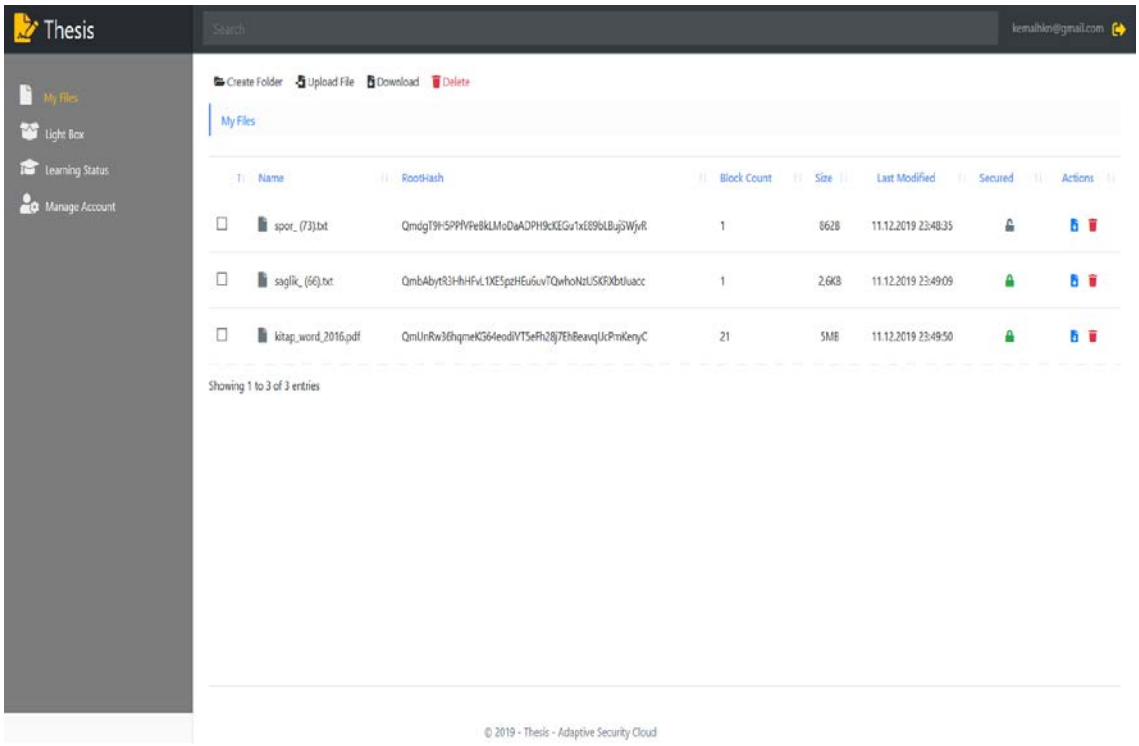
8.1.4 Kayıtlı Dosya Listeleri

Bir kullanıcı sisteme login olma işlemini tamamladıktan sonra belgelerini yükleyebileceği ekran ile karşılaşır. Bu ekran Benim Dosyalarım (My Files) ekranıdır. Bu kısımda yeni klasörler oluşturulabilir, belgeler indirilebilir ya da silinebilir.Bu ekranda yüklenen her bir dosyanın ismi, boyutu, oluşturulduğu tarihi, güvenli olarak işaretlenip işaretlenmediği bilgileri de dosya tablosunda görüntülenmektedir. Güvenli olarak işaretlenmiş bir dosyanın indirilmek istenmesi ya da silinmek istenmesi durumunda 2FA devreye girecektir. Güvenli olarak işaretlenmemiş bir dosya ise

herhangi bir güvenlik denetimine tabi tutulmadan indirilebilir, silinebilir niteliktedir.

Bu dosyalar IPFS tarafından 256kb'lık parçaya bölünmüş olarak saklanmaktadır. Herbir parça hash bilgisi ile tutulduğundan tüm parçalar IPFS doğrulamasından geçmektedir. Bu doğrulama veri üzerinde herhangi bir değişiklik olup olmadığı bilgisini kullanıcıya vermektedir.

Benim dosyalarım ekranı Şekil 8.4'te gösterilmiştir.



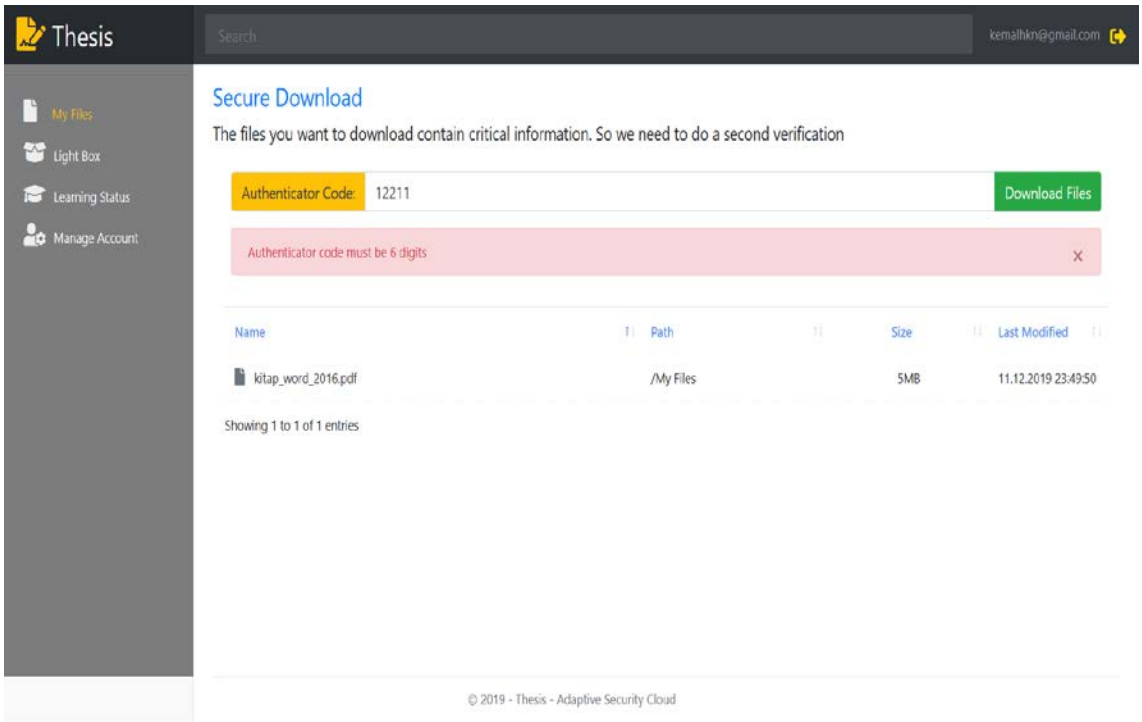
Şekil 8.4 Benim Dosyalarım Ekranı

8.1.4 Güvenli İndirme

Güvenli olan bir belge indirilmek istendiğinde karşılaşılan ekrandır. Kullanıcı güvenli olarak işaretlediği dosyayı indirmek istediğinde 2FA devreye girer ve kullanıcıdan mobil cihazında 30 saniyede bir güncellenen şifre istenir. Kullanıcı şifreyi doğru girmesi durumunda download(indirme) işlemi aktif olur ve işlem başarılı bir şekilde gerçekleşir. Yanlış olması durumunda kullanıcıya kullanıcı doğrulamasının yapılamadığına ilişkin bilgi mesajı verilir.

Örnek olarak bir senaryo üzerinden gidecek olursak, bir saldırganın çeşitli saldırılar sonucu kendisine hedef seçtiği kurbanın mail bilgisine ve oturum açma şifresine

ulaştığını varsayalım. Bu mail ve şifre bilgisiyle oturum açabilir ancak kullanıcının güvenli olan dosyasına ulaşması için kullanıcının mobil cihazına tanımladığı 2FA şifresine de ihtiyacı olacaktır. Cihaza ulaşamadığı takdirde anahtarı elde edemeyecek ve belge indirme işlemini gerçekleştiremeyecektir. Belgeler bloklar halinde tutulduğundan bir ftp hizmetine sızma durumunda dahi hangi bloğun hangi dosyaya ait olduğunu sistemi kullanmadan bilemeyeceğinden dolayı dosya bloklarına erişmesi mümkün olsa da görüntülenmesi mümkün olmayacaktır. Bu bağlamda 6 karakterli şifre yanlış girildiği varsayılmış ve uyarı mesajı alınmıştır. Belgeyi indirmek isteyen kullanıcı dosyaya ulaşamamıştır. Şekil 8.5'te Güvenli belge indirme ekranı gösterilmiştir.

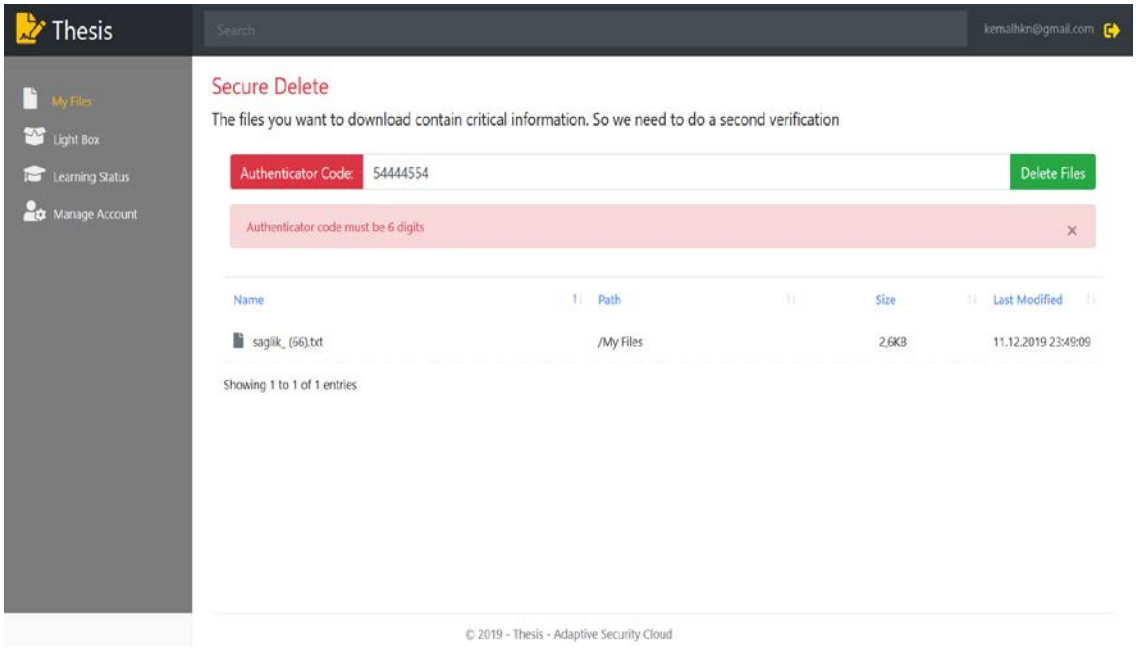


Şekil 8.5 Güvenli İndirme Ekranı

8.1.5 Güvenli Silme

Belgelerin görüntülenmesi, indirilmesi ne kadar önemli ise silinmek istenmesi durumunda belgenin önemine göre işlemin yapılması da bir o kadar önemlidir. Belgelerin indirilmesi aşamasında alınmış 2FA önlemine benzer olarak silinmek istenmesi durumunda da 2FA devreye girmektedir. Bu önlem ile belgelere ulaşmak ve yok etmek isteyen saldırganlar için ek güvenlik tedbiri olarak uygulanmıştır. Aynı zamanda kullanıcı belgenin kişisel veri içerdiğini bu doğrulama ile anlamakta ve bu sayede yanlışlıkla silinmesinin önüne geçilmektedir. Silme işleminin gerçekleşmesi için kullanıcı mobil cihazında üretilen 6 haneli kodu girmek zorundadır. Kod'un doğru bir şekilde girilmesinin ardından IPFS dosya bütünlüğünü sağlayarak doğrulama işleminden geçirir ve belgenin silinmesi işlemini gerçekleştirir.

