T.C.

İSTANBUL MEDENİYET ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİYOLOJİK VERİ BİLİMİ

GENOMİK ÇALIŞMALARDA KULLANILAN MAKİNE ÖĞRENMESİ MODELLERİNİN BLOKZİNCİR TEMELLİ BİR EKO SİSTEMDE FEDERE ÖĞRENME YÖNTEMİYLE GELİŞTİRİLMESİ İÇİN KAVRAMSAL BİR ÇALIŞMA

YÜKSEK LİSANS TEZİ

ERKAN OKTAY

HAZİRAN 2022

T.C.

İSTANBUL MEDENİYET ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİYOLOJİK VERİ BİLİMİ

GENOMİK ÇALIŞMALARDA KULLANILAN MAKİNE ÖĞRENMESİ MODELLERİNİN BLOKZİNCİR TEMELLİ BİR EKO SİSTEMDE FEDERE ÖĞRENME YÖNTEMİYLE GELİŞTİRİLMESİ İÇİN KAVRAMSAL BİR ÇALIŞMA

YÜKSEK LİSANS TEZİ

ERKAN OKTAY

DANIŞMAN

DR. ÖĞR. ÜYESİ ARAFAT SALİH AYDINER

EŞ DANIŞMAN

DR. ÖĞR. ÜYESİ MUHAMMED ERKAN KARABEKMEZ

HAZİRAN 2022

BİLDİRİM

Hazırladığım tezin tamamen kendi çalışmam olduğunu, akademik ve etik kuralları gözeterek çalıştığımı ve her alıntıya kaynak gösterdiğimi taahhüt ederim

ERKAN OKTAY

Danışmanlığını yaptığım işbu tezin tamamen öğrencinin çalışması olduğunu, akademik ve etik kuralları gözeterek çalıştığını taahhüt ederim.

Dr. Öğr. Üyesi Arafat Salih Aydıner

ONAY

ERKAN OKTAY tarafından hazırlanan Genomik Çalışmalarda Kullanılan Makine Öğrenmesi Modellerinin Blokzincir Temelli Bir Eko Sistemde Federe Öğrenme Yöntemiyle Geliştirilmesi İçin Kavramsal Bir Çalışma başlıklı bu yüksek lisans tezi, Biyolojik Veri Bilimi Bilim Dalında hazırlanmış ve jürimiz tarafından kabul edilmiştir.

JÜRİ ÜYELERİ **İMZA**

Tez Danışmanı:

Dr. Öğretim Üyesi Arafat Salih Aydıner

Kurumu: İstanbul Medeniyet Üniversitesi

Tez Eş Danışmanı:

Dr. Öğr. Üyesi Muhammed Erkan KARABEKMEZ

Kurumu: İstanbul Medeniyet Üniversitesi

Üyeler:

Prof. Dr. Kazım Yalçın ARGA

Kurumu: Marmara Üniversitesi

Dr. Öğr. Üyesi Ahmet KAPLAN

Kurumu: İbn Haldun Universitesi

Prof. Dr. Handan ANKARALI

Kurumu: İstanbul Medeniyet Üniversitesi

Tez Savunma Tarihi: 29.06.2022

TEŞEKKÜR

Tez çalışmasını yakından takip ederek akademik ve deneyimlerini benimle paylaşan, tezin her aşamasında bilimsel katkı, ilgi ve desteğini hiçbir zaman esirgemeyen, sabırla yönlendiren, kendisinden çok şey öğrendiğim değerli danışman hocam, Sayın Dr. Arafat Salih Aydıner'e ve Sayın Dr. Muhammed Erkan KARABEKMEZ'e çok teşekkür ederim.

Tez çalışmam boyunca bana destek olan ve sabır gösteren aileme, dostlarıma ve özellikle eşim Filiz Hasasu Oktay'a zaman zaman ihmal ettiğim çocuklarım Bahar, Yusuf Aybar ve Maysa'ya, bu yüksek lisans programına katılmama vesile olan değerli arkadaşım Ahmet Şener'e en derin duygularımla teşekkür ederim.

İÇİNDEKİLER

KISALTMALAR

ABI	: Uygulama İkili Arayüz (Application Binary Interface)
ACLU	: Amerikan Sivil Özgürlükler Birliği
CPU	: Merkezi İşleme Ünitesi
BFT	: Bizans Hata Toleransı
BT	: Bilgisayarlı Tomografi
ÇTGH	: Çok Taraflı Güvenli Hesaplama
Dapps	: Merkeziyetsiz uygulamalar
DAG	: Yönlendirilmiş Döngüsel Grafikler
DAO	: Merkeziyetsiz Özerk Organizasyon
DHT	: Dağıtık Hash Tabloları
DNA	: Deoksiribonükleik Asit
DoS	: Hizmet Reddi (Denial of Service)
DPoS	: Delegasyonlu Hissenin İspatı
EVM	: Ethereum Sanal Makinesi
FL	: Federe Öğrenme
FBI:	: Federal Soruşturma Bürosu
GB	: Gigabyte
GDPR	: Genel Veri Koruma Regülasyonu
GPU	: Grafik İşleme Ünitesi
GWAS	: Genome Çapında İlişki Analizi
HDFS	: Hadoop Dağıtık Dosya Deposu
HE	: Homomorfik Şifreleme
HGP	: İnsan Genom Projesi
HIPPIA	: Taşınabilirlik ve Sorumluluk Yasası
IoT	: Nesnelerin İnterneti
IoMT	: Tıbbi Nesnelerin İnterneti
IPFS	: Gezegenler Arası Dosya Sistemi (interplanetary File System)

ISP	: İnternet Sağlayıcı Firma
KVKK	: Kişisel Verilerin Korunması Kanunu
ML	: Makine Öğrenmesi
NFT	: Takas Edilmez Token
P2P	: Peer to peer
PBFT	: Pratik Bizans Hata Toleransı
PoA	: Aktivitenin İspatı
PoS	: Hissenin İspatı (proof of stake)
PoW	: Çalışmanın İspatı (proof of work)
RNA	: Ribonucleic Asit
RPC	: Uzaktan Yordam Çağrısı
SGD	: Stokastik Gradyan İnişi
SGX	: Yazılım Koruma Uzantısı
T.B.M.M	: Türkiye Büyük Millet Meclisi
UML	: Birleşik Modelleme Dili
YZ	: Yapay Zekâ

TABLO LİSTESİ

ŞEKİL LİSTESİ

ÖZET

GENOMİK ÇALIŞMLARDA KULLANILAN MAKİNE ÖĞRENMESİ MODELLERİNİN BLOKZİNCİR TEMELLİ BİR EKO SİSTEMDE FEDERE ÖĞRENME YÖNTEMİYLE GELİŞTİRİLMESİ İÇİN KAVRAMSAL BİR ÇALIŞMA

Oktay, Erkan

Yüksek Lisans Tezi, Lisansüstü Eğitim Enstitüsü; Biyolojik Veri Bilimi Programı

Danışman: Dr. Öğr. Üy. Arafat Salih Aydıner

Eş Danışman: Dr. Öğr. Üyesi Muhammed Erkan KARABEKMEZ

Haziran, 2022, 103 sayfa

Sağlık teknolojilerindeki son gelişmelerle genetik verinin toplanması, kullanılması ve paylaşımı gittikçe hız kazanmaya başladı. Genetik veri, hastalıkların önlenmesi, doğru teşhisler koyma ve kişiselleştirilmiş tedaviler için fırsatlar sunmaktadır. Özellikle makine öğrenmesi (ML) alanındaki ilerlemelerle beraber genetik alandaki ML uygulamaları da hız kazanmaktadır. Ancak genetik alanda ML modellerinin geliştirilmesi için gerekli verinin kullanımında yasal ve teknolojik engeller vardır. Genetik veri kişisel veri sınıfına girer ve bu verilerin kullanımı ancak yasaların izin verdiği ölçülerde olabilir. Bir diğer engel ise genetik verilerin yüksek hacimli veriler olmasıdır. Bundan dolayı işlenmesi ve paylaşılması için güçlü altyapı gerektirmektedir. Genetik alandaki ML çalışmaları genelde bir araştırma kuruluşunun elindeki veri ve bilgi ile sınırlı kalmaktadır. ML alanında benzer çalışmalar yürüten, farklı araştırma kuruluşları arasında ortak çalışmaya imkân tanıyan, güvenli bir iş birliği platformunun kurulması tekrar kullanılabilirlik açısından ciddi verimlilik sağlayabilir. Ayrıca farklı bölgelerdeki kurumların bu tarz çalışmalara dahil edilmesi durumunda, bölgesel genetik farklılıkların sebep olduğu yanılma oranlarının azaltılabileceği düşünülmektedir. Bu çalışmada, yukarıda bahsedilen engellerin aşılarak, araştırma kuruluşları arasında iş birliğini destekleyecek blokzincir temelli bir

platform tasarımı sunulmaktadır. Çalışmanın mimari tasarımında “yazlımın 4+1 görünümü” referans alınmış olup dizayn, senaryolar, mantıksal ilişkiler, geliştirme adımları ve süreçler UML diyagramları ile açıklanmaktadır. Blokzincir ile kurumlar arasındaki koordinasyon merkeziyetsiz bir şekilde sağlanmış olacak ve geliştirilen ML modelleri federe öğrenme (FL) yardımı ile birleştirilerek ana modeller üretilebilecektir. İş birliği içerisinde üretilen modeller farklı kurumlar tarafından yeniden kullanılabilir ve tekrar tekrar eğitilerek güçlendirilebilir olacaklardır. Blokzincir alandaki çalışmaların birçoğu birey verisinin güvenli bir şekilde diğer kurumlar ile paylaşılması, veri sahipliğinin kullanıcıya verilmesi ve izinsiz veri kullanımının engellenmesine yönelik yapılmıştır. FL alanındaki çalışmalarda ise ML eğitim süreçlerinde cihazların iş birliği yapabileceği modeller sunulmuştur. Bizim önerdiğimiz tasarım bireyden ziyade kurumlar arasında ki iş birliğine odaklanır. Bu çalışma kurumların sunucuları üzerinde çalışan ML modellerinin bir araya getirilerek daha güçlü modellerin geliştirilebileceği bir ortam önerisi sunar. Bu çoklu iş birliği yaklaşımının performans ve verimliliği arttıracağı düşünülmektedir.

Anahtar Kelimeler: Blokzincir, Makine Öğrenmesi, Federe Öğrenme, Genetik, İş Birliği, Gizlilik Koruma

ABSTRACT

A CONCEPTUAL STUDY FOR DEVELOPING MACHINE LEARNING MODELS USED IN GENOMIC STUDIES BY FEDERATED LEARNING METHOD IN A BLOCKCHAIN-BASED ECO SYSTEM

Oktay, Erkan

Master Thesis, Graduate School of Education; Biological Data Science Program

Supervisor: Asst. Prof. Dr. Arafat Salih Aydiner

Pair Supervisor: Asst. Prof. Dr. Muhammed Erkan KARABEKMEZ

June, 2022,103 pages

Towards the recent developments in healthcare technologies, it is started to accelerate genomic data collecting, using, and sharing. Genetic data promises opportunities for disease prevention, accurate diagnoses, and personalized treatments. Especially with the advances in the field of machine learning (ML), its applications in the field of genetics have also accelerated. However, the usage of genomic data for the development of ML models has big obstacles in technological and legal. Genetic data is counted in the category of personal data and the use of this data can only be within the scope of permitted by law. Another obstacle is the high volume of genetic data. Therefore, it requires a strong infrastructure for processing and sharing ML studies in the field of genetics are generally limited by the data and information available at a single research institution. Having a secure cooperation environment between different research institutions carrying out similar ML application studies may provide significant efficiency because of the reusability of applications. In addition, it is thought that when institutions from different regions are included in such studies, the rate of the error caused by regional genetic differences will decrease. In this study, we propose a blockchain-based platform design that will support cooperation between research institutions by overcoming the aforementioned obstacles. Scenarios, logical relationships, development steps, and processes are explained with UML diagrams with reference to the design “4+1 architectural view model”. The collaboration of research institutions will be provided by blockchain network as decentralized. Furthermore, thanks to the federated learning (FL) method main models will be improved by combining the developed ML models. The Models developed in the edge

institutions will be reusable by other institutions and the models can be strengthened with the contribution of the stakeholders by training repeatedly. Most of the studies in the field of blockchain are related to secure data sharing, giving data ownership to the user, and preventing unauthorized data use. On the other hand, the studies in FL field have been focused on the cooperation of devices in ML training processes. According to our study, the collaboration of institutions is taken into consideration rather than individuals. It offers an environment for aggregating ML models, which are trained in the local servers of the institutions, to generate high performed models. It is thought that this multi-collaboration approach will increase performance and productivity.

Keywords: Mechine Learning, Blockchain, Federated Learning, Genetic, Collaboration, Privacy-Preserving

GİRİŞ

Günümüzde kişisel verilerin korunması ve bu verilerin kullanımları ile ilgili ciddi önlemler alınmıştır. Birçok ülkede olduğu gibi ülkemizde de yayınlanan kişisel verilerin korunması kanununda, bu tezin de kapsamına giren biyometrik ve genetik veriler özel nitelikli kişisel veriler olarak sınıflandırılmıştır. Bu verilerin işlenmesinin ancak ilgili kişinin açık rızası ile mümkün olabileceği açık bir şekilde belirtilmiştir (Türkiye Büyük Millet Meclisi (T.B.M.M), 2016). Ancak kanunlarımızda genetik veri kullanımının açık rızaya tabi tutulması, genetik veri bilimi araştırmaları ve özellikle de yepay zeka temelli çalışmalarda önemli bir engel teşkil etmektedir (Trinidad ve diğerleri, 2010). Kişilik hakları açısından genetik verilerin korunması her ne kadar önemli ise akademik alandaki gelişmelerin devamlılığını sağlamak da bir o kadar önemlidir. Bundan dolayı kişisel veriyi korumak ve yapay zeka (YZ) modellerinin güvenli bir şekilde geliştirilmesi arasında bir denge kurmak elzemdir (Mitrou, 2018). YZ ve makine öğrenmesi (ML) bilgisayarların veriyi kaynak olarak kullanabildiği, eğitilebilen programlardır. Verinin çokluğu, kalitesi ve niteliği başarılı ML algoritmalarının en önemli şartlarındandır (Geron, 2017). Son yıllarda bir ML tekniği olarak ortaya çıkan federe öğrenme (FL) sayesinde, verileri merkezi bir sistemde toplamadan modeli eğitmek mümkün olmuştur (McMahan ve diğerleri, 2017). Bu sayede yerel sistemlerdeki veriler güvende tutulmakta, ortak çalışma ile mevcut bir model geniş katılımcılar ile eğitilebilmektedir. Bu çalışmanın temel amacı genetik alanda geliştirilen ML modellerinde FL yöntemini kullanarak birçok araştırma kuruluşunun model eğitimi için iş birliği sağladığı merkeziyetsiz bir platform önerisi sunmaktır.

Sağlık alanındaki veriler özel niteliktedir. Sağlık verileri; sağlık alanında kullanılan çeşitli cihazlar tarafından üretilen metrikler, resim, video ya da genetik dizilim gibi verilerden oluşabilir (Fost vediğerleri, 2014). Bu farklı tipteki verilerin işlenmesi özel bir bilgi ve deneyim gerektirmektedir. Analitik çalışmalarda etiketlenmiş veri, denetimli öğrenme modelleri için çok kıymetlidir. Etiketleme, hangi tür şartlarda ya da özelliklerde hangi sonucun çıktığının veri üzerinde belirtilmesidir (Geron, 2017). Örneğin bir hastanın sağlık durumunu ele alarak uygulanan tedavinin işe yarayıp yaramamasının belirtilmesi ya da hangi röntgen resimlerinde kanser görülüp

görülmediğinin işaretlenmesi gibi düşünülebilir. Kurumların büyük efor ve maliyetlerle topladıkları etiketli verileri farklı çalışmalarda da kullanmak, araştırma çalışmalarında büyük faydalar sağlayabilir. Genetik verilerin sağlık verileri içerisinde işlenmesi, etiketlenmesi, anlamlandırılması ve bulunması en zor veri kaynaklarındandır. Genetik veri, dizileme cihazları tarafından oluşturulur. Bu cihazların oluşturduğu veriler referans genomlarla kıyaslanarak genom üzerindeki varyantlar belirlenir ve son olarakta her genin açıklaması yapılır. Tüm bu süreç muazzam miktarda hesaplama kaynağı gerektirir. Bu zorlu işlemler sonucunda oluşan yüksek hacimli verilerin depolanması ve paylaşılması oldukça zordur (Ozercan ve diğerleri, 2018). Ayrıca ülkemizde birçok sağlık kurumunun bu tür verileri elde etme konusunda imkanları sınırlıdır. Veriyi elde etmek için harcanan efor tekrarlanmadan yeniden kullanım yapılması mümkündür (Chan ve diğerleri, 2014). Fakat veri paylaşımı yapılırken kişisel verinin gizliliği açığa çıkabilir (Torkzadehmahani ve diğerleri, 2021) ve bu durumunda yukarıda açıklandığı gibi kanunlarımızda cezası vardır (Türkiye Büyük Millet Meclisi (T.B.M.M), 2016).

Veri paylaşımının sağladığı verimliliği başka yaklaşımlarla da sağlamak mümkündür. Bu tezde özel nitelikli genetik verilerin paylaşımına gerek kalmadan, sağlık alanında araştırma yapan hastane, üniversite, bilim merkezleri gibi kuruluşların analitik çalışmalarda iş birliği yapabilmelerine olanak tanıyan blokzincir temelli makine öğrenmesi tekniği önerilmiştir. Bu yapı ile direk açık veriyi paylaşmak yerine öğrenilmiş modellerin paylaşımını sağlayan bir mimari tasarım sunulmuştur. Bu sayede kurumlar ML yöntemi ile geliştirdikleri modelleri blokzincir ağı üzerinden paylaşabilecek ve diğer kuruluşların modellerini de kullanabilir olacaktır. Daha da fazlası, analitik çalışmalara katkı sağlayanlar için bir ödüllendirme mekanizması tasarlanarak iş birliğinin devamlılığının sağlanması da kurgulanmıştır. Tezde genetik veri çalışmaları konu edilse de, önerilecek yöntem genel amaçlı olarak farklı sektör ve veri modellemesi çalışmalarında da kullanılabilir olacaktır. Bu yaklaşım ile aşağıdaki alt sorunlara çözüm sağlanması hedeflenmektedir:

a) *Genetik veri kullanan ML çalışmalarında kişisel verinin korunması*: Bu çalışmada kişisel verinin korunmasi için bir ML tekniği olan Federe öğrenme (Federated Learning: FL) yöntemi kullanılmıştır. FL birçok varlığın (kurum, cihaz, kişi) merkezi sunucular ya da servis sağlayıcılar tarafından koordine

edilerek iş birliği içinde bir problemi çözmesine olanak tanır. Her bir varlık ham verisini kendi üzerinde tutar ve diğer taraflar ile ham veriyi direk payşamak yerine, öğrenme hedefine ulaşmak için, öğrenilen bilgi diğer taraflar ile paylaşılarak bilginin güncellenmesi sağlanır (Kairouz ve diğerleri, 2019). Bu yöntemde eğitilmiş model ağırlıkları paylaşılır. Eğitilmiş modellerin ağırlıkları ham veriyi barındırmadığı için kişisel verilerin güvenliği sağlanmış olur. Biz bu modeli farklı bir mimari ile kullanarak merkezi sunucu ihtiyacını kaldırmayı hedefliyoruz. Merkezi sunucunun yaptığı görevleri blokzincir ağına yıkarak merkeziyetsiz bir yapıda modellerin birleştirilmesini amaçlıyoruz.

b) *Sağlık kuruluşları arasında merkeziyetsiz iş birliğinin sağlanması*: Merkeziyetsiz yapılar ve bu yapıların yol açabileceği bazı sorunlar ve riskler BitCoin' in mucidi Satoshi Nakamato' nun 2009' da yayınladığı "Bitcoin: A Peer-to-Peer Electronic Cash System" makalesindeki çözüm ile minimuma indirilmiştir (Nakamoto, 2009). Sonrasında daha gelişmiş blokzincir alt yapıları oluşturulmuştur. Bunlardan bir tanesi de Ethereum blokzincir ağıdır (Wood, 2014). Sağlık kuruluşları arasındaki koordinasyonun sağlanması için, taraflar arasında belirli protokoller ile koordinasyonu mümkün kılan akıllı kontrat yapısının kullanıldığı, Ethereum temelli bir çözüm aranmıştır.

c) *Genişleyebilir bir mimari ile yeni katılımcıların sürece dahil edilebilmesi*: Blokzincir ağları doğası gereği genişleyebilir yapılardır. Halka açık Ethereum ağının kullanılması planlanarak çok geniş bir ağ üzerinde sistemin çalışması planlanmıştır. Ayrıca bu sisteme kimlerin katılımcı olabileceği, hangi roller ile hangi fonksiyonları yerine getirebilecekleri çalışılmış ve sistemde yüksek işlem gücü gerektiren öğrenme süreçleri katılımcıların yerel sunucularına bırakılarak optimum performans sağlanması düşünülmüştür.

d) *Katkı sağlayanların ödüllendirilmesi*: Son olarakta model geliştirmede faaliyet gösteren taraflara teşvik verilmesi düşünülmüştür. Bu sayede sistemin aktif tutulması ve yeni katılımcıların blokzincir ağına katılması için bir motivasyon sağlanması hedeflenmiştir. Blokzincir ağları genelde teşvik sorununa token bazlı ödüllendirme yöntemi ile çözüm sağlamaktadır. Buradan esinlenerek fayda sağlayan her katılımcıya ödüllendirme mekanizması tasarlanmıştır.

Bu çalışmanın önemi, kurgulanan sistemle araştırma kuruluşlarının merkeziyetsiz, sürdürülebilir ve güvenilir bir çerçevede iş birliği yapabilmesini sağlayarak, açık bilim felsefesini desteklemesidir. Bu sayede yapılan çalışmalar tekrar kullanılabilir olacak ve limitli kaynaklarla yürütülen çalışmalar daha geniş kitleler tarafından ele alınarak daha doğru sonuç üretmek mümkün olacaktır. Dolaylı olarak araştırmalar için ayrılan bütçe, zaman ve emek daha verimli bir şekilde kullanılmış olacaktır. Verimliliğin sağlanmasının yanı sıra kişinin en hassas verisi olabilecek genetik verinin mahremiyeti de güvence altına alınmış olacaktır.

BÖLÜM I

1. LİTERATÜR ÖZETİ

Blokzincir temelli federe öğrenme platformunun tasarımı için gerekli akademik literatüre bu bölümde yer verilmiştir. Genetik verinin tanımı, genetik veri mahremiyetinin etik ve yasal olarak nasıl ele alındığı, genetik mahremiyetin korunması için teknik yöntemler, merkeziyetsiz sistemler, blokzincir temelli genetik veri paylaşımı sağlayan çalışmalar, FL ve blokzincir teknolojileri ve bu teknoljilerin beraber kullanımı ile sağlanan çözümler ele alınmıştır. Bu çalışma etik, yasal ve teknolojik olarak birçok noktaya dokunduğu için literatür taraması geniş tutulmuştur

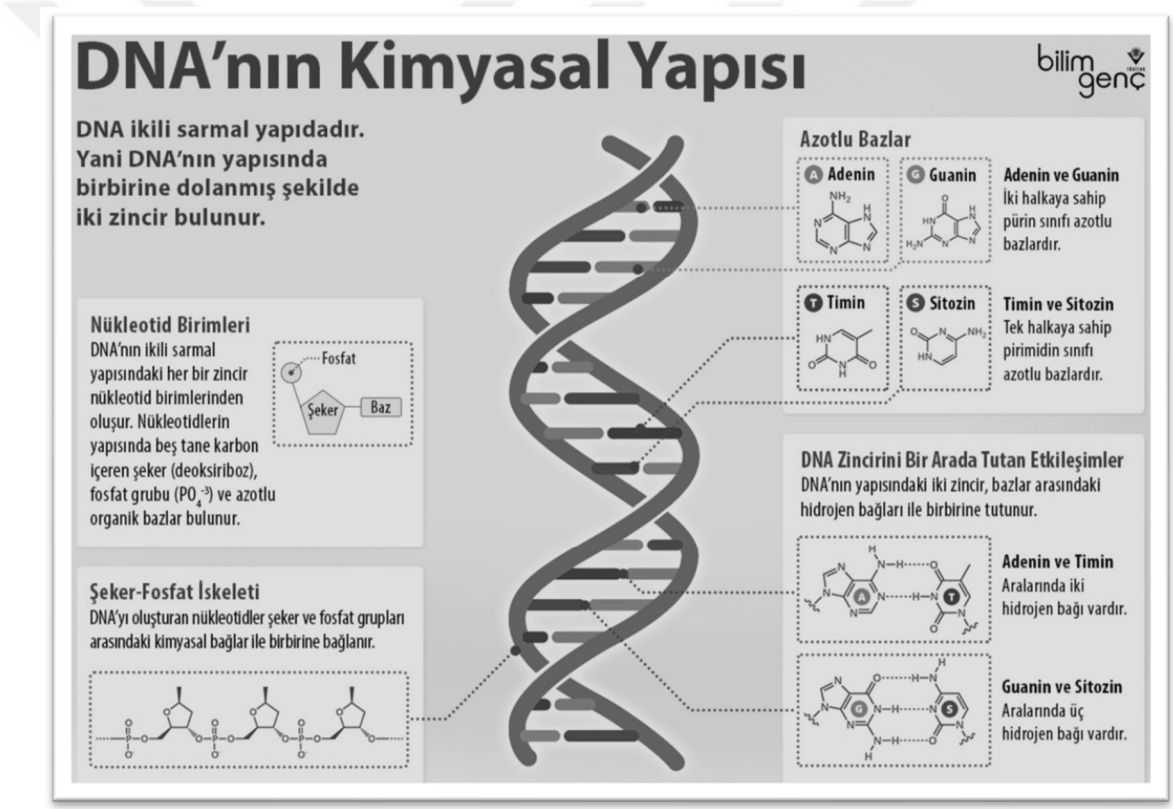
1.1 GENETİK VERİ TANIMI, MAHREMİYETİ VE YASAL ZEMİNİ

Bu bölümde genetik verinin tanımı verilerek tezin konusu olan genetik verinin işlenmesi ve paylaşılması ile ilgili yasal düzenlemeler ele alınacaktır. Genel bir bakış olarak genetik verinin tanımı, dünya genelinde genetik verinin nasıl ele alındığı, Türkiye'de yasal olarak genetik verinin kullanımı ile ilgili düzenlemeler, genetik verinin saklanması, paylaşılması ve kullanımıyla ilgili şartlar yer alacaktır.

1.1.1 Genetik Veri

Yeryüzündeki yaşam oldukça zengin ve çeşitlidir. Basit yapılardaki tek hücreli canlılardan, insan gibi gelişmiş çok hücreli canlılara kadar birçok yaşam formu dünyamızdaki hayatı oluşturmaktadır. Büyük ölçüde bu kadar geniş bir çeşitlilik olmasına rağmen mikroskobik düzeyde tüm canlıların temel moleküler özellikleri birbirine benzemektedir. Esasen yer yüzündeki yaşam formları çevrelerinden farklılaşma, bilgiyi kalıcı olarak tutabilme, bilgiyi bir sonraki nesillere aktarabilme ve enerji üretebilme gibi anahtar temalar ile açıklanabilmektedir. Bu fonksiyonlar yaşamın ortak en temel parçası olan hücreler tarafından gerçekleştirilmektedir. Hücre nükleik asit, protein, karbonhidrat ve yağlardan oluşmaktadır. Hücre içerisinde

biyolojik bilgi aktarımı kendini kolayca kopyalayabilen ve sonraki nesillere aktarılabilen bir çeşit nükleik asit olan deoksiribonükleik asit (DNA) tarafından gerçekleştirilir. Nükleotid olarak bilinen adenin(A), guanin (G), sitozin (C) ve timin (T) birbirlerine zayıf bağlarla tutunarak doğrusal polimer yapılar oluştururlar. Bu yapılar hücre çekirdeğinde kromozom adı verilen ikili sarmal yapılar şeklinde bulunurlar. Kromozom ipliklerindeki nükleotid dizileri canlıların genetik kimliğini belirler. DNA, hücre ve organizmanın hayatta kalması, çoğalması için gerekli tüm fonksiyonel moleküllerin sentezlenmesinde sorumludur. DNA üzerindeki nükleotidlerin dizilimi yaşam türleri ile ilgili bizlere birçok bilgi vermektedir (N. L. Craig, 2010). Şekil 1'de DNA'nın kimyasal yapısı ve temsili görseli paylaşılmıştır.

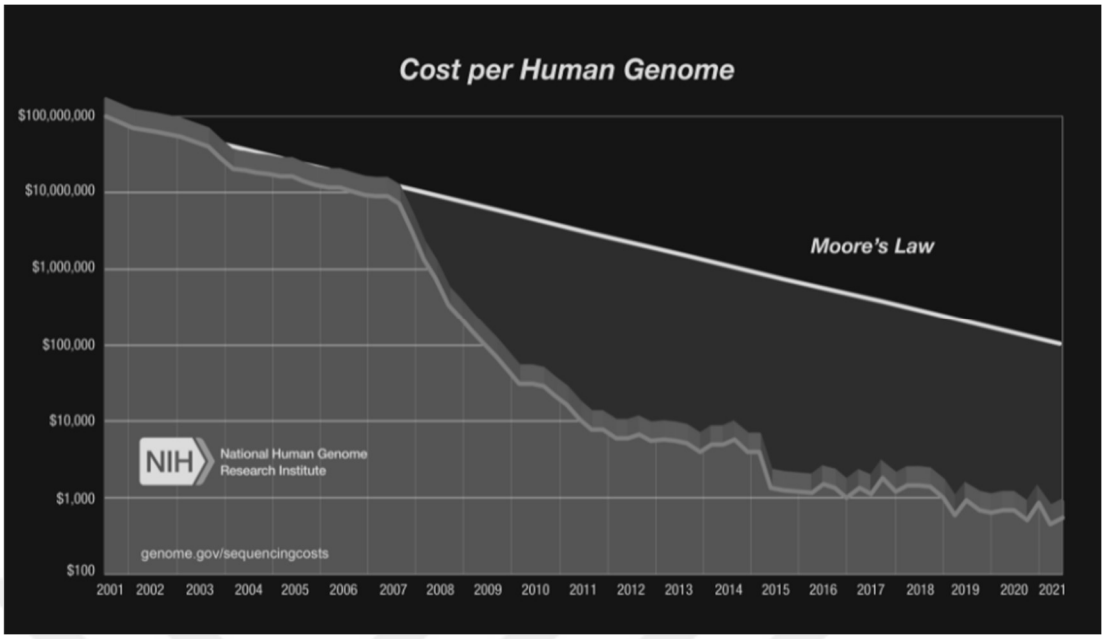


Şekil 1. DNA' nın Kimyasal Yapısı (url16).

Gen, geçmişte soyut anlamda anne babadan çocuğa özellikleri taşıyan kalıtımsal yapılar olarak tanımlanırdı. Bu yapılar ise hücre içerisindeki protein ve enzimler olarak düşünülürdü. Moleküler biyolojinin gelişmesi ile proteinleri oluşturan sistemin, DNA'nın bir ipliğinden sentezlenen haberci RNA (mRNA) sayesinde oluştuğu

keşfedilerek, gen fiziksel bir kavrama dönüştü. Genel anlamda "gen" için bu tanım kullanılsa da son zamanlarda bu tanımın da yetersiz kaldığı düşünülmektedir. Çünkü çalışmalar ilerledikçe kromozomlar üzerindeki bilginin düşünüldüğünden daha kompleks olduğu ve RNA'nın sadece genetik bilgiyi taşıyan pasif unsurlar olmadığı, aynı zamanda hücre fonksiyonlarında düzenleyici rol oynadığı görüldü. Hatta genetik bilginin RNA tarafından aktarıldığı durumlarda tespit edildi (Pearson, 2006).