Şekil 8.6 Güvenli Belge Silme ekranında "Yeni Microsoft Word Belgesi.docx" isimli belge senaryo gereği yanlış şifre girilerek silinmek istenmiştir ve hatalı girilen 6 haneli kod ile belgenin silinemediği gösterilmiştir. Kullanıcıya bilgi mesajı olarak "İndirmek istediğiniz bu dosya kritik bilgiler içermektedir. Bu yüzden 2. Bir doğrulamaya ihtiyaç duyuyoruz." şeklindedir. Hata mesajı olarak ta "Üzgünüm! Kim olduğunuzu anlamayadım." şeklindedir. Bunun sebebi 2FA doğrulamada mobil cihaz ile oturum açılmasıdır.



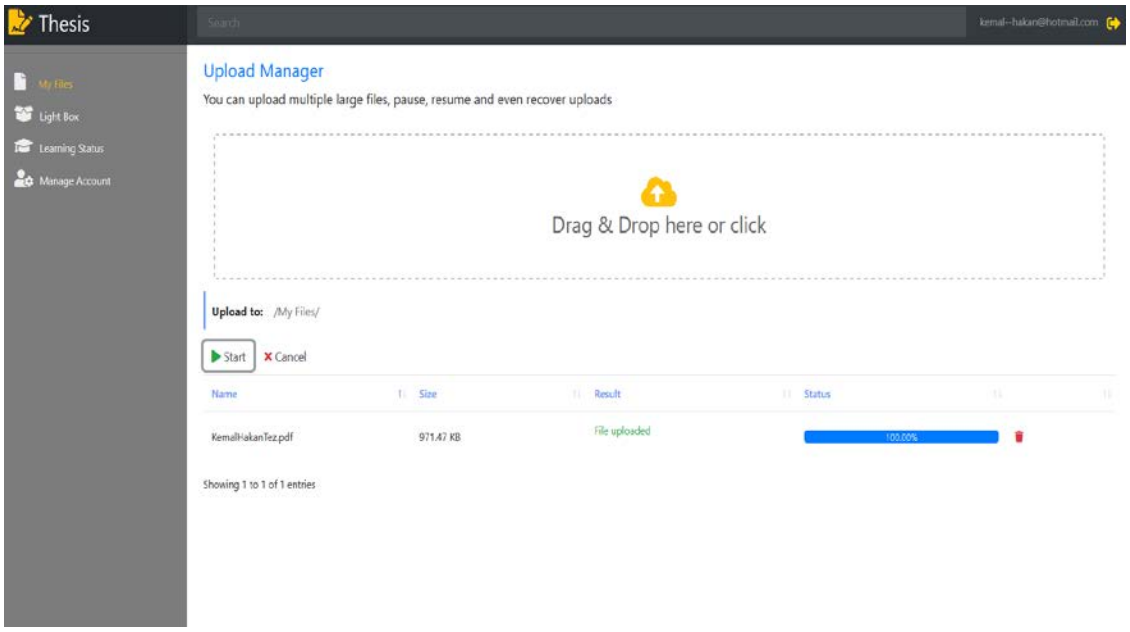
Şekil 8.6 Güvenli Belge Silme

8.1.6 Dosya Yükleme

Kullanıcılar MyFiles ekranından Upload File sekmesine tıkladıklarında karşılaştıkları ekran Şekil 8.7 Dosya Yükleme Ekranın da gösterilmiştir. Kullanıcı sürükle bırak yöntemiyle ya da "Drag & Drop here or click" alanına tıklayarak dosya seçme işlemine yönlendirilmektedir. İlgili dosya seçildikten sonra start butonuna tıklanır ve dosyanın yüklenmesi beklenir. Bu süreçte dosya IPFS ortamına parçalanmış bir şekilde aktarılır ve bu dosya tüm süreçlerde doğruluğu kontrol edilerek işleme alınır. Bu ekranımızda Status alanı dosyanın yüklenme yüzdesini göstermektedir.

Bu alanda dosya yüklemesi %100'e ulaştığında makine öğrenmesi devreye girmektedir. Daha önce oluşturulmuş eğitim seti ile belgenin hangi doğrulukla hangi sınıfa ait olduğu belirlenmektedir. Eğer sınıflama sonucu kararsız durumda ise yani 2 eğitim setide farklı sonuçlar vermiş ise belge "Light Box" ekranına düşer.

Şekil 8.7.'de "KemalHakanTez.pdf" isimli belgenin boyutu, yüklenme durumu ve yüzdesi gösterilmektedir.



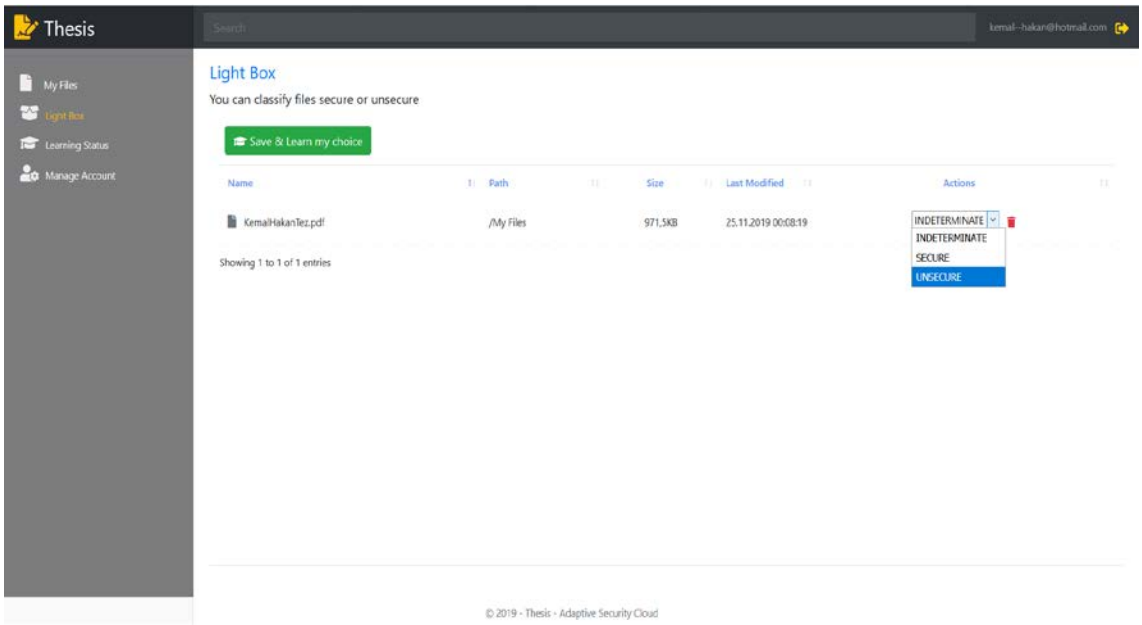
Şekil 8.7 Dosya Yükleme Ekranı

8.1.7 Eğitim Ekranı

Hafif Kutu (Light Box)' ekranımız dosya yüklenmesinin ardından makine öğrenmesi sonucunun kararsız kaldığı durumda belgelerin kullanıcı onayına güvenli ya da güvensiz belge olarak işaretlemesi için talepte bulunan ekranımızdır.

Örnek olarak Şekil 8.8 Hafif Kutu Ekranı'ndan yola çıkacak olursak dosya yükleme adımından sonra işleme alınan "KemalHakanTez.pdf" dosyası listede görebiliriz. Bu belgemiz makine öğrenmesi tarafından kararsız kalmıştır. Bu durumda önemli ya da önemsiz evrak olarak belirlenmesi kullanıcı bilgisine sunulmuştur.

Kullanıcı güvenli ya da güvensiz olarak belgeyi işaretlediğinde mevcut eğitim seti bu belgenin işaretleme bilgisine göre tekrar ekleme yapılarak geliştirilir. Böylece doyum noktasına ulaşılıncaya dek adaptif öğrenme süreci devam ettirilir. Belge işaretlendikten sonra "Kaydet ve Benim Seçimimi Öğren" (Save & Learn my choice) butonu tıklanır. Bu işlemden sonra kısa bir süre eğitim için beklenir ve işlem tamamlandıktan sonra belge Benim Dosyalarım alanına güvenli ya da güvensiz durumuna göre yerleşir. İşaretlenmeyip bekletilen dosyalar Benim Dosyalarım alanına yerleşmez mutlaka bir kategori seçilip eğitime sokulmalı ya da işlem yapılmayacaksa silinmelidir. Bilginin güvenliği tamamıyla kullanıcının eğitim setine ayırdığı test verileri ve adaptif öğrenme sürecinde gireceği doğru bilgiler ile paraleldir.



Şekil 8.8 Hafif Kutu Ekranı

8.1.8 Makine Öğrenmesi Sonuçları

Şekil 8.9.'da gösterilen öğrenme durum ekranımız 'da makine öğrenmesine ilişkin temel veri seti bilgileri, kullanıcı tanımlı veri seti bilgileri ve öğrenme durumu bilgileri yer almaktadır. Bu kısımları kısaca açıklayacak olursak;

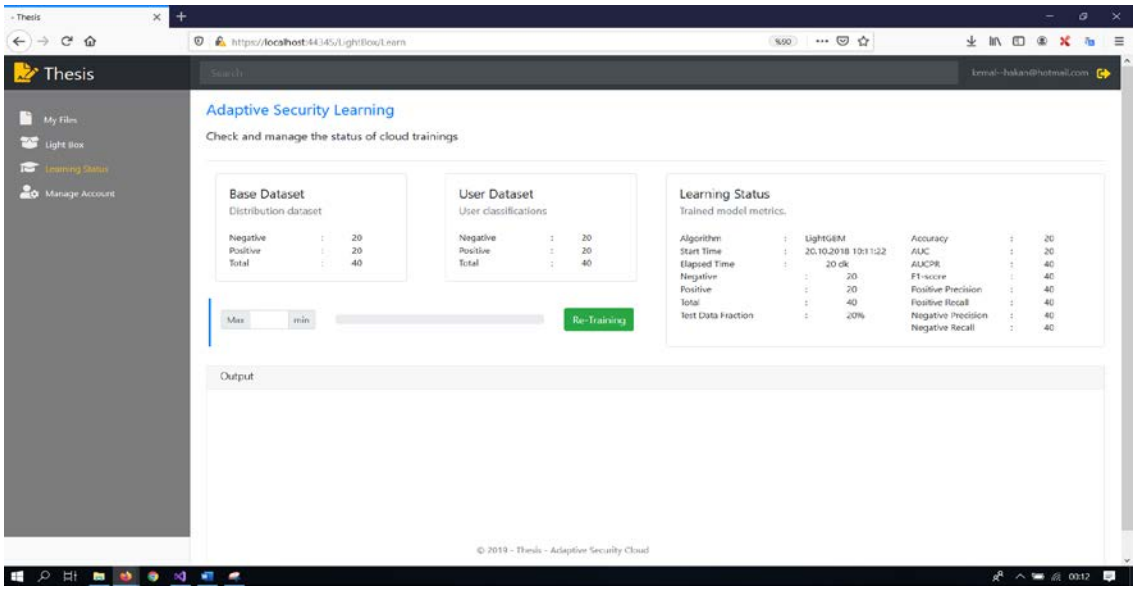
Temel Veri Seti (Base Dataset): Tezimizde ikili sınıflandırma yöntemi kullandığımızı daha önce belirtmiştik. Bu doğrultuda bilgilerin pozitif ve negatif olarak ayırdık. Pozitif veri için kişisel veri içeren, negatif için ise kişisel veri içermeyen evraklar olarak tanımladık. Çeşitli kaynaklardan edindiğimiz 20 adet pozitif ve 20 adet negatif belge varsayılan olarak bulunmaktadır. Bu ifadeyle toplam 40 adet evrağın temel olarak eğitildiği bilgisini veren kavramdır.

Kullanıcı Veri Seti (User Dataset):

Yükleme aşamasından sonra eşik değeri altında kalan algoritma kullanıcı işaretleme işlemine gerek duyar. Kullanıcının her işaretlediği veri user dataset olarak tanımlanmıştır. Bu örneğimizde kullanıcı 20 adet pozitif ve 20 adet negatif işaretleme yapmıştır.

Öğrenim Durumu (Learning Status):

Bu kısımda son eğitim zamanında kullanılan parametreler tanımlanmıştır. Bu öğrenmede makine öğrenmesi otomatik olarak LightGM algoritmasını seçmiş ve eğitim sürecini 20 adet pozitif ve 20 adet negatif belge ile 20 dk tamamlamıştır.



Şekil 8.9 Öğrenme Durum Ekranı

8.1.9 Bölümlendirilmiş Dosya Blokları

Uygulamamızda blokların incelenmesi amacıyla yerel IPFS depolama kullandık. Örnek blok boyutları Şekil 8.10s. de gösterilmiştir. IPFS verileri boyutlarına göre maksimum 256kb'lık parçalara ayırır ve bir tane verisi boş olan nesne yaratır. Bu nesne oluşturulan tüm nesnelerin linklerini tutar ve dosya bu nesneden talep edilir.
Örneğin: 612kb'lık bir dosya IPFS'e aktarıldığında 256kb*2 + 100kb*1 'lik toplam 3 bloktan oluşur. Şekil 8.10'da uygulamamızda yer alan blokların bir kesiti paylaşılmıştır.

Ad	Değiştirme tarihi	Tür	Boyut
ciqa3h4q2azprwjdpm3kkvfma5jw2fazggy46jqvyvysed27zdxtsoy	13.11.2019 20:48	Dosya	1 KB
ciqa5scgovnkcyac3g3yx3z44arubyuypcecegeftwyu275weeencla	24.11.2019 22:03	Dosya	4 KB
ciqa6jmuj4xso6jltrjkf2j67c4ax2qzi645jlmu3vf5d6lumc57iwi	24.11.2019 22:03	Dosya	1 KB
ciqa42ukuggundcjsxhqxzgkz4bstkhb3lmhbm667dt6e6duq5a6nwa	13.11.2019 20:48	Dosya	257 KB
ciqaadqxacqhbxsq2pzm5xsgjy3hdieysxate4nd442dhfnax2l3c2y	13.11.2019 20:48	Dosya	257 KB
ciqabfhcwtskbfo3jxxfwdnkv4dk74wr22lv3ujw3i6oyj4huwh4ulq	13.11.2019 20:48	Dosya	257 KB
ciqadshbrxzzxpfxxmvomzbmat6anls25fyecxisc4mth7g6otmzdmq	13.11.2019 20:48	Dosya	257 KB
ciqak3z3j5c5qby4dlkdeyzls7xxdkmad42xakipcf2ypnburzvv7my	13.11.2019 20:48	Dosya	2 KB
ciqamkvaonczl5qwh4wtaa64e6dxvdcgk4m2yme7jnuqpmqtzzgafra	24.11.2019 22:03	Dosya	1 KB
ciqanplurimxr2dxfuqxvrweyfjbob3u7zxjbtkb23gtbxrbirkkisq	25.11.2019 00:08	Dosya	1 KB
ciqas6upd3jpezoh62s37qcd3ettkdarqwghjrytmgygiytzsses5wi	13.11.2019 20:48	Dosya	257 KB
ciqasqshyap5wbaytkljeu5xhuffonk7p5a2i6on6qparcoelwtosgy	13.11.2019 20:48	Dosya	257 KB
ciqaw5d4atrxpswo5kcikyucyacfdq5ogvb4v6xbmuc4roy6qhhx2ja	13.11.2019 20:48	Dosya	257 KB
ciqawjzydzn3r5rhxkv2wztqhylgqmmo42hcvsniwuaxdwhy75zosxq	24.11.2019 22:03	Dosya	2 KB
ciqax5exhcwsx7symgqzydbeko3mywnguikoybrvm2lqscboebmrxji	24.11.2019 22:03	Dosya	2 KB
ciqaxhoulepcqpmrdltxyatagac2ohqo7mtp2kkxb3rua4jtybql6vy	13.11.2019 20:48	Dosya	163 KB
ciqaym3nas4mukbxyevzwd2nhzb4h7fwfjpmqcors6i5nmsi6utf3kq	24.11.2019 22:03	Dosya	3 KB
ciqb77oevifgwsg4pjwbem5yylbx3vgis32hhtpascesxwxw2f6u64y	13.11.2019 20:48	Dosya	257 KB
ciqbba7wj7td6bbppgybukldlomfdx3fwnhttuevu22dhfuc2z6tqpq	13.11.2019 20:48	Dosya	257 KB
ciqbcnimdkbr7tdx4n5tojs6ehj3lb5x3yi7tcea4madkecc7myxqea	13.11.2019 20:48	Dosya	37 KB
ciqbcxlyvwpic2zshea5gl6lv675f7lmerd46v5tt3bm2hag4vtf7fy	13.11.2019 20:48	Dosya	257 KB
ciqbdfxkuqjyzpimler4cfwi5kfs62cw7n3ttq73fwgp7oldnlmpyqi	24.11.2019 22:03	Dosya	2 KB
ciqbeowb7ybaol2czfsutfrweqliretaa26wg3br2zbjfjjnsjhix2a	13.11.2019 20:48	Dosya	257 KB
ciqbgn46edtfpajtnsdcikkuymm7c7pvjlrmvpzayw5dqy4lec6xe6q	24.11.2019 22:03	Dosya	5 KB
ciqbi3ngrq6devtabrbesvanjq7fvn7hjq2bvwve3dvyhgpyv6qanvi	24.11.2019 22:03	Dosya	4 KB
ciqblck6756m3fqlhkhgr7pam74ojhm3jkg4yct4dq7geyc5hhiergq	13.11.2019 20:48	Dosya	257 KB
ciqblpzdf7cj43gype5b2n4rwp65qed4dmcuxwp4w2w22fwdicgyyma	13.11.2019 20:48	Dosya	257 KB
ciqbn2qgxrlh5ijrcf5vdtipmuiuzrsqtjlskoplqw4xvl2zqiak5ca	24.11.2019 22:03	Dosya	1 KB
ciqbp7nfgdogmqozctrvmdnie2xdrqe5sdwed34i5k6ypylos3qbn3a	13.11.2019 20:48	Dosya	257 KB
ciqbqrpryznijlirkfnc6jorgwgio5idrs3jjyvhgn5vfyqnzvdtjdq	24.11.2019 22:03	Dosya	5 KB
ciqbruygozda7s34dmjlurelhyduz4shdhs7ye6urp2jpvzrpbw4ipq	24.11.2019 22:03	Dosya	4 KB
ciqbt6bx5ru677lpnxwtrxbxjjpko6u5yefvitb5hx6gyfxulrscqqy	13.11.2019 20:48	Dosya	257 KB

Şekil 8.10 Bloklanmış Veriler

8.2 Sistem Diyagramları ve Şemaları

8.2.1 Uml Diyagramı

Tasarladığımız sistem temel 3 bileşenden oluşmaktadır bunlar: IPFS, Makine Öğrenmesi ve 2FA dır. Bu bileşenleri özellikleriyle açıklayacak olursak;

1. **IPFS**

Güvenli Belge Depolama: Dağıtılacak tüm veriler IPFS tarafından şifrelenerek saklanır.

Belge Doğrulama: Hash'lenerek dağıtılan tüm veri bloklarının doğruluk ve tutarlık kontrolünün testi yapılır.

Dağıtık Depolama: Swarm ile tanımlı sunuculara veri bloklarının dağıtılması.