İnsan genomu projesi (HGP) ile insan DNA'sı üzerindeki genler tanımlanmış ve yaklaşık 3 milyar baz çifti içerisinden 30.000 civarında gen tespit edilmiştir. Ayrıca genlerin %99,9'unun tüm insanlarda aynı olduğu görülmüştür. Bununla beraber yaklaşık 1,5 milyon civarında tek nükleotid değişikliği tespit edilmiştir (V. J. Craig ve diğerleri, 2001). Genlerin %50'sinden fazlasının işlevi bilinmemektedir ve genlerin önemli bir kısmı sinir sisteminin gelişiminden sorumludur. İnsanlar arasında %0.1 gen farklılıklarının insanlar arasındaki farklılaşmayı, hastalıkların nedenlerini, farklı kişilerin hastalıklara verdikleri tepkileri, insan davranışlarındaki farklılıkları, madde bağımlılığına yatkınlığı ve ilaçların etkisinin kişiden kişiye nasıl değiştiğini açıklayabileceği düşünülmektedir (Başaran ve Aras, 2010). Bazı hastalıkların aile ve yakın akrabalar arasında daha yaygın olduğu bilinmekteydi. HGP tıbbi nedenlerle kalıtsal olan bu tür hastalıkların gizemini çözmek için başlatıldı (Collins ve Mansoura, 2001). Günümüzde genetik veri dizileme teknolojisinin gelişimi ile bir genomun çıkarılması 1000$ dan daha ucuz seviyelere indi. Şekil 2 de yıllara göre DNA dizisi oluşturma maliyetleri verilmektedir. 2001 -2007 birinci nesil yöntemlerin, 2008 sonrası ise ikinci nesil yöntemlerin maliyetleri grafikte gösterilmiştir. Genom dizileme maliyetlerinin azalması sayesinde, genetik analizler, tıbbi amaçların yanında ticari amaçlar için de kullanılmaya başlanmıştır. Özellikle kişiye özel ilaç geliştirilmesi, anne karnındaki embriyonun genetik bozuklukları, adli alanda şüpheli tespiti ve tanınamayacak durumda olan bedenlerin kimlik tespiti gibi konularda aktif olarak kullanılmaktadır (Kavak Konrat, 2020).



Şekil 2. Genom verisi başına maliyet (url2).

1.1.2 Genetik Mahremiyet

Mahremiyet teriminin tanımı yıllar içerisinde değişmekte ve hala kapsayıcı bir tanımının olmadığı düşünülmektedir. Mahremiyet ilk olarak Warren ve Brande tarafından geçen yüzyılda "yalnız bırakılma hakkı" olarak tanımlanmıştır. Sonrasında Pound ve Freund tarafından kişiliğin uzantısı, Westin ve birçokları tarafından bilgi kontrolünün bozulması gibi tanımları yapılmıştır. Yasal olarak mahremiyet davaları izinlerin ihlali, özel şeylerin açığa çıkarılması, kişilik ihlali yaparak yanıltma ve izinsiz fayda sağlama başlıkları altında işlenmektedir (Moore ve Moore, 2017). Genetik mahremiyet kavramı ise HGP ile beraber ortaya çıkmaya başlamıştır. Bu proje sayesinde insandan alınan doku, kan, tükürük, kıl gibi biyolojik örneklemler ile kalıtsal özelliklerimiz üzerinde önemli bilgiler elde edilmiştir. Kalıtsal özelliklerimizin sırlarını çözmek ve toplumsal özelliklerin gen ile ilişkisini ortaya koymak için bu alandaki çalışmalar gittikçe hız kazanmakta, bu anlamda birçok proje yürütülmektedir. Yürütülen bu projeler bilimsel anlamda yeni keşiflere ve bilimin ilerlemesine yol açsa da genetik mahremiyet ihtiyacını ortaya çıkarmıştır. Genetik mahremiyet genel mahremiyet tanımlarının içerisinde oldukça sınırlı ve spesifik bir alanı göstermektedir (Hayırlıoğlu, 2019). HGP çalışmalarının başlarında çalışmanın etik alt yapısı hazırlanmamıştı. Fakat sonrasında, beş yıllık bir çalışmanın ardından uluslararası bir

mutabakata varılarak beş prensip üzerinde anlaşıldı(Knoppers ve Chadwick, 1994). Bunlar:

a) *Özerklik*: Genetik testler ve genetik bilgi son derece kişiseldir. Bu tür bilgiler sosyal, ekonomik ayrımcılığa sebep olabilir. Örneğin birini işe alırken, göçmen seçiminde ya da sigorta uygulamalarında genetik seçimler yapılabilir.

b) *Mahremiyet*: Kişilerin genetik bilgilerinin gizliliğine ve mahremiyetine saygı göstermek çok önemlidir. Çünkü bir genetik test ile sadece ilgili kişinin değil aynı zamanda yakın akrabalarının da sağlık riskleri ortaya çıkabilmektedir. Bu kayıtların sigorta firmalarının ya da şirketlerin eline geçmesi ayrımcılığa sebep olabilir.

c) *Adalet*: Kendi başlarına karar alamayan, bakıma muhtaç kişiler ve çocuklar için genetik gelişmeler endişe vericidir. Onların da mümkün olduğunda karar verme süreçlerine dahil edilmesi gerekir. Gelişen teknoloji ile genlerde değişiklik yapılması ve öjeni çoğu yasaya göre yasaklanması yönündeyse de, bu konuda önlemler alınarak çalışmaların yapılabileceği ile ilgili fikirlerde temkinli olunmuştur.

d) *Eşitlik*: Açık olarak ortaya konmasa da eşitliğin nasıl sağlanacağı tartışma konusudur. Bu konuda sosyal eşitsizlikler ortaya çıkabilmektedir.

e) *Kalite*: Bu prensipte de ortak bir görüş olmasa da bu alanda çalışan laboratuvarlar ve personeller etik anlamda bilinçli ve hassas olmalıdır.

1.1.3 Mahremiyet İhlalleri

Etik anlamda genetik bilginin diğer sağlık verilerinden farklı olup olmadığı tartışma konusudur (Shoenbill ve diğerleri, 2014). Genetik İstisnacılara (Genetic Exceptionalism) göre bu konudaki tüm yasalar genetik ayrımcılığı engellemeye yöneliktir. Genetik analizlerin yakın akrabalarla ilgili bilgileri açığa çıkarması, sonraki nesillerle ilgili tahminler yapılmasına imkân tanıması, öngörücü olması, öjeni, ırkçılık,

soykırıma neden olma ihtimali, benzersiz olması gibi sebeplerden dolayı hukuksal olarak ayrı ele alınması gerektiği savunulmaktadırlar. (Rothstein, 2007).

Genom verisi gibi biyometrik verilerin dijital ortamlarda biriktirilmesinde ve kullanılmasında, kişisel ve kamusal çıkarlar bağlamında, etik sorunlar çıkabilir. Bireyler genetik verilerinin izinsiz kullanımından dolayı olumsuz etkilenebilirler (Karabekmez, 2021). Özellikle de bireysel genetik verileri, kişilerin paylaşmak istemeyecekleri sağlık sorunlarını açığa çıkarabilir. Bu bilgi kişilerin aleyhine işlerini kaybetmelerine ya da bir iş konusunda tercih edilmemelerine, sigorta firmalarının bu bilgiler ışığında kapsamlarını belirlemelerine ya da başka ekonomik etkilere sebep olabilir. Ayrıca genetik bilgi, çoğu zaman başka veri kümeleri ile birleştirilerek kullanılır. Örneğin moleküler ya da klinik araştırmalarda genetik bilgi demografik, sosyal, çevresel veya fenotip özellikler ile beraber kullanılır. Bu kritik bilgiler, iyi korunmayan bir bilgisayar ya da sunucu üzerinden, kötü niyetli kişiler tarafından çalınarak açığa çıkarılabilir. Bundan dolayı genetik bilginin güvenliği oldukça önemlidir. Kötü amaçlı saldırılar genel anlamda yeniden tanımlama, üyelik çıkarımı, aile ve yakın akraba tespiti başlıkları altında özetlenebilir (Z. Wan ve diğerleri, 2022).

a) *Yeniden Tanımlama*: Bir kişiye ait genetik bilgiyi kullanarak kişinin kimliğini tespit etme yoludur. Açık tanımlayıcılar olmadan genomik veriler paylaşılsa bile yeniden tanımlama mümkün olabilir. Genel yaklaşım tanımlayıcı verileri şifreleme, anonimleştirme ya da kriptolama yöntemleri ile tutmak yönündedir. Fakat gelişmiş algoritmalar ile hasta ziyaretlerinden yola çıkarak kimlik tespit etmek mümkündür (Malin ve Sweeney, 2004). Farklı bir çalışmada kişisel bilgi içermeyen, halka açık bir veri tabanından alınan örneklemden Y kromozomu verisi ile başka veri tabanlarında arama yapılarak ilgili kişinin soyadı bulunmuştur (Gymrek ve diğerleri, 2013).
b) *Sınıf Tespiti:* Bu gruptaki ihlaller, gelişen istatistik ve makine öğrenmesi yöntemleri ile genetik bilgi havuzlarındaki örneklemlerin gruplandırılması olarak açıklanabilir. Bu yöntemle bir kişinin hastalık yatkınlıkları tespit edilebilir (Z. Wan ve diğerleri, 2022).

c) *Aile/Akraba İfşası*: Genetik bilgi yakın akrabalar arasında daha az farklılık gösterdiği için, bir kişiden yola çıkarak diğer yakın akrabaların kimliklerinin ve hastalık yatkınlıklarının tespit edilmesi yöntemidir (Humbert ve diğerleri , 2013). Örneğin 2018 yılında FBI bir seri katilin genetik bilgisinden yola çıkarak suçlunun kuzenine ulaşmış, sonrasında aile gözlem altına alınarak suçlu yakalanmıştır (Callaway, 2018).

Bu tür saldırılar yasal ve teknolojik yöntemlerle engellenmeye çalışılmaktadır. Aşağıdaki bölümde dünya ve Türkiye genelindeki veri koruma kanunları incelenecek ve bir sonraki bölümde de teknolojik olarak genetik veri güvenliğinin nasıl ele alındığı incelenecektir.

1.1.4 Yasal Düzenlemeler

Yasal düzenlemeler sayesinde kişisel verilerin kişilik hakları korunarak, araştırmalarda hangi şartlarda kullanılabileceğinin sınırları çizilmiştir. Bu bağlamda Avrupa da 'Genel Verileri Korunma Tüzüğü (GDPR)', Amerika'da 'Sağlık Sigortası Taşınabilirlik ve Sorumluluk Yasası (HIPAA)'(Z. Wan ve diğerleri, 2022) ve Türkiye' de 'Kişisel Verilerin Korunması Kanunu (KVKK)' düzenlemeleri genel anlamda kişisel verilerin kullanım şartlarını belirler (url3). Bu düzenlemeler genel olarak veri sahiplerinin kendi verilerine erişiminin ve verilerini kullanım hakkının yanında, kendi rızalarının dışındaki erişimlerin ihbar edilmesine yönelik kaideleri belirler.

GDPR' a göre kişisel veri kimliği belirli ya da belirlenebilir gerçek kişilerle ilişkili her türlü bilgidir ve bu yönetmeliğin kapsamı kişisel verilerdir. Genetik, biyometrik, sağlık kayıtlarının yanında etnik, dini, politik ve ideolojik eğilimler kişisel verilerden sayılmıştır. Bu yönetmeliğe göre kişiyi belirlemek amaçlı genetik bilginin kullanılması yasaklanmıştır. Bu yasak, açık rızanın alınması, açık rıza alınamayacak kişilerin hayati menfaatlerinin korunması, kişinin kendi verisini alenileştirmesi, yasaların uygulanabilmesi, mahkeme kararları aracılığı olması, güvenlik önlemlerinin alınması koşulu ve kamu sağlığını koruma amaçlı gerekliliği gibi durumlarda istisnai olarak kullanılabilmektedir.

Amerika Birleşik Devletleri, moleküler genetik veri işleme yollarına sıkça başvuran ülkelerdendir. Özellikle suçlar arasındaki bağlantıların ortaya çıkarılması, kimlik tespiti ya da kayıp kişinin bulunması amacıyla FBI dünyanın en büyük genetik veri tabanını kurmuştur. Bu veri tabanı gönüllüler, suçlu ve şüphelilerden alınan genetik bilgiler ile oluşturulmuştur. Fakat bu veri tabanından veri silinmesi istendiğinde, kişinin yazılı başvurusu sonucu veri silinebilmektedir. Bu şekilde yazılı başvuru yapanların sayısı çok az olduğu için bu veriler çoğalmakta ve Amerikan Sivil Özgürlükler Birliği (ACLU) tarafından bu teknolojinin kullanılması gerekli görülmekle beraber, genel kanı bu veri tabanının büyümesinin gizlilik ihlali yaratabileceği yönündedir (Atalay, 2020). Hasta bakımı ve biyomedikal araştırmaları yöneten en önemli düzenlemeler sağlık hizmeti sağlayıcıları, sağlık planları ve sağlık hizmet odaları ile kısıtlanan HIPAA yasasıdır. HIPAA genellikle bu kuruluşların, korunan sağlık bilgilerinin tedavi, ödeme ve sağlık hizmetleri operasyonları dışında kullanımları ve ifşaları için hasta izni almasını gerektirir ve erişim haklarını bireylere devreder. Fakat HIPAA da bazı istisnalar barındırmaktadır (url5). Özellikle isim, sosyal güvenlik numarası, adres gibi kimlik belirten bilgilerin silinmiş olması ya da kimlik tespiti riskinin düşük olduğu istatistiksel uygulamaların kullanımında, genetik bilginin de içinde bulunduğu sağlık kayıtlarının işlenmesinde özel izin gerektirmemektedir (url6).

Türkiye' de ise KVKK kişisel verilerin işlenmesi ile ilgili düzenlemeleri içerir. Bu kanuna göre genetik bilgiler özel nitelikli kişisel veriler kapsamında değerlendirilmektedir. Özel nitelikli kişisel verilerin, ilgilinin açık rızası olmaksızın işlenmesi yasaklanmıştır. Ancak KVKK 'da genetik verinin rıza dışı işlenmesinde bazı istisnalar bırakılmıştır. Sağlık alanında kullanılan verilerin, kamu sağlığının korunması, koruyucu hekimlik, tıbbî teşhis, tedavi ve bakım hizmetlerinin yürütülmesi, sağlık hizmetleri ile finansmanının planlanması ve yönetimi amacıyla, sır saklama yükümlülüğü altında bulunan kişiler veya yetkili kurum ve kuruluşlar tarafından, ilgilinin açık rızası aranmaksızın işlenmesi uygun görülmüştür (url3). İstisnaların belirtildiği KVKK kanunu Madde 6/3 ikinci fıkra bölümünde "Birinci fıkrada sayılan sağlık ve cinsel hayat dışındaki kişisel veriler, kanunlarda öngörülen hâllerde ilgili kişinin açık rızası aranmaksızın işlenebilir" ifadesi karmaşa yaratmaktadır. Birinci fıkrada ayrı olarak ifade edilen genetik veri üçüncü fıkrada ayrı

olarak belirtilmediği için, rıza aranmaksızın genetik bilginin kullanılabileceği yönünde bir anlam çıkarılabileceği düşünülmektedir (Hayırlıoğlu, 2019).

1.2 GENETİK MAHREMIYETİN TEKNİK YOLLARLA KORUNMA YÖNTEMLERİ

Genetik verinin mahrem olduğundan ve kullanımı için birçok ülkenin yasal düzenlemeler yaptığından önceki bölümlerde bahsedilmişti. Fakat her ne kadar yasal anlamda bazı tedbirler alınmış olsa da, kötü niyetli saldırılır konusunda, verilerin güvenliğini sağlayacak teknolojik koruma yöntemlerine ihtiyaç vardır. Kötü niyetli saldırıların dışında KVKK kanununda ki istisnai durumlar ve bu istisnai durumlardan faydalanarak, kişisel genetik verilerimizin rızamız dışında kullanılması riski için de teknik önlemler koruyucu olacaktır. Bu tür durumların önüne ancak teknolojik önlemler ile geçilebilir. Bu bölümde, teknik anlamda genetik veri mahremiyetinin nasıl korunabileceği incelenecektir.

Teknik olarak hassas içerikli verilerin korunması genel anlamda beş tür içinde incelenebilir. Bunlar pertürbasyon, otantikasyon, difarensiyel gizlilik, kriptografik ve blokzincir temelli çözümlerdir (Keshk ve diğerleri, 2020). Bu yöntemler aşağıda kısaca açıklanmıştır.

1.2.1 Pertürbasyon Temelli Koruma

Bu yöntem kişisel kayıtların gizliliğini koruyarak veriye gürültü eklenmesidir. Bu sayede veri kullanıcıları, güvenlik ihlali oluşturmadan, ilgili veri kümesi içerisinde ihtiyaç duydukları özet bilgiyi alabilirler. Veri tabanındaki önemli bilgiler arasındaki ilişkiler korunurken, bireysel gizliliği korumak için hassas veriler maskelenir (Wilson ve Rosen, 2003). Bu yöntem ile orijinal veri işlenerek istatistiksel değerler gibi farklı bir veriye dönüştürülebilir. Fakat bu yöntemde kullanışlılık ve gizlilik arasında ters bir ilişki vardır. Veri ne kadar değiştirilirse kullanışlılığı o kadar azalır (Keshk ve diğerleri, 2020). Ayrıca ham veriyi işleyenler ister istemez hassas içerikli verilere erişmiş olurlar.

1.2.2 Otantikasyon Temelli Koruma

Veri güvenliğini koruma yöntemlerinden birisi de kimlik doğrulama ve yetkilendirmedir. Kimlik doğrulama, modern bilgisayar teknolojileri sistemlerinin temel bir unsurudur. Bir servise kullanıcı adı ve şifreyle erişilmesi tek kullanımlık şifre, akıllı kartlar, sertifika doğrulaması vb. gibi birçok yöntemi içeren, karmaşık yöntemlerden oluşur. Yetkilendirme ise kişiler sisteme giriş yaptıktan sonra ne tür verilere erişebileceği ile ilgilidir (Vielhauer, 2006). Örneğin Avrupa'da Biyoinformatik alanından veri hacmini kontrol altında tutarak, iş birliğini arttırmak için, ELIXIR denilen bir servis kullanılmaktadır. Bu servis kimlik doğrulama yöntemi kullanarak, üye devletler arasında biyoinformatik kaynakları entegre ve koordine ederek, bu alanda hizmet veren araştırmacıların kolay erişimine olanak tanır (Drysdale ve diğerleri, 2020). Kimlik doğrulama yönteminin çok yaygın olması, onu kötü niyetli saldırılara açık hale getirmiştir. Fiziksel gözlem, şifre tahmin etme, entegrasyonlardaki açıklar, e-dolandırıcılık, sunucuların ele geçirilmesi gibi bir çok yöntem ile sistem güvenlikleri aşılabilmektedir (Barkadehi, Nilashi, Ibrahim, Zakeri Fardi ve Samad, 2018).

1.2.3 Diferansiyel Gizlilik Yoluyla Koruma

Diferansiyel gizlilik, kişisel verilerin veri kümesinde olup olmadığına bakılmaksızın, kullanılan algoritmanın sonucunun çok değişmeyeceğini garanti eder (Ruogang, 2021). Örneğin bir veri kümesi üzerinde yapılan toplu analizlerde, bir kaydın çıkarılması, popülasyon analizlerinin sonucunu çok etkilemeyeceği gibi düşünülebilir. Dünya Sağlık Örgütü bireyin minör alelleri üzerinden kişilik tespiti yapılabileceğini keşfettikten sonra, halka açık veri tabanlarında uygulanan prosedürü güncelleyerek Diferansiyel Gizlilik yönteminin kullanılmasını önerdi (Mohammed Yakubu ve Chen, 2020). Diferansiyel gizlilik kişisel veriyi korurken, aynı zamanda fayda dengesini de dikkate alır. Veriye eklenen gürültü ile gizliliğin ne kadar korunduğunu matematiksel olarak hesaplayabilir (Yüksek, 2021). Kullanışlı bir yöntem olduğu için bu konuda çok araştırma yapılmış ve çeşitli algoritmalar geliştirilmiştir (Ruogang, 2021).

Denklem 1'de ifade edildiği gibi diferansiyel mahremiyet, bir veri setine gürültü eklendiğinde bir fonksiyonun ne kadar benzer sonuç çıkardığına göre diferansiyel mahremiyetin sağlanıp sağlanmadığını matematiksel olarak açıklar. D1 ve D2, en fazla bir satırda farklılık gösteren iki veri tabanı olsun ve S, veri tabanlarındaki olası bir

sorgu çıktıları kümesi olsun. Sonra bir rastgele algoritma A seçelim, aşağıdakiler geçerliyse, D1 ve D2 üzerinde ε-diferansiyel değeri özeldir. Bu formüle göre D1 ve D2 benzer sonuçlar üretmelidir. ε'nun küçük olması, veriye fazla gürültü eklendiği ve sonuçları olumsuz etkileyebileceği anlamına gelir (Mohammed Yakubu ve Chen, 2020).

Denklem 1. Diferansiyel mahremiyet denklemi

$$\Pr[A(D1) \in S] \leq e^{\varepsilon} \times \Pr\,[A(D2) \in S]$$

Bu mekanizma birçok alanda kullanıldığı için farklı mekanizmalar kurulmuştur. İlk olarak Laplace ve sonrasında üstel, medyan, çarpımsal ağrılıklar, geometrik, merdiven, gauss ..vs. gibi bir çok mekanizma geliştirilmiştir (Yüksek, 2021).

1.2.4 Kriptografik Yöntemler İle Koruma

Kriptografik yöntemler genetik veri gizliliğini korumak için kullanılan önemli tekniklerden biridir. Bu yöntemler homomorfik şifreleme, güvenli çok taraflı hesaplama ve güvenli çalıştırma ortamları başlıkları altında toplanabilir.

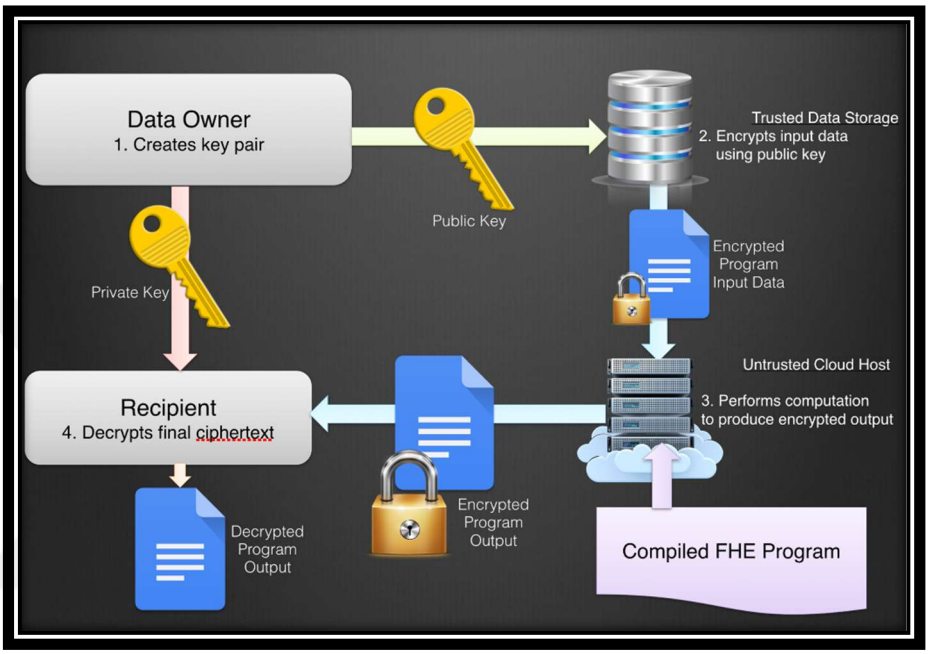
Homomorfik Şifreleme (HE): Denklem 2 de ifade edildiği gibi HE yöntemi, şifrelenmiş veri kümesi üzerinde şifrelenmemiş hali ile aynı sonucu üreten işlemler yapabilmeye olanak tanır. Matematiksel olarak aşağıdaki şekilde ifade edilir. E verinin şifrelenmesini, D ise verinin açık hale getirilmesini ifade eder. Eşitlikte verilen o_e ise toplama, çarpma gibi bir operatörü ifade eder. Aşağıdaki eşitliğin sağlanması durumu M veri kümesinde o_e operatörünün homomorfik özellik taşıdığı şeklinde ifade edilir (Özdemir, 2008).

Denklem 2. Homomorfik şifreleme matematiksel ifadesi

$$\forall m1, m2 \in M\; D\big(E(m1)\; o_e\; E(m2)\big) = m1\; o_p\; m2$$

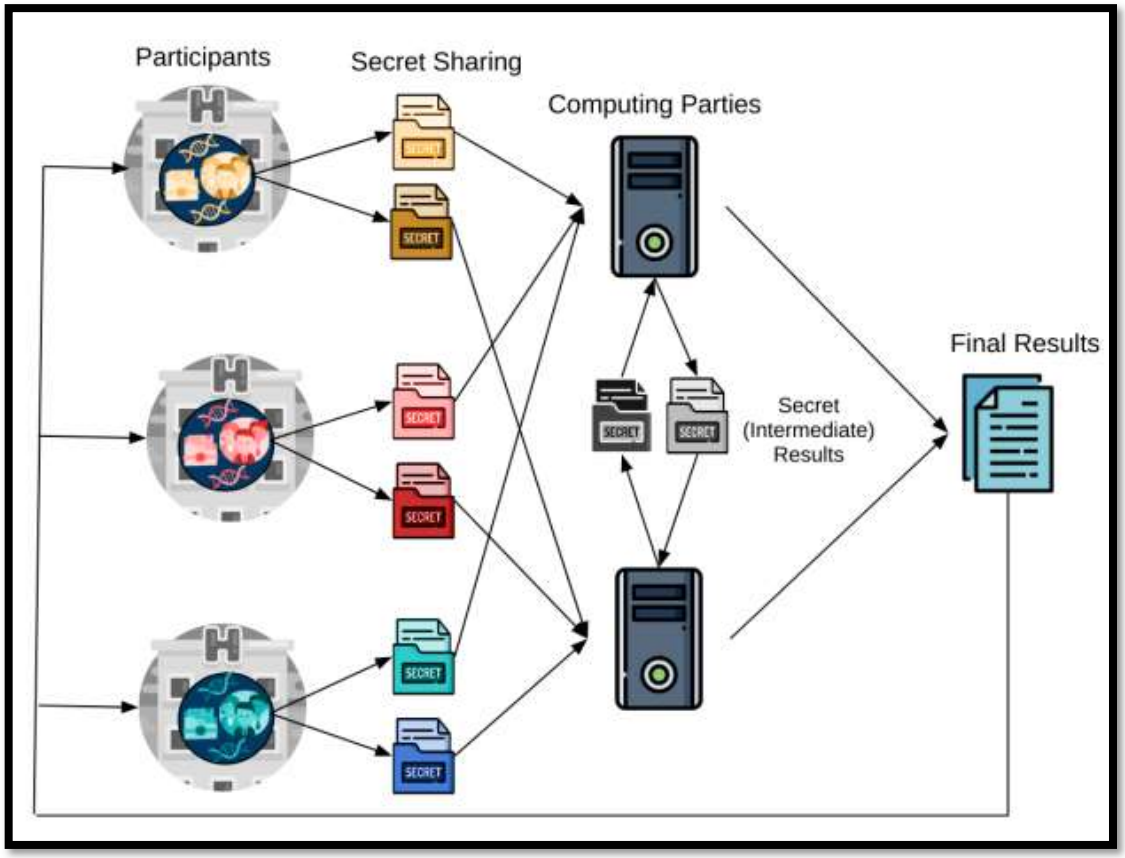
HE üç kategoride sınıflandırılabilir. Bunlar sadece seçilen bir matematiksel işleme izin veren kısmi HE, belirli sayıda işleme izin veren hem de hesaplama zamanlarının sayısıyla sınırlı olan sınırlı HE ve son olarak sınırsız sayıda hesaplamayı destekleyen

tam HE yöntemleridir (Tang ve diğerleri, 2016). Bu çalışmada HE detaylarına girilmeyecektir. Genel anlamda iş akışı Şekil 3'de resmedilmiştir. Burada gösterildiği gibi veri sahibi veriyi bir anahtar ile şifreler ve bulut sistemleri gibi güvenli olmayan bir sisteme aktarır. Güvenli olmayan bu sistemde, şifreli veri üzerinde işlemler yapılarak, şifreli şekilde alıcıya teslim edilir. Alıcı, özel anahtarı ile çıktıyı açık hale getirerek sonucu görebilir.



Şekil 3. Homomorfik şifreleme yöntemi (Archer, D., 2017).

Çok taraflı güvenli hesaplama (Multi-party Secure Computition): Çok taraflı güvenli hesaplama (ÇTGH), tarafların ortak bir F() işlemi üzerinde anlaşarak, taraflar arasında dağıtık depolanan girdilerin gizliliğinin korunması yoluyla ortak olarak hesaplamaların yapılmasıdır (Bogetoft, Christensen ve Damg, 2008). Sisteme katılan taraflar P → P1…Pn ve her bir taraf Pi ile ifade edilir. Bu taraflar n tane girdisi olan bir F fonksiyonu üzerinde anlaşırlar. Bu iş birliğinin amacı $y = f(x1, \dots, xn)$ işlemini doğruluk ve güvenliği garanti ederek çözmektir. ÇTGH Şekil 4 de resmedildiği gibi üç adım da uygulanır. İlk olarak, her bir taraf ayrı ayrı gizli girdileri hesaplama sunucularına gönderir. İkinci adımda hesaplama sunucuları ara sonuçları üretir. Bu sonuçları birbirleri arasında güvenli bir şekilde paylaşırlar. Son olarak ara sonuçlar toplanarak sonuç çıktısı elde edilir (Torkzadehmahani ve diğerleri, 2021).



Şekil 4. Çok taraflı güvenli hesaplama (Torkzadehmahani ve diğerleri, 2020).

Güvenli Hesaplama Ortamı: Bu yöntemde, özel donanımlar ve yazılımlar sayesinde şifrelenmiş bir veri kümesinin açık hale getirilmeden işlenmesine olanak tanıyan sistemler kullanılır. Intel firmasının geliştirdiği yazılım koruma uzantıları (SGX) gibi ortamlarda merkezi işlem birimi (CPU) desteği ile hesaplama süreci şifreli bir yerleşim bölgesinde yalıtılır. Böylece kötü niyetli işletim sistemi yazılımları veri içeriğini göremez (Canim, Kantarcioglu ve Malin, 2012).

1.2.5 Blokzincir Temelli Koruma

Blokzincir teknolojisi kriptografik temelli bir sistem olsa da onu bir araç olarak kullanır. Bundan dolayı ayrı başlıkta ele alınmıştır.

Blokzincir sistemleri, güvenliği garanti ederek, genetik veri paylaşımını teşvik eden sistemlerdir. Blokzincir teknolojisi finansal işlemler için tasarlansa da dağıtık, merkeziyetsizlik, değiştirilemezlik, kriptografik güvenlik ve taraflar arasındaki

senkronizasyonun otomatik sağlanması özellikleri farklı alanlarda da kullanımını mümkün kılmıştır. Blokzincir yapılarının temel yaklaşımı, hassas veriye erişim için erişim kontrolü mekanizmasını ve veri bütünlüğünün korunmasını sağlamaktır. Geleneksel şifreleme yöntemlerinde, şifrelenmiş veriyi çözerek hesaplama yapılabilmektir. Homomorfik şifreleme, şifrelemeyi çözmeden veri işlemeye olanak tanısa da gerçek hayatta uygulamak çok pratik değildir. Çünkü bu yöntemde sınırlı türde işlem yapılmasına izin verilir (Ozercan ve diğerleri, 2018).

Özellikle blokzincir teknolojisi, hassas veri erişimleri için izin yönetiminin yapılması ve ilgili erişimlerin değiştirilemez şekilde kayıt altında tutulabilmesi için imkanlar sunar. Şifrelenmiş bir verinin genel anahtar bilgisi, blokzincir ağı üzerinden, veri sahibinin izni ile veriye erişmek isteyenlere iletilebilir (Grishin, Obbad ve Church, 2019). Bununla beraber daha küçük veri kümeleri direk blokzincir üzerinden paylaşılabilir. Örneğin genome çapında ilişki analizi (GWAS) çalışmalarında, veri paylaşmadan makine öğrenmesi parametreleri paylaşılarak iş birliği sağlanabilir (Y. Zhang ve diğerleri, 2019).