Yerel Depolama: IPFS veri bloklarının uygulamanın çalıştığı sunucuda depolanması.

2. **Makine Öğrenmesi**

Adaptif Öğrenme: Eğitim işlemi tamamlandıktan sonra makine öğrenmesi sonuçlarının belirli bir eşik değeri altında kalması durumunda kullanıcıdan bilgi istemesi ve bu bilgiye göre eğitimi tekrar güncellemesidir.

3. **2FA Koruması**

İstemci Taraflı Şifreleme: Kullanıcı sisteme kayıt zamanında girdiği mail adresi ile oluşan eşsiz anahtar ile verinin şifrelenmesi ve 2FA sistemi tarafından üretilen bu anahtarın zaman bazlı deşifrelemede kullanılması için kullanıcının mobil cihazında yer alan Microsoft Authenticator ya da Google Authenticator gibi uygulamalara tanımlanmasıdır.

Güvenli Belge Silme: Önemli olarak işaretlenmiş belgelerin silinebilmesi için 2FA şifresine ihtiyaç duyulmasıdır.

Güvenli Belge İndirme: Önemli olarak işaretlenmiş belgelerin indirilebilmesi için 2FA şifresine ihtiyaç duyulmasıdır.

Sistemin tüm özellikleri Şekil 8.11 Uml Diyagramında gösterilmiştir.



Şekil 8.11 Uml Diyagramı

Sistemimiz genel olarak bilgi güvenliğinin belgeler üzerinde sağlanması amacıyla oluşturulmuştur. IPFS sayesinde dağıtık ya da yerel depolamada belgenin şifreli bir şekilde saklanması ve 2FA birleşimi sayesinde de güvenli erişim imkânı sunulmuştur. Makine öğrenmesiyle de kullanıcının yapabileceği hataların en aza indirilmesi hedeflenmiştir. Burada amaç sistem çalışma zamanında kullanıcı belge tiplerine ne kadar az karar verirse performans ve doğruluğun o kadar attırılmasıdır.

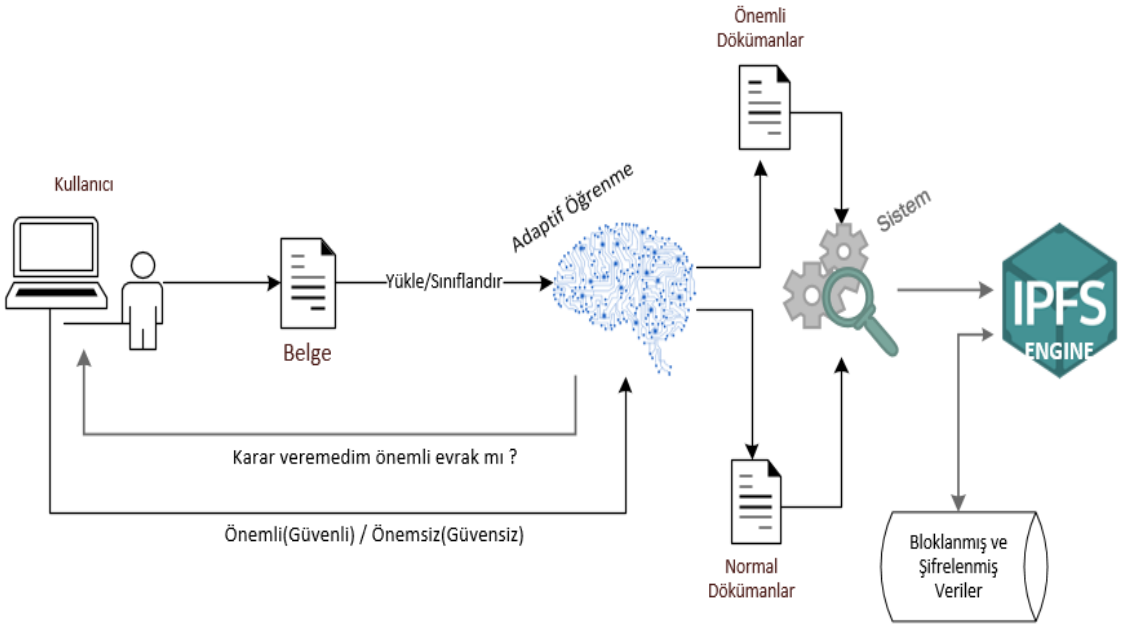
Kullanıcının kimler olabileceği konusunda bir kısıtlama getirilmemiştir. Ancak kullanıcı genel itibari ile bilgi ve belge yönetimi üzerine işlem yapan kurum ve kuruluşlar olabilir. Örneğin, değerli evrak kayıtları alan bankalar, öğrenci bilgilerini sisteme giren üniversiteler, fatura kayıtlarının gizli tutulmasını isteyen muhasebe departmanları ya da kişisel verilerin, hisselerin gizli tutulmasını isteyen borsalar olabilir. Diğer bir deyişle belgelerinin tutarlı ve değişmediğine emin olmak isteyen kurumlar ve kuruluşlar kullanıcı profili olarak değerlendirilebilir. Kurum ve kuruluşlar dışında kullanıcı, mevcut sistem bulut ortamında depolama hizmeti veren bir uygulamaya dönüştürüldüğünde son kullanıcı olabilir.

8.2.2 Belge Yükleme Şeması

Doküman yükleme işleminin sistematik olarak nasıl gerçekleştiği Şekil 8.12 Belge yükleme şeması diyagramında gösterilmiştir. Bu diyagrama göre kullanıcı belgeyi yüklediği anda makine öğrenmesi eğitim bilgisi durumuna göre bu belgeyi önemli ya da önemsiz olarak ayıracak ancak makine öğrenmesi kararsız kalırsa kullanıcıdan belgenin işaretlenmesi yani önemli ya da önemsiz durumuna karar verilmesi istenecek. Kullanıcı işaretleme işlemini tamamladığında adaptif olarak eğitim seti güncellenecek ve yeni eğitim bu eğitim seti üzerinden tekrar güncellenecek. Bu işlem eğitim belli bir doyum noktasına ulaşıncaya dek sürdürülecek.

Karar işlemi tamamladıktan sonra IPFS engine ile dosya bloklara ayrılacak ve istenilen ortamda tutulacak. Bu ortamlar lokal ya da swarm özelliği ile dağıtık ortamlara aktarılabilecek. Biz bu çalışmamızda dosyaların blokları, içerdiği değeri, linklerini ve üretilen hash değerlerini gözlemlemek adına lokal depolama yapacağız.

Belgelerin önemli ve önemsiz olarak ayrılmasında adaptif öğrenme ve IPFS ile gizlilik, güvenlik ve performans artırımı sağlayacağız

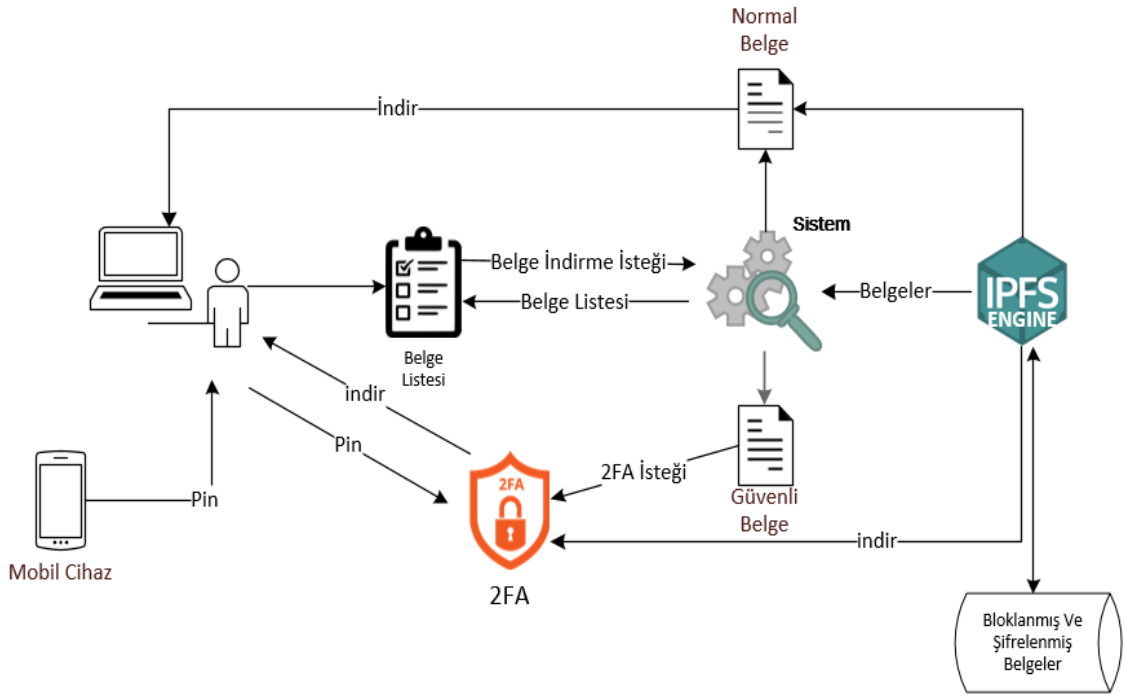


Şekil 8.12 Belge Yükleme Şeması

8.2.3 Belge İndirme Şeması

Belge yükleme aşaması tamamlandığında indirme işlemine hazır olacaktır. Belge indirme süreci Şekil 8.13. Belge indirme diyagramında gösterilmiştir. Bu diyagrama göre kullanıcı kendisine bağlı doküman listesini ilgili sayfada bir tablo halinde güvenli veya güvensiz olarak kayıt edildiğini görebilmektedir.

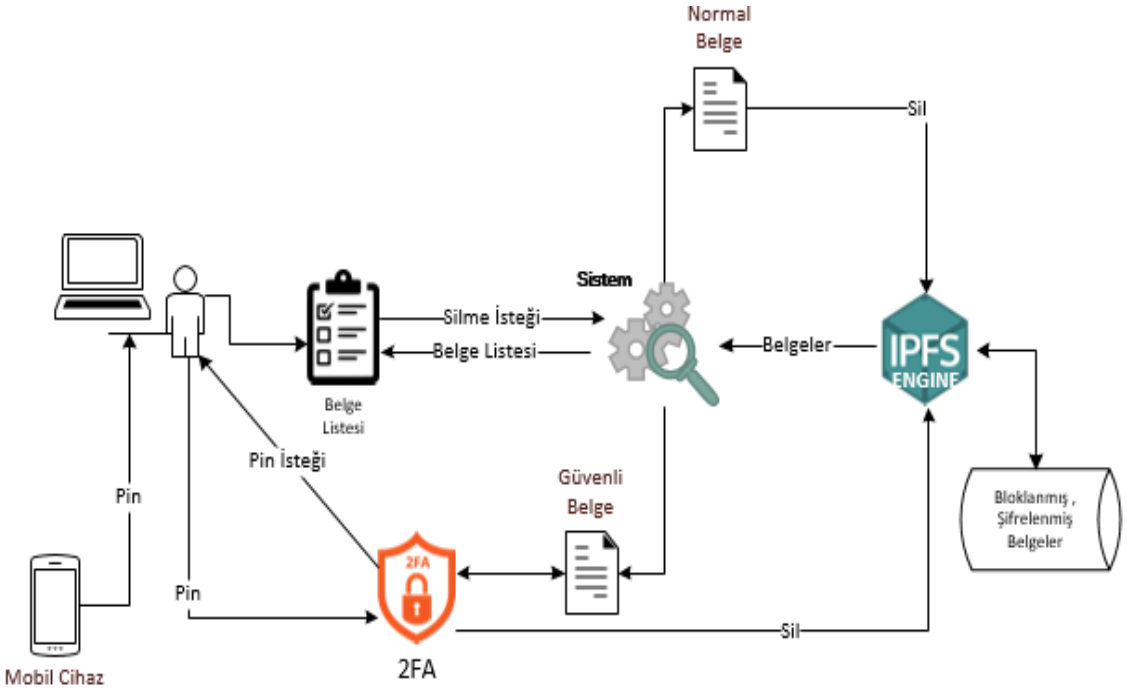
Güvensiz bir belge seçilip indirilmek istendiğinde kısıtlama yapılmaksızın IPFS den çekilerek indirme işlemi gerçekleştirilir. Eğer kullanıcı güvenli bir belge indirmek istiyorsa 2FA aşamasını geçmek zorundadır. Mobil cihazdan üretilen 6 haneli kod 2FA tarafından doğrulanması durumunda belge indirme işlemi başarı ile gerçekleştirilir.



Şekil 8.13 Belge İndirme Şeması

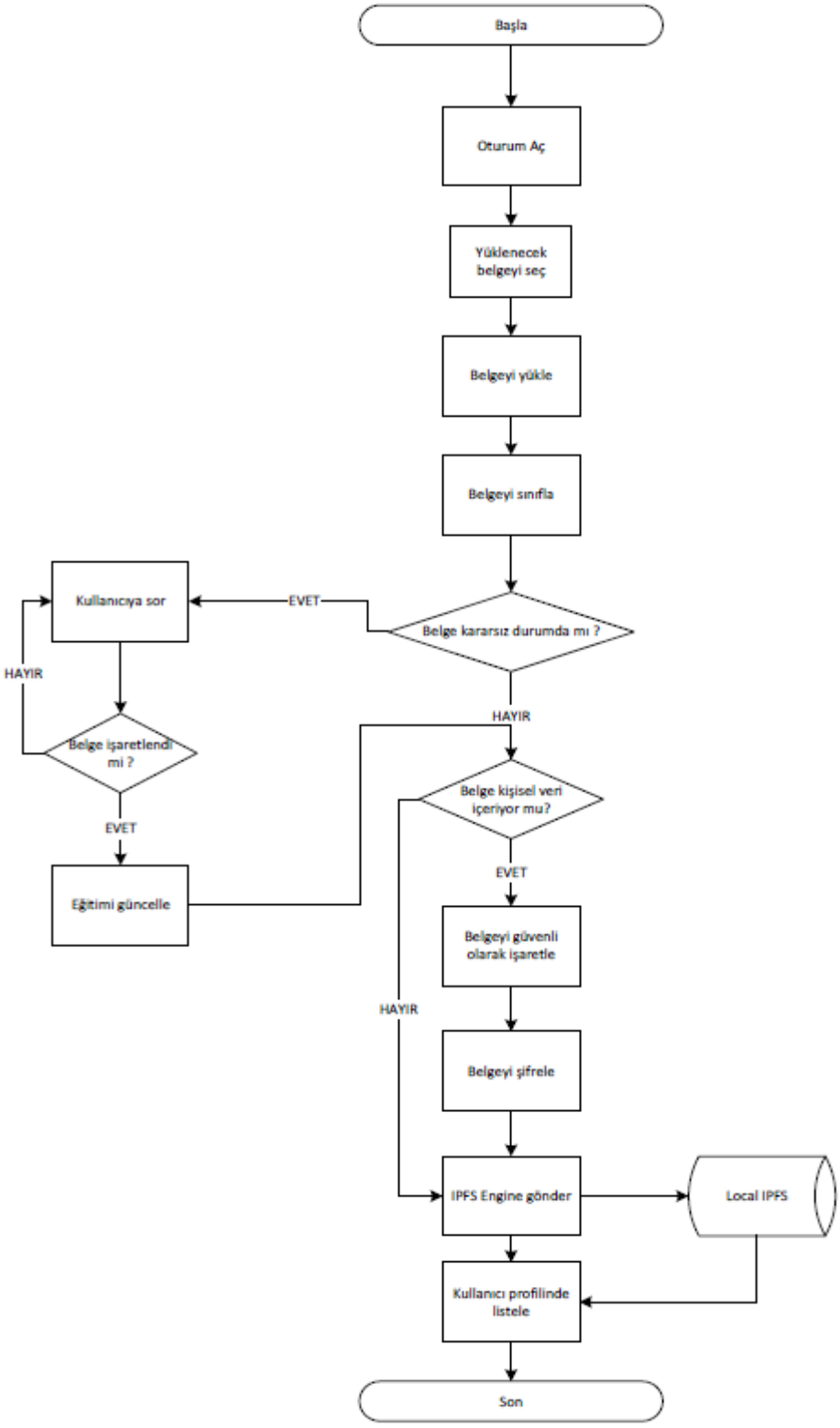
8.2.4 Belge Silme Şeması

Belge silmeye ilişkin süreç Şekil 8.14. Belge silme diyagramında gösterilmiştir. Belge listesinden kullanıcı silmek istediği belgeyi seçtiğinde belgenin güvenli olarak işaretlenip işaretlenmediğine bakılır. Eğer kullanıcı güvenli olarak işaretlenmemiş bir belge seçti ise belge doğrudan silebilir. Kullanıcı güvenli bir belge silmek istiyorsa 2FA aşamasını geçmek zorundadır. Mobile cihazdan zaman bazlı üretilen pin 2FA ekranına doğru girmesi durumunda silme işlemini başarılı bir şekilde gerçekleştirebilecektir.