1.3 GENETİK ALANDA BLOKZİNCİR TEKNOLOJİSİNİN KULLANIMI

Bölüm 1.1 de anlatıldığı gibi genetik verilerin gizemi tamamen çözülmese de, genetik veri analizleri hastalıkların sebeplerinin tespiti, kişiselleştirilmiş tedavi yöntemleri ya da hastalık mutasyonlarının belirlenmesi gibi birçok konuda katkıda bulundu. Şekil 2 den de hatırlanacağı gibi 2007 sonrasında dizileme maliyetleri oldukça azalmıştır ve günümüzde bir Illumina sistemi ile 18.000 dizileme işlemi, genom başına 1000 dolardan daha az maliyetle yapılabilmektedir (Goldfeder ve diğerleri, 2017). Bir insan genomu yaklaşık 200Gb yer kaplamaktadır. Tahminlere göre 2025'e kadar 2 milyar insanın genom dizilerini çıkartması bekleniyor, ki bu durumda oluşacak genetik veri depolama ihtiyacının video içeriklerinin bile önüne geçeceği tahmin ediliyor (Stephens ve diğerleri, 2015). Yüksek hacimli verilerin depolanması, bulut sistemlerinin genişleyebilir ve ölçeklenebilir özellikleri sayesinde başa çıkılabilir olsa da, veri sahibi ve kullanıcı arasında bir aracı kaçınılmaz olmaktadır. Bununla beraber bulut

sistemleri, verileri veri merkezlerinde tutar. Bundan dolayı veri güvenliği, saldırılara ve hatalara açık olmaktadır (Ozercan ve diğerleri, 2018). Genetik veri alanındaki diğer bir zorluk ise genetik verinin yönetimidir. Verinin toplanması, paylaşılması, sahipliği, ve depolama maliyetleri çözülmesi gereken zorluklardandır (Stephens ve diğerleri, 2015).

Ozercan ve diğerleri (2018) genomik alanda blokzincir teknolojisinin kullanımını üç ana başlık altında toplamıştır. Bunlar (i) dağıtık hesaplama, (ii) veri depolama ve dağıtık oylama, (iii) merkezi olmayan özerk organizasyon (DAO) ile kimlik ve sahiplik kontrolü olarak belirtilmiştir (Ozercan ve diğerleri, 2018). Bu çalışmada ise blokzincirin genetik alanda kullanımı dağıtık hesaplama, veri ve yetki paylaşımı, blokzincir-yapayzekâ uygulamaları olarak incelenecektir.

Tablo 1. Genomik alanda olası blokzincir kullanım alanları (Ozercan ve diğerleri, 2018).

Table 1. Possible use cases for blockchain in genomics

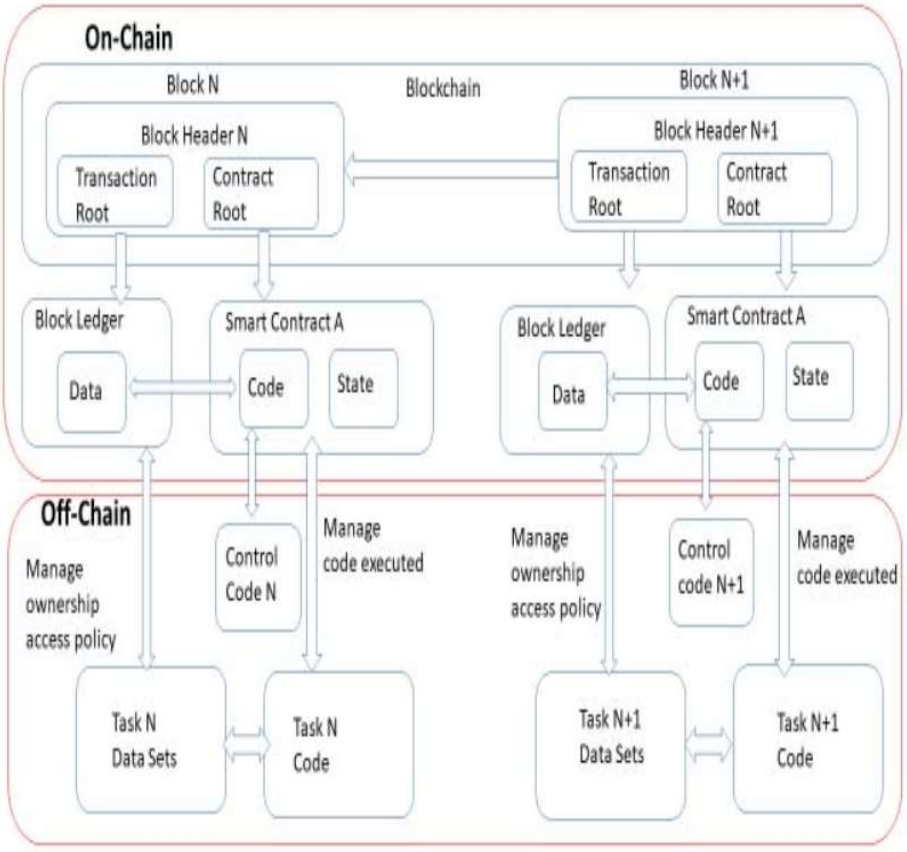
Use case	Examples
Distributed computation	Coinami, Gridcoin, Curecoin, FoldingCoin, Single instruction multiple data parallelization (smart contracts working on different data instances)
Data storage and distribution	Filecoin, CGT, CrypDist, Gene-chain, Nebula Genomics
Voting	Standardization teams (vote on proposals), Crowdsourced solutions (manual curation, gene-drug interactions)
Identity and ownership	Decentralized researcher identification databases, personal data adoption, crediting data ownership
Decentralized Autonomous Organization (DAO)	Predefined rules governing large organizations and projects such as GA4GH, ELIXIR, TCGA, ICGC

1.3.1 Dağıtık hesaplama

Dağıtık hesaplama, blokzincir ağındaki sunucuların veri işleme gücünden faydalanarak, zorlu hesaplamaları hızlı bir şekilde yapabilmeyi amaçlamaktadır. FoldingCoin ve Curecoin gibi projeler bu sınıfa giren örnek projelerdendir. Bu projelerin amacı, blokzincir ağında yüksek enerji kullanarak oluşturulan yeni blok maliyetini faydalı bir amaç için kullanmaktır. Bu projeler, katılımcıların CPU ve GPU kaynaklarını, protein katlanması işlemi sonunda proteinin aldığı fiziksel şekli ortaya çıkarmak için kullanır. Protein şekli kanser ve hücre fonksiyonlarının açıklanmasında önemli bir bilgidir (Ross ve diğerleri, 2018). Bu projeler destekçilerine kripto para kazanma imkânı da sağlar. Dağıtık hesaplama projeleri, geniş bant ağ kullanımı,

sunucu tarafında yüksek hacimli giriş/çıkış işlemleri ve hesaplama teşvik yaklaşımlarında ölçeklenebilirlik gibi sorunları çıkarabilir. Bu tür sorunlar her zaman dağıtılmış hesaplama platformlarında karşılaşılabilir (Ozercan ve diğerleri, 2018).

Dağıtık blokzincir yapısını anlamak için, günümüzde kurumsal firmalarda yaygın olarak kullanılan büyük veri ve bulut teknolojilerini anlamak faydalı olacaktır. Büyük veri sistemleri, birçok sunucudan oluşan bir sunucu havuzunda işlerin sunuculara dağıtılıp işlemlerin koşulduğu sistemlerdir. Ana sunucular hangi verinin hangi sunucu da olduğunu bilir. Bir iş sisteme gönderildiğinde, ana sunucular, verinin konumunu işin koşulacağı sunuculara bildirir ve işlem başlar. Bulut sistemleri ise benzer bir yapıyı sanal sunucularda gerçekleştirir. Fakat burada iki problem vardır. Birincisi veri HDFS ve S3 gibi merkezi ortamlardadır. Diğer sorun ise işi koşan sunucular herhangi bir veri erişim kontrolü olmadan ilgili veriye erişebilir. Bir diğer yöntem ise grid hesaplama yöntemidir. Grid hesaplamada bir iş bir şebeke üzerinde çalıştırılmaya çalışılır. Bu şebekede yer almak isteyen sunucular, sisteme dahil olarak yapılacak işin tamamlanmasına katkı sağlar (Creel ve Goffe, 2008). Bu tarz yapılar blokzincir mimarisinede uyarlanabilir. Akıllı kontratlar üzerinde tanımlanan bir işe destek vermek isteyen katılımcılar, kullanmadıkları kaynaklarını sisteme bildirir ve iş talebinde bulunur. İşini tamamlayan katılımcılar blokzincir ağına çıktıları geri gönderir (Shae ve Tsai, 2018). Shae ve Tsai (2018) makalesinde Şekil 5’ te ki mimari ile blokzincir tabanlı bir model önermiştir. Modele göre sistem blokzincir içi ve dışı olarak iki katmana ayrılır. Blokzincir içindeki her bir akıllı kontrat yapılması gereken işlemleri blokzincir dışındaki sisteme gönderir ve sonuçları toplar. Her akıllı kontrat, blokzincir dışındaki yapıyla ayrı olarak iletişim sağlar ve ayrı ayrı işler çalıştırabilirler.



Şekil 5. Blokzincir dağıtık işleme sistem mimari paradigması (Shae ve Tsai, 2018).

1.3.2 Veri ve Yetki Paylaşımı

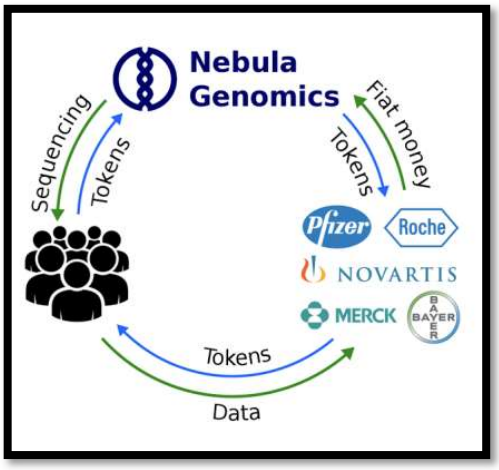
Son yıllarda açık bilgi paylaşımı bilimsel ve endüstriyel alandaki ilerlemenin önemli bir parçası olmuştur. Yazılım alanında kullanılan açık kaynak kodu yaklaşımı ile internet teknolojilerinde vazgeçilmez olan birçok işletim sistemi ve uygulama hayata geçirilmiştir. Örneğin Linux, Python, Firefox, PHP gibi popüler yazılımlar bu yaklaşımla üretilmiştir. Açık kaynak kodlama çalışma mantığı 'Katedral ve Pazar' anolojisi ile açıklanır (Raymond, 1999). Bilim ve endüstri alanındaki önemli gelişmelerin birçoğu katedral modelinde çalışan kapalı bir yetenek grubu ile yapılmaktadır. Katedral modelinde bir otorite gruba dahil olacak kişiyi belirler. Pazar modelinde ise bir gruba dahil olma maliyeti daha düşük ama organizasyonu kaotik ve kendi kendini yöneten bir yapıdadır. Bu sistem etkilidir fakat çalışmaların duyurulması maliyetli olabilir (Woelfle ve diğerleri, 2011). Bilimsel araştırmalarda ise açık kaynak kodu mantığının uygulanması ile açık bilim yaklaşımı ortaya konulmuştur. Açık bilim,

birey aklının toplum aklından daha üstün olamayacağını düstur edinerek, araştırmaların halka açık bir şekilde yapılmasıdır. Yapılan araştırmalarda açık bilim metodu ile akademik araştırmaların ilerlemesinin hızlandığı görülmüştür. Bununla beraber toplumdan alınan vergiler ile yapılan çalışmalar şeffaf hale gelmiş, kişiye olan bağımlılıklar azalmış, araştırmanın fonlanması kolaylaşmış, çalışma çok daha fazla kişi tarafından kontrol edilmiş, yayınlanması ve atıf yapılmada kolaylık olması gibi birçok faydası olduğu görülmüştür (Woelfle ve diğerleri, 2011).

Veri paylaşımı her zaman bilimsel gelişimin mihenk taşı olmuştur. Açık bilim yaklaşımı veriye erişmek, analiz etmek ve farklı kaynaklardan öğrenmek, anlamlı sonuçlar çıkarmak için kaçınılmazdır (Ozercan ve diğerleri, 2018). Fakat birinci bölümde bahsedildiği gibi genetik verinin mahremiyeti ve genetik verinin paylaşımı ile ilgili yasal düzenlemeler, veriyi özgürce kullanabilmenin önünde bürokratik engeller çıkarmaktadır. Genetik alanında kamuya açık veri tabanları, bilimsel araştırmalarda en sık kullanılan veri kümeleridir. Bu amaçla hazırlanmış, konuya özel birçok veri tabanı vardır. Örneğin dbSNP, GenBank, GEO, EBI, HSMD, Oncomine vs. gibi örnekler en yaygın bilinenlerdendir. Bu veri tabanları, verileri belirli standartlarda tutarak, farklı çalışmalar için tekrar kullanılabilir veri kümeleri sağlamaktadır (Rung ve Brazma, 2013). Başka bir örnek olarak federe veri paylaşımını model alan ELIXIR platformu gösterilebilir. ELIXIR, platforma üye olan Avrupa ülkelerinin biyoinformatik kaynaklarını bir araya getiren veri tabanı, yazılım araçları, eğitim materyalleri, depolama araçları ve süper bilgisayarları içeren bir alt yapı sunar (Harrow ve diğerleri, 2021). Bu sisteme erişim ise kimlik ve erişim doğrulama metodu ile yapılmaktadır (Linden ve diğerleri, 2018).

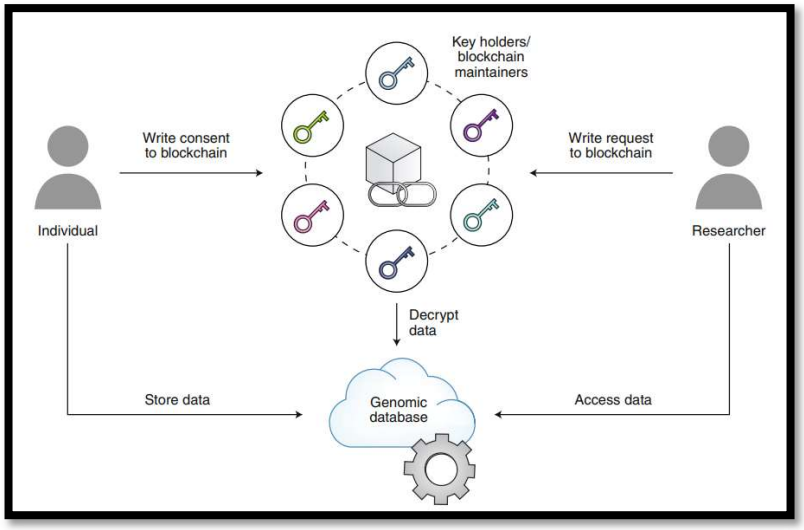
Bahsettiğimiz bu veri tabanları ve platformlar bilimsel araştırmalarda iş birliğini mümkün kılar ancak kişisel verilerin kontrolü merkezi bir otoritededir. Blokzincir bu tür platformların güvenilir merkeziyetsiz bir rıza mekanizması sağlaması konusunda fırsatlar sunabilir. Bu yöntemle oluşturulmuş platformlarda, veri erişim kontrollerinin veri sahiplerine verilmesi, veri sahiplerine bir kazanç imkanı sağlanması, token bazlı ödüllendirme sistemleri, daha fazla verinin toplanması ve bilimsel kuruluşlar arasında iş birliğinin artması mümkün olmaktadır (Adanur Dedeturk ve diğerleri, 2021). Aşağıda genetik veri paylaşımında blokzincir teknolojisi kullanan bazı örneklere yere verilmiştir.

Nebula Genomics: Nebula Genomics, genetik veri analizinin tüm ihtiyaçlarını karşılayabilecek blokzincir temelli bir platform sunar. Amerika' da genomik alanda faaliyet gösteren, şeffaflıktan yoksun bazı kuruluşların ücret karşılığı verdikleri genetik dizilimeyi ve test hizmetleri ile elde ettikleri verileri, veri sahiplerinin bilgisi olmadan, ilaç firmalarına sattığı görülmüştür. Ayrıca hukuksal konularda suçlu ya da suçsuz bir bireyin genetik bilgilerine hükümet dilediğinde erişebildiği için kararlarda ayrımcılık riskini ortaya çıkarmaktadır. Nebula Genomics müşterilerinin gizliliklerinden ödün vermeden, biyomedikal araştırmalardan faydalanmaları için, genomik verilerin şifrelenerek veri tabanlarına yazılmasını sağlar ve erişim izninin blokzincir platformu üzerinden bireye verildiği bir alt yapı sunar (url7). Nebula tokenleri, Nebula ağının para birimidir. Şekil 6 da taraflar arasındaki ticari ilişki gösterilmiştir. Bireyler ücret karşılığında Nebula Genomics' ten dizilime hizmeti alır, genetik veri kullanıcıları nebula token ile genetik veriye erişim hakkını satın alır ve Nebula Genomics tokenleri veri kullanıcılarına satar.



Şekil 6. Nebula iş modeli (Grishin ve diğerleri, 2018)

Firma alt yapısını Şekil 7 de önerilen modele göre geliştirmiştir. Bu modele göre genetik veriye erişim, elinde parçalanmış şifreleme anahtarı bulunan birden fazla tarafın kontrolünde gerçekleşir. Ayrıca sistemdeki taraflar erişim taleplerini ve kullanıcı onaylarını blokzincir üzerinde şeffaf bir şekilde kayıt altında tutar (Grishin ve diğerleri, 2019).



Şekil 7. Çoklu veri erişim kontrolü ve kayıtların tutulması (Grishin ve diğerleri, 2019).

Nebula Genomics, genetik veriyi merkezi veri tabanlarında şifreli olarak tutar. Bu veriler üzerindeki analizler '*çok taraflı güvenli hesaplama*' prensibi ile çalışan Intel SGX işlemcilerine sahip sunucular üzerinde yapılır. Böylece şifrelenmiş veri deşifre edilmeden, üzerinde analizler yapılabilmektedir (Adanur Dedeturk ve diğerleri, 2021).

EncrypGen(Gene-Chain): EncrypGen, genetik veri satıcıları ve alıcıları için blokzincir ağı üzerinde bir market alanı sunar. Temel amacı şahıslara genetik verilerini depolama ve paylaşma yetkisi vererek, şahısların kendi verilerini, bu verilere ihtiyaç duyan kurumlarla güvenli bir şekilde, gelir elde ederek paylaşabilmesini sağlamaktır. Bir sebeple genetik test yaptırmak durumunda kalmış kişiler bu ağ üzerinde kendi verilerini kaydedebilir. Sistem ham veriden kişisel verileri arındırarak veriyi depolar. Veri satın alacak taraflar projeleri için uygun profildeki verileri aratabilir ve kimliksizleştirilmiş veriyi satın alabilir (Ahmed ve Shabani, 2019). Nebula Genomics' ten farklı olarak merkeziyetsiz bir çözüm sunar.

Platform özel bir blokzincir ağında çalışmaktadır. Ağ katılımcıları gerekli güvenlik şartlarını sağladığı taktirde lisans alarak, ağa bir düğüm olarak eklenebilir. Veri kaydetmek isteyenler ise ücretsiz olarak sisteme verilerini kaydedebiliyor. Platformun temel özellikleri aşağıdaki gibidir (Shabani, 2019);

a) Veri saklama, arama ve alım-satım yapabilme
b) Bireylerin arama ve satın alma için izin yetkisinin olması
c) Bireyin kendi genetik verilerinin sahibi olması

LunaDNA: LunaDNA, Luna Public Benefit Corporation adındaki bir topluluk tarafından oluşturulmuş bir platformdur. Platform genetik ve sağlık kayıtlarını paylaşan kişilere şirket hisselerini sunar. Firma sağlık alanında başardığı geliştirmelerden elde ettiği karı hisse sahipleri ile paylaşır. Şahıslar paylaştıkları verilerin içeriğine ve büyüklüğüne göre farklı miktarda hisse kazanabilir. Örneğin belirli hedef gen dizilerini paylaşanlar bir birim hisse alırsa, tam genom verisi paylaşanlar 100 birim hisse kazanabilir. Platform blokzincir teknolojisi kullanarak, kişilerin rızasıyla, araştırmacılara toplu veriler sağlar (Ahmed ve Shabani, 2019). Kullanıcılar diledikleri zaman verilerini platform üzerinden indirebilir, görebilir ya da silebilir. Kişisel veriler gizlenerek platformda saklanır ve bu topluluk, verilerin satılmayacağını, sadece araştırma geliştirme amaçlı kullanılacağını taahhüt eder (url8).

Zenome: Zenome projesi, ethereum ağında akıllı kontrat yoluyla veri paylaşımını ve hatta aynı platformda verinin işlenebilmesini de mümkün kılan bir sistemdir. Kişilerin kaydettiği veriler platformda dağıtık bir mimaride tutulur. Biyoinformatik analizler hesaplama düğümlerinde gerçekleştirilir. Veriyi saklayan ve analizleri koşan düğümler, emeklerinin karşılığında 'Zenome DNA token' kazanırlar. Bireysel katılımcılar kendi verileri üzerinde tam yetkiye sahiptir. Fakat verilerinin saklanması ve analiz edilmesi için ücret ödemeleri gerekir. Son kullanıcılar dilediklerinde verilerine erişim ve kullanım hakkını satabilir (Ozercan ve diğerleri, 2018). Zenome projesi aşağıdaki temel prensip üzerine kurulmuştur (Kulemin ve diğerleri, 2017). Bunlar;

a) Bireyler kişisel verilerinin sahibidir.
b) Seçme özgürlüğü: Kişisel verilerin hangi amaçla kullanılacağını katılımcılar karar verebilir.
c) Paylaşma hakkı: Katılımcılar veri kopyalanmasına izin vermeden, verilerini kurumların erişimine izin verebilir.
d) Mahremiyet: Veriler şifreli bir şekilde saklanır.

e) Dağıtık veri depolama: Dağıtık veri depolama erişilebilirlik, düşük hata toleransı ve ölçeklendirilebilirlik özellikleri ile verileri sağlıklı bir şekilde depolar.
f) Dağıtık veri işleme: İşler hesaplama düğümlerinde aynı anda paralel çalışabilir.
g) Ölçeklenebilirlik: Platform genişlemesi rahat ve esnektir.

Genocoin: Genecoin, Ethereum ağı üzerinde oluşturulmuş kriptografik bir para birimidir. Temel amacı bilim adamlarını, yatırımcıları, servis sağlayıcılarını, laboratuvarları, araştırmacıları bir araya getirerek hızlı, güvenli ve ucuz bir yolla biyoekonomiye teşvik etmektir (url9). Kullanıcılar bu platforma yeni üye kazandırdığı takdirde teşvik ödülü kazanmaktadır. Sistem, genetik bilgi verilerini tutar ve bu veriler dağıtık mimaride depolanır. Genetik bilgi, firmanın anlaşmalı olduğu genom dizileme merkezileri tarafından sağlanır. Kişilere gönderilen kitler ile alınan örnekler, anlaşmalı kurumlar vasıtasıyla genom verisine dönüşür. Veriler sisteme şifreli olarak yazılır. Ancak platformun sağladığı yazılımlar ile veriler, blokzincir ağından çekilerek deşifre edilebilir (Adanur Dedeturk ve diğerleri, 2021).

1.3.3 Blokzincir ve Yapay Zekanın Beraber Kullanıldığı Çalışmalar

Blokzincir teknolojisi ve yapay zekanın beraber kullanımı henüz yaygın değildir. Genomik alanda ise blokzincir ve yapay zekanın beraber kullanımı oldukça yenidir ve bu konuda çok az kaynak vardır. Blokzincir rıza – onay süreçlerinin yönetilmesi, oylanması, yetkilendirilmesi, küçük veri setlerinin kaydedilmesi ya da patentlenme gibi fonksiyonların yönetilmesinde elverişli olsa da, enerji tüketiminin yüksek olması işlem ücretlerini arttırmaktadır (Golosova ve Romanovs, 2018). Ethereum akıllı kontratları sayesinde dağıtık uygulamalar (Dapps) geliştirilebilmktedir (Zhouve diğerleri, 2020). Ancak, bir yapay zekâ modelinin ağ üzerinde eğitilmesi gibi kompleks hesaplama gerektiren işlemler, ağa çok fazla yük bindirdiği için hem teknik hem de ekonomik anlamda maliyetli oluyor. Sonraki bölümlerde anlatılacağı gibi, blokzincir ağlarında bir uygulamanın çalıştırılması için ağ katılımcılarına bir ücret ödemek gerekiyor. Örneğin Ethereum ağında bu ücretlere ‘GAS’ denilmektedir

(url10). Görüntü işleme, yapay zekâ modeli eğitme ya da ağ analizleri gibi işlemci maliyeti yüksek işlemler için alternatif olarak hibrit yaklaşımların kullanıldığı görülmektedir. Hibrit yaklaşımda maliyetli hesaplamalar blokzincir ağı dışında (off-chain) yapılır, güvenli iletişimi sağlayacak gereksinimler ise blokzincir ağı içerisinde (on-chain) yapılmaktadır (Acharya ve diğerleri, 2019).

Bu iki teknolojinin beraber kullanımı genelde nesnelerin interneti ile ilgili makalelerde öne çıktığı görülmüştür. Yapay zekâ ve blokzincirin beraber kullanımında, eğitilen modellerin enthereum akıllı kontratlarına gömülerek blok zincir üzerinde çalıştırılması şeklinde olabilir (url11). Ek olarak FL teknikleri ile eğitilen modellerin ağırlıkları ağ üzerinde paylaşılarak, bir topluluğun yerel verilerini koruyarak ortak model eğitimine katıldıkları çalışmalar da mevcuttur (Lu ve diğerleri, 2020).

Blokzincir ile YZ' nın beraber kullanımı, sağlık sektöründe hastaların takibi veya YZ destekli tanı konulmasında yararlanılabilir (Hou ve diğerleri, 2021). Lu ve diğerleri(2020), çalışmasında, endüstriyel IoT'de mahiremiyet korumalı veri paylaşımı için blokzincir ve FL teknolojilerini kullanmıştır. Çok taraflı katılımcıların güvenli ve dağıtık veri paylaşımı için blokzincirden faydalanmıştır. Bu paylaşım ağında, gerçek veri yerine eğitilmiş modellerin paylaşımı sağlanmıştır. Ayrıca blokzincir ağına yeni bir blok eklenmesi için gerekli fikir birliği protokolü (consensus), FL için harcanan efor ile sağlanmıştır. Böylece blokzincir teknolojisinin dezavantajlarından biri olan 'çalışma ispatı(PoW)' na harcanan enerji israfı engellenerek bu enerji FL için değerlendirilmiştir. Bu çalışma başka alanlarada uygulanabilecek bir yaklaşım sunmaktadır.

Bir diğer çalışmada, pandeminin başında Kovid hastaları ile ilgili yapılan test kitlerinin yetersizliğine bir çözüm olarak önerilen güncel bilgisayarlı tomografi (BT) resimleri üzerinden Kovid teşhisi yapabilen bir model önerilmiştir. Bu modele göre katılımcılar en güncel BT verilerini kullanarak eğittikleri YZ modellerini blokzincir ağında paylaşarak, iş birliği içerisinde hastalık teşhisinin daha hızlı yapılabileceği bir ortam kurmayı amaçlamıştır (Kumar ve diğerleri, 2020).

Son zamanlarda giyilebilir cihazların artmasıyla, elektronik cihazlar kişilerin sağlık kayıtlarını toplayabilmektedir. Bu cihazlar genel olarak nesnelerin sağlık interneti (IoMT) olarak adlandırılıyor. Cihazlar üzerinden toplanan sağlık verileri, federe

öğrenme metotları ile bazı hastalıkların tahmin edilmesinde kullanılabilir. IoMT de kişisel verilerin korunması için blokzincir – YZ nın beraber kullanımı önerilmiştir (Rahman ve diğerleri, 2020).

IoMT alanındaki başka bir çalışmada da cihaz verilerinin veya eğitimli sınıflandırıcı modellerin takas edilerek ML kullanımına ve ayrıca blok zincir ile kullanıcı veri güvenliğine odaklanılır. Bu çalışmada, çok taraflı bir mimari ile genel olarak belirli görevleri ayırarak, birimlere atanmasına izin verir. Taraflar FL ile bir arada çalışır. Birçok kaynaktan elde edilen sonuçları sınıflandırmak için konsorsiyum mekanizmasını kullanır, blok zinciri ile özel veri süreçlerinde paylaşım ve korumayı gerçekleştirir (Połap ve diğerleri, 2021).

Başka bir çalışmada blokzincir temelli şeker hastalığının erken teşhisi için çeşitli ML sınıflandırma modelinin çalıştırılabildiği bir yapı önerilmiştir. Veriler giyilebilir teknolojiler ile toplanarak modeli tahminleme oluşturulmuş ve tahminleme modeli ile blokzincir ve IPFS in beraber çalıştığı bir model önerilmiştir. Sistemde hasta ve doktor, öncelikle ilgili tüm bilgileri toplayarak, bir özel anahtar ve ID ile, Kayıt Merkezi'ne talepte bulunarak kayıt olur ve daha sonra talepleri Yönetim Birimine iletilir. Yönetim Birimi blokzincir üzerinde işlem yapacak kullanıcılar için merkezi kontrolör görevini yapar. Blokzincir yazma ya da okuma için yönetim birimi açık anahtar bilgisini kontrol eden bir akıllı sözleşme ile okuma ya da yazma yetkisinin olup olmadığına bakar. Doğrulaması yapılan kullanıcılar IPFS sistemine erişebilir ve işle koşabilir. Modeller ile ilgili performans metriklerini hesaplayabilir(Adanur Dedeturk ve diğerleri, 2021).

1.4 BLOKZİNCİR VE FEDERE ÖĞRENME TEMELLERİ

Bu bölümde araştırmamıza konu olan iki önemli teknoloji ile ilgili temel bilgilere yer verilecektir. Blokzincir ağlarının özellikleri, mimarisi, riskleri ve çalışmadaki öneminden bahsedilecektir. Sonrasında ise bir makina öğrenmesi yöntemi olan federe öğrenme (FL) yaklaşımının ne demek olduğu ve ne gibi sorunlara çözüm sağlayacağı anlatılacaktır.

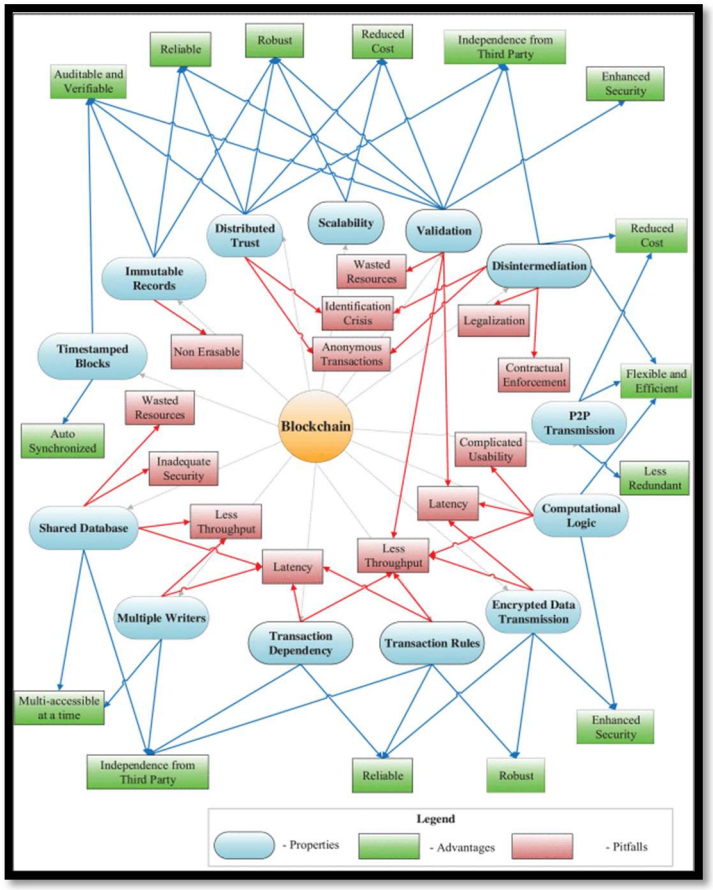
1.4.1 Blokzincir Yapısı ve Özellikleri

Nakamoto, BitCoin adını verdiği kripto para birimi ile, ekonomik krizlerden sonra güvenilirliği tartışılan aracı kurumlara ihtiyaç duymadan, online olarak para transferinin ve ödemelerin nasıl yapılabileceğini anlattı. Kripto paralar BitCoin' den önce de vardı. Fakat bu paraların temel sorunu çift harcama (double spending) problemini ortadan kaldırmak için merkezi sistemlere ihtiyaç duymasıydı. Nakamato, makalesinde bu probleme çözüm sağlayacak bir protokol ortaya koydu (Nakamoto, 2009). Önerdiği yöntem ile kişiler arası para transferi gerçek zamana yakın bir şekilde gerçekleştirilebilir hale geldi.