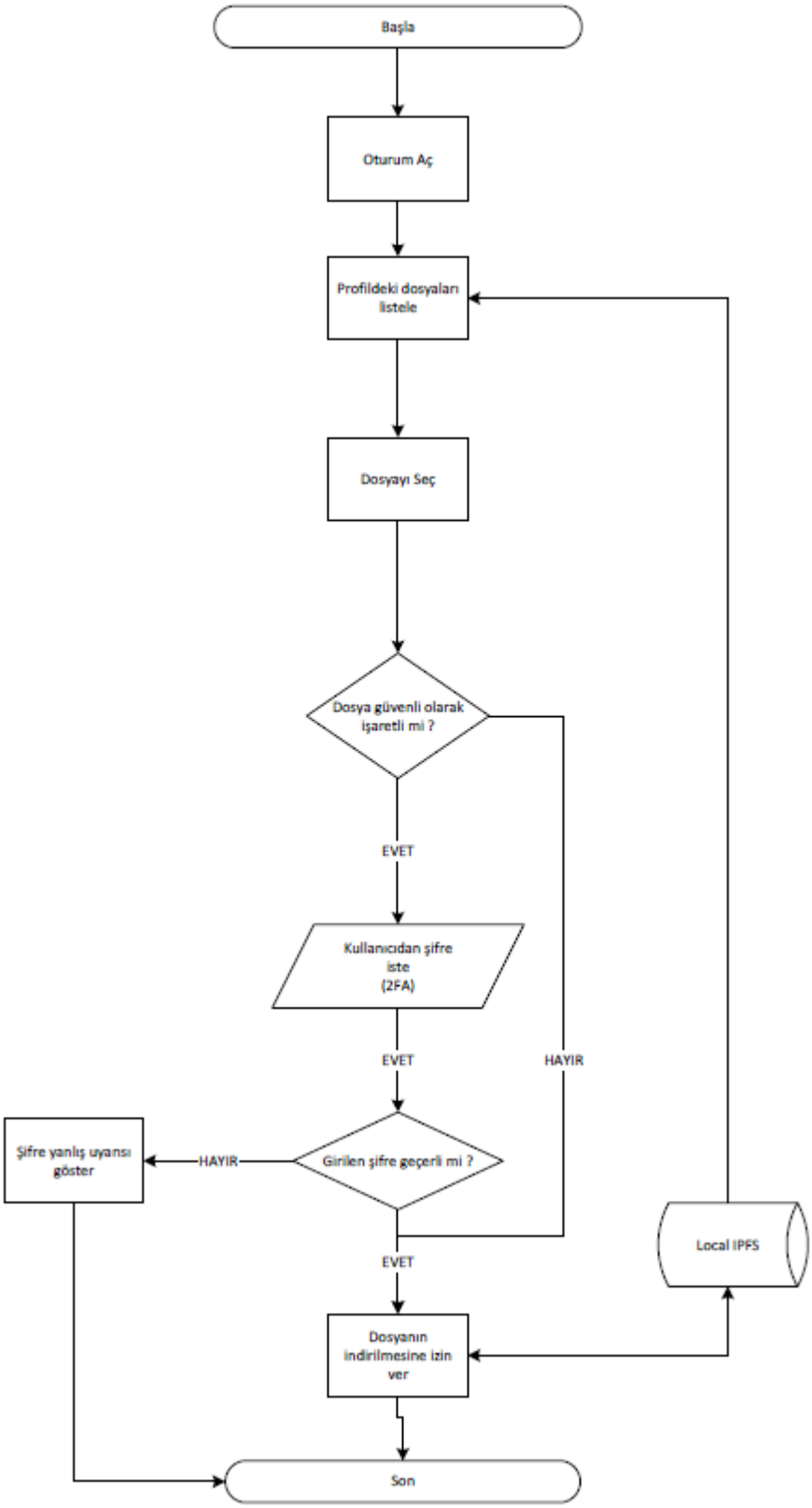
Şekil 8.14 Belge Silme Şeması

8.2.5 Belge Yükleme Akış Diyagramı



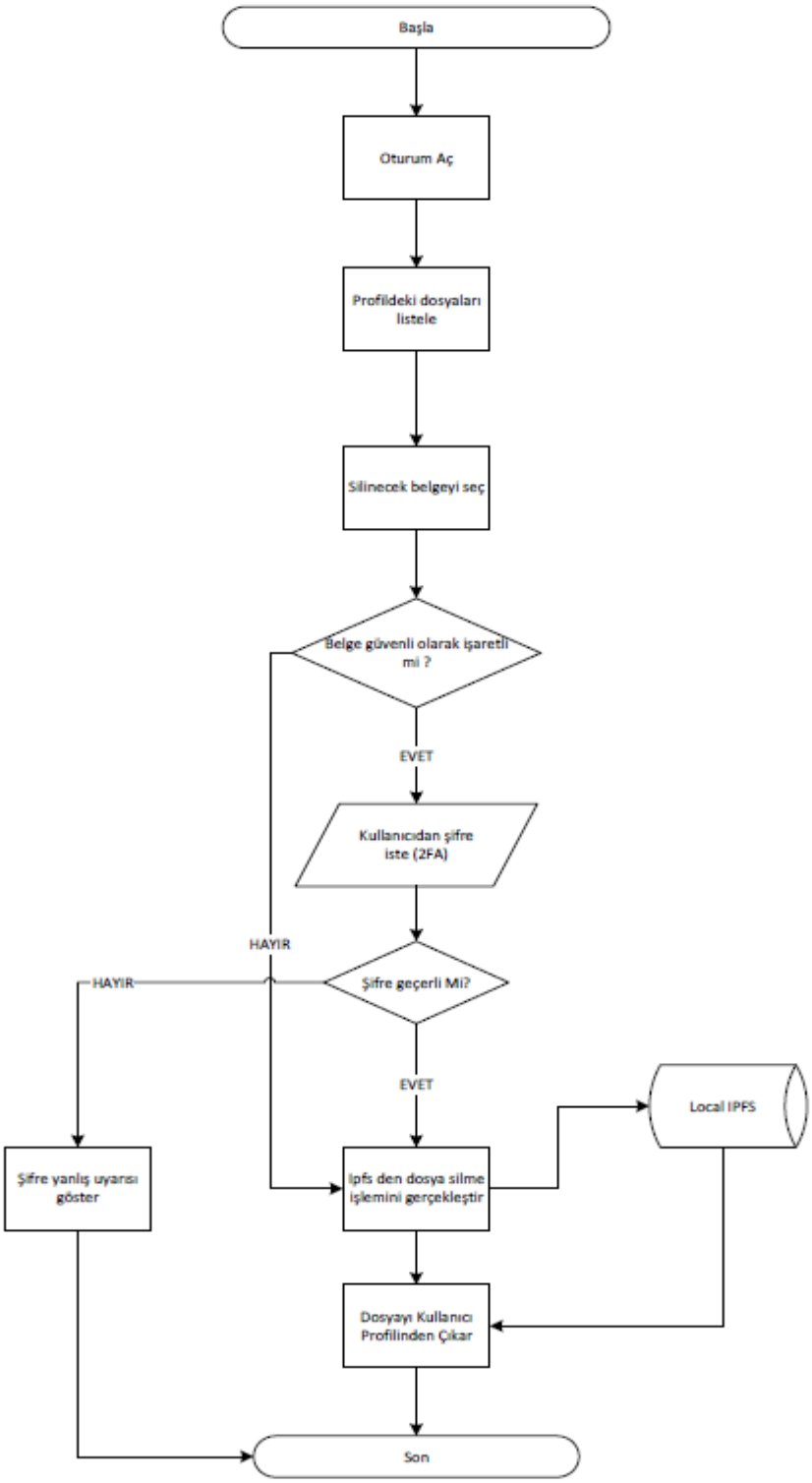
Şekil 8.15 Belge Yükleme Akış Diyagramı

8.2.6 Belge İndirme Akış Diyagramı



Şekil 8.16 Belge İndirme Akış Diyagramı

8.2.7 Belge Silme Akış Diyagramı



Şekil 8.17 Belge Silme Akış Diyagramı

BÖLÜM 9

SONUÇLAR VE TARTIŞMA

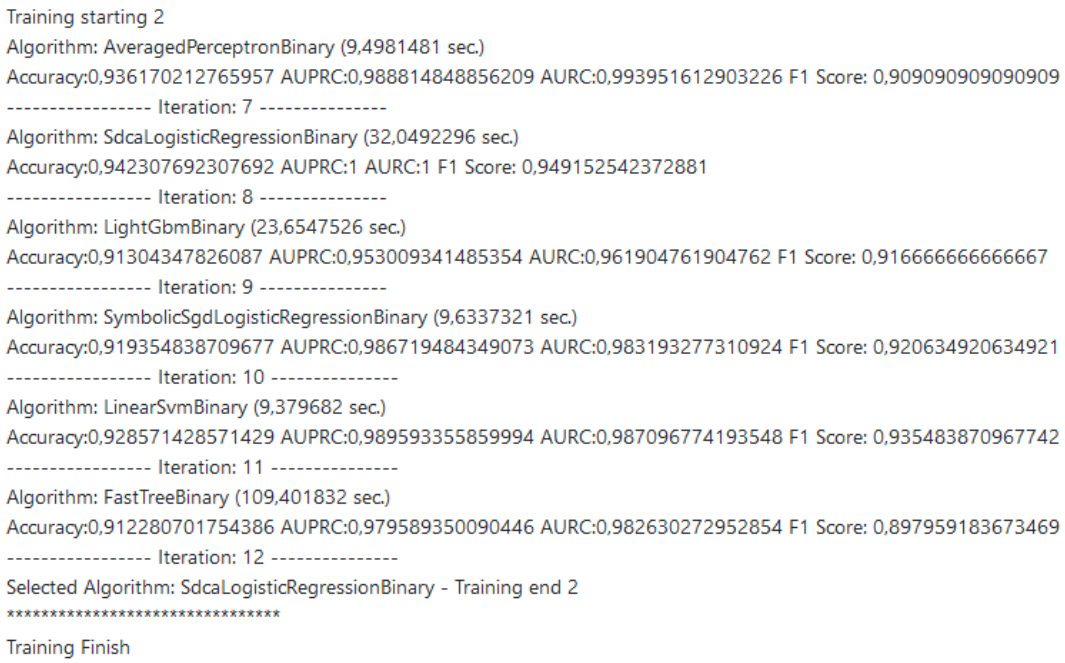
Merkezi depolama yöntemi kullanan kurum ve kuruluşların güvenlik endişeleri her geçen gün artmaktadır. Kurumların bilgi güvenliklerine yönelik aldıkları tedbirler çoğu zaman stratejik bir çözüm olmamakla birlikte yapılan donanımsal masraflar maliyeti de arttırmaktadır. Donanımların artması beraberinde bakım masraflarını getirmekte ve yönetimini zorlaştırmaktadır. Blok zinciri ile hayatımıza giren dağıtık uygulamalar (Dapps) kavramı birçok alanda kullanım imkânı sağlamıştır.

Çalışmamızda güvenliği arttırıcı birçok yöntemin entegre bir yapıda çalışması sağlanmıştır. Belge sınıflama işleminde güvenli ve güvensiz olarak ayrılmış 600 belgenin %90'ı 540 tanesi eğitim %10'u yani 60 tanesi ise test verisi olarak ayarlanmıştır. Sınıflandırma için toplam 2 adet veri seti oluşturulmuştur. Her eğitim 600 evrak içerisinden rastgele seçilen 540 adet evrak ile eğitilmiş ve en iyi algoritma otomatik olarak seçilmiştir. Toplam 6 algoritma üzerinde eğitim denenmiş ve 6. İterasyondan sonra belirlenmiştir. Şekil 9.1'de Eğitim Test Sonucu gösterilmiştir.

```
Preparing Learning Environment
********************************
Loading user dataset...
Loaded user dataset count: 0
Loading base dataset...
Loaded base dataset count: 600
Clearing old trainings...
-----------------------------------
Training starting 1
Algorithm: AveragedPerceptronBinary (10,5497508 sec.)
Accuracy:0,918918918918919 AUPRC:0,988713686982935 AURC:0,987573099415205 F1 Score: 0,923076923076923
---------- Iteration: 1 ----------
Algorithm: SdcaLogisticRegressionBinary (30,9385065 sec.)
Accuracy:0,941176470588235 AUPRC:0,952503426729799 AURC:0,959627329192547 F1 Score: 0,936170212765957
---------- Iteration: 2 ----------
Algorithm: LightGbmBinary (24,3486847 sec.)
Accuracy:0,928571428571429 AUPRC:0,987321720181276 AURC:0,987228607918263 F1 Score: 0,925925925925926
---------- Iteration: 3 ----------
Algorithm: SymbolicSgdLogisticRegressionBinary (9,9380593 sec.)
Accuracy:0,894736842105263 AUPRC:0,970334453539812 AURC:0,95875 F1 Score: 0,903225806451613
---------- Iteration: 4 ----------
Algorithm: LinearSvmBinary (9,6670905 sec.)
Accuracy:0,891304347826087 AUPRC:0,969186585007986 AURC:0,962121212121212 F1 Score: 0,893617021276596
---------- Iteration: 5 ----------
Algorithm: FastTreeBinary (107,6566321 sec.)
Accuracy:0,934782608695652 AUPRC:0,978289354480039 AURC:0,971590909090909 F1 Score: 0,933333333333333
---------- Iteration: 6 ----------
Selected Algorithm: SdcaLogisticRegressionBinary - Training end 1
********************************
```

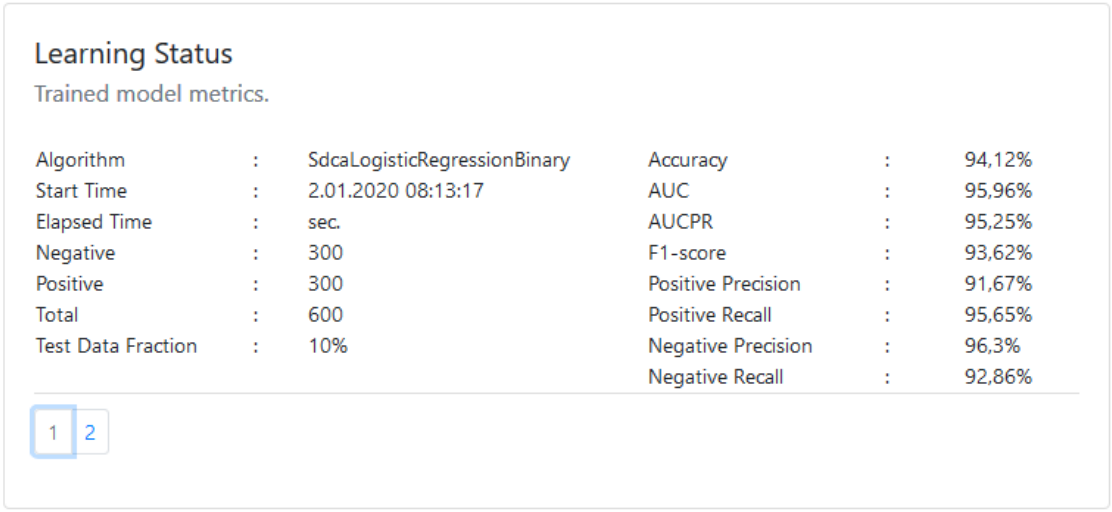
Şekil 9.1 Eğitim Test Sonucu 1

İlk eğitim seti oluşturulup tamamlandığında 2. eğitim için tekrar rastgele seçilen 600 evraktan %90'ı eğitim %10 test için seçilip işlem başlatılmıştır. 2. eğitim seti de "SdcaLogisticRegression" algoritmasını seçip işlemi tamamlamıştır. Şekil 9.2 de sonuçlar gösterilmiştir.

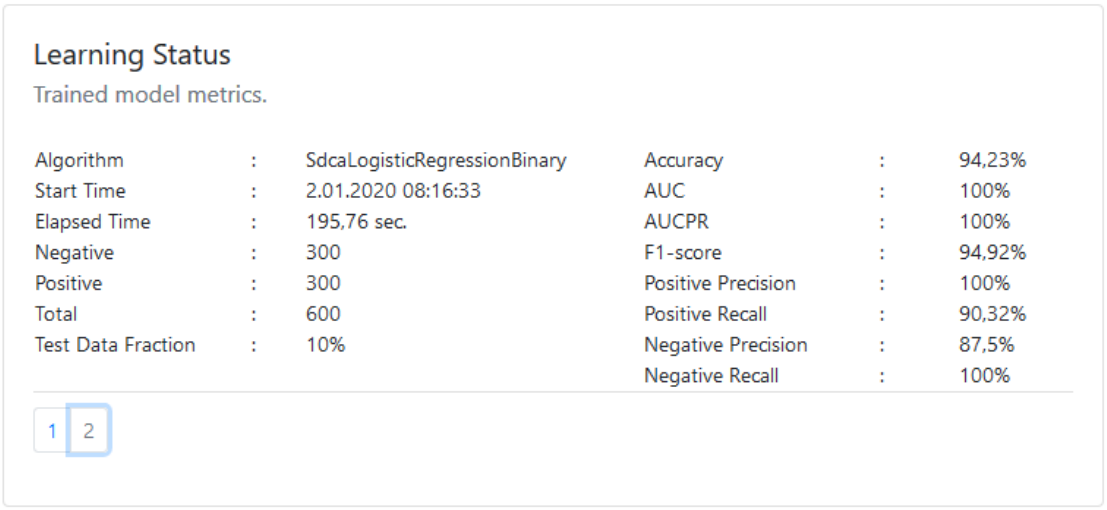


Şekil 9.2 Test Eğitim Sonucu 2

Eğitimler tamamlandıktan ve belgeler sisteme yüklendikten sonra her belge için 2 eğitim seti devreye girmiş true ya da false olarak sonuçlar dönmüştür. Sonuçlar “True - True” olması durumunda güvenli, “False - False” dönmesi durumunda ise güvensiz belge olarak kayıt edilmiştir. “False - True” ya da “True – False” olması durumunda ise kararsız kaldığı kabul edilmiş ve kullanıcı bilgisine sunulmuş karar kullanıcıya bırakılmıştır. Kullanıcının verdiği karar neticesinde eğitim seti yeni belge ile birlikte yeniden eğitilmiştir. Kullanıcıya bırakılan kararlar önemli ve önemsiz olarak denenmiş ve silinip tekrar eklendiğinde tanımlandığı sınıfa otomatik olarak aktarıldığı gözlemlenmiştir. Eğitimler tamamlandıktan sonra genel sonuçlar Şekil 9.3 ve Şekil 9.4 te gösterilmiştir.



Şekil 9.3 Genel Eğitim Sonuçları 1



Şekil 9.4 Genel Eğitim Sonuçları 2

Şekil 9.3 ve Şekil 9.4 eğitim setleri üzerinde bir test işlemi yapılmıştır. Bu test işlemi için 6 kategoriden rastgele 10'ar evrak seçilmiştir. İkili sınıflandırma işlemi yaptığımızdan pozitif (Önemli) ve negatif (önemsiz) olarak ayırdığımız belgeler için sonuçlar Çizelge 9.1 Test Sonuçları'ın da gösterilmiştir.

Çizelge 9.1 Test Sonuçları

Döküman	Önemli Belge Mi ?	1.Eğitim Seti Kararı	Doğruluk	2.Eğitim Seti Kararı	Doğruluk	Son Karar
Ekonomi_(135).txt	Doğru	**Doğru**	%45.00	**Doğru**	%53,29	**Doğru**
Ekonomi_(365).txt	Doğru	**Yanlış**	%-17.91	**Yanlış**	%-26.81	**Yanlış**
Ekonomi_(539).txt	Doğru	**Doğru**	%23.42	**Doğru**	%62,66	**Doğru**
Ekonomi_(190).txt	Doğru	**Doğru**	%53,46	**Doğru**	%64.03	**Doğru**
Ekonomi_(400).txt	Doğru	**Doğru**	%16,78	**Doğru**	%50.03	**Doğru**
Ekonomi_(550).txt	Doğru	**Yanlış**	%-17.56	**Yanlış**	%-1.42	**Yanlış**
Ekonomi_(125).txt	Doğru	**Doğru**	%28,69	**Doğru**	%50,52	**Doğru**
Ekonomi_(362).txt	Doğru	**Doğru**	%13.06	**Doğru**	%23.06	**Doğru**
Ekonomi_(531).txt	Doğru	**Yanlış**	%-1.84	**Doğru**	%10.38	**Kulanıcı Kararı**
Ekonomi_(270).txt	Doğru	**Doğru**	%22.27	**Doğru**	%28,50	**Doğru**
K&S_(244).txt	Doğru	**Doğru**	%70.07	**Doğru**	%73,65	**Doğru**
K&S_(121).txt	Doğru	**Doğru**	%39,74	**Doğru**	%45.03	**Doğru**
K&S_(230).txt	Doğru	**Doğru**	%54,60	**Doğru**	%63.03	**Doğru**
K&S_(300).txt	Doğru	**Doğru**	%18.36	**Doğru**	%73,24	**Doğru**
K&S_(316).txt	Doğru	**Doğru**	%44.09	**Doğru**	%54,51	**Doğru**
K&S_(481).txt	Doğru	**Doğru**	%42,16	**Doğru**	%86.07	**Doğru**
K&S_(526).txt	Doğru	**Doğru**	%19,85	**Doğru**	%31,67	**Doğru**
K&S_(600).txt	Doğru	**Doğru**	%61,95	**Doğru**	%75.00	**Doğru**
K&S_(178).txt	Doğru	**Doğru**	%40,45	**Doğru**	%39,80	**Doğru**
K&S_(288).txt	Doğru	**Doğru**	%86,20	**Doğru**	%51.00	**Doğru**
Saglik_(288).txt	Doğru	**Doğru**	%79,70	**Doğru**	%70,10	**Doğru**
Saglik_(172).txt	Doğru	**Doğru**	%30,80	**Doğru**	%19,89	**Doğru**
Saglik_(406).txt	Doğru	**Doğru**	%113,86	**Doğru**	%74,70	**Doğru**
Saglik_(508).txt	Doğru	**Doğru**	%86.04	**Doğru**	%68,47	**Doğru**
Saglik_(304).txt	Doğru	**Doğru**	%21.49	**Doğru**	%15,83	**Doğru**
Saglik_(454).txt	Doğru	**Doğru**	%34,17	**Doğru**	%32,79	**Doğru**
Saglik_(543).txt	Doğru	**Doğru**	%100.00	**Doğru**	%52,50	**Doğru**
Saglik_(377).txt	Doğru	**Doğru**	%100.00	**Doğru**	%90,79	**Doğru**

Saglik_(107).txt	Doğru	**Doğru**	%71,46	**Doğru**	%76,32	**Doğru**
Saglik_(199).txt	Doğru	**Doğru**	%91,42	**Doğru**	%71,84	**Doğru**
Siyaset_(199).txt	Yanlış	**Yanlış**	%-33.02	**Yanlış**	%-37.25	**Yanlış**
Siyaset_(319).txt	Yanlış	**Yanlış**	%-32.04	**Yanlış**	%-30.94	**Yanlış**
Siyaset_(553).txt	Yanlış	**Yanlış**	%-9.44	**Yanlış**	%-7.83	**Yanlış**
Siyaset_(401).txt	Yanlış	**Yanlış**	%-100.0	**Yanlış**	%-22.71	**Yanlış**
Siyaset_(265).txt	Yanlış	**Yanlış**	%-44.51	**Yanlış**	%-38.06	**Yanlış**
Siyaset_(110).txt	Yanlış	**Yanlış**	%-32.92	**Yanlış**	%-22.75	**Yanlış**
Siyaset_(135).txt	Yanlış	**Yanlış**	%-9.8	**Yanlış**	%-2.89	**Yanlış**
Siyaset_(487).txt	Yanlış	**Doğru**	%12,84	**Doğru**	%23,61	**Doğru**
Siyaset_(203).txt	Yanlış	**Yanlış**	%-91.38	**Yanlış**	%-7.78	**Yanlış**
Siyaset_(269).txt	Yanlış	**Yanlış**	%-16.47	**Yanlış**	%-9.5	**Yanlış**
Spor_(124).txt	**Yanlış**	**Yanlış**	%-39.78	**Yanlış**	%-37.58	**Yanlış**
Spor_(501).txt	**Yanlış**	**Yanlış**	%-30.18	**Yanlış**	%-15.39	**Yanlış**
Spor_(600).txt	**Yanlış**	**Yanlış**	%-89.04	**Yanlış**	%-99.85	**Yanlış**
Spor_(229).txt	**Yanlış**	**Yanlış**	%4,89	**Doğru**	%0,75	**Kullanıcı Kararı**
Spor_(445).txt	**Yanlış**	**Yanlış**	%-13.11	**Yanlış**	%-4.90	**Yanlış**
Spor_(127).txt	**Yanlış**	**Yanlış**	%-29.20	**Yanlış**	%-53.24	**Yanlış**
Spor_(488).txt	**Yanlış**	**Yanlış**	%-8.40	**Yanlış**	%-12.30	**Yanlış**
Spor_(489).txt	**Yanlış**	**Yanlış**	%-33.09	**Yanlış**	%-20.60	**Yanlış**
Spor_(202).txt	**Yanlış**	**Yanlış**	%-52.08	**Yanlış**	%-72.23	**Yanlış**
Spor_(105).txt	**Yanlış**	**Yanlış**	%-34.76	**Yanlış**	%-57.82	**Yanlış**
Teknoloji_(111).txt	**Yanlış**	**Yanlış**	%-48.98	**Yanlış**	%-30.20	**Yanlış**
Teknoloji_(178).txt	**Yanlış**	**Yanlış**	%-46.12	**Yanlış**	%-45.43	**Yanlış**
Teknoloji_(332).txt	**Yanlış**	**Yanlış**	%-21.53	**Doğru**	%0.23	**Kullanıcı Kararı**
Teknoloji_(137).txt	**Yanlış**	**Yanlış**	%-77.17	**Yanlış**	%-58.58	**Yanlış**
Teknoloji_(560).txt	**Yanlış**	**Doğru**	%6.31	**Doğru**	%26,13	**Doğru**
Teknoloji_(436).txt	**Yanlış**	**Doğru**	%11.10	**Doğru**	%33,14	**Doğru**
Teknoloji_(555).txt	**Yanlış**	**Yanlış**	%-9.92	**Yanlış**	%-4.65	**Yanlış**