Nakamato çift harcama sorununu blokzincir yapısı ile çözmüştür. Blokzincir P2P (peer-to-peer), eşler arası ağ protokolü ile modellenmiş bir ağ üzerinde tutulan, işlemleri değitirilemez, merkeziyetsiz olarak yönetilen ve kriptografik algoritmalar ile güvenliği sağlanan bir hesap defteri ya da veri tabanıdır (Chowdhury et al., 2020). Ağda dağıtılan veriyi tutan her üye sunucuya düğüm (node) adı verilir. İki kere harcama sorununu engellemek için düğümlere kendi başlarına veriyi değiştirme yetkisi verilmez. Blokzincir, çeşitli fikir birliği (consensus) algoritmaları aracılığıyla uygulanan farklı doğrulama mekanizmalarını kullanır. Bu algoritmalar, bir grup katılımcı arasında ağdaki işlemlerin gerçekten yapıldığının doğrulanmasını sağlayan anlaşmalardır. Çoğunluğun doğruladığı işlemler ağdaki bloklara yazılmaya hak kazanır (Mingxiao ve diğerleri, 2017).

Tanımından da anlaşıldığı gibi blokzincirlerin değiştirilemezlik, dağıtık veri tabanı, güvenlik, şeffaflık ve merkeziyetsizlik özellikleri ön plana çıkmaktadır. Bu özellikler sayesinde blokzincir teknolojisi sadece para transferlerinde değil, değerli görülen herhangi bir varlığın merkezi bir otoriteye gerek kalmadan paylaşımında kullanışlı olmaktadır. Örneğin devlet yönetim işlerinin bazıları, elektronik oylama, tapu, diploma, patent, elmas sertifikaları, güvenli sağlık veri tabanı, tedarik zinciri takibi gibi bir çok alanda bu teknoloji kullanılabilir (Aydiner, 2021). Blokzincir teknolojisi birçok avantaj getirse de bazı negatif tehlikeleri olabilir. Şekil 8 de blokzincirin özellikleri, avantajları ve bu avantajlar karşısında nelerden feragat edilmesi gerekebileceği resmedilmiştir. Örneğin, dağıtık veri tabanı özelliği sayesinde aynı anda çoklu erişim ve merkeziyetsizlik sağlanabilir. Fakat böyle bir özellik aynı zamanda kaynakların verimsiz kullanılması, düşük hızda veri işleme, yetersiz güvenlik

ve gecikmelere yol açar. Benzer şekilde blokzincirin diğer özelliklerinin getirdiği avantajlar karşılığında bazı dezavantajlar oluşabilmektedir (Islam ve diğerleri, 2020).



Şekil 8. Blokzincir özellikleri, avantajları ve tuzakları arasındaki ilişki (Islam ve diğerleri, 2020).

Aşağıda bu çalışmanın dayandığı en önemli blokzincir yapılarından bahsedilecektir. Bunlar;

a) Kriptografik güvenlik
b) Merkeziyetsiz
c) Değiştirilemezlik
d) Akıllı Kontratlar

1.4.1.1 Kriptografik Güvenlik

Kriptografi, bir ağ üzerinde katılımcılar arasındaki mesajların üçüncü şahıslar tarafından okunmasını engelleyerek oluşturulmuş güvenli iletişim tekniklerinin bütünüdür (Coron, 2006). Blokzincir mekanizması kriptografi biliminin temelleri üzerinde kurgulanmıştır. Kriptografi, blokzincir üzerinde katılımcıların mahremiyetini koruyarak ağ, işlem verileri ve hesapların güvenliğini sağlar (Kalogeropoulos, 2018).

Hash fonksiyonları, blokzincir ağında blokların oluşturulmasında, mutabakat ve zincir yapısının bağlanmasında kullanılır (Nakamoto, 2009). Hash fonksiyonları herhangi bir anahtar bilgisine ihtiyaç duymadan, verinin boyutundan bağımsız olarak sabit uzunlukta bir değer üretir. Blokzincir ağı üzerinde yaygın olarak SHA256 algoritması kullanılmaktadır. Hash algoritmaları aşağıdaki özelliklere sahiptir (Conley, 2019).

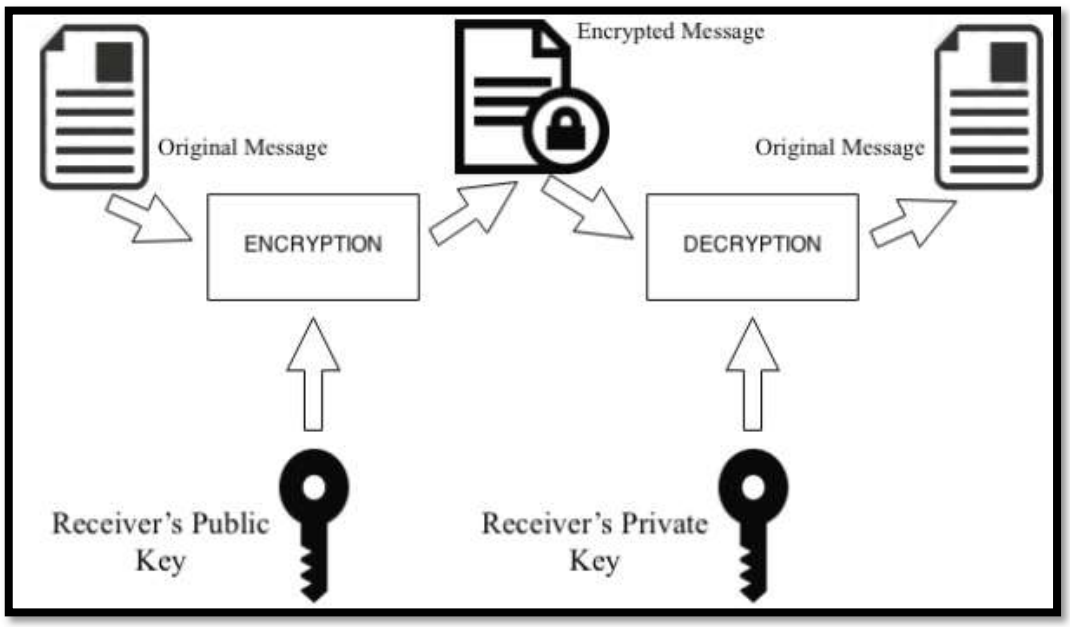
a) Mesaj boyutundan bağımsız, sabit uzunlukta bir sonuç üretir. Ayrıca bilgisayar işlemcilerine çok yük getirmeden, hızlıca hesaplanabilir.
b) Deterministiktir. Yani aynı girdi için her zaman aynı sonuç alınır.
c) Mesajlar birbirine çok benzese bile hash sonuçları tamamen farklı olur.
d) Geri döndürülemezdir. Bir hash değerine bakılarak hangi veriden bu hash değerinin üretildiği bulunamaz.

Tablo 2 de görüldüğü gibi, bir karakterin değişmesi ya da eklenmesi sonucu oluşan hash çıktılarında tamamen farklı sonuçlar çıkmıştır. Her bir blok bir hash değeri ile ifade edilir. Bu hash değeri blok içindeki mesaj, bir önceki bloğun hash değeri ve tarih-zaman gibi verilerin tamamını kullanarak oluşturulur. Bu sayede blokzincir üzerindeki bir blokta küçük bir değişiklik yapıldığında bloklar arasındaki ilişki bozulmuş olur. Blokzincirin veri güvenliği böyle sağlanmaktadır.

Tablo 2. SHA256 algoritması kullanılarak şifrelenmiş veri örnekleri.

GİRDİ	ÇIKTI(SHA256)
123456789	15e2b0d3c33891ebb0f1ef609ec419420c20e320ce94c65fbc8c3312448eb225
023456789	3ad519f9744c1c847d1336095e8f948a7b5b10d69d11a0f1f3faf1b83b3aae6c
1234567890	c775e7b757ede630cd0aa1113bd102661ab38829ca52a6422ab782862f268646

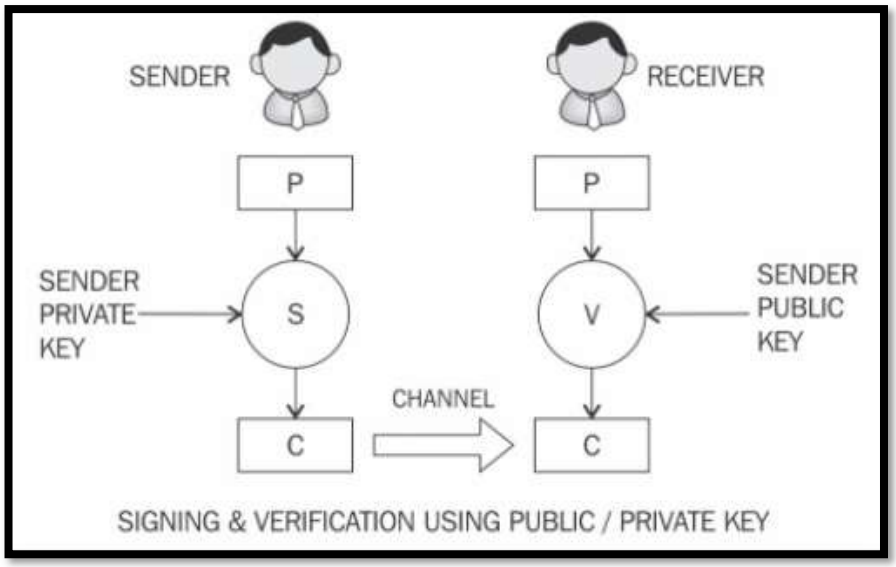
Mesaj alış-verişlerinde blokzincir, asimetrik şifreleme yönteminden faydalanır. Asimetrik şifrelemede genel ve özel olarak iki anahtar kullanılır. Genel anahtarlar özel anahtar kullanılarak türetilebilir ama tersi mümkün değildir. Özel anahtar ile şifrelenen bilgi genel anahtar ile okunabilir. Bir mesajın doğru kişiden geldiğini anlamak için mesajlar e-imza yöntemi ile iletilir (Kalogeropoulos, 2018).



Şekil 9. Asimetrik şifreleme (Bhatia ve Sumbaly, 2014).

Asimetrik şifreleme Şekil 9' da gösterildiği gibi, mesajı gönderen, mesajı alıcının genel anahtarı ile şifreler. Şifreli mesaj aracılar vasıtası ile alıcıya iletilir. Alıcı kendi özel anahtarı ile mesajı açabilir (Bhatia ve Sumbaly, 2014). Fakat iletilen mesajın gerçekten gönderici tarafından iletildiğini anlamak için dijital imza kullanılır. Dijital imza, göndericinin özel anahtarı ile mesaj şifrelenerek oluşturulur. Alıcı, göndericinin açık anahtarı ile gelen dijital imzayı kontrol ederek mesajı doğrular. Blokzincir ağında

işlemler, dijital imzalı olarak bloklara yazılır. Gönderilen bir paranın geçekte ilgili hesap tarafından gönderildiği bu yolla kontrol edilebilmektedir. Dijital imza mekanizması Şekil 10 da resmedilmiştir (Bashir, 2020).



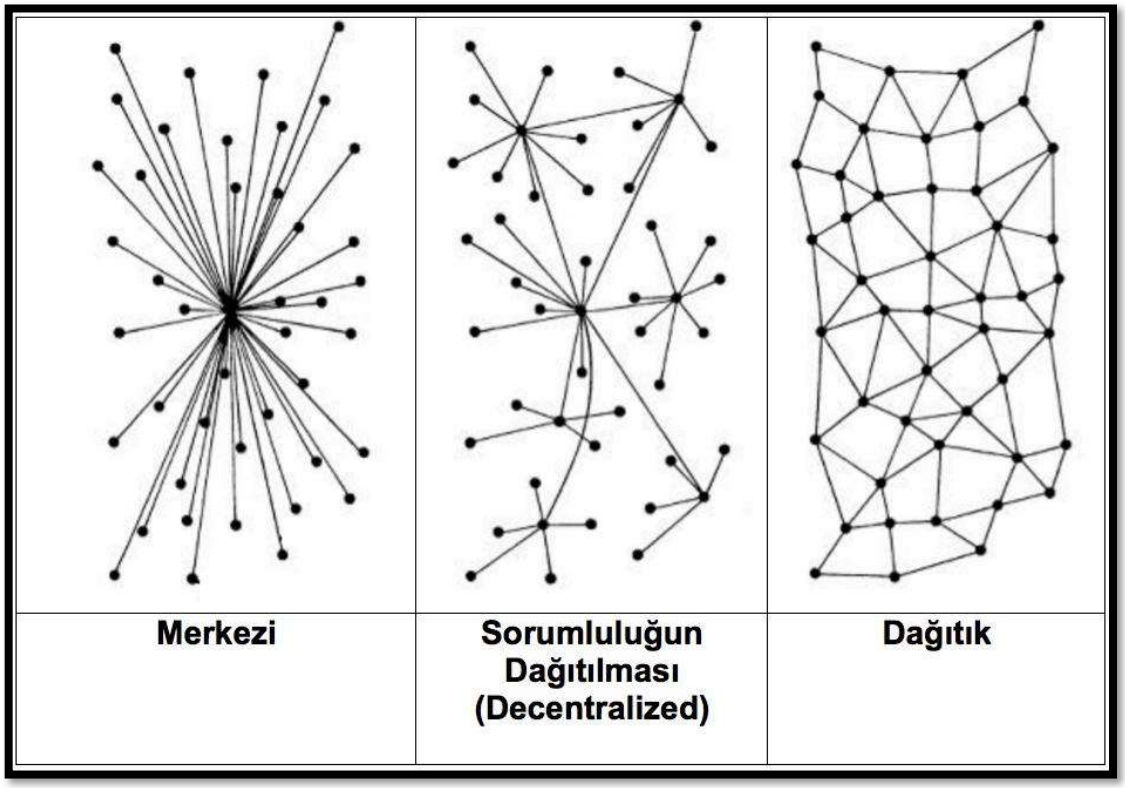
Şekil 10. Dijital imza (Bashir, 2020)

1.4.1.2 Merkeziyetsizlik

Merkeziyetsizlik, tüm kontrolün tek merkezde toplanmasının aksine, kontrol ve otoritenin organizasyondaki taraflar arasında dağıtıldığı sistemlerdir. Bu yapıların verimlilik, hızlı karar alma, daha iyi motivasyon sağlama, üst yönetimin sorumluluğunu azaltma gibi faydaları vardır (Bashir, 2020). Bilgisayar bilimleri geleneksel olarak, veri tabanı veya uygulamaların yöneticisi merkezi bir otoritenin kontrolü altında toplandığı bir paradigmaya dayanmaktadır. Bitcoin ve blokzincir teknolojisinin ortaya çıkmasıyla birlikte bu paradigma değişmeye başladı. Daha az insan müdahalesinin gerektiği, hatanın tek noktada (single point of failure) toplanmadığı ve tek bir güvenilir otoriteye ihtiyaç kalmadığı yapılar popülerlik kazanmaya başladı.

Şekil 11 de farklı ağ modelleri resmedilmiştir. Merkezi sistemler geleneksel olarak tasarlanan sunucu – müşteri uygulamalarıdır. Tüm hizmetler merkez tarafından sağlanır. Dağıtık sistemlerde ise verilerin depolanması ve işlemler, ağdaki katılımcılar

arasında dağıtık olarak yürütülmektedir. Paralel işleme uygulamalarına bulut ve big data sistemleri örnek olarak gösterilebilir. Dağıtık sistemlerde işlerin çalışması yönetimi koordine edilmelidir. Yani doğası gereği bu sistemler merkezi sistem gibidir. Sorumluluğun dağıtıldığı sistemlerde ise ana bir düğüme gerek kalmadan, sorumluluk ağdaki diğer üyeler arasında paylaşılır. Bu yapılarda her departmanın kendi veri tabanını ve işlemcilerini yönetmesi gibi düşünülebilir (url12). Bitcoin sayesinde, sorumluluğun dağıtıldığı sistemlerde iş birliği sağlayabileceğimiz bir mutabakat (consensus) yapısı sunularak, merkeziyetsiz yapılarda koordinasyon mümkün olmuştur (Nakamoto, 2009). Blokzincir yapısında merkeziyetsizlik depolama, iletişim, hesaplama ve mutabakat başlıklarında incelenebilir.



Şekil 11. Farklı tipteki ağ sistemleri (URL12).

1.4.1.3 Depolama

Blokzincir ağlarında veriler direk olarak bloklara yazılabilir. Bloklar her düğümde tutulmak zorundadır. Böylece doğal olarak merkeziyetsiz bir kayıt sistemi oluşmuş olur. Fakat blokzincirler, yüksek hacimli veriyi bloklarda tutmak için elverişli değildir.

Küçük kayıt verileri bloklar üzerinde tutulabilir ancak video, resim ya da çok büyük karakter verileri bloklarda tutmak uygun değildir.

Yüksek hacimli veriler dağıtık olarak, dağıtık hash tablolarında (DTH) tutulabilir. Bu yapılara genişleyebilir ve hızlı bire-bir(p2p) veri paylaşımı yapılabilen torrent sistemleri örnek gösterilebilir. Fakat p2p veri paylaşımının dezavantajı katılımcıların ellerindeki veriyi saklamak için bir motivasyonlarının olmamasıdır. Blokzincir sistemlerinde birinci öncelik tutarlı erişilebilir veri ve bağlantı kararlılığıdır. Yani veriye gerektiğinde erişilebilmeli ve ağdaki iletişim sürekli devam edilebilmelidir (Bashir, 2020). Bu amaçla Gezegenler Arası Dosya Sistemi (IPFS) bu özelliklerin her ikisini de kapsayacak şekilde geliştirilmiştir. Asıl hedefi ise http protokolünü değiştirerek, merkezi olmayan bir WWW ağı sağlamaktır. IPFS, depolama ve arama işlevselliği sağlamak için Kademlia DHT ve Merkle Yönlendirilmiş Döngüsel Grafikleri (DAG) yapılarını kullanır (Franco ve diğerleri, 2021).

Alternatif olarak Filecoin, dağıtık veri depolama hizmeti sunarken, ağ katılımcılarına gönderdiği ve aldığı verilerin boyutuna göre teşvik verir. Bu sistem IPFS sistemi üzerinde çalışır ve veriyi şifreli olarak depolar. Ethereum kendi ekosisteminde Swarm, Storj ve MaidSafe'i dağıtık veri depolama yapıları olarak sunmaktadır (Zahed Benisi ve diğerleri, 2020). Ayrıca BigChainDB de geleneksel bir dosya sisteminin aksine ölçeklenebilir, hızlı merkeziyetsiz veritabanı hizmeti sağlamayı amaçlayan başka bir depolama projesidir (El-Hindi ve diğerleri, 2019).

1.4.1.4 İletişim

İnsanları birbirine bağlayan en önemli iletişim aracı internettir. İnternet ilk başlarda merkeziyetsiz bir felsefe ile hayata geçse de, günümüzde bu geniş ağ üzerindeki kontrolün hükümetler, sosyal ağ ve servis sağlayıcılarının eline geçtiğini söyleyebiliriz. Mevcut internet yapısının demokrasi yoksunluğu, sansür, bant genişliği ve güvenlik sorunları vardır (Alabdulwahhab, 2018). Bu sorunları örneklerle açıklayacak olursak ilk olarak, internet hizmetini belirli bir internet sağlayıcı firmadan (ISP) almak zorundayız. Anlaşmalı olduğumuz ISP firması dilediği zaman internet hizmetimizi bir teknik ya da ekonomik sebepten kesebilir. Ayrıca internet üzerinden hareketlerimizi takip edebilir, verilerimizi toplayarak ticari amaçlarla kullanabilir.

Bununla beraber verilerimiz sadece ISP firmaları tarafından değil, aynı zamanda internet altyapısını kullanan sosyal medya, akıllı telefon uygulamaları, çevrimiçi oyunlar ya da Google, Amazon gibi büyük bulut firmaları tarafından da toplanmaktadır. Bu firmalar, kullanıcılarından topladıkları verileri kendi çıkarları için kullanabilmektedirler. Bir diğer sorun da hükümetler dilediklerinde internet erişimini engelleyebilmekte ve bazı uygulamalara erişim yasağı getirebilmektedir. Daha da fazlası mevcut internet alt yapısı siber saldırılara son derece açıktır. Bilgisayar korsanlarının kişisel hesapları ele geçirerek maddi ve manevi, firmalara ve kişilere zararlar verdiği görülmektedir. Bu sorunlar çeşitli protokoller ile oluşturulmuş blokzincir ağları üzerinde çözülebilmektedir. Bu yeni yaklaşım web 3.0 olarak adlandırılır ve bireylerin kendi verileri ve aldıkları hizmetler üzerinde tam kontrole sahip olmasını amaçlar (Alabdulwahhab, 2018). Tablo 3' de web in tarihsel olarak gelişimi ile beraber ortaya çıkan özellikleri listelenmiştir.

Tablo 3. Web versiyonları arasındaki farklar (URL15).

(1990 – 2000)	(2000 – 2010)	(2010 – 2020)	(2020 – 2030)
Web 1.0	**Web 2.0**	**Web 3.0**	**Web 4.0**
Salt okunabilir	Okunup yazılabilir	Taşınabilir ve kişisel	Kendi kendine öğrenebilen, kendi kendini organize edebilen
Doküman odaklı	İnsan odaklı	Bilgi odaklı	Sanal gerçeklik ve büyük veri odaklı
Bireysel bilgiye erişim	İnternet deneyimini başkalarıyla paylaşım	Sanal bilgisayar asistanları yardımcı olur	Bilgisayarlar isteklerinizi anlar. Faaliyetlerinizi yönetir.
Statik web	Sosyal web	Anlamsal (semantik) web	Simbiyotik web / Akıllı web
Web formları	Web uygulamaları	Akıllı uygulamalar	Yapay zeka, makine öğrenmesi, Sanal gerçeklik, IoT, Otonom hizmetler
Şirket odaklı	Toplum odaklı	Bireysel odaklı	Yapay zeka odaklı
Şirket siteleri	Blog siteler, Sosyal medya	Yarı sosyal ağ yarı blog siteler, Canlı yayınlar	Yapay zeka ile otonom çalışan siteler
Anonim kullanıcılar	Kayıtlı kullanıcılar	Yetkili kullanıcılar	Kişiselleştirilmiş verilere sahip kullanıcılar
Haber bağlantı ağı	İnsan bağlantı ağı	Bilgi bağlantı ağı	Yapay zeka bağlantı ağı
Tüm hakları saklı ©	Creative commons lisansı (CC)	Açık bilgi	Gizli bilgi

1.4.1.5 Hesaplama

Bitcoin, taraflar arasında merkeziyetsiz para transferi yapan belirli bir servisi sağlamaktadır. Ethereum gibi ağlar ise, ağ üzerinde ethereum sanal makine (EVM) ve akıllı kontrat yapısı ile daha farklı hizmetlerin sağlanabildiği bir blokzincir ağı olmuştur. Ancak blokzincir ağları kompleks ve yüksek işlemci gücü gerektiren

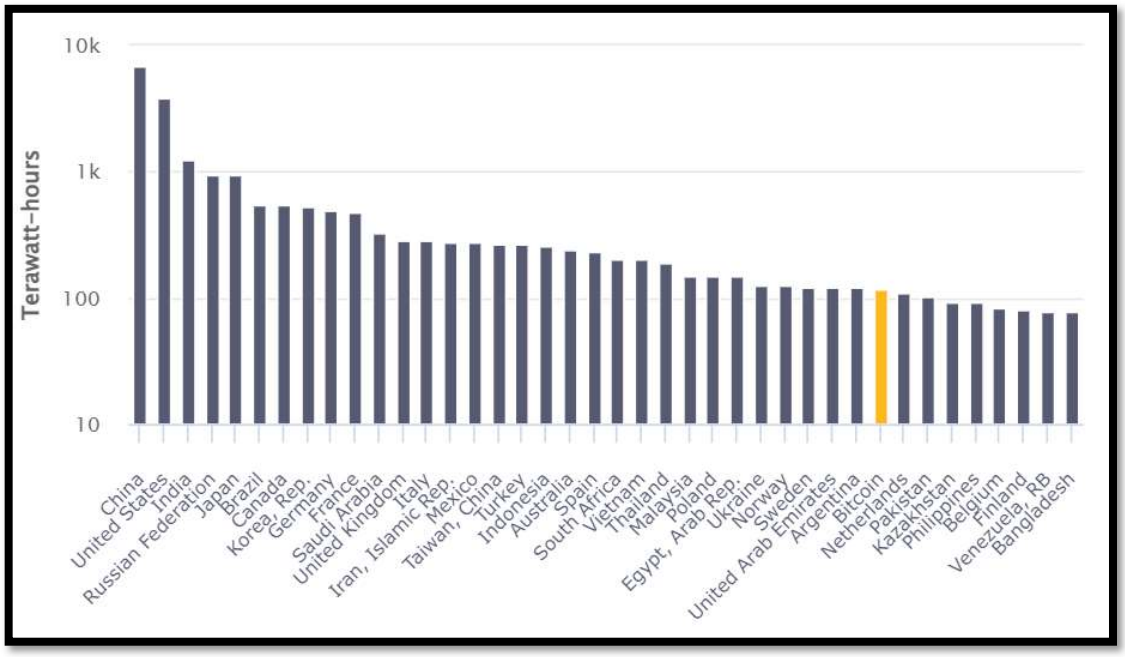
işlemlerin çalıştırılması için elverişli ortamlar değildir (Alacam ve Sencer, 2021). Yüksek işlemci gücü gerektiren işlemler, anlatıldığı gibi on-chain/off-chain mimarisi ile çözülebilmektedir. Akıllı kontratlar sayesinde belirli bir işin yapılması için ağ üzerindeki düğümlerden kaynak kiralaması yapılarak paralel işleme yapmakta mümkündür (Cai ve diğerleri, 2018). Örneğin FoldingCoin mimarisinde, bir proteinin üç boyutlu şeklini çıkarmak için ağ üzerinde bulunan düğümlerin kaynakları kiralanarak çalışan bir mekanizma tasarlanmıştır (Ross ve diğerleri, 2018).

1.4.1.6 Mutabakat (Consensus)

Blokzincir uygulamalarının sağlıklı çalışabilmesi için gerekli iki önemli problem vardır. Bunlar çift harcama sorunu ve Bizanslı Generaller Problemidir (Reischuk, 1985). Çift harcama sorunu, bir dijital para biriminin ya da fonun aynı anda iki farklı işlemde kullanılmasıdır. Blokzincir ağında işlermlerin kontrolü düğümler tarafından yapılır. Fakat bu yaklaşımda düğümlerin güvenilir olması gerekmektedir. Dağıtık sistemlerde katılımcıların güvenilir olup olmadığı problemi Bizanslı Generaller Problemi olarak adlandırılır. İki taraf arasında yapılan bir işlemin farklı düğümler tarafından kontrol edilmesi ve bu işlemin iletişim ağı üzerinde herhangi değişikliğe uğramadan gerçekleştiğinin kontrol edilmesi gerekmektedir (Mingxiao ve diğerleri, 2017). Bitcoin ile bu sorunun çözümü, ağ katılımcılarının %51 inin güvenilir olduğu kabul edilerek çözülmüştür (Nakamoto, 2009). Yani çoğunluğun mutabık kaldığı transfer işlemleri ancak blokzincir ağına yazılmaktadır. Farklı blokzincir ağlarında farklı mutabakat yapıları mevcuttur. En bilinen mutabakat yapıları, ağdaki katılımcıların belirli ve güvenilir olup olmadığına göre farklılık gösterir. Bu çalışmada bir çok yöntem içerisinden en çok bilenen PoW (Proof of Work), PoS (Proof of Stake), DPoS (Delegated Proof of Stake), PBFT (Practical Byzantine Fault Tolerance) ve PoA (The Proof of Activity) yöntemlerinden bahsedilecektir.

PoW (Proof of Work): PoW blokzincir sistemlerinde kullanılan ilk ve en yaygın mutabakat protokolüdür. Basitçe PoW bir iş için gereken eforun harcandığının ispatıdır. PoW ilk olarak 1993 yılında, spam mail problemini çözmek için önerilmiştir. Mail göndericileri mail göndermeden önce, mail gönderebilmek için bir efor harcadığını ispatlamalıdır. Bu yöntem harcanan eforun, ispatlanması kolay ve tekrar gönderimi engelleyecek kadar çözümü zor olan bir matematik probleminin çözümü

şeklinde çalışmaktadır (Dwork ve Naor, 1993). Nakamoto bitcoin makalesinde düğümlerin tutarlılığı için bu yöntemi geliştirerek kullanmıştır. Nakamoto, mutabakat protokolüne göre blok içerisindeki bilgilerin ve 256 bit boyutunda bir Nonce değerini değiştirerek protokolün belirlediği sayıda başında sıfır olan bir hash değerinin bulunması şeklinde tasarlanmıştır. Nonce, şanslı sayı olarak adlandırılır. Ağdaki düğümler farklı Nonce değerlerini deneyerek bulmacayı çözmeye çalışırlar. Protokolün belirlediği özellikte bir hash değeri bulunursa bu işlemi başaran düğüme bir ödül verilir ve bu işlem *madencilik*(mining) olarak adlandırılır (Nakamoto, 2009; Wan ve diğerleri, 2020). PoW ün en büyük dezavantajı, yüksek enerji tüketimine sebep olmasıdır. Şekil 13 de Bitcoin ağının madencilik için harcadığı enerji miktarının diğer ülkeler ile kıyaslaması gösterilmiştir. Günümüzde Bitcoin madenciliği için harcanan enerji Hollanda, Belçika, Finlandiya gibi birçok ülkenin enerji tüketiminin önüne geçerek 32. sırada yer almıştır. Bu durum sürdürülebilirlik açısından önemli sorun teşkil etmektedir.



Şekil 12. Bitcoin saatlik enerji tüketiminin ülkelere göre sıralaması (url13).

PoS (Proof of Stake): PoS de PoW gibi bir sonraki bloğun kim tarafından zincire yazılacağını belirler. PoW ün en büyük dezavantajı madencilik işlemleri için çok fazla enerji tüketilmesidir. PoS mutabakat protokolü, fazla enerji tüketimine ve %51

saldırısına çözüm olacak alternatif bir çözüm olarak ortaya atılmıştır (S. Wan ve diğerleri, 2020). PoS in en büyük farklılığı yüksek enerji maliyetinin olmamasıdır. PoS de düğümler, yeni blok yazmak için ortaya belirli miktar hisse koyarlar. En yüksek hisseyi teklif edenin blok yazma şansı daha fazla olur (Ozercan ve diğerleri, 2018). PoS de madenci kavramı yerine onaylayıcı(validator) terimi kullanılır. Yeni blok yazma hakkını kazananlar için bitcoindeki gibi bir ödüllendirme yoktur. Ancak onaylayıcı olarak seçilen düğüm, işlem ücretlerini alır. PoS de zengin olan katılımcıların onaylayıcı olarak seçilme şansı daha fazla olduğu için bir haksızlık söz konusudur. Ağdaki zengin katılımcıların daha da zenginleşmesi olasıdır. Algoritma bu haksız durumu azaltmak için, seçim yaparken tarafların elindeki hisselerin ne kadar süredir tutulduğunu da hesaba katar. Bu süreye elektronik para yaşı(coin-age) denir (Nicolas, 2014).