Teknoloji_(597).txt	**Yanlış**	**Yanlış**	%-10.97	**Doğru**	%33,38	**Kullanıcı Kararı**
Teknoloji_(222).txt	**Yanlış**	**Yanlış**	%-11.96	**Doğru**	%4.53	**Kullanıcı Kararı**
Teknoloji_(333).txt	**Yanlış**	**Yanlış**	%-28.18	**Yanlış**	%-12.26	**Yanlış**

Doğruluk değeri 0 (Sıfır) dan büyük pozitif değerler **önemli** olarak, 0 (sıfır) dan küçük negatif değerler ise **önemsiz** evrak olarak sınıflandırılmıştır. Eğitim seti sonuçlarının farklı olması durumu kararsız olarak kabul edilmiş ve kullanıcı bilgisine adaptif olarak danışılmıştır. Çizelge 9.1'de ki sonuçlar değerlendirildiğinde Çizelge 9.2'de ki genel sonuçlar elde edilmiştir.

Çizelge 9.2 Genel Sonuçlar

	1.Eğitim Seti- Algoritma: Ortamalama Algılayacı (Avaraged Perceptron)					
	Kategori	**Doğru Tahmin Sayısı**	**Yanlış Tahmin Sayısı**	**Başarı Oranı %**	**Kullanıcı Karar Sayısı**	**Kullanıcı Kararından Sonra Başarı Oranı %**
Önemli	Ekonomi	7	3	%70	1	**^%80**
	Kültür	10	0	%100	0	%100
	Sağlık	10	0	%100	0	%100
Önemsiz	Polika	9	1	%90	0	%90
	Spor	10	0	%90	0	%90
	Teknoloji	8	2	%80	0	%80
	2. Eğitim Seti – Algoritma: Sdca Lojistik Regresyon (Logistic Regression)					
Önemli	Ekonomi	8	2	%80	0	%80
	Kültür	10	0	%100	0	%100
	Sağlık	10	0	%100	0	%100
Önemsiz	Politika	9	1	%90	0	%90
	Spor	9	1	%90	1	**^%100**
	Teknoloji	5	5	%50	3	**^%80**

Sonuçlar incelendiğinde1. Eğitim Seti test sonucunda 6 kategori için %88,33'lük genel bir başarı elde edilmiş ve kullanıcı desteği ile bu başarı %90 'a çıkartılmıştır.2.Eğitim Seti test sonucunda ise %85'lik bir başarı elde edilmiş ve bu oran kullanıcı desteği ile %91,66 ya çıkartılmıştır. Bu sonuçlar belgelerin güvenli ve güvensiz olarak

ayrılmasında ve güvenli belgeler için 2FA'nın devreye alınmasında başarılı olduğunu göstermiştir.

Sonuç olarak yapılan test çalışmasında önemli ve önemsiz olarak makine öğrenmesi ile belirlenen tüm evraklar IPFS motoruna gönderilmiş ve başarılı bir şekilde bloklara ayrılmıştır. Bloklardaki en ufak veri değişikliğinin dosya yapısında bozulmaya yol açtığı gözlemlenmiş silinmesi durumunda ise blok doğrulaması yapamadığından belgeye ulaşmak mümkün olmamıştır. Bu sonuçlara göre bloklardaki en ufak değişikliğin belgenin orjinalliği bozduğu ve erişilemediğini göstermiştir. Belge güvenliği etkenleri IPFS ile arttırılmıştır.

KAYNAKLAR

Akleylek, S., Yıldırım, H., & Yüce Tok, Z. (2011). *Kriptoloji ve uygulama alanları: açık anahtar altyapısı ve kayıtlı elektronik posta. Akademik Bilişim*, 713-718.

Automated Machine Learning.(2019). 10 Aralık 2019 tarihinde https://en.wikipedia.org/wiki/Automated_machine_learning adresinden erişildi.

Bambara, J., & Allen, P. (2018). *BLOCKHAIN A Practical Guide to Developing Business, Law, and Technology Solutions.* https://books.google.com.tr/books

Benet, J. (2014). *Ipfs-content addressed, versioned, p2p file system*. ArXiv:1407.3561.

BeyazNet.(2019). 5 Aralık 2019 tarihinde https://www.beyaz.net/tr/guvenlik/makaleler/kvkk_ya_neden_ihtiyac_duyulmustur.html adresinden erişildi.

Binary Classification.(2019). 10 Kasım 2019 tarihinde https://en.wikipedia.org/wiki/Binary_classification adresinden erişildi.

BinaryClassification_AutoML(2019). 1 Ekim 2019 tarihinde https://github.com/dotnet/machinelearning-samples/tree/master/samples/csharp/getting-started/BinaryClassification_AutoML?fbclid=IwAR2I-W-LWNJVtApXcHRD6ouulo54_HgZEVfYpGUFv_5pa62CKQBclr4-BZ4 adresinden erişildi.

Comparison Of Cryptographic Hash Functions.(2019).11 Aralık 2019 tarihinde https://en.wikipedia.org/wiki/Comparison_of_cryptographic_hash_functions adresinden erişildi.

Çetin, H. (2014). *Kişisel veri güvenliği ve kullanıcıların farkındalık düzeylerinin incelenmesi.(29) 2014, 86-105*.

Dağsalı, A. (2019).31 Ekim 2019 tarihinde http://yasaizleme.org.tr/kisisel-verilerin-korunmasi-kanununun-uygulanma-sureci/ adresinden erişildi.

Farmer, C. (2018). 29 Ekim 2019 tarihinde https://medium.com/textileio/whats-really-happening-when-you-add-a-file-to-ipfs-ae3b8b5e4b0f adresinden erişildi.

Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow:*

Concepts, Tools, and Techniques to Build Intelligent Systems. https://books.google.com.tr/books

Google Authenticator.(2019). 8 Kasım 2019 tarihinde https://en.wikipedia.org/wiki/Google_Authenticator adresinden erişildi.

Ketizmen, D., & Ülküderner, Ç. (2007). *E-Devlet Uygulamalarında Kişisel Verilerin Korun(ma)ması*. XII."Türkiye'de İnternet" Konferansı.

K-nearest Neighbors Algorithm.(2019). 6 Kasım 2019 tarihinde https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm adresinden erişildi.

Lantz, B. (2013). *Machine Learning with R Kindle Edition.* https://books.google.com.tr/books

MicrosoftSDCA.(2019). 10 Aralık 2019 tarihinde Https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.sdcalogisticregressionbinarytrainer?view=ml-dotnet adresinden erişildi.

Naive Bayes Sınıflandırıcı.(2019). 23 Kasım 2019 tarihinde Https://tr.wikipedia.org/wiki/naive_bayes_sınıflandırıcı adresinden erişildi.

Patsakis, C., & Casino, F. (2019). *Hydras and IPFS: A Decentralised Playground for Malware International Journal of Information Security*. doi:10.1007/s10207-019-00443-0".

Random Forest (2019). 6 Kasım 2019 tarihinde https://en.wikipedia.org/wiki/Random_forest adresinden erişildi.

Regresyon Analizi. (2019).10 Kasım 2019 tarihinde https://tr.wikipedia.org/wiki/Regresyon_analizi adresinden erişildi.

SHA-1. (2019).21 Ekim 2019 tarihinde https://tr.wikipedia.org/wiki/SHA-1 adresinden erişildi.

SHA-2.(2019). 22 Ekim 2019 tarihinde https://tr.wikipedia.org/wiki/SHA-2 adresinden erişildi.

Singhal, B., Dhmeja, G., & Panda, P. (2018). *Beginin Blockchain.* https://books.google.com.tr/books

sirantonakis, G., Petsas, T., Athanasopoulos, E., & Ioannidis, S. (2015). *Two-factor Authentication: Is the World Ready?Quantifying 2FA Adoption*. *doi:* 10.1145/2751323.2751327.

Two-Class Averaged Perceptron. (2019). 10 Aralık 2019 tarihinde

Https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/two-class-averaged-perceptron adresinden erişildi.

Two-factor-authentication.(2019).11 Aralık 2019 tarihinde https://searchsecurity.techtarget.com/definition/two-factor-authentication adresinden erişildi.

Watts, C. (2019). Hash and Mac Algorithms. Contents Hash Functions Secure Hash Algorithm HMAC [PowerPoint slaytı]. 19 Ekim 2019 tarihinde https://slideplayer.com/slide/8199025/ adresinden erişildi.

ÖZGEÇMİŞ

KEMAL HAKAN
Doğum Tarihi: 02.04.1986

EĞİTİM

2007-2009: Çukurova Üniversitesi, Adana MYO, Bilgisayar Programcılığı (Adana)

2011-2014: Trakya Üniversitesi, Mühendislik-Mimarlık Fakültersi, Bilgisayar Mühendisliği (Edirne)

2018-……: Trakya Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı (Edirne)