PoS her ne kadar enerji tüketimini düşürse de tamamen sorunsuz bir sistem değildir. Bu algoritmanın aşağıdaki gibi sorunları vardır (S. Wan ve diğerleri, 2020).

a) İlk paranın dağılımı sorunludur. PoS kullanan sistemler ilk paraların dağılımı için iki yöntem kullanır. İlki PoW ile ağı başlatıp sonra PoS ile devam etmek. Diğer yöntem ise halka arz yöntemi ile ilk ücretlerin dağılımıdır. Fakat bu yöntemde geliştiricilerin ve bazı diğrlerinin elinde para yoğunlaşması durumu ortaya çıkabilir.
b) İstiflemeyi teşvik eder. Hisse olarak önerilen paraların elektronik para yaşının yanında, bloklarda tutulan diğer paraların da yaşı sıfırlanır ve bu durumda onaylayıcılar bir fayda elde etmez.
c) Elektronik para yaşının, düğümler çevrim dışı olduğunda da artması sorunu. Düğümler çevrim dışı olduğunda da ellerindeki elektronik paraların yaşı artar. Bundan dolayı düğümler, elektronik para yaşı belirli bir seviyeye gelene kadar masraflarını azaltmak için çevrim dışı kalmayı tercih edebilir. Çevrim içi düğümlerin azalması ağı saldırılara açık ve veri senkronizasyonu sorunlarına neden olabilir.
d) Ortaya konan bir şeyin olmaması (nothing at stake) problemi. Blokzincir ağlarında ataklar ya da yazılım güncellemeleri ile çatallaşma olabilmektedir. PoS de onaylayıcılar her çatal için teklif sunabilir. Bu

durum illegal olarak yaratılmış çatalların devamına ya da alternatif çatal oluşturmaya sebep olabilir.

e) Bir diğer sorun ise uzun menzil atağı sorunudur. Uzun menzilli saldırılarda, son bilgilere erişmeden iki alternatif geçmiş arasında ayrım yapmaya çalışan bir kurban düğümü vardır. Ağda uzun süredir çevrimdışı olan ya da yeni dahil olan düğümlerin güncel durumu yakalayabilmesi için uzun süre geçebilir.

DPoS (Delegated Proof of Stake): PoS ve DPOS arasındaki en büyük fark, PoS'un doğrudan demokratik, DPOS'un ise temsili demokratik olmasıdır. Bu yöntemde delegeler ve şahitler olarak iki rol vardır. Paydaşlar blok oluşturmak ve doğrulamak için delegelerini seçerler. Delegeler blokları doğrular. Bloğu doğrulamak için önemli ölçüde daha az düğümle blok hızlı bir şekilde onaylanabilir ve bu da işlemlerin hızlı bir şekilde onaylanmasına yol açar (Zheng ve diğerleri, 2017).

PBFT (Practical Byzantine Fault Tolerance): Yukarıda bahsedilen mutabakat yöntemlerinin hepsi, dağıtık ağ katılımcılarının sınırsız olduğu ağlarda kullanıma elverişlidir. PBFT ise düğüm sayısının belli olduğu özel blokzincir ağlarında kullanılır(S. Wan ve diğerleri, 2020). Bizanslı generaller probleminde kaleyi kuşatmış generaller aynı anda saldırmalı ya da aynı anda geri çekilmelidir. Diğer durumlarda savaş kaybedilecektir. Bizans Hata Toleransı (BFT), hain olduğu bilinen bir generaller topluluğunda da savaşın kazanılması durumunu açıklar. Çoğunluğun dürüst olduğu ağlarda sistem başarılıdır. PBFT, senkron olmayan sistemlerde BFT probleminin çözümü için geliştirilmiş hızlı ve performanslı bir yöntemdir. Bu yöntemde ağda bir düğüm lider, diğerleri ise takipçi olur. Bütün düğümler iletişimdedir ve amaç dürüst çoğunluğun ağın durumu hakkında anlaşmaya varmasıdır. Ağdaki düğümler çok sık iletişim kurarak, mesajın ağda kabul edilen bir düğümden geldiği ve işlemin ağ üzerinde değiştirilmediğini ispatlarlar. Sistemin düzgün çalışması için kötü niyetli katılımcıların oranı 1/3'den daha az olmalıdır. Mutabakat algoritması komutan – teğmen ilişkisi ile çalışır ve her turu dört fazdan oluşur. (i) Talepçi talebini lider düğüme iletir. (ii) Lider talebi takipçi düğümlere iletir. (iii) Takipçi düğümler işlemi koşar ve talepçiye geri bildirir. (iv) Talepçi düğümlerin birbirinden farklı f+1 (f: maksimum güvenilmez düğüm sayısı) adedinde aynı sonucun dönmesini bekler.

Çoğunluğun döndüğü sonuç doğru kabul edilir. Lider sırayla bir düğümden diğerine geçer (Castro, 2001).

Blokzincir teknolojisi üzerine çalışmalar devam ettikçe alternatif mutabakat protokolleri geliştirilmeye devametmektedir. Her bir protokolün olumlu ve olumsuz özellikleri vardır. Tablo 4 de en yaygın olarak kullanılan dört protokolün karşılaştırılması düğüm sayısı, ağa katılımın izin gerektirip gerektirmediği, ölçeklenebilirlik, enerji tüketimi, güvenlik önceliği, gecikme, hız ve kullanılan ağlar kapsamında listelenmiştir.

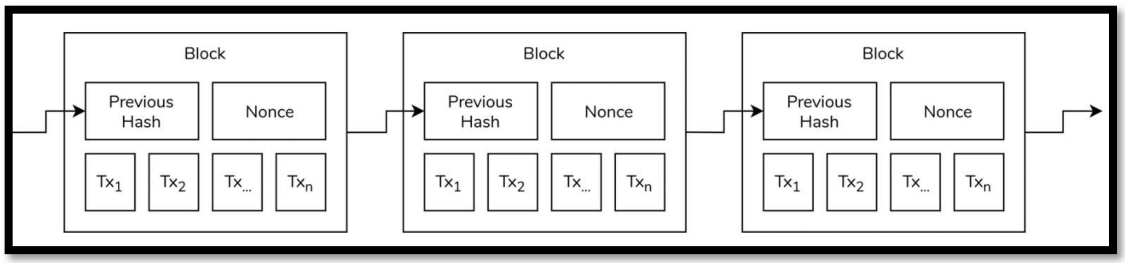
Tablo 4. Mutabakat protokollerinin karşılaştırılması(S. Wan ve diğerleri, 2020).

Consensus	PoW	PoS	DPoS	PBFT
Number of nodes	Unlimited	Unlimited	Unlimited	Limited
Permission	No	No	No	Yes
Scalability	High	High	High	Low
Energy consumption	High	Low	Low	Low
Safety preference	Liveness	Liveness	Liveness	Consistency
Latency	High	High	Low	Low
Throughput	Low	Low	High	High
Example	Bitcoin	Peercoin	BitShares	Hyperledger Fabric v0.6

1.4.1.7 Değiştirilemezlik

Blokzincirlerde kayıtlar, son gelen en sona eklenecek şekilde bloklara yazılır. Yani her yeni veri geldiğinde, zincirin sonuna veriler bloklar halinde yazılır. Şekil 13' de gösterildiği gibi işlemler listesi blokların içerisinde yer alır. Bir blok en temelde tarih-zaman, işlemler listesi, Nonce ve bir önceki bloğun hash değerini tutar. Ağa kaydedilen herhangi bir blok içerisindeki veride küçücük bir değişiklik yapılsa bile hash değeri değişeceği için bloklar arasındaki zincir yapısı, dolayısı ile entegrasyon bozulmuş olur (Kalogeropoulos, 2018). Hash değeri bir bloğun kimliğini tanımlar ve kendinden sonra gelen blok içerisine yazılarak bloklar birbirine bağlanmış olur.

İşlemler geldikçe yeni bloklar oluşturulur ve bloklar birbirlerine hash değerleri ile bağlanarak gittikçe büyüyen bir zincir yapısı oluşmuş olur (Chowdhury, 2019).



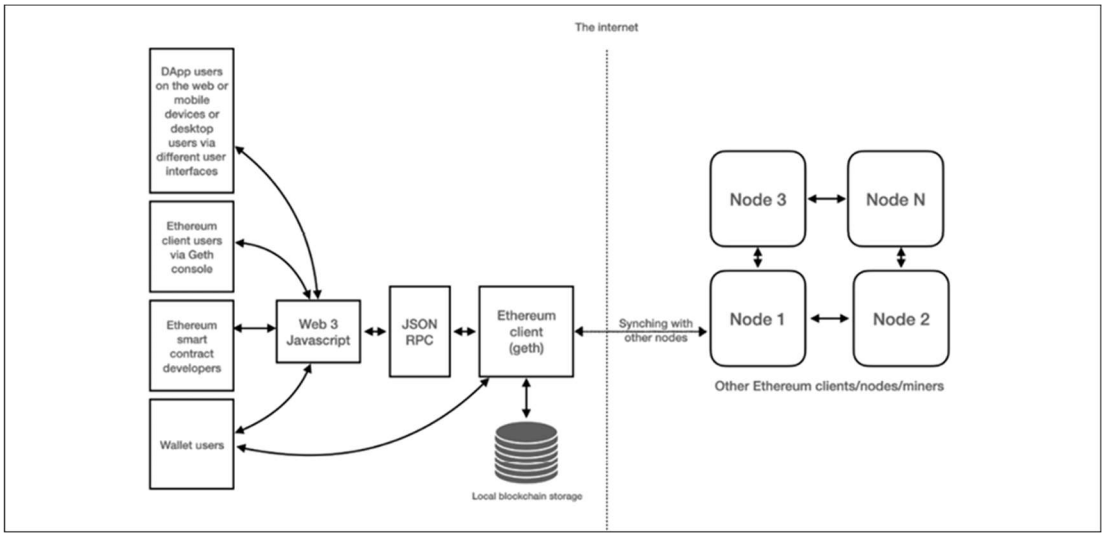
Şekil 13. Blokzincir yapısı(Mamo ve diğerleri, 2020)

1.4.1.8 Ethereum ve Akıllı Kontratlar

Akıllı kontrat terimi, ilk olarak 1997 yılında, Washington üniversitesinde bilgisayar ve kriptografi alanında çalışan Nick Szabo tarafından önerilmiştir (Szabo, 1997). Szabo (1997)' ya göre kurumlar ve bu kurumları oluşturan ilişkileri resmileştirmenin yeni yolları artık dijital devrimle mümkün hale gelmiştir. Bu yeni sözleşmeler "akıllı" olarak adlandırılmıştır. Çünkü bunlar, kâğıt temelli cansız atalarından çok daha işlevseldir. Akıllı sözleşme, tarafların vaatler üzerinde gerçekleştirdikleri protokoller de dahil olmak üzere, dijital biçimde belirtilen bir dizi sözdür (Szabo, 1997). "Akıllı kontratlar", bir anlaşmanın tamamını veya bir kısmını otomatik olarak yürüten ve blokzincir tabanlı bir platformda depolanan bilgisayar kodunu tanımlamak için kullanılan bir terimdir (Schär, 2021). Akıllı kontratlar bir blokzincir ağındaki düğümler arasında çoğaltılarak tutulur ve bu nedenle blokzincir teknolojisinin sunduğu güvenlik, kalıcılık ve değişmezlik özelliklerinden yararlanır. Dağıtık olarak tutulan bu programlar, belirli şartlar yerine getirildiğinde devreye girerek üstlendikleri görevleri yerine getirirler ve çalışan bu kod parçacıkları son durumlarını blokzincir ağı üzerinde tutarlar (Bashir, 2020). Bu tarz uygulamalar Ethereum blokzincir ağının hayata geçirdiği Solidity gibi özel programlama dilleri ile geliştirilir. Akıllı kontratlar deterministiktir. Yani aynı girdiler için her zaman aynı sonucu döndürür. Başka bir deyişle, programlar çalıştığında kodda programlanan gereksinimlerle tamamen uyumlu, güvenilir ve doğru bir iş yerine getirilmiş olur. Akıllı kontratlar aşağıdaki özelliklere sahip olmalıdır (Bashir, 2020).

a) *Otomatik çalışabilme*: Herhangi bir müdahaleye gerek kalmadan kendiliğinden çalışabilmelidir.
b) *Ugulanabilir:* Tüm kontrat koşulları otomatik gerçekleşmelidir.
c) *Güvenli:* Üzerinde çalıştığı blokzincir ağı genellikle güvenlik garantisi sağlar; ancak akıllı kontrat kodu doğru, geçerli ve doğrulanmış olmalıdır.
d) *Deterministik:* Belirli bir girdi için her zaman aynı sonucu üretmelidir.
e) *Anlamsallık:* Hem insanlar hem de bilgisayarlar için tam ve anlamlı olmalıdır.
f) *Durdurulamaz:* Saldırılar veya elverişsiz koşullar, akıllı bir sözleşmenin yürütülmesini olumsuz etkilememelidir. Akıllı kontratlar başlatıldığında kendisi için belirlenen süre içinde deterministik olarak çalışmasını tamamlarlar.

Bitcoin, Nakamoto (2009) tarafından spesifik bir programı çalıştıracak şekilde tasarlandı. Vitalik Buterin, Ethereum ile bitcoin fikrini geliştirerek, blokzincir ağları üzerinde merkezi olmayan uygulamaların (DApps) geliştirilmesine olanak tanıyan 'Turing-complete' bir dilin geliştirilmesini önerdi (Buterin, 2014). Ethereum da diğer blokzincir ağları gibi en son durumun kayıt altına alındığı bir durum makinasıdır(State machine) (Wood, 2014). Şekil 14' de gösterildiği gibi Ethereum blokzincir ekosistemi birkaç bileşenden oluşur. Bu bileşenler, durum kayıtlarının tutulduğu Ethereum ağı ve bu ağ ile iletişim kurmak için gerekli araçlardan oluşur. Bir Ethereum düğüm sunucusunun merkezi ağ ile iletişim kurmak, ağdaki güncellemeleri yerel sunucuya kopyalamak ve yerel güncellemeleri de ağa göndermek için bir istemci uygulamaya ihtiyacı vardır (Geth). Geth, hesap yönetimi ve madencilik fonksiyonlarının yürütülmesini sağlamaktadır. Diğer bir bileşen, Geth istemcisi ile Uzaktan Yordam Çağrısı (RPC) arabirimi aracılığıyla etkileşime izin veren web3.js kütüphanesidir. Ethereum hesaplar, anahtar ve adresler, işlem ve mesajlar, tokenlar, ethereum sanal makine(EVM) ve akıllı kontratların bütününden oluşmaktadır (Bashir, 2020).



Şekil 14. Ethereum genel ekosistemi (Bashir, 2020).

Ethereum ağında bireyler anonim olsa da işlemler herkes tarafından görülebilir. Akıllı kontratlarda iki tip hesap vardır. Bunlar kontrat ve sahibinin kendi hesabıdır. Ethereum sistemi akıllı kontratların ve DApps uygulamaların çalıştırılması için EVM yapısını kullanır. Kullanıcılar, Hizmet Reddi (DoS) gibi saldırıların engellenmesi ve düğümlere teşvik olarak EVM in yakıtı olarak kabul edilen EVM GAS'ı harcamalıdır. GAS, EVM bir iş çalıştırdığında ilgili işlemi yapan düğüme aktarılan bir çeşit kripto paradır (Wood, 2014).

Ethereum halk sağlığı, farmakoloji ve ilaç sektörlerindeki projelerde yaygın olarak kullanılan bir blokzincir ağıdır. Ayrıca genetik alanda da, bir önceki bölümde anlatılan Nebula, Zenome ve GeneCoin gibi hizmetlerin temelini, Ethereum akıllı kontrat yapısı oluşturmaktadır (Kuo ve diğerleri, 2019). Bu çalışmada önerilen modelin merkezinde de Ethereum bulunmaktadır. Sistemin hesap yönetimi, model paylaşımı, ödüllendirme ve onaylama mekanizmaları akıllı kontratlar üzerinde tasarlanmış. Genişleyebilir bir yapıda olması için genel ethereum ağının kullanılması düşünülmüştür.

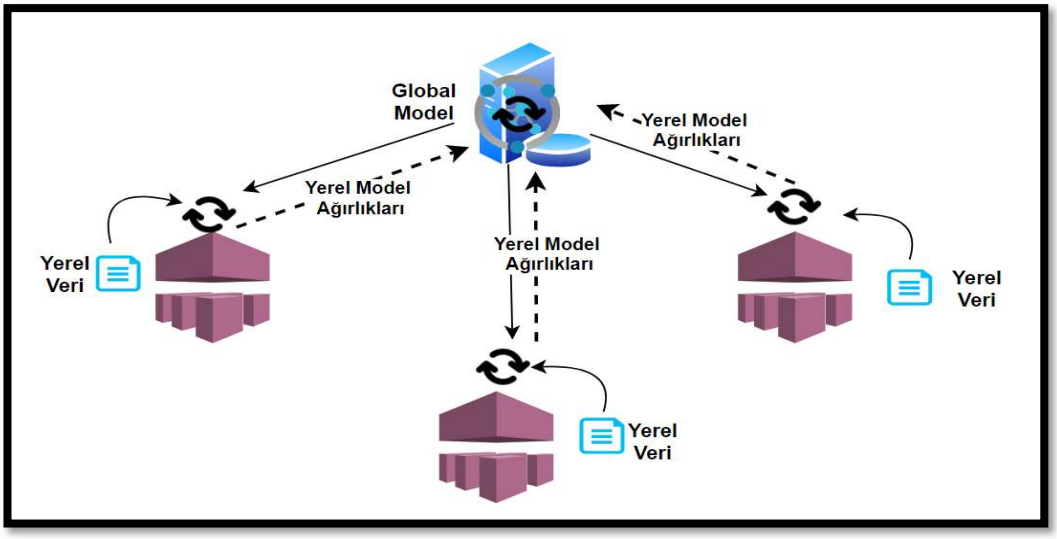
1.4.2 Federe Öğrenme (FL)

Bu bölümde FL ile ilgili temel kavramlar ve çalışma sisteminden bahsedilecektir. FL ilk defa mcMahan (2017) tarafından önerilmiş ve Google tarafından Android işletim

sistemli cihazlar üzerinde kullanılmıştır. Bir ML yöntemi olan FL, merkezi sunucular üzerinde geliştirilen modellerin uzak cihazlara gönderilip orada eğitilmesini ve eğitilen modellerin tekrardan merkezde toplanarak merkezi modelin güncellenmesini içeren adımları barındırır (McMahan ve diğerleri, 2017).

1.4.2.1 Federe Öğrenme Temelleri

Günümüzde kendi depolama ve işlemci donanımları olan cihazların kullanımları yaygınlaşmaktadır. Bu cihazlardaki kamera ve GPS gibi sensörler sayesinde kişisel içerikli birçok veri üretilmektedir. Kişisel cihazlarda üretilen veriler akıllı sistemlerin geliştirilmesi için fırsatlar sunsa da, hassas verilerin merkezi sistemlere alınarak işlenmesi oldukça risklidir. FL sayesinde merkezi sistemler üzerinde geliştirilen yüksek kaliteli ML modelleri uzak kullanıcı cihazlarında eğitilerek, veri güvenliğini riske atmadan yapay zekâ uygulamaları geliştirmek mümkün olmuştur (Li ve diğerleri, 2020). Şekil 15 de gösterildiği gibi FL merkezi sistem tarafından koordine edilen bir yaklaşım vardır. Geliştirilen ML modelleri merkezi sistem tarafında uzak sistemlere dağıtılır. Uzak sistemler, aldıkları modelleri kendi işlemcilerini ve yerel verilerini kullanarak eğitirler. Daha sonra eğitilen modellerin parametreleri merkezi sisteme gönderilir. Merkezi sistem topladığı parametreleri çeşitli formüllerle konsolide ederek modelin genelleştirilmesini sağlar. FL uygulamaları Google'ın geliştirdiği makine öğrenme kütüphanesi TensorFlow ile geliştirilebilir. TensorFlow, modellerin güncellenmesi ve güvenli bir şekilde genelleştirilmesi için FedAvg ile entegredir. FedAvg sayesinde toplanan parametreler, ortalama alma yöntemi kullanılarak, daha genel parametrelere dönüştürülür. Böylece her bir uzak sistem öğrenme işinin bir kısmını yapmış ve merkezi sistemde bu sistemi yönetmiş olur (McMahan ve diğerleri, 2017).



Şekil 15. FL sisteminin gösterimi.

Bu çalışmada, FL kullanarak blokzincir ağı üzerinde modellerin genelleştirilebildiği bir yöntem önerilmektedir. FL'in sadece akıllı cihazlarda değil, aynı zamanda farklı kurumların yerel sunucularını kullanarakta çalışma potansiyeli olduğundan bahsedilmişti. Biz de çalışmamızda bu potensiyeli genetik modellerin genelleştirilmesinde değerlendirdik.

1.4.2.2 Federe Öğrenme Yönteminin Zayıflıkları

FL, hala gelişmeye devam eden ve sorunları olan bir metottur. Bu sorunlar aşağıdaki gibi listelenebilir (Hou ve diğerleri, 2021).

a) *Tek noktadan hata alınması (single-point of failure):* FL merkezi sunucular tarafından koordine edilir. Bu sunuculara yapılan saldırılar, model parametrelerinin güncellenmesinde hatalara sebep olur ve model performanslarını düşürür (Bonawitz ve diğerleri, 2019). Bununla beraber aynı anda çok fazla güncellemenin merkezi sisteme iletilmesi sistemin aşırı yüklenmesine sebep olabilir (Yng ve diğerleri, 2020).

b) *Zehirleme saldırısı (poison attack)*: Sistemdeki katılımcılar bilinçli ya da bilinçsiz, merkezi modele bozuk parametreler göndererek ana

modelin başarısını bozabilir. FL'in, modelleri eğiten katılımcıların tespiti için bir yöntemi yoktur (J. Zhang ve diğerleri, 2019).

c) *Motivasyon eksikliği (lack of motivation)*: FL, her katılımcının gönüllü olarak verilerini ve donanım kaynaklarını kullanmasını bekler. Ancak bu pratik bir çözüm değildir. Bencil katılımcılar kendilerine finansal bir fayda getirmeyen modellerin eğitilmesi konusunda isteksiz davranabilir (Hou ve diğerleri, 2021).

d) *Gizlilik sızıntısı (privacy leakage)*: FL yönteminde veri eğitim kaynakları yerel olarak uzak sistemlerde tutulsa da bazı gizli bilgileri sızdırmak mümkün olabilir. Ayrıca kötü niyetli merkezi sistem Üretken Düşman Ağı (Generative Adversarial Network) yöntemi ile hassas bilgilerden yararlanabilir (Song ve diğerleri, 2017).

1.4.2.3 Federe Öğrenmenin Blokzincir ile Kullanımı

FL yönteminin bazı zaaflarını blokzincir ile çözmek mümkündür. Blokzincir, FL yönteminin merkezi sisteme olan bağımlılığını ortadan kaldırabilir. Blokzincir katılımcılar arasında merkeziyetsiz güvenli iletişim ve koordinasyon sağlar (Hou ve diğerleri, 2021). Tablo 5 blokzincir üzerinde kurgulanan FL yapılarının çözdüğü sorunları göstermektedir. Blokzincir sayesinde tek noktadan hata alınması, motivasyon eksikliği, gizlilik sızıntıları ve zehirleme saldırıları çözülebilmektedir.

Tablo 5. Farklı blokzincir FL yapılarının çözdüğü sorunlar (Hou ve diğerleri, 2021).

Architecture	Solved issues	Type of blockchain	Consensus mechanism
BlockFL [7]	Single point failure, lack of motivation	Public	PoW
FLChain [8]	Single point failure, lack of motivation	Public	–
FLChain [9]	Single point failure	Public	PBFT/PoW
RFL [10]	Poison attack, lack of motivation	Consortium	PBFT
DeepChain [11]	Single point failure, lack of motivation, privacy leakage	Public	Blockwise-BA
FedBC [12]	Single point failure, privacy leakage	–	–
BC-FL [13]	Single point failure, lack of motivation	Public	Pow
BlockFLA [14]	Single point failure, poison attack, lack of motivation	Public & private	PoW & PBFT
Chain FL [15]	Single point failure	Private	PoA
PSFL [16]	Poison attack	–	–
BFEL [17]	Single point failure, poison attack	Public & consortium	DPoS/PBFT & PoV
Biscotti [18]	Single point failure, poison attack, privacy leakage	public	PoF
VFChain [19]	Single point failure, poison attack	Public	Algorand
BLADE-FL [20]	Single point failure	Public	PoW
BFLC [21]	Single point failure, poison attack, lack of motivation	Public	Committee
VBFL [22]	Single point failure, poison attack, lack of motivation	Public	PoS

FL'in merkezi yapısı yerine blokzincir konumlandırılabilir. Yerel cihazlarda hesaplanan model parametreleri, blokzincir ağı üzerindeki düğümler arasında merkeziyetsiz bir şekilde transfer edilebilir. Madenci düğümler, özelleştirilmiş mutabakat protokolleri sayesinde, yerel cihazlardan toplanan model ağırlıklarının doğrulanmasını yapabilir. Son olarak güncellenen modeller dağıtık ve değiştirilemez şekilde blokzincir ağına eklenebilir (Kim ve diğerleri, 2020). FL modellerini bir blokzincir ağında kullanmak için özel bir blokzincir ağı kurmak zorunlu değildir. Bitcoin ve Ethereum ağları bu tarz mekanizmalar kurmak için uygundur. Sağlık kuruluşları, hastaların veri güvenliğini en üst seviyede tutarak araştırmalarda kullanılmasını destekler. Fakat sağlık alanında ender rastlanan ya da veri toplaması güç olan durumlarda sınırlı bir örneklem üzerinden araştırmalar yapılabilir. FL ve blokzincir sayesinde eğitilen modeller kurumlar arasında herhangi bir veri güvenlik sorununa sebep olmadan paylaşılabilir (Połap ve diğerleri, 2021).

1.5 ÇALIŞMANIN LİTERATÜR DEĞERLENDİRMESİ

Önceki bölümlerde genetik verinin ne olduğu ve neden hassas içerikli veri sınıfına girdiği açıklanmıştı. Genetik verinin kişisel veri sınıfına girdiğinden ve bu anlamda Avrupa, Amerika ve ülkemizde belirlenen yasal düzenlemelerden bahsedildi. Fakat verinin özgürce kullanılamaması bilimsel araştırmaların önünde önemli bir engel oluşturmaktadır. Bu çalışmada yasal engellerin de aşılarak, genetik verilerin daha özgür bir şekilde kullanılabileceği, YZ temelli akademik araştırmaların blokzincir altyapılı bir model üzerinde çalışılması hedeflendi. Bu kapsamda yaptığımız literatür taramalarından önerdiğimiz model tasarımı için kullanacağımız aşağıdaki çıkarımlar elde edildi.

Bölüm 1.2' de genetik mahremiyeti koruma yöntemleri olarak pertürbasyon, otantikasyon, diferansiyel gizlilik, kriptografik ve blokzincir temelli koruma ele alındı. Bu yöntemler içerisinden blokzincir temelli korumanın düşündüğümüz iş birlikçi araştırma modeli için kişisel veri güvenliğini sağlayan en iyi yaklaşım olduğuna karar verilmiştir. Çünkü blokzincir hassas veriye erişmek için erişim kontrolü, kriptografik veri güvenliği ve veri bütünlüğünü sağlayabileceğimiz bir alt yapıyı bizlere

sunmaktadır. Diğer yöntemler içerisinde pertürbasyon ve diferansiyel gizlilik veri bütünlüğünü bozmaktadır. Ayrıca veri gizliliği ile veri bütünlüğü arasında seçim yapılmasını zorlayan bir sistemdir. Otantikasyon ise merkezi bir sisteme bağımlılığı gerektirir. Merkezi sistemin güvenliğinin ihlal edilmesi durumunda, kişisel gizlilik sorunu ortaya çıkacaktır. Kriptografik yöntemler içerisinde en güvenilir yöntemlerden birisi ise HE tekniğidir. Bu yöntem oldukça güvenilir olsa da her tür çalışmaya uyarlamak güçtür.

Bölüm 1.4' de blokzincir teknolojisinin güvenli, merkeziyetsiz ve değiştirilemez olduğundan bahsedildi. Blokzincirin bu özellikleri sayesinde genetik alandan yürütülen çalışmalara Bölüm 1.3.3'de değindik. FoldingCoin gibi yapılar, yüksek işlemci gücü gerektiren proteinin üç boyutlu şeklini çıkarmak için blokzincir temelli bir çözüm önermiştir. Biz de çalışmamızda yüksek işlemci gücü gerektiren ML modelleri için benzer bir model üzerinde çalıştık. FoldingCoin tasarım olarak merkezi bir sunucu tarafından belirlenen işleri ağ katılımcıları üzerine dağıtıp işin paralelleştirilerek daha kısa zamanda tamamlanması amaçlanmıştır ve bu işlemleri yapan sunuculara da token bazlı ödül verilmesi sağlamıştır. Önerdiğimiz sistemde katılımcılar kendi modellerini geliştirerek blokzincir ağı üzerinde paylaşacak ve bu modelin eğitilmesine gönüllü katkı sağlamak isteyen diğerleri kendi yerel verilerini kullarak bu sürece destek olacaklardır. Benzer şekilde token bazlı ödüllendirme yapılarak emek karşılıksız kalmayacaktır. Bu bağlamda tasarladığımız sistem, iş paylaşımı için FoldingCoin' den farklı olarak, iş dağıtma yaklaşımı yerine iş talep etme yöntemi kullanır.

Blokzincir genetik veri paylaşımı için de fırsatlar sunmaktadır. Bu anlamda Bölüm 1.3.2' de bahsedilen Nebula, Zenome, EncrypGen gibi ticari sistemler kurulmuştur. Bu sistemler kişilerin kendi genetik bilgilerini ücreti karşılığında paylaşmalarını sağlar. Bahsedilen bu sistemlerin temel hedefi, kişilere kendi verilerinin sahipliğini vererek, kimler ile verilerini paylaşabileceklerinin özgürlüğünü vermektir. Bunu yaparken de mahremiyetin korunmasını ve dağıtık veri depolaması ile veri güvenliğini sağlamaktır. Çalışmamızda bu projelere yer vermemizin sebebi, ticari olarak blokzincir temelli genetik veri paylaşımının sağladığı imkanlardan esinlenerek, farklı bir yaklaşım ortaya koymaktır. Önerdiğimiz sistem bir veri paylaşım sistemi olarak tasarlanmamıştır. Bunun yerine ML model paylaşımını mümkün kılacak bir ortam

sağlanacaktır. Bu sayede geliştirilen modeller tekrar kullanılabilir şekilde blokzincir ağına kaydedilebilecektir. Modellerden faydalanmak isteyen ilaç, hastene ya da akademik kuruluşlar, modelleri blokzincir ağından çekerek kullanabileceklerdir. Model kullanımı ücretli olacak şekilde tasarlanmaktadır. Böylelikle model geliştirmek için emek harcayanlar, kendi ürünlerini bu sistem üzerinden pazarlayabilir olacaklardır. Bu sayede model gelişimine katkı sağlayan katılımcıların sistemde kalma motivasyonlarının artacağı düşünülmektedir. Modelleri sadece kullanmakla yetinenlerin ise bir çalışmayı baştan sona yapmalarındansa, önceden hazırlanan modelleri kullanarak zaman ve masraftan tasarruf etmeleri amaçlanmaktadır.

Blokzincir farklı tarafların bir arada iletişim sağlayabilecekleri bir ortam sunmaktadır. Fakat veri paylaşımı olmaksızın, ML modellerinin iş birliği içerisinde geliştirilebilmesi için bir yöntem belirlemek gerekmektedir. Bu yöntem FL yöntemidir. FL sayesinde yerel sistemlerde eğitilen modeller birleştirilerek daha doğru sonuçlar üreten modeller üretmek mümkündür. Literatür de bahsedildiği gibi FL' in bazı zayıflıkları vardır. Bu zayıflıklardan tek noktadan hata alma, motivasyon eksikliği ve gizlilik sızıntısı blokzincir çözümleri ile aşılabilmektedir. Ancak zehirleme saldırısına ayrı bir çözüm sağlamak gerekmektedir. Önerdiğimiz modelde bu zayıflığı aşmak için akıllı kontratlar üzerinde çalışan kodlara çeşitli kurallar eklenmesi düşünülmektedir. Bu sayede model başarısını bozabilecek durumların önüne geçmeye çalışılacaktır.

Blokzincir ve YZ' nın beraber kullanıldığı çalışmalar oldukça yenidir. Bu birliktelik genelde FL ile mümkün olmuştur. Rahman ve diğerleri (2020) çalışmasında IoMT cihazlarının bazı hastalıkların tahmin edilmesi için FL ile blokzincirin beraber kullanılabileceğini önermiştir. Önerilen bu yaklaşımdan esinlenerek, IoMT cihazları üzerinde kurgulanan bu sistemin daha büyük kurumsal sunucularda da uygulanabilir olduğu düşünülmüştür.

Tasarlanan sistemde, FL' in merkezi sunucu ihtiyacının blokzincir ağı ile merkeziyetsiz olarak karşılanması düşünülmüştür. Model eğitimi ve model birleştirme süreçleri için FL yöntemleri kullanılacaktır. Ethereum ağı, akıllı kontratları sayesinde Dapps uygulamalarını da çalıştırabildiği için tercihimiz olmuştur. Çünkü modellerin birleştirilmesini merkeziyetsiz olarak yapabilmek için akıllı kontratlar kullanışlıdır.

Blokzincir ağı üzerinde model birleştirme, taraflar arasında koordinasyon ve token transferleri gibi işlemlerin yapılması düşünülmektedir. Ayrıca modellerle ilgili güncellemeler de blokzincir ağında tutulacaktır. Fakat daha büyük dosyalar ve veriler için blokzincir ağları çok uygun değildir. Bu amaçla modellerin kodlarını ve detaylarını IPFS sistemi üzerinde tutmanın daha uygun olduğuna karar verilmiştir. Böyle bir sistem kurmak için gerekli temel yapıların detaylarına Bölüm 1.4.1 de yer verilmiştir. Bir sonraki bölümde genetik veri gibi yüksek hacimli kişisel veri özelliği taşıyan verilerin, merkezi sistemlerde toplanmasına gerek kalmadan, genetik veri modellerinin geliştirildiği ve paydaşlar arasında iş birliğinin sağlandığı bir kavramsal tasarım sunulmuştur.

BÖLÜM II

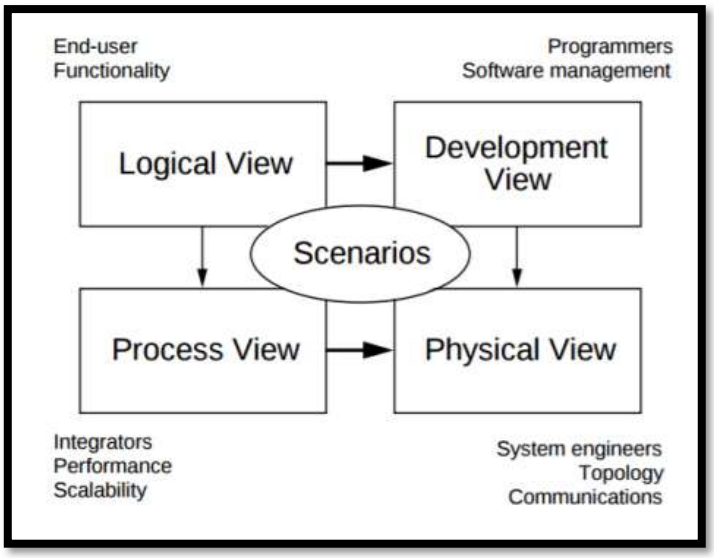
Bu bölümde çalışmanın yöntemine değinilmiştir. Önerdiğimiz kavramsal tasarımın modellenmesi için izlediğimiz yolun detayları açıklanmıştır.

2. YAZILIM MİMARİSİNİN 4+1 GÖRÜNÜMÜ

Açık bilim yaklaşımı ile yürütülen bilimsel çalışmalar, farklı bilim insanları tarafından erişilebilir ve çıktılarının kullanılabilir olması, bilimsel araştırmaların hızlı ilerlemesini sağlamaktadır. MIT üniversitesinin hayata geçirdiği Kipoi (Avsec ve diğerleri, 2019) platformu, genomik eğitimli modellerin paylaşımını ve yeniden kullanımını teşvik eden bir açık bilim girişimidir. Platform üzerinde, genetik veri kullanarak geliştirilen ML modellerinin paylaşımı yapılmaktadır. Bu platform, kromatin erişilebilirliği tahmini, transkripsiyon faktörü bağlanması ve DNA dizisinden alternatif ekleme gibi genetikteki önemli tahmin görevlerini kapsayan birçok model barındırmaktadır. Platform tekrar kullanılabilirliği sağladığı için akademik araştırmalarda vakit tasarrufu sağlamaktadır. Bu tezde ise Kipoi yaklaşımına alternatif bir öneri getirilerek, blokzincir ve FL in beraber kullanımı sayesinde geliştirilen modellerin eğitim süreçleri devam ettirilebildiği, katılımcıların teşvik kazanabildiği ve kişisel veri güvenliğinin ön planda tutulduğu taraflar arası bir iş birliği konsepti önerilmektedir. Önerilen bu yapının dizaynı yazılım mimarisinin 4+1 görünümü baz alınarak modellenmiştir.

Yazılım mimarisi bir yazılımın üst seviyede dizayn ve hayata geçirilmesi ile ilgili yapıyı ortaya koyar. Yazılım mimarisini açıklamak için birden çok görünüm ve bakış açısından oluşan modeller kullanılır. Bu çalışmanın mimari tasarımında kullanılan "Yazılım Mimarisinin 4+1 Görünümü" Şekil 16' da resmedilmiştir. Mimari tasarım mantıksal, geliştirme, süreç ve fiziksel görünüm olarak ele alınmıştır (Kruchten, 1995). Mantıksal görünüm, tasarımın nesne modelini ortaya koyar ve nesneye yönelik tasarım yöntemi kullanılır. Süreç görünümü, yazılımın senkron ve asenkron faaliyetlerini gösteren tasarımı içerir. Fiziksel görünüm, yazılımın fiziksel donanımlar üzerinde nasıl yayıldığını açıklayan dizaynı gösterir. Geliştirme görünümü, yazılımın

statik organizasyonunu geliştirme ortamı üzerinde anlatır. Ancak bir yazılımın mimarisinin olmazsa olmaz bir parçası daha vardır. Bu parça senaryolar ya da kullanım durumlarıdır (use cases).



Şekil 16. Yazılım mimarisinin 4+1 görünümü (Kruchten, 1995).

Nesne yönelimli gösterimlerin ve yöntemlerin çoğalmasıyla birlikte, Birleşik Modelleme Dili (UML), nesne yönelimli modelleri tanımlamak için, standartlaştırılmış bir gösterim sağlamak için ortaya çıkmıştır (Gomaa, 2006). Bizde bu çalışmada temel bileşenler arasındaki etkileşimi UML notasyonu ile açıklamaya çalışacağız.

BÖLÜM III

Tezin bu bölümünde, iş birlikçi genetik ML modelleri geliştirmek için önerilen blokzincir temelli FL sisteminin dizayn ve tasarımına yer verilecektir.

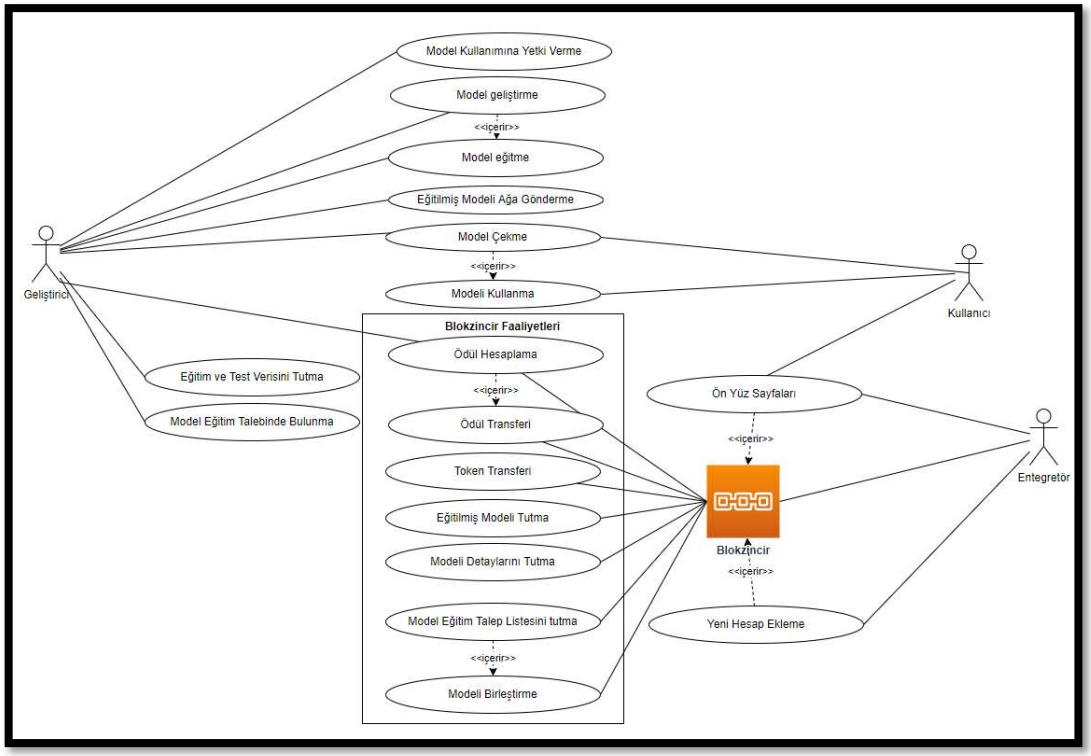
3. BLOKZİNCİR TABANLI GENOMİK VERİ MODELİ PAYLAŞIMI PLATFORMU KAVRAMSAL TASARIM

Bu bölümde önerdiğimiz sistemin kavramsal tasarımı 2. Bölüm' de anlatılan "Yazılım Mimarisinin 4+ Görünümü" metodu ile açıklanacaktır. Sırasıyla kullanım seneryoları, mantıksal tasarım geliştirme tasarımı ve süreç tasarımı anlatılacaktır. Fakat fiziksel tasarıma bu çalışmada yer verilmemiştir. Hedeflenen kullanım miktarları ve kapasite planlama çalışmaları netleşmeden bir fiziksel tasarımı ortaya koymanın uygun olmayacağı düşünülmüştür.

3.1 SENARYOLAR

Bu bölümde senaryolar, UML diyagramı ile tasarladığımız sistem üzerindeki faaliyetler açıklanacaktır.

Yapımızda temel olarak dört farklı tip aktör varıdır. Bunlar geliştiriciler, kullanıcılar, entegretörler ve blokzincir olarak sınıflandırılabilir. Geliştiriciler sistem üzerinde yeni makine öğrenme modelleri geliştirenler ve bu modellerin eğitilmesini sağlayan aktörlerdir. Bu aktörler blokzincir ağı üzerinde bir araya getirilerek, bunların iş birliği içerisinde ürün geliştirebilmeleri sağlanacaktır. Geliştiriciler aldıkları sorumluluklara göre farklı rollere sahip olabilmektedir. Kullanıcılar model gelişimine destek vermeden, geliştirilmiş modelleri ihtiyaçları doğrultusunda kullanabileceklerdir. Kullanıcılar bu sistemin müşterisi ya da tüketicisi gibi düşünülebilir. Sistemdeki tüm aktörlerin bir araya getirilmesi ve arka plandaki karmaşıklığı basitleştirmek için entegre edicilerin olması gerekmektedir. Entegretörler bu sistemin geliştiricileridirler. Aktörlerin sistem üzerindeki kabiliyetleri Şekil 17 ile açıklanmaya çalışılmıştır.



Şekil 17. Senaryolar

Tablo 6’ da detaylandırıldığı gibi sistemde yer alacak aktörler listelenmiştir. Bu aktörlerin yetenekleri, sistem üzerinde yapabilecekleri faaliyetleri ifade etmektedir. Geliştiriciler bu sistem üzerinde hem geliştirici hem de son kullanıcı gibi davranırlar. Blokzincir ağı canlı bir aktör olmasa da kendi başına bazı faaliyetler yürüttüğü için bu tabloya eklenmiştir. Direk ağ üzerinde koşulacak işlemleri yürütür. Entegretörler merkeziyetsiz yapının bir parçası değillerdir. Kullanıcılar ve geliştiriciler direk ağa bağlanarak faaliyetlerini yürütebilirler ama entegratörler bu görevi alarak sistemin daha rahat kullanılmasını sağlayacaklardır.

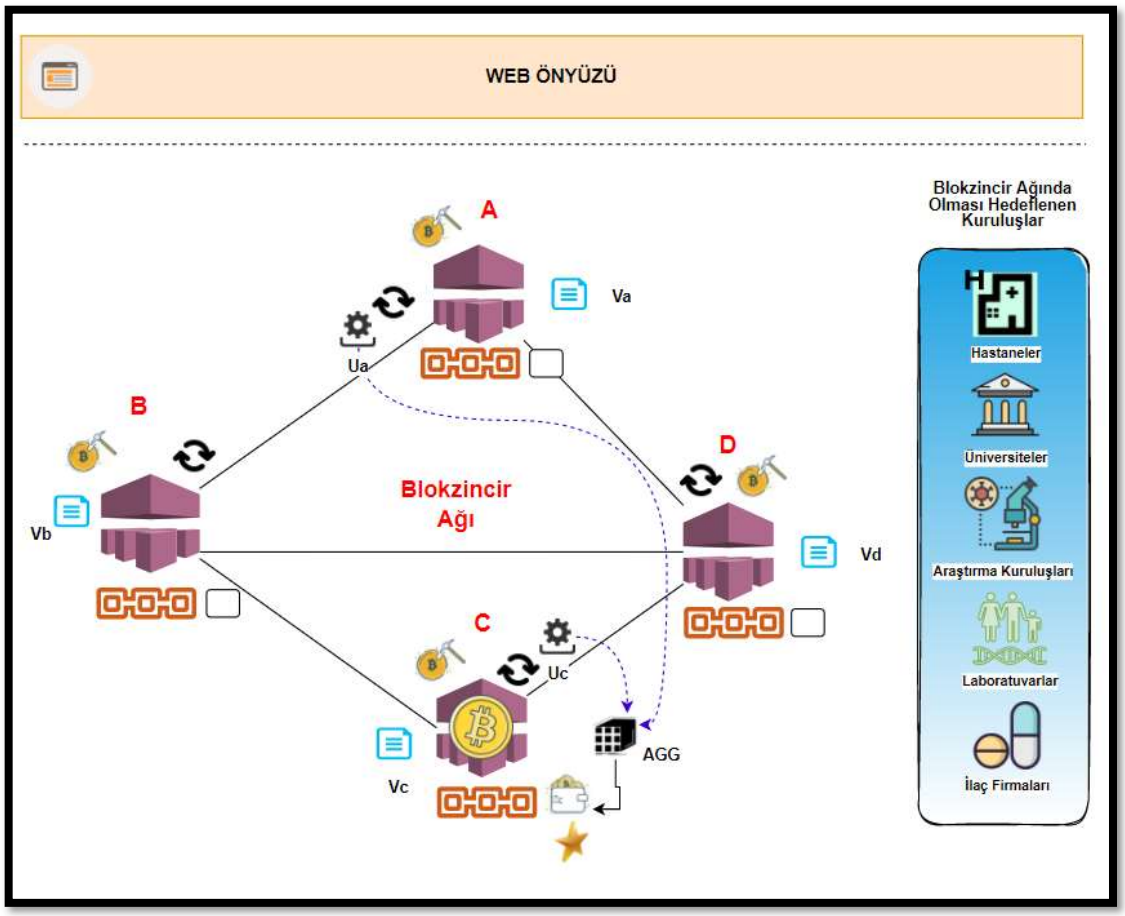
Tablo 6. Model için önerilen aktör-yetenek listesi.

Aktör	Yetenek
Geliştirici	Makine öğrenmesi modellerini geliştirme ve paylaşabilme.
	Başkasının oluşturduğu modelleri kullanabilme.
	Modelleri yerel veri kümeleri ile yeniden eğitebilme.
	Sahip olduğu modellerin kullanımı ile ilgili yetkilendirme yapabilme.
	Emeğinin karşılığında bir teşvik kazanabilme.
	Ara yüz servislerini kullanabilme.
Kullanıcı	Başkasının oluşturduğu modelleri teşvik ücreti karşılığında kullanabilme.
	Bir ara yüz üzerinden modellerle ilgili bilgilere ulaşma ve sistem üzerinde yetki talebi gibi temel işlemleri yapabilme.
Entegretör	Son kullanıcılar ile blokzincir ağı arasındaki bağlantıyı kuran uygulamalar geliştirme.
	Blokzincir ağı üzerindeki işlemleri yönetebilme.
	Yeni kullanıcı ve model girişlerinin yapılabilmesi için ara yüzler sunma.
Blokzincir	Eğitilmiş model ağırlıklarının birleştirilmesi.
	Dağıtılacak ödülün hesaplanması.
	Ödül ve token ücretlerinin transfer edilmesi.
	Modeller üzerinde yapılan işlemlerin kayıt altına alınması.

3.2 MANTIKSAL TASARIM

Mantıksal tasarım bölümünde genel yapının hangi temeller üzerinde kurgulanacağı nesne temelli bir yaklaşımla açıklanacaktır. Sistem ön yüz uygulamaları ve arka planda çalışacak uygulamalardan oluşacaktır. Ön yüz, sistemin kullanıcıları için kullanıcı dostu bir ekran sağlamaktadır. Arka planda çalışan karmaşık blokzincir ağı, modeller, depolama araçları üzerindeki işlemleri basitleştiren bir araç olacaktır. Blokzincir ağı

üzerinde akıllı kontratlar ve işlemler olacaktır. Ayrıca merkeziyetsiz veri depolama sistemleri (IPFS), sistemin bir parçası olarak tasarlanmıştır.



Şekil 18. Önerilen modelin mantıksal tasarımı.

Şekil 18 de gösterildiği gibi, önerdiğimiz blokzincir ağında, genetik bilgi üzerinde araştırma yapması muhtemel hastane, üniversite, araştırma kuruluşları, laboratuvarlar, ilaç firmaları gibi kurumsal yapılar ve bunların yanında halka açık ağda bulunan ve para kazanma amaçlı akıllı kontratları çalıştıran düğümler de sistemin bir parçası olacaktır. Ayrıca bu sistemin son kullanıcılar tarafından rahat kullanılabilmesi için ön yüz bulunacaktır.

Genel kurgu blokzincir ağı üzerinde tasarlandığı için olası sahte modellerin paylaşılması ya da sahte veri ile modellerin eğitilerek faydanın yok edilme riski(poisining) vardır. Bu riskleri yetkilendirme ile azaltmak mümkündür. Bu amaçla model paylaşımı yapacak katılımcılar için bir onay mekanizması kurgulanmalıdır.

Ağda model geliştirenler, modelleri eğitenler, modelleri kullananlar ve bu ağ içindeki akıllı sözleşmeleri çalıştırarak GAS ücreti kazanmak isteyen katılımcılar sistemin paydaşlarıdır.

Model geliştirenler ve modelleri eğitenler için özel yetkilendirmenin yönetildiği akıllı sözleşme tasarlanmıştır. Modeli sadece ağdan çekerek kullanmak için teşvik ücreti ödenmesi gerekecek ve bu teşvik ücreti model geliştiren ile eğitenler arasında paylaştırılacaktır. Buradaki ücret transferleri ve teşvik ücretinin dağıtılması için akıllı kontratlar kullanılacaktır. Bu kapsamda sistemde aşağıdaki rollere ve işlemlere gerek olacağı düşünülmüştür.

Roller:

a) *Model geliştirenler*: Bu tip katılımcıların yeni model geliştirerek ağda paylaşım hakları vardır. Bu rolü alabilmek için ağda bu statüye hali hazırda sahip olan diğer katılımcıların onayı gerekmektedir.
b) *Model Eğiticileri:* Bu katılımcılar yeni model geliştirmeseler bile, ağdaki mevcut modelleri kendi yerel sistemlerine çekerek, kendi veri kümeleri ile modeli eğitebilirler. Eğitilen modelleri ağa geri gönderirlerse teşvik ücreti kazanırlar. Eğitilen modelleri ağa geri göndermek için özel bir yetki gerekir ve diğer katılımcıların onayı gerekmektedir.
c) *Model Kullananlar*: Sistem üzerinde modelleri sadece çekip, kendi verileri ile test yapmak isteyen gruptur. Ağdan modelleri çekmenin bir ücreti olacaktır. Bu ücret, ağa katkı sağlayan diğer katılımcılara dağıtılacaktır. Özel bir yetkilendirme gerekmemektedir.
d) *Sözleşme Çalıştıranlar:* Ağda akıllı sözleşmeleri çalıştırarak, GAS ücreti kazanmak isteyen gruptur. Yetkilendirme sözleşmesinde bulunmazlar. Kullanılmak istenen Ethereum ağının kendi parçalarıdır. Sisteme donanım kaynağı sağlamaktadırlar.

Tasarladığımız model üzerindeki işlemler:

a) *Model Geliştirme:* Her katılımcı yerel sistemlerinde modellerini geliştirir ve eğitir. Modeller FL prensiplerine göre geliştirilmelidir.

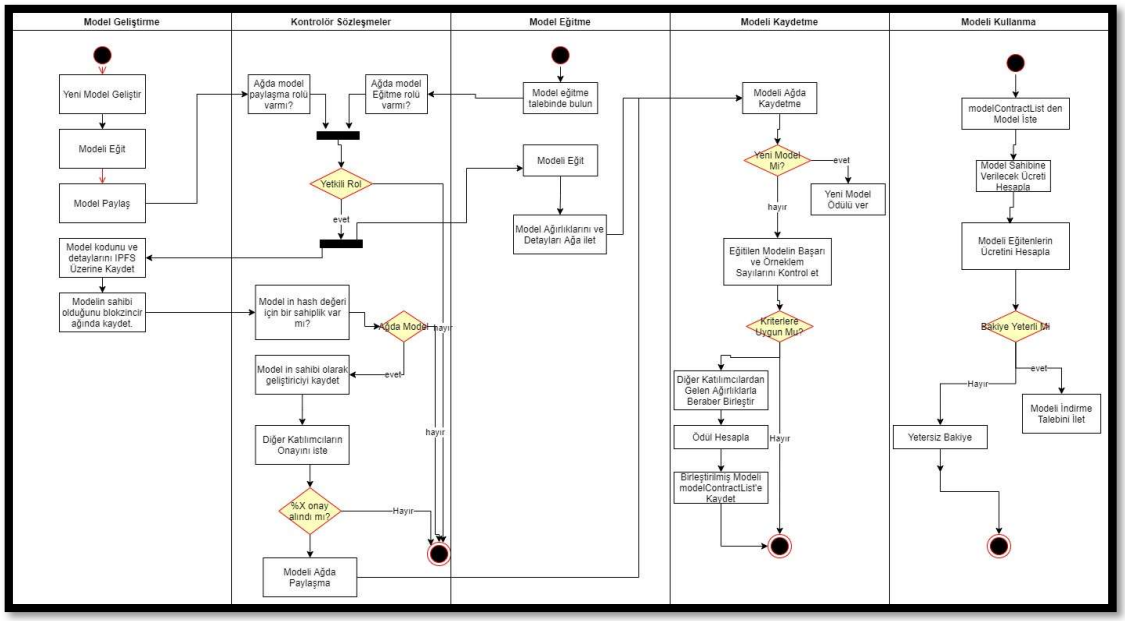
b) *Modellerin Sisteme Kaydedilmesi:* Geliştirilen modellerin kaynak kodları IPFS sistemi üzerinde kaydedilir.

c) *Model Paylaşımı:* Tasarlanan modeller ağda paylaşılır. Model paylaşımı bir akıllı kontrat (*modelContractList*) üzerinden yapılır. Kontrat modelin sahibini, amacını, model kodunu tutan IPFS adresini, verinin formatını, ne zaman eklendiği ve yerel sistem üzerindeki doğruluğu gibi bilgileri gösteren verileri içermelidir. Paylaşılan modellerle ilgili bilgiler ön yüz sayfalarında görülebilecektir.

d) *Model Çekme:* Son kullanıcılar *ModelContractList* üzerinden modeli kendi sistemlerine indirebilirler. Model çekilirken belirlenen teşvik ücretini model sahibine iletmek için gerekli işlemler tetiklenir.

e) *Model Kullanımı için Ödeme: ModelContractList* üzerindeki bir fonksiyon aracılığı ile ödeme gerçekleştirilir.

f) *Modellerin Eğitimi:* Modeller eğitici rolüne sahip katılımcılar tarafından eğitilir. Eğitilmiş modellerin ağırlıkları, blokzincir üzerindeki *ModelAggregator* kontratına gönderilir. Bu kontrat, aynı modele ait diğer düğümlerden gelen tüm ağırlıkları alarak birleştirir.



Şekil 19. Önerilen modelin aktivite Diyagramı.

Şekil 19 tasarımın aktivite diagramını göstermekte olup daha büyük görseli EK-1' de de bulunmaktadır. Bu diyagramda, blokzincir ağı üzerindeki işlemler ve birbirleri ile olan ilişkileri görsel olarak açıklanmıştır. Sistemde model geliştiriciler, ürettikleri kodları hash algoritmasından geçirip imzalayarak, modelin sahibi olduklarını garanti altına alırlar. Kodların kapladığı boyut, blokzincir için fazla büyük olabilir. Bundan dolayı kodların bir IPFS sistemi üzerinde tutulması planlanmıştır. IPFS, merkeziyetsiz dosya dağıtım sistemidir. P2P protokol üzerinden dosya paylaşımı yapar. Her dosyanın içerisindeki bilgiye göre benzersiz bir hash değeri oluşturulur. Bu hash değeri dosyanın adı olur ve bu değer üzerinden dosya erişim talebi yapılır (Huang, Chang ve Wu, 2020).

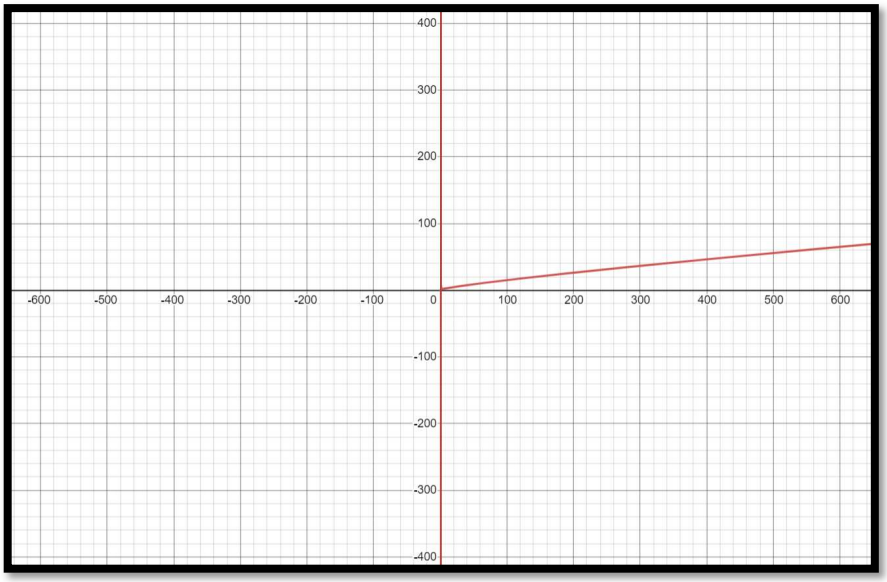
3.2.1 Model Geliştirme ve Ödüllendirme

Yeni modeller sisteme yazılmadan önce çeşitli kontrollerden geçirilir. Öncelikle model geliştiricisi ağ üzerinde yeni model geliştirme rolüne sahip mi diye bakılır. Eğer bu role sahipse model dosyasını IPFS sistemine kaydedebilir ve kaydettiği dosyanın hash değeri ile modelin kodunu paylaşabilir. Yeni modellerin, blokzincir sistemindeki kontrat yapısına girebilmesi için, ağda doğrulayıcı rolüne sahip belirli bir sayıdaki katılımcı tarafından onaylanması gerekmektedir. Buradaki sayı aşağıdaki formül ile belirlenecektir.

Denklem 3. Doğrulayıcı sayısının hesaplanma yöntemi

$$\boldsymbol{X \geq 2 \; için}$$

$$\boldsymbol{v = X * 1/\log_2 X}$$



Şekil 20. $v = X * 1/\log_2 X$ grafiği

Denklem 3' e göre v, talep edilen rolün ağ üzerinde kaç onaylayıcı düğüm tarafından onaylanması gerektiğidir. X ise, ağda mevcutta bulunan onaylayıcı sayısıdır. Ağ üzerinde model geliştirici onayı alanlar, aynı zamanda onaylayıcı rolünü de almış olurlar. Formüldeki logaritmik ifadenin kullanılma nedeni, onaylayıcı sayısı arttıkça yüzdesel olarak etkinin yavaş yavaş düşürülmesini sağlamaktır. Formülde logaritma kullanıldığı için X in en az 2 olması gerekir. Sonrasında X sayısı arttıkça onaylayıcı sayısı aşağıdaki gibi yavaş yavaş artacaktır.

$$X = 2 \; için \; v = 2$$
$$X = 256 \; için \; v = 32$$
$$X = 1024 \; için \; v = 102$$
$$X = 32768 \; için \; v = 2184$$

Fakat performans için onaylayıcı sayısının sürekli artması, yeni bir modelin ağa dahil edilmesi süresini arttıracağı için, X değerine bir limit konulması uygundur. Bu limit ise bizim önerdiğimiz ağda onaylayıcı sayısının 10.000'i geçmeyeceği düşünülerek, bu sayı limit olarak kullanılacaktır. Bu durumda ağdaki sayı ne kadar artarsa artsın bu limitten sonra bir rolün onaylanması için $v = 753$ onaylayıcı gerektirecektir.

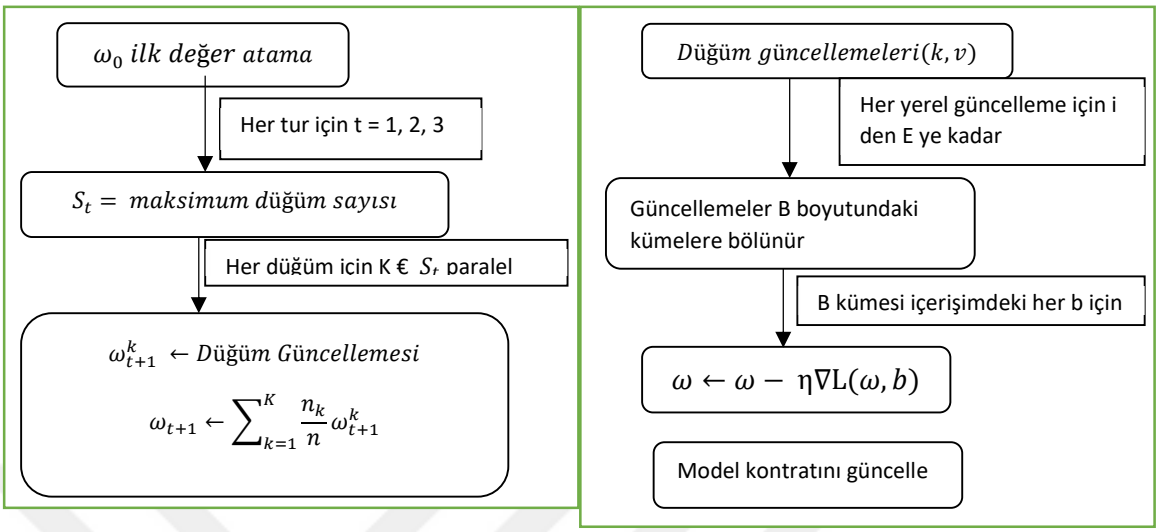
Model eğiticiler, model kodlarını geliştiricilerin onayı ile IPFS sisteminden alabilir. Bunu yapabilmek için öncelikle onay kontrolü yapılmalıdır. Eğer eğiticinin gerekli rolü varsa, kendi veri kümesi üzerinde modeli eğiterek, eğitilen modelin ağırlıklarını model kontratına gönderir.

3.2.2 Model Eğitme ve Ödüllendirme

Model kontratı öncelikle, gelen ağırlıklarının yeni bir model mi yoksa ağda bulunan bir modelin eğitilmiş ağırlıkları mı diye kontrol eder. Eğer mevcut bir model eğitilmiş ise kontrata gönderilen örneklem sayılarının, doğruluk değerlerinin, hassaslık gibi parametrelerin belirlenen kriterlere uygun olup olmadığını kontrol eder. Kriterler yeterli ise diğer düğümlerden gelen ağırlıklar da alınarak model birleştirilir. Model birleştirme işlemi FL yöntemleri ile yapılır. FedAvg algoritması model birleştirmede kullanılan bir yöntemdir. Bu algoritma yaygın kullanılan yöntemlerden biridir. Fakat performans sorunları bulunmaktadır. Bu konuda alternatif yöntemler de önerilmiştir (Jiang ve diğerleri, 2020). Biz bu çalışmada genel konsept çalışıldığımız için FedAvg algoritmasını referans alacağız.

Birçok ML modeli, genelleştirme yapma optimizasyonu için stokastik gradyan inişi (SGD) algoritmasını kullanır. Bu algoritma, büyük örneklem kümelerinde standart gradyan inişinden daha iyi performans gösterir. FL yöntemlerinde de bu algoritma tercih edilir. Bu yöntemin performanslı çalışması için birçok kez örneklem kümelerinde tekrarlanması gerekir. FL bu yöntemin her bir düğümde uygulanması ve sonrasında da evrensel model üzerinde güncellenmesini sağlar. Sabit öğrenme oranı η için ve k düğümünde $g_k = \nabla F_k(\omega_t)$ hesaplanır. Her bir düğümdeki modelin ortalama gradyanları ω_t merkezi sunucuda birleştirilerek ana model $\omega_{t+1} \leftarrow \omega_t - \omega g_k$ yani $\omega_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \omega_{t+1}^k$. Böylece her düğüm yerel verilerini kullanarak mevcut model üzerinde bir adım gradyan inişi sağlar. Merkezi sunucu, bizim modelimizde blokzincir üzerindeki akıllı kontrat, her modelden gelen gradyanların ağırlıklı ortalamasını alır. Bu yönteme "Birleşik Öğrenme Ortalamaları (FederatedAveraging yada FedAvg)"

denilmektedir (McMahan ve diğerleri, 2017). Yukarıdaki açıklamaya göre algoritmanın akış çizelgesi Şekil 21' de ki gibi olacaktır (Jiang ve diğerleri, 2020).



Şekil 21. Önerilen modelin FedAvg algoritması

Bu işlem sonrasında eğiticiye token bazlı bir ödül verilir. Bu ödül hem yeni model geliştirenlere hem de model eğiticilerine verilir. Ethereum ağında iki tip token bulunur; takas edilebilir (fungible tokens) ve takas edilemez tokenlar (non-fungible tokens - NFT). Bu çalışmada bahsedilen tokenlar ERC-20 standartlarına göre hazırlanmış takas edilebilir tokenlardır. Bu tip tokenlar para gibi alış-verişlerde kullanılabilir ve küçük parçalara ayrılabilir (Schär, 2021). Yeni model geliştirenler için, model kontrat listesine yazılan yeni modeller karşılığında, ilk başta sabit bir ödül verilir. Fakat sonraki yayınlanan modeller için ilgili düğümün ya da katılımcının etkisine göre bir ödül verilmesi planlanmıştır. Bu tarz bir ödül sistemi için, her katılımcı için, bir etki skoru hesaplanır ve bu bilgi ağ üzerindeki kontratlarda tutulur. Etki skoru Denklem 4' de ki gibi yapılması planlanmaktadır. M, bir katılımcı için ağda bulunan modellerin sayısıdır, m her bir modelin kaç kişi tarafından kullanıldığını ve t ise modellerin kaç eğitmen tarafından eğitildiğini gösterir. Bu formül ile modellerin ağa marjinal etkisinin ne olduğu hesaplanmıştır.

Denklem 4. Modellerin Ağa Marjinal Katkısı

$$es = \frac{1}{M} \sum_{i=0}^{M} (m_i + t_i) \times 100$$

Hesaplanan etki skoru(es) ilgili izinler alındıktan sonra model kontratına yazılırken hesaplanır ve geliştiricinin bakiyesine yüklenir.

Bir diğer ödül verilmesi planlanan konu ise model eğiticilerinin harcadıkları efordan dolayı bir fayda sağlamalarıdır. Bu ödül mekanizması yeni model geliştirenler için düşünülen ödüllendirme sistemine görece daha zor bir sorundur. Bu ödül verilirken, merkezi modele ne kadar katkıda bulunulduğu hesaplanarak ödülün hesaplanması uygun olur. Shapley değeri tam olarak bu işe yaramaktadır. Bir iş birliğinde katılımcıların hangi rolü üstlendiklerini ve iş birliğinde ne kadar fayda sağlandığını gösteren bir değerdir (Nagalapatti ve Narayanam, 2021). Fakat bu değer de katılımcıların rollerine göre etkilerinin farklılaşacağı da hesaba katılır. Bizim kurgulamak istediğimiz yapı buradan esinlenilerek daha basit bir yöntem üzerinde tasarlanmıştır. Önerdiğimiz modelde, her bir 'model eğiticinin' merkezi model başarısında ne kadar etkili olduğunu hesaplayarak, bir ödül mekanizması kurmaya çalıştık. Blokzincir ağında bireysel katkıyı hesaplamak oldukça maliyetli olabilir. Bundan dolayı model eğiticilerinin merkezi modele olan katkılarını (Gk) kendilerinin hesaplaması beklenmektedir. K eğitici sayısı olmak üzere, her bir katılımcının modeli ne kadar geliştirdiği (Gk), etki skoru (es), yerel data miktarları (Dl) ve aynı model için bir önceki seferde toplam ne kadar ödül dağıtıldığı (Pin) dikkate alınarak aşağıdaki Denklem 5' de ki formül ile ödül verilmesi planlanmıştır.

Denklem 5. Ödüllendirme Formülü

$$i = es \times \frac{G_k}{\sum_{k=1}^{K} G_k} \times \frac{Dl_l}{\sum_{k=1}^{K} Dl_k} + \frac{Pin}{K}$$

Formülde katılımcıların daha fazla sisteme hizmet etmeleri için bazı çarpanlar kullanılmıştır. Etki skoru bir çarpan olarak şimdiye kadar sisteme hizmet etmiş katılımcıların daha fazla token kazanması için eklenmiştir. Bir diğer çarpan ise model eğitimi için kullanılan veri miktarıdır ($\frac{Dl_l}{\sum_{k=1}^{K} Dl_k}$). Katılımcıların bir model için kullandıkları verilerin miktarını arttırmasını teşvik etmek için bu çarpan eklenmiştir. Son olarak da formülde, bir önceki seferde dağıtılan ödülün katılımcılara pay edilerek yeniden verilmesi sağlanmıştır. Bunun amacı ise modellerin bir seviyeden sonra performanslarının arttırılması zorlaşacaktır. Katılımcıların mevcut modellerin

performanslarını iyileştirmeye devam etmeleri için ödül artarak verilmeye devam edilecektir.

Model eğiticilerin ağa gönderdikleri eğitilmiş modellerin ayrıca doğrulanması gerekmektedir. Bölüm 1.4.2.2 de anlatılan FL yönteminin "*zehirlenme saldırısı*" karşısında olan zayıflığı sebebi ile blokzincir üzerinde bir kontrolün yapılması elzemdir. Bizim belirlediğimiz algoritmada, yerel eğitim ve test kümesinin, IPFS üzerindeki model kataloğunda belirtilen minimum değerden büyük olması ve G fayda skorunun pozitif değerde olması kontrol edilmiştir. Kısaca bu yapı aşağıdaki algoritma ile çalışacaktır;

Tablo 7. Model eğitme ve ödüllendirme algoritması.

1. *Modeli blokzincir üzerinden yerel sunucuya indir.*
2. *Modeli yerel test kümesi (**Dt**) üzerinde test et.*
3. *Modelin performansını (doğruluk, kesinlik) hesapla (**Pm**).*
4. *Modeli yerel eğitim örneklemi (**Dl**) ile yeniden eğit.*
5. *Eğitilmiş yeni modeli aynı test kümesi üzerindeki performansını hesapla (**Pl**).*
6. *Testler arasındaki farkı alarak faydayı hesapla. **G= Pl-Pm***
7. ***Dl, Dt, Pm, Pl, G ve es** değerlerini blokzincir ağına gönder.*
8. *Doğrulama fonksiyonunu çalıştır*
    a. ***Dl > minD** (IPFS üzerinde model kataloğunda model geliştiren tarafından belirlenen en az ne kadar veri ile modelin eğitilmesi gerektiğini gösteren değer).*
    b. ***Dt>minT** (IPFS üzerinde model kataloğunda model geliştiren tarafından belirlenen en az ne kadar veri ile modelin test edilmesi gerektiğini gösteren değer).*
    c. ***G>0***
9. ***Eğer** geliştirilen model doğrulamadan geçer ise*
    a. *Model ağırlıklarını kaydet.*
    b. *Tüm katılımcılardan gelen değerleri bir listeye yaz.*
    c. *Model birleştirme işlemini yap. (FedAvg)*

3.2.3 Model Kullanma

Kurguladığımız yapıda, ağda paylaşılan hazır modelleri kullanmak isteyenler için de bir rol tanımlanmıştır. Bu kullanıcılar herhangi bir model geliştirmezler ve eğitmezler. Sadece eğitilmiş modelleri ağdan çeker ve kendi verilerini modele vererek sonuçlarını alırlar. Bir anlamda sistemin müşterisi gibi konumlandırılacaktır. Ağdan aldıkları bu hizmetin karşılığında ise modeli kullanma bedelini sisteme ödemelidirler. Ödemeler modelin kullanım ücretine göre *ModelContractList* hesabına yapılacaktır. Bölüm 1.4.1.8' de belirtildiği gibi Ethereum ağında kontratların kendine ait bir hesap numaraları ve bakiyeleri mevcuttur. *ModelContractList* sistemde merkezi bir görev alarak ödül dağıtma ve model kullanma için gerekli ücret transfer işlemlerini de yönetecektir. Modellerin kullanım ücretleri ise geliştirme maliyetleri referans alınarak hesaplanacaktır. Geliştirme maliyeti, geliştiriciye verilen ödül ve her eğitimden sonra dağıtılan toplam ödül miktarları üzerinden hesaplanır. Bir ürünün fiyatının belirlenmesinde bir çok farklı parametrenin ve piyasa koşullarının etkisi vardır (Simon ve Fassnacht, 2019) . Fakat biz karmaşık hesaplara girmeden, dağıtılan toplam teşvik ücreti üzerinden Denklem 6'yı kullanarak bir fiyat belirleyeceğiz. Denklem 6 temsili bir formül olup bu hesaplama farklılaştırılabilir.

Denklem 6. Model kullanım ücreti formülü

$$Ei = \sum_{e=0}^{E} I_e$$

$$P = \frac{Ei}{T}$$

My=Yeni model ödülü

Ei=Eğitme ödülleri toplamı

e = Her bir eğitim çalışması

I = Her bir eğitimde ne kadar ödül verildiği

E = toplam eğitim sayısıdır.

T = Tahmini toplam talep

P = Ödenmesi gereken ücret

Şeklinde hesaplanabilir. Bu hesaplama yöntemi mantıksal olarak bu çalışmaya eklenmiştir. Blokzincir ağı üzerindeki ürünlerin fiyatının nasıl belirleneceği konusu, kapsamlı incelenmesi gereken bir konudur. Bundan dolayı sonraki çalışmalarda bu konuların netleştirilmesi hedeflenecektir.

3.2.4 Nesneye Yönelik Sınıf Tasarımı

Kurgulanan yapıda işlemlerin bir kısmı sistem katılımcılarının yerel sunucularında, bir kısmı IPFS sisteminde ve bir kısmı da blokzincir ağında olacaktır. Modellerin geliştirilmesi, eğitilmesi ve kullanılması yerel sunucular tarafında olurken, modellerin paylaşılması, ödeme ve ödüllendirmenin yapılması, veriler üzerindeki okuma ve yazma işlemleri blokzincir tarafından yapılacaktır. Ayrıca büyük veri kümelerinin kaydedilmesi için dağıtık depolama birimlerinden faydalanılır. Blokzincir ağında işlemleri yöneten ve koordine eden akıllı kontratlar olmalıdır. Model geliştirme ve eğitme rolleri üzerindeki yetkilendirmeleri yöneten *ValidatorAccounts* kontratı, yeni modellerin kodlarını merkezi olmayan bir sisteme kaydederek modeller üzerinde yapılan her işlemi yöneten ve kaydeden *ModelContractList,* model ağırlıklarını toplayarak modelleri birleştiren *ModelAggregator* ve eğitim taleplerini yöneten *FLContratRequests* kontratları ilk etapta tasarlanacak ve aralarındaki ilişki anlatılacaktır. Ayrıca token bazlı bir ödüllendirme yapabilmek için ERC20 standartlarına göre çalışan *GenoModelToken* kontratının geliştirilmesi planlanmaktadır. Bu kontrat, ücretlerin hesaplar arasında transferinden sorumlu olacaktır.

a) *ValidatorAccounts:* "*Accounts*" kontratı hesapların yönetilmesinden sorumlu olan ana kontrattır. Bu kontrattan "*ValidatorAccounts*" ve "*UserAccounts*" kontratları türetilecektir. Ana kontrat ve türetilen akıllı

kontratlar "*newAccount*", "*checkNewAccIsValid*" ve "*checkAccoutRole*" metotlarını barındırmaktadır. "*newAccount*" metodu sisteme yeni hesap eklemek için kullanılacaktır. Sisteme katılmak isteyen hesaplar gerekli bilgiler ile bu metodu çağırarak katılımcı talebinde bulunur. İlgili hesabın kimlik kontrolleri ve rollerin onaylanması gibi işlemler "*checkNewAccIsValid*" metodu ile yapılacaktır. "*checkAccoutRole*" metodu ise sisteme kayıtlı olan hesapların rollerini döner. Yeni model eklenirken ve eğitilirken bu metot çağrılarak kontroller yapılır.

b) *ModelContractList:* "*newModel*" metodu yeni model kaydetmek için kullanılacaktır. Kaydedilmek istenen modellerin uygunluk kontrolleri "*checkModelIsUniq*" ve "*checkModelIsValid*" metotları ile sağlanacaktır. "*checkModelIsUniq*" metodu sisteme kaydedilecek modelin daha önce paylaşılmayan bir model olup olmadığını kontrol eder. "*getModel*" ile eğitilmiş model ağdan çekilebilir ve "*getModelCode*" ile modelin kodu çekilebilir.

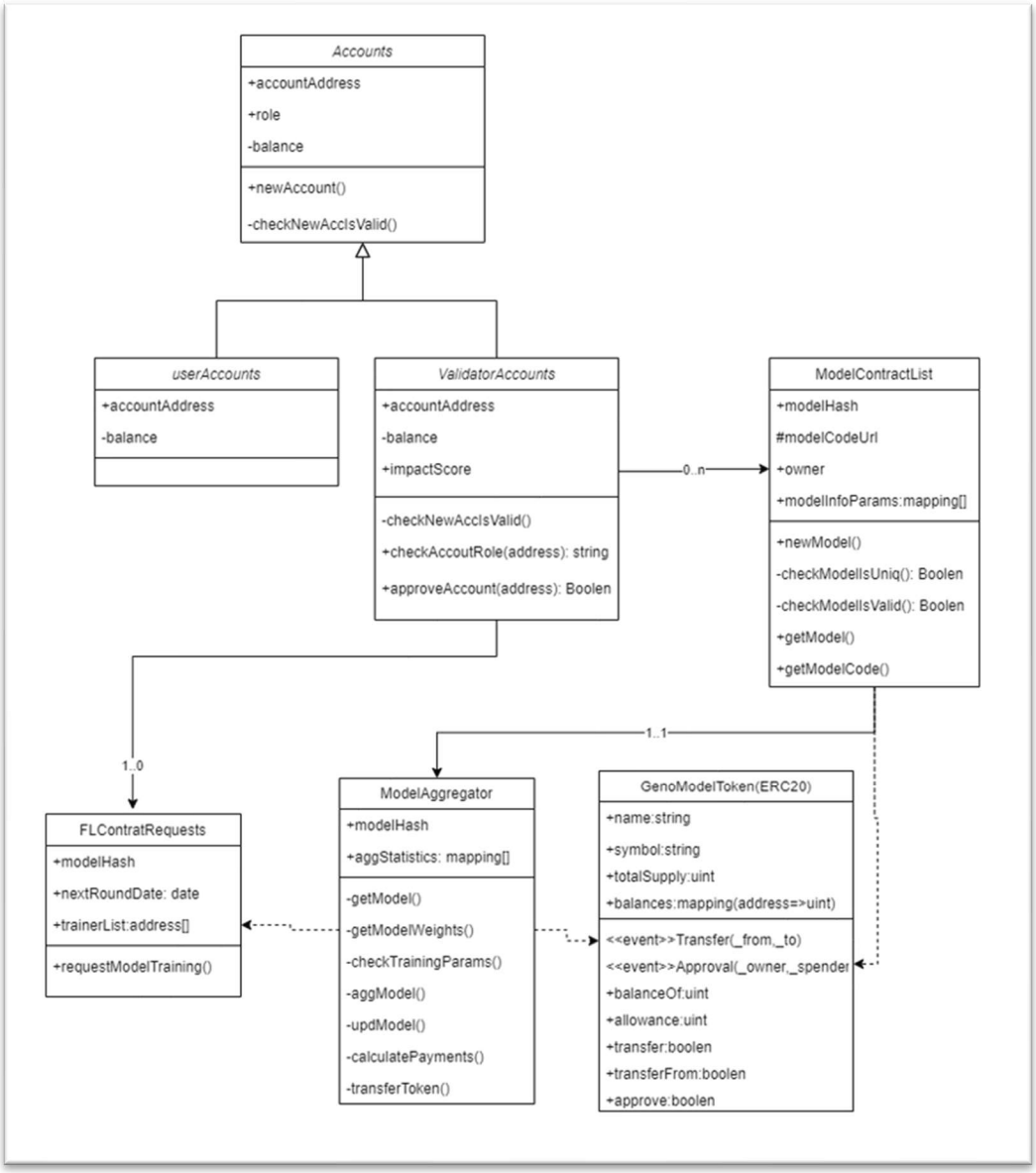
c) *ModelAggregator:* Bu kontrat modellerin eğitilmesi işlemlerini yönetecek kontrattır. Her bir eğitim işlemlerinin başlatılması, ağırlıkların toplanması, model ağırlıklarının birleştirilmesi ve model bilgilerinin güncellenmesi işlerini yapar. "*FLContratRequests*" yardımıyla eğitime katılacak hesapları ve birleştirme tarihine göre süreci başlatır. *"getModel"* metodu ile son güncel model çekilir. "*getModelWeights*" metodu ile katılımcı hesaplardan ağırlıklar alınır. "*checkTrainingParams*" metodu katılımcılardan gelen başarı, data miktarı, test datası gibi bilgileri alarak merkezi modeli olumsuz yönde etkileyecek bir durum var mı diye kontrol eder. "*aggModel*" metodu ile kontrolden başarılı geçen güncellemeler alınarak modeller birleştirilir. "*updModel*" metodu ile son güncellemeler kaydedilir. "*calculatePayments*" metodu ile verilecek ödül hesaplanır ve "*transferToken*" ile ödüller dağıtılır.

d) *FLContratRequests:* Bizim modelimizde eğitim sürekli olamayacağı için talep doğrultusunda eğitim işlemi başlatılacaktır. Bu kontrat

modele göre eğitim taleplerini alarak, eğitime katılım sağlayacak katılımcıları belirler.

e) *GenoModelToken:* Bu kontrat ise token üreterek, transfer işlemlerini kontrol eden kontrattır.

Şekil 22' deki görselin daha büyük bir hali EK-2' ye de konulmuş olup, bu görsel nesneye yönelik programlama yöntemine göre tasarlanmış sınıflar ve aralarındaki ilişkiler gösterilmektedir. "*Accounts*" sınıfından iki yeni alt sınıf türetilmiştir. Bunlar sistemin kullanıcılarını tanımlayan "*userAccounts*" ve yetkilendirme rolüne sahip "*ValidatorAccounts*" sınıflarıdır. "*ValidatorAccounts*" sınıfı ile "*ModelContratList*" arasında 0..n ilişki kurulmuştur. Çünkü model geliştirici ve eğiticileri tanımlayan bu sınıfa ait varlıklar, 0 ya da daha fazla model geliştirebilir ve sistem üzerinde paylaşabilir. "*ModelContratList*" ile "*ModelAggregator*" arasında bire bir ilişki vardır. Her modelin eğitilme sürecini temsil eden bir obje olacaktır. "*ModelAggregator*" çalışabilmesi için ağda bir talep oluşmalıdır. "*FLContratRequests*" bu talepleri toplar ve bu durum "*ModelAggregator*" çalışmasının ön koşuludur. "*GenoModelToken*" token kontratı ise yeni model geliştirme ve eğitim sürecine bağlı olduğu için, "*ModelAggregator*" ve "*ModelContratList*" sınıfları arasında bağımlı ilişki tanımlanmıştır.



Şekil 22. Önerilen modelin sınıf diyagramı

3.3 GELİŞTİRME TASARIMI

Geliştirme tasarım bölümünde, sistem üzerinde kullanılacak teknolojik modüllerin ve bileşenlerin açıklandığı kısım olacaktır. Ayrıca bu bileşenler arasındaki ilişki de gösterilecektir.

a) *Son Kullanıcı Grafiksel Ara yüzü*: Bu bileşen son kullanıcılar için karmaşık arka plan işlemlerinin basitleştirilmesi için kullanılır. Model paylaşımı, ağ üzerinde paylaşılan modellerle ilgili bilgilerin görülebilmesi, ücret transfer talep girişlerinin yapılması, model eğitim taleplerinin girilmesi gibi işlemler bu ekranlar üzerinden yapılacaktır. Ön yüz ekranları web temelli teknolojiler ile geliştirilecektir.

b) *Uygulama Arka Planı:* Son kullanıcı ekranlarının arka planında çalışan bileşenlerin toplamıdır. Kayıtların tutulduğu veri tabanı, kodlar, ekranlarda kullanılan dosya ve resimlerin bulunduğu depolama birimlerinden oluşur. Bölüm 1.4.1.4 de anlatıldığı gibi Blokzincir ağı ile iletişim web3 ara yüzleri ile gerçekleştirilebilir. Bu bileşenler de arka planın parçaları olarak düşünülmüştür.

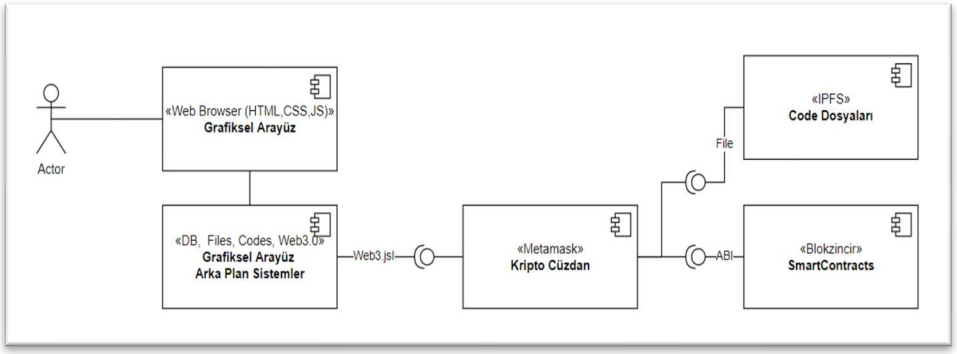
c) **Ethereum Blokzincir**: Akıllı kontratların çalıştırıldığı genel blokzincir ağıdır. Eğitilen modellerin ağırlıklarının toplanması, ödül hesaplama ve token transferleri gibi işlemler bu sistem üzerinde çalışan akıllı kontratlar sayesinde yürütür. Modeller ile ilgili her işlem ve hesap bakiye değişimleri gibi işlemlerin kayıtları tutulur. Modellerle ilgili model kodunun IPFS sisteminde nerede tutulduğu, model özellikleri, model başarısı, ne kadar ödül dağıtıldığı gibi bilgiler de burada tutulur. Ethereum blokzincir ağı, akıllı kontratlar için bir entegrasyon ara birimi (ABI) sağlar. Ön yüz uygulamaları ABI vasıtasıyla blokzincir üzerindeki işlemleri yapabilir. Ayrıca son kullanıcılar da bu ara yüzleri kullanarak kendi ön yüz uygulamalarını tasarlayabilir ya da kendileri bu ara yüzün sağladığı fonksiyonları kullanarak işlem yapabilirler.

d) **IPFS**: Bölüm 1.4.1.3 de bahsedildiği gibi paylaşılmak istenen dosyaların dağıtık bir yapıda tutulmasını sağlayan bir depolama birimidir. Ara yüz ile bu sisteme erişilerek, gerekli dokümanlara IPFS bağlantı adresleri üzerinden ulaşılabilir.

e) **Kullanıcı Cüzdanı-Metamask**: Ethereum blokzincir üzerinde herhangi bir işlem yapabilmek için blokzincir cüzdanına ihtiyaç vardır. Metamask, ethereum ağı için tasarlanmış cüzdan uygulamasıdır. Bu

sistemde yer alabilmek için her katılımcının bir kripto cüzdan adresi olmalıdır.

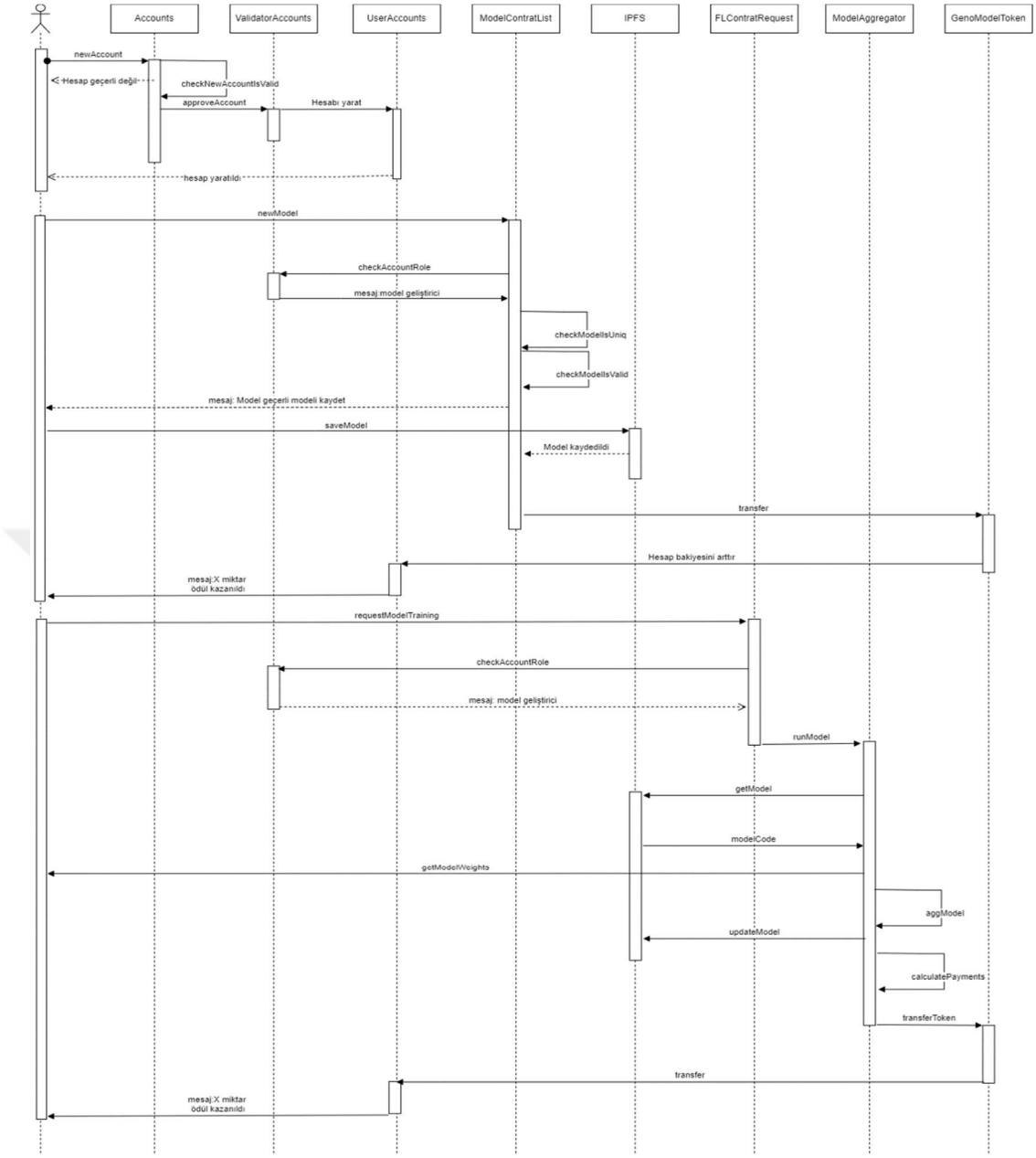
Şekil 23’ te tasarlanan sistemin temel bileşenleri ve bağıntıları resmedilmiştir.



Şekil 23. Önerilen modelin bileşen diyagramı.

3.4 SÜREÇ TASARIMI

Şekil 24’ te ve EK-3’ te süreç tasarımı resmedilmiştir. Bu gösterimde aktörlerin birbirleriyle ve fonksiyonlarla olan ilişkileri anlatılmıştır. Süreç tasarımı tüm proje süreçlerini içermeyip temel süreçleri içermektedir. Bu süreçler yeni hesap oluşturma, yeni model geliştirme ve mevcut modeli eğitmektir. Bahsedilen bu süreçlerin dışında para transferleri, ön yüz etkileşimleri, onaylama süreçleri gibi detaylandırılması gereken süreçler vardır. Ancak bu tez kapsamında ele alınmayıp sonraki çalışmalarda değerlendirilecektir.



Şekil 24. Süreç tasarım diyagramı.

Konsept çalışmamızdaki ana süreçlerin etkileşimleri aşağıda açıklanmıştır.

Yeni hesap oluşturma: Yeni hesap oluşturma ile ilgili temel fonksiyonlar "*Accounts*" ve bu sınıftan türetilen diğer sınıflar tarafından yönetilir. Yeni hesap oluşturmak için "*newAccount*" prosedürü çağrılır. Bu prosedür öncelikle talepçinin gerçek bir şahıs olup olmadığını sorgular ve diğer katılımcılar tarafından onaylanması için bu talebi

tüm düğümlere dağıtarak cevap bekler. Onaylama işlemi tamamlandıktan sonra "*UserAccount*" sınıfı üzerinden yeni hesap nesnesi türetilir.

Yeni model geliştirme: Yeni model geliştirme işlemi "*newModel*" fonksiyonun çağrılması ile başlar. Bu fonksiyon, modelle ilgili bilgileri parametre olarak bekler. İlgili parametrelerle fonksiyon çağrıldıktan sonraki ilk adım, talepçinin rolünü kontrol etmektir. "*ValidatorAccounts*" sınıfının bir fonksiyonu olan "*checkAccountRole*" fonksiyonu ile talepçinin sisteme yeni model ekleme hakkının olup olmadığı kontrol edilir. Talepçi yetkili bir hesap ise "*ModelContratList*" içerisindeki modelin tekilliğini ve gerçek bir model olup olmadığını kontrol eden prosedürler çalıştırılarak model onaylanır. Model onayını alan talepçi, modelini IPFS sistemine kaydetmeye hak kazanır. IPFS modeli üzerine kaydetme işlemi tamamlanınca "*GenoModelToken*" sınıfı içerisindeki "*transfer*" fonksiyonu ile hak edilen ödül ilgili hesabın bakiyesine eklenir. Transfer işlemi gerçekleşince talepçi bilgilendirilir.

Mevcut modeli eğitme: Son olarak model eğitim süreci ise, "*FLContratRequest*" sınıfında bulunan "*requestModelTraining*" fonksiyonu aracılığı ile başlatılır. Bir önceki süreçte olduğu gibi talepçinin öncelikle "*checkAccountRole*" prosedürü ile rol kontrolü yapılır. Talepçi uygun role sahipse "*FLContratRequest*" modelini kimin çalıştıracağını ve FL çalışma mantığında kaç tur eğitilen modellerin toplanacağı ile ilgili parametreleri kaydeder. İlk tur "*ModelAggregator*" sınıfında bulunan "*runModel*" fonksiyonu çağrılarak başlatılır. "*ModelAggregator*" çalıştırılmak istenen modelle ilgili bilgileri IPFS üzerinden alarak, katılımcılardan model ağırlıklarını toplar. Toplanan model ağırlıkları birleştirilerek ana model oluşturulur ve eğitilen model ile ilgili bilgiler IPFS üzerinde güncellenir. Kaydetme işlemi tamamlandıktan sonra, katılımcıların ödülleri hesaplanarak "*GenoModelToken*" aracılığı ile transfer işlemleri gerçekleştirilir.

BÖLÜM IV

4. TARTIŞMA ve SONUÇ

Bu tezde, teorik olarak blokzincir ve FL'in beraber kullanımı ile araştırma kuruluşlarının beraber çalışabilecekleri bir yöntem sunuldu. Genetik veri analizi üzerinde çalışan araştırma kuruluşlarının iş birliği ile daha doğru sonuçlar üreten ML modellerini geliştirmeleri ve üretilen modelleri kripto para karşılığında satabildikleri, merkeziyetsiz bir referans mimari önerilmektedir. Önerilen mimari tasarımda, veri güvenliği konusunda hassasiyet ile durulmuş olup, FL ve blokzincir sayesinde ham veriyi paylaşmadan, farklı tarafların geliştirdikleri modelleri birleştirebildiğimiz yapı önerilmektedir. Sistem tasarımında, ödüllendirme kurgusu ile sistemin devamlılığı ve taraflar arasında bir sinerji oluşturmanın yollarına çözümler önerilmiştir. Çalışmadaki farklılık FL ve blokzincirin genetik veri bilimi alanında beraber kullanımıdır. Literatürde bahsedilen makalelerde bu iki teknolojinin genelde Iot ve IoMT cihazları üzerinde kullanımına değinildi. Ayrıca enerji sektörü gibi farklı alanlarda da bu iki teknolojinin beraber kullanıldığı çalışmalara rastlanmaktadır (Ali, 2020). Fakat "Veri ve Yetki Paylaşımı" başlığı altında değinilen genetik alandaki çalışmaların birçoğu birey verisinin güvenli bir şekilde diğer kurumlar ile paylaşılması, veri sahipliğinin kullanıcıya verilmesi ve izinsiz veri kullanımının engellenmesine yönelik yapılmış çalışmalardır. Fakat biz bu çalışmada, bireyden ziyade araştırma kuruluşlarının verimliliğini arttıracak bir tasarım çalıştık. Aynı zamanda tasarımda, Kipoi (Avsec ve diğerleri, 2019) projesindeki gibi tekrar kullanılabilirliği olan model sağlamanın yanısıra, ML çalışmalarına farklı ülkelerdeki katılımcıların katkı sağlayarak ML model performanslarının geliştirilebileceği bir alt yapı sunulmuştur.

Çalışmada tasarım, yazılımın 4+1 görünümüne göre oluşturulmuştur. Bu anlamda kullanımdaki senaryolar, mantıksal tasarım, geliştirme ve süreç tasarımları UML diyagramları ile açıklanmıştır. Senaryo bölümünde temel aktörler ve sistem üzerinde yapabilecekleri işlemler belirlenmiştir. Mantıksal tasarım bölümünde model eğitme, geliştirme, ödül mekanizması ve hazır model kullanımı ile ilgili algoritmalara yer

verilmiştir. Ödül ve kullanım ücreti hesaplamaları için çeşitli denklemler önerilmiştir. Ayrıca mantıksal tasarım bölümünde, sistemin ana yapıları nesneye yönelik sınıf tasarımı hazırlanmıştır. Geliştirme tasarımında önerilen modelde geliştirilmesi gereken temel modüller ve teknolojiler açıklanmıştır. Son olarakta süreç tasarımı ile sistemin temel fonksiyonları arasında etkileşimler belirlenmiştir. Kullandığımız tasarım yöntemi ile önerdiğimiz platformun ana bileşenleri farklı bakış açıları ile modellenerek yazılıma hazır hale getirilmiştir. Tasarım ile belirlenen çerçeve sayesinde yazılım geliştirme sürecinin daha hızlı ilerleyebileceği düşünülmektedir.

Sonraki çalışmalarda bu tasarımın kodlanması ve performans göstergelerinin ölçülmesi gerçekleştirilebilir. Platformun ön yüz tasarımına bu çalışmada yer verilmemiştir. Ön yüz için ayrıca kullanıcı dostu bir ekran tasarımına ihtiyaç vardır. Bu tasarım, modellerin listelenebileceği, yeni model taleplerinin girilebileceği ve modellerin satın alınması için ekranların olabileceği şekilde geliştirilebilir. Ayrıca geliştirilen modeller NFT tokenlara dönüştürülerek, kodların güvenliğini sağlamak mümkün olacaktır. Ek olarak ön yüz ile arka plan arasındaki iletşimler için ayrı arayüzleri ve entegrasyonları tasarlamak gerekebilir. Bunlarla beraber tasarımda önerilen FL için FedAvg yöntemi ile model birleştirme yapılmıştır. Fakat bu algoritmanın bazı performans sorunları olduğuna da değinilmiştir. Bundan dolayı sonraki çalışmalarda, farklı model birleştirme algoritmalarının da karşılaştırılabileceği şekilde ilerlenebilir. Geliştirilmeye uygun noktalardan bir tanesi de model kullanım ücretinin belirlenmesidir. Mevcut çalışmada kullanılan hesaplama, maliyetler ön plana alınarak yapılmıştır. İlerideki çalışmalarda arz talep dengesine göre model kullanım ücretleri belirlenebilir. Son olarak çalışmada sadece belirli süreçlere yer verildi. Muhtemel tüm süreçler için daha geniş kapsamlı çalışmalara ihtiyaç vardır.

KAYNAKÇA

Acharya, V. ve Yerrapati, Anand Eswararao Prakash, N. (2019). *Oracle Blockchain Quick Start Guide*.

Adanur Dedeturk, B., Soran, A. ve Bakir-Gungor, B. (2021). Blockchain for genomics and healthcare: a literature review, current status, classification and open issues. *PeerJ*, *9*, e12130–e12130. doi:10.7717/peerj.12130

Ahmed, E. ve Shabani, M. (2019). DNA Data Marketplace: An Analysis of the Ethical Concerns Regarding the Participation of the Individuals. *Frontiers in Genetics*, *10*. doi:10.3389/fgene.2019.01107

Alabdulwahhab, F. A. (2018). Web 3.0: The Decentralized Web Blockchain networks and Protocol Innovation. *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)* içinde (ss. 1–4). doi:10.1109/CAIS.2018.8441990

Alacam, S. ve Sencer, A. (2021). Using Blockchain Technology to Foster Collaboration among Shippers and Carriers in the Trucking Industry: A Design Science Research Approach. *Logistics* . doi:10.3390/logistics5020037

Ali, F. S. (2020). SynergyGrids : Blockchain Assisted Peer-to-Peer Energy Trading and Prosumer Management Framework by.

Atalay, A. Ö. (2020). Ceza Muhakemesi Hukukunda Moleküler Genetik İncelemelerin Özel Nitelikli Kişisel Verilerin Korunması Açısından Değerlendirilmesi. *Journal of Penal Law & Criminology*, *7*(2), 127–184. doi:10.26650/jplc2019-0018

Avsec, Ž., Kreuzhuber, R., Israeli, J., Xu, N., Cheng, J., Shrikumar, A., … Gagneur, J. (2019). The Kipoi repository accelerates community exchange and reuse of predictive models for genomics. *Nature Biotechnology*, *37*(6), 592–600. doi:10.1038/s41587-019-0140-0

Aydiner, A. S. (2021). New Approach to A Disruptive Business Model with Dynamic Capability Under the Blockchain Technology BT - Management Strategies to Survive in a Competitive Environment: How to Improve Company Performance. H. Dincer ve S. Yüksel (Ed.), (ss. 17–32). Cham: Springer International Publishing. doi:10.1007/978-3-030-72288-3_2

Barkadehi, M. H., Nilashi, M., Ibrahim, O., Zakeri Fardi, A. ve Samad, S. (2018). Authentication systems: A literature review and classification. *Telematics and Informatics*, *35*(5), 1491–1511. doi:https://doi.org/10.1016/j.tele.2018.03.018

Başaran, E. ve Aras, S. (2010). General Outlook and Applications of Genomics, Proteomics and Metabolomics. *Turkish Bulletin of Hygiene and Experimental Biology*, *67*(2), 85–96.

Bashir, I. (2020). *Mastering Blockchain 3rd Edition*.

Bhatia, P. ve Sumbaly, R. (2014). Framework for Wireless Network Security Using Quantum Cryptography. *International journal of Computer Networks & Communications*, *6*. doi:10.5121/ijcnc.2014.6604

Bogetoft, P., Christensen, D. L. ve Damg, I. (2008). 15_Secure_MPC_goes_live.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., … Roselander, J. (2019). Towards Federated Learning at Scale: System Design. A. Talwalkar, V. Smith ve M. Zaharia (Ed.), *Proceedings of Machine Learning and Systems* içinde (C. 1, ss. 374–388). https://proceedings.mlsys.org/paper/2019/file/bd686fd640be98efaae0091fa301e613-Paper.pdf adresinden erişildi.

Buterin, V. (2014). A next-generation smart contract and decentralized application platform. *Etherum*, (January), 1–36. http://buyxpr.com/build/pdfs/EthereumWhitePaper.pdf adresinden erişildi.

Cai, W., Wang, Z., Ernst, J. B., Hong, Z., Feng, C. ve Leung, V. C. M. (2018). Decentralized Applications: The Blockchain-Empowered Software System. *IEEE Access*, *6*, 53019–53033. doi:10.1109/ACCESS.2018.2870644

Callaway, E. (2018, Ekim). Supercharged crime-scene DNA analysis sparks privacy concerns. *Nature*. England. doi:10.1038/d41586-018-06997-8

Canim, M., Kantarcioglu, M. ve Malin, B. (2012). Secure management of biomedical data with cryptographic hardware. *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, *16*(1), 166–175. doi:10.1109/TITB.2011.2171701

Castro, M. (2001). Practical Byzantine Fault Tolerance. *Proceedings of the Third Symposium on Operating Systems Design OSDI '99*, (February), 1–172. http://pmg.csail.mit.edu/papers/osdi99.pdf adresinden erişildi.

Chan, A.-W., Song, F., Vickers, A., Jefferson, T., Dickersin, K., Gøtzsche, P. C., … van der Worp, H. B. (2014). Increasing value and reducing waste: addressing inaccessible research. *Lancet (London, England)*, *383*(9913), 257–266. doi:10.1016/S0140-6736(13)62296-5

Collins, F. S. ve Mansoura, M. K. (2001). The Human Genome Project. *Cancer*, *91*(S1), 221–225. doi:https://doi.org/10.1002/1097-0142(20010101)91:1+<221::AID-CNCR8>3.0.CO;2-9

Conley, J. P. (2019). Encryption, Hashing, PPK, and Blockchain: A Simple Introduction, (2019). http://jpconley.wordpress.com/ adresinden erişildi.

Coron, J.-. (2006). What is cryptography? *IEEE Security & Privacy*, *4*(1), 70–73. doi:10.1109/MSP.2006.29

Craig, N. L. (2010). *Molecular Biology: Principles of Genome Function*. Molecular Biology: Principles of Genome Function. OUP Oxford. https://books.google.com.tr/books?id=iX6sAQAAQBAJ adresinden erişildi.

Craig, V. J., D., A. M., W., M. E., W., L. P., J., M. R., G., S. G., … Xiaohong, Z. (2001). The Sequence of the Human Genome. *Science*, *291*(5507), 1304–1351. doi:10.1126/science.1058040

Creel, M. ve Goffe, W. L. (2008). Multi-core CPUs, clusters, and grid computing: A tutorial. *Computational Economics*, *32*(4), 353–382. doi:10.1007/s10614-008-9143-5

Drysdale, R., Cook, C. E., Petryszak, R., Baillie-Gerritsen, V., Barlow, M., Gasteiger, E., … McEntyre, J. (2020). The ELIXIR Core Data Resources: fundamental infrastructure for the life sciences. *Bioinformatics (Oxford, England)*, *36*(8), 2636–2642. doi:10.1093/bioinformatics/btz959

Dwork, C. ve Naor, M. (1993). Pricing via Processing or Combatting Junk Mail BT - Advances in Cryptology — CRYPTO' 92. E. F. Brickell (Ed.), (ss. 139–147). Berlin, Heidelberg: Springer Berlin Heidelberg.

El-Hindi, M., Binnig, C., Arasu, A., Kossmann, D. ve Ramamurthy, R. (2019). BlockchainDB: A Shared Database on Blockchains. *Proc. VLDB Endow.*, *12*(11), 1597–1609. doi:10.14778/3342263.3342636

Franco, M., Killer, C., Rafati, S., Rodrigues, B., Scheid, E., Ribeiro, R. ve Huertas, A. (2021). Communication Systems XIV, (June).

Geron, A. (2017). *[(PDF)] Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O'Reilly Media.

Goldfeder, R. L., Wall, D. P., Khoury, M. J., Ioannidis, J. P. A. ve Ashley, E. A. (2017). Human Genome Sequencing at the Population Scale: A Primer on High-Throughput DNA Sequencing and Analysis. *American journal of epidemiology*, *186*(8), 1000–1009. doi:10.1093/aje/kww224

Golosova, J. ve Romanovs, A. (2018). The Advantages and Disadvantages of the Blockchain Technology. *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)* içinde (ss. 1–6). doi:10.1109/AIEEE.2018.8592253

Gomaa, H. (2006). Designing concurrent, distributed, and real-time applications with UML. *Proceedings - International Conference on Software Engineering*, *2006*, 1059–1060. doi:10.1145/1134285.1134504

Grishin, D., Obbad, K. ve Church, G. M. (2019). Data privacy in the age of personal genomics. *Nature Biotechnology*, *37*(10), 1115–1117. doi:10.1038/s41587-019-0271-3

Grishin, D., Obbad, K., Estep, P., Cifric, M. ve Zhao, Y. (2018). NEBULA_whitepaper_v4.52.

Gymrek, M., McGuire, A. L., Golan, D., Halperin, E. ve Erlich, Y. (2013). Identifying personal genomes by surname inference. *Science (New York, N.Y.)*, *339*(6117), 321–324. doi:10.1126/science.1229566

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, B. A. y A. (2017). 1. 谷歌FL方案Communication-Efficient Learning of Deep Networks from Decentralized Data.pdf, *54*, 10.

Harrow, J., Hancock, J., Community, E.-E. ve Blomberg, N. (2021). ELIXIR-EXCELERATE: establishing Europe's data infrastructure for the life science research of the future. *The EMBO journal*, *40*(6), e107409–e107409. doi:10.15252/embj.2020107409

Hayırlıoğlu, G. H. (2019). Genetik Verilerin Korunması. *Yüksek Lisans Tezi*.

Hou, D., Zhang, J., Man, K. L., Ma, J. ve Peng, Z. (2021). A Systematic Literature Review of Blockchain-based Federated Learning: Architectures, Applications and Issues. *2021 2nd Information Communication Technologies Conference*

(ICTC) içinde (ss. 302–307). doi:10.1109/ICTC51749.2021.9441499

Huang, H. S., Chang, T. S. ve Wu, J. Y. (2020). A secure file sharing system based on IPFS and blockchain. *ACM International Conference Proceeding Series*, 96–100. doi:10.1145/3409934.3409948

Humbert, M., Ayday, E., Hubaux, J.-P. ve Telenti, A. (2013). Addressing the Concerns of the Lacks Family: Quantification of Kin Genomic Privacy. *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security* içinde , CCS '13 (ss. 1141–1152). New York, NY, USA: Association for Computing Machinery. doi:10.1145/2508859.2516707

Islam, I., Munim, K. M., Oishwee, S. J., Islam, A. K. M. N. ve Islam, M. N. (2020). A Critical Review of Concepts, Benefits, and Pitfalls of Blockchain Technology Using Concept Map. *IEEE Access*, *8*, 68333–68341. doi:10.1109/ACCESS.2020.2985647

Jiang, J. C., Kantarci, B., Oktug, S. ve Soyata, T. (2020). Federated learning in smart city sensing: Challenges and opportunities. *Sensors (Switzerland)*, *20*(21), 1–29. doi:10.3390/s20216230

Kairouz, P., Brendan McMahan, H., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., … Zhao, S. (2019). Advances and open problems in federated learning. *arXiv*, 1–105.

Kalogeropoulos, A. (2018). A Reference Architecture for Blockchain-based Resource-intensive Computations managed by Smart Contracts Master ' s Thesis in Informatics A Reference Architecture for Blockchain-based Resource-intensive Computations managed by Smart Contracts Anastasios K, (October), 20. https://www.researchgate.net/profile/Anastasios_Kalogeropoulos/publication/328342750_A_Reference_Architecture_for_Blockchain-based_Resource-intensive_Computations_managed_by_Smart_Contracts/links/5bc74f34458515f7d9bff64d/A-Reference-Architecture-for-Block adresinden erişildi.

Karabekmez, M. E. (2021). Data Ethics in Digital Health and Genomics. *The New Bioethics*, *27*(4), 320–333. doi:10.1080/20502877.2021.1996965

Kavak Konrat, E. G. (2020). *GENETİK VERİLERİN KORUNMASI VE GENETİK AYRIMCILIK*. Eskişehir Anadolu Üniversitesi Sosyal Bilimler Enstitüsü.

Keshk, M., Turnbull, B., Moustafa, N., Vatsalan, D. ve Choo, K. K. R. (2020). A Privacy-Preserving-Framework-Based Blockchain and Deep Learning for Protecting Smart Power Networks. *IEEE Transactions on Industrial Informatics*, *16*(8), 5110–5118. doi:10.1109/TII.2019.2957140

Kim, H., Park, J., Bennis, M. ve Kim, S.-L. (2020). Blockchained On-Device Federated Learning. *IEEE Communications Letters*, *24*(6), 1279–1283. doi:10.1109/LCOMM.2019.2921755

Knoppers, B. M. ve Chadwick, R. (1994). The human genome project: Under an international ethical microscope. *Science*, *265*(5181), 2035–2036. doi:10.1126/science.8091225

Kruchten, P. (1995). The 4 + 1 View of Software Architecture. *IEEE Software*, *12*(6), 42–50.

Kulemin, N., Popov, S. ve Gorbachev, A. (2017). The Zenome Project: Whitepaper blockchain-based genomic ecosystem. *Zenome. io*, A.

Kumar, R., Khan, A. A., Zhang, S., Wang, W. Y., Abuidris, Y., Amin, W. ve Kumar, J. (2020). Blockchain-Federated-Learning and Deep Learning Models for COVID-19 detection using CT Imaging. *arXiv*, *14*(8), 1–12.

Kuo, T. T., Zavaleta Rojas, H. ve Ohno-Machado, L. (2019). Comparison of blockchain platforms: A systematic review and healthcare examples. *Journal of the American Medical Informatics Association*, *26*(5), 462–478. doi:10.1093/jamia/ocy185

Li, L., Fan, Y., Tse, M. ve Lin, K. Y. (2020). A review of applications in federated learning. *Computers and Industrial Engineering*, *149*(September). doi:10.1016/j.cie.2020.106854

Linden, M., Procházka, M., Lappalainen, I., Bucik, D., Vyskocil, P., Kuba, M., … Nyrönen, T. (2018). Common ELIXIR Service for Researcher Authentication and Authorisation. *F1000Research*, *7*, 1199. doi:10.12688/f1000research.15161.1

Lu, Y., Huang, X., Dai, Y., Maharjan, S. ve Zhang, Y. (2020). Blockchain and Federated Learning for Privacy-Preserved Data Sharing in Industrial IoT. *IEEE Transactions on Industrial Informatics*, *16*(6), 4177–4186. doi:10.1109/TII.2019.2942190

Malin, B. ve Sweeney, L. (2004). How (not) to protect genomic data privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems. *Journal of biomedical informatics*, *37*(3), 179–192. doi:10.1016/j.jbi.2004.04.005

Mamo, N., Martin, G. M., Desira, M., Ellul, B. ve Ebejer, J.-P. (2020). Dwarna: a blockchain solution for dynamic consent in biobanking. *European Journal of Human Genetics*, *28*(5), 609–626. doi:10.1038/s41431-019-0560-9

Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W. ve Qijun, C. (2017). A review on consensus algorithm of blockchain. *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* içinde (ss. 2567–2572). doi:10.1109/SMC.2017.8123011

Mitrou, L. (2018). Data Protection, Artificial Intelligence and Cognitive Services: Is the General Data Protection Regulation (GDPR) 'Artificial Intelligence-Proof'?, 1–90. http://dx.doi.org/10.2139/ssrn.3386914 adresinden erişildi.

Mohammed Yakubu, A. ve Chen, Y. P. P. (2020). Ensuring privacy and security of genomic data and functionalities. *Briefings in Bioinformatics*, *21*(2), 511–526. doi:10.1093/bib/bbz013

Moore, A. D. ve Moore, A. D. (2017). 64.Privacy : Its Meaning and Value. *American Philosophical quarterly*, *40*(3), 215–227.

Nagalapatti, L. ve Narayanam, R. (2021). Game of Gradients: Mitigating Irrelevant Clients in Federated Learning. http://arxiv.org/abs/2110.12257 adresinden erişildi.

Nakamoto, S. (2009). *Bitcoin: A Peer-to-Peer Electronic Cash System*. https://bitcoin.org/bitcoin.pdf adresinden erişildi.

Nicolas, H. (2014). It Will Cost You Nothing to "Kill" a Proof-of-Stake Crypto-Currency. *SSRN Electronic Journal*, *34*. doi:10.2139/ssrn.2393940

Özdemir, S. (2008). Kablosuz algılayıcı ağlarinda Homomorfik Şifreleme ile Güvenli Veri Kümeleme, *23*(2), 365–373.

Ozercan, H. I., Ileri, A. M., Ayday, E. ve Alkan, C. (2018). Realizing the potential of blockchain technologies in genomics. *Genome Research*, *28*(9), 1255–1263. doi:10.1101/gr.207464.116

Pearson, H. (2006). What is a gene? *Nature*, *441*(7092), 398–401.

doi:10.1038/441398a

Połap, D., Srivastava, G. ve Yu, K. (2021). Agent architecture of an intelligent medical system based on federated learning and blockchain technology. *Journal of Information Security and Applications*, *58*(February), 102748. doi:10.1016/j.jisa.2021.102748

Rahman, M. A., Hossain, M. S., Islam, M. S., Alrajeh, N. A. ve Muhammad, G. (2020). Secure and Provenance Enhanced Internet of Health Things Framework: A Blockchain Managed Federated Learning Approach. *IEEE Access*, *8*, 205071–205087. doi:10.1109/ACCESS.2020.3037474

Raymond, E. (1999). The cathedral and the bazaar. *Knowledge, Technology & Policy*, *12*(3), 23–49. doi:10.1007/s12130-999-1026-0

Reischuk, R. (1985). A new solution for the Byzantine generals problem. *Information and Control*, *64*(1), 23–42. doi:https://doi.org/10.1016/S0019-9958(85)80042-5

Ross, R., Molina, M. ve Beard Jr., B. (2018). Waste Not, Want Not : Your Computer Could Help Cure Cancer, 1–8. https://foldingcoin.net/index.php/whitepaper adresinden erişildi.

Rothstein, M. A. (2007). *Genetic exceptionalism and legislative pragmatism. The Journal of law, medicine & ethics : a journal of the American Society of Law, Medicine & Ethics*. England.

Rung, J. ve Brazma, A. (2013). Reuse of public genome-wide gene expression data. *Nature Reviews Genetics*, *14*(2), 89–99. doi:10.1038/nrg3394

Ruogang, T. (2021). Privacy Preserving in Big Data: Summary of Classification techniques. *SSRN Electronic Journal*, (July), 1–7. doi:10.2139/ssrn.3909074

Schär, F. (2021). Decentralized finance: on blockchain-and smart contract-based financial markets. *Federal Reserve Bank of St. Louis Review*, *103*(2), 153–174. doi:10.20955/r.103.153-74

Shabani, M. (2019). Blockchain-based platforms for genomic data sharing: a de-centralized approach in response to the governance problems? *Journal of the American Medical Informatics Association*, *26*(1), 76–80. doi:10.1093/jamia/ocy149

Shae, Z. ve Tsai, J. (2018). Transform Blockchain into Distributed Parallel

Computing Architecture for Precision Medicine. *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)* içinde (ss. 1290–1299). doi:10.1109/ICDCS.2018.00129

Shoenbill, K., Fost, N., Tachinardi, U. ve Mendonca, E. A. (2014a). Genetic data and electronic health records: A discussion of ethical, logistical and technological considerations. *Journal of the American Medical Informatics Association*, *21*(1), 171–180. doi:10.1136/AMIAJNL-2013-001694

Shoenbill, K., Fost, N., Tachinardi, U. ve Mendonca, E. A. (2014b). Genetic data and electronic health records: a discussion of ethical, logistical and technological considerations. *Journal of the American Medical Informatics Association*, *21*(1), 171–180. doi:10.1136/amiajnl-2013-001694

Simon, H. ve Fassnacht, M. (2019). Decision: Multidimensional Prices BT - Price Management: Strategy, Analysis, Decision, Implementation. H. Simon ve M. Fassnacht (Ed.), (ss. 209–257). Cham: Springer International Publishing. doi:10.1007/978-3-319-99456-7_6

Song, C., Ristenpart, T. ve Shmatikov, V. (2017). Machine Learning Models That Remember Too Much. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* içinde , CCS '17 (ss. 587–601). New York, NY, USA: Association for Computing Machinery. doi:10.1145/3133956.3134077

Stephens, Z. D., Lee, S. Y., Faghri, F., Campbell, R. H., Zhai, C., Efron, M. J., … Robinson, G. E. (2015). Big Data: Astronomical or Genomical? *PLoS biology*, *13*(7), e1002195–e1002195. doi:10.1371/journal.pbio.1002195

Szabo, N. (1997). Formalizing and Securing Relationships on Public Networks. *First Monday*, *2*(9 SE-). doi:10.5210/fm.v2i9.548

Tang, H., Jiang, X., Wang, X., Wang, S., Sofia, H., Fox, D., … Ohno-Machado, L. (2016). Protecting genomic data analytics in the cloud: state of the art and opportunities. *BMC medical genomics*, *9*(1), 63. doi:10.1186/s12920-016-0224-3

Torkzadehmahani, R., Nasirigerdeh, R., Blumenthal, D. B., Kacprowski, T., List, M., Matschinske, J., … Baumbach, J. (2021). Privacy-Preserving Artificial Intelligence Techniques in Biomedicine. *Methods of Information in Medicine*,

(July). doi:10.1055/s-0041-1740630

Torkzadehmahani, R., Nasirigerdeh, R., Blumenthal, D., Kacprowski, T., List, M., Matschinske, J., … Baumbach, J. (2020). *Privacy-preserving Artificial Intelligence Techniques in Biomedicine*.

Trinidad, S. B., Fullerton, S. M., Bares, J. M., Jarvik, G. P., Larson, E. B. ve Burke, W. (2010). Genomic research and wide data sharing: Views of prospective participants. *Genetics in Medicine*, *12*(8), 486–495. doi:10.1097/GIM.0b013e3181e38f9e

Türkiye Büyük Millet Meclisi (T.B.M.M). (2016). Kişisel verilerin korunmasi kanunu. *Resmi Gazete*, 12301–12317. https://www.mevzuat.gov.tr/MevzuatMetin/1.5.6698.pdf adresinden erişildi.

Vielhauer, C. (Ed.). (2006). Fundamentals in User Authentication BT - Biometric User Authentication for it Security: From Fundamentals to Handwriting (ss. 77–115). Boston, MA: Springer US. doi:10.1007/0-387-28094-4_4

Wan, S., Li, M., Liu, G. ve Wang, C. (2020). Recent advances in consensus protocols for blockchain: a survey. *Wireless Networks*, *26*(8), 5579–5593. doi:10.1007/s11276-019-02195-0

Wan, Z., Hazel, J. W., Clayton, E. W., Vorobeychik, Y., Kantarcioglu, M. ve Malin, B. A. (2022). Sociotechnical safeguards for genomic data privacy. *Nature Reviews Genetics*, *0123456789*. doi:10.1038/s41576-022-00455-y

Wilson, R. ve Rosen, P. (2003). Protecting Data through Perturbation Techniques: The Impact on Knowledge Discovery in Databases. *J. Database Manag.*, *14*, 14–26. doi:10.4018/978-1-59140-471-2.ch003

Woelfle, M., Olliaro, P. ve Todd, M. H. (2011). Open science is a research accelerator. *Nature Chemistry*, *3*(10), 745–748. doi:10.1038/nchem.1149

Wood, G. (2014). Ethereum: a secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 1–32.

Yng, Q., Kong, H., Yu, H., Brachman, R. J., Rossi, F., Stone, P. ve Yang Yang Liu Yong Cheng Yan Kang Tianjian Chen Han Yu, Q. (2020). Federated Learning. *Federated Learning*, 69–82.

Yüksek, Ş. A. (2021). DERİN ÖĞRENME YÖNTEMİ İLE DİFERANSİYEL MAHREMİYETLİ MEDİKAL GÖRÜNTÜ SINIFLANDIRMA.

Zahed Benisi, N., Aminian, M. ve Javadi, B. (2020). Blockchain-based decentralized storage networks: A survey. *Journal of Network and Computer Applications*, *162*, 102656. doi:https://doi.org/10.1016/j.jnca.2020.102656

Zhang, J., Chen, J., Wu, D., Chen, B. ve Yu, S. (2019). Poisoning Attack in Federated Learning using Generative Adversarial Nets. *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)* içinde (ss. 374–380). doi:10.1109/TrustCom/BigDataSE.2019.00057

Zhang, Y., Zhong, M., Zhao, X., Curtis, C., Li, X. ve Chen, C. (2019). Enabling privacy-preserving sharing of genomic data for GWASs in decentralized networks. *WSDM 2019 - Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, 204–212. doi:10.1145/3289600.3290983

Zheng, Z., Xie, S., Dai, H., Chen, X. ve Wang, H. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. *2017 IEEE International Congress on Big Data (BigData Congress)* içinde (ss. 557–564). doi:10.1109/BigDataCongress.2017.85

Zhou, Q., Huang, H., Zheng, Z. ve Bian, J. (2020). Solutions to Scalability of Blockchain: a Survey. *IEEE Access*, *8*, 16440–16455. doi:10.1109/aCCESS.2020.2967218

Beach, D. (2003, December). A Problem of Validity in Education Research. Qualitative Inquiry, Vol. 9, 859-873. Web: http://www.qix.sagebub.com/cgi/reprint/9/6/859 adresinden erişilmiştir

Okatan, Ayşenur, Balıkçı Erhan. "DNA' nın Kimyasal Yapısı." https://bilimgenc.tubitak.gov.tr/dnanin-kimyasal-yapisi. Erişim [29.03.2022]

Wetterstrand, Kris A. "DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP)." www.genome.gov/sequencingcostsdata. [Erişim 14.03.2022]

Archer, D. (2017). "Revolution and evalution: fully homomorphic encryption." www.genome.gov/sequencingcostsdata. [Erişim 14.04.2022]

url1:https://www.genome.gov/About-Genomics/Introduction-to-Genomics

url2:www.genome.gov/sequencingcostsdata. Erişim [14.03.2022]

url3:https://www.mevzuat.gov.tr/mevzuat?MevzuatNo=6698&MevzuatTur=1&MevzuatTertip=5. Erişim[20.03.2022]

url4: https://gdpr-info.eu/. Erişim[20.03.2022]

url5: https://www.ecfr.gov/current/title-45/subtitle-A/subchapter-C/part-164/subpart-E/section-164.514#p-164.514(b)(2). Erişim[20.03.2022]

url6:https://www.ecfr.gov/current/title-45/subtitle-A/subchapter-C/part-164/subpart-E/section-164.514#p-164.514(b)(1). Erişim[20.03.2022]

url7: https://nebula.org/technology/. Erişim[07.05.2022]

url8: https://www.lunadna.com/individuals/. Erişim[07.05.2022]

url9: https://github.com/genecoin-science/genecoin-science.github.io. Erişim[07.05.2022]

url10: https://ethereum.org/en/whitepaper/. Erişim[08.05.2022]

urd11:https://www.ibm.com/topics/blockchain-ai#:~:text=Using%20blockchain%20to%20store%20and,to%20blockchain%2Dbased%20business%20networks. Erişim[08.05.2022]

url12: https://blokzincir.bilgem.tubitak.gov.tr/blok-zincir.html adresinden 15.05.2022 tarihinde erişilmiştir.

url13: https://ccaf.io/cbeci/index/comparisons. Erişim[19.05.2022]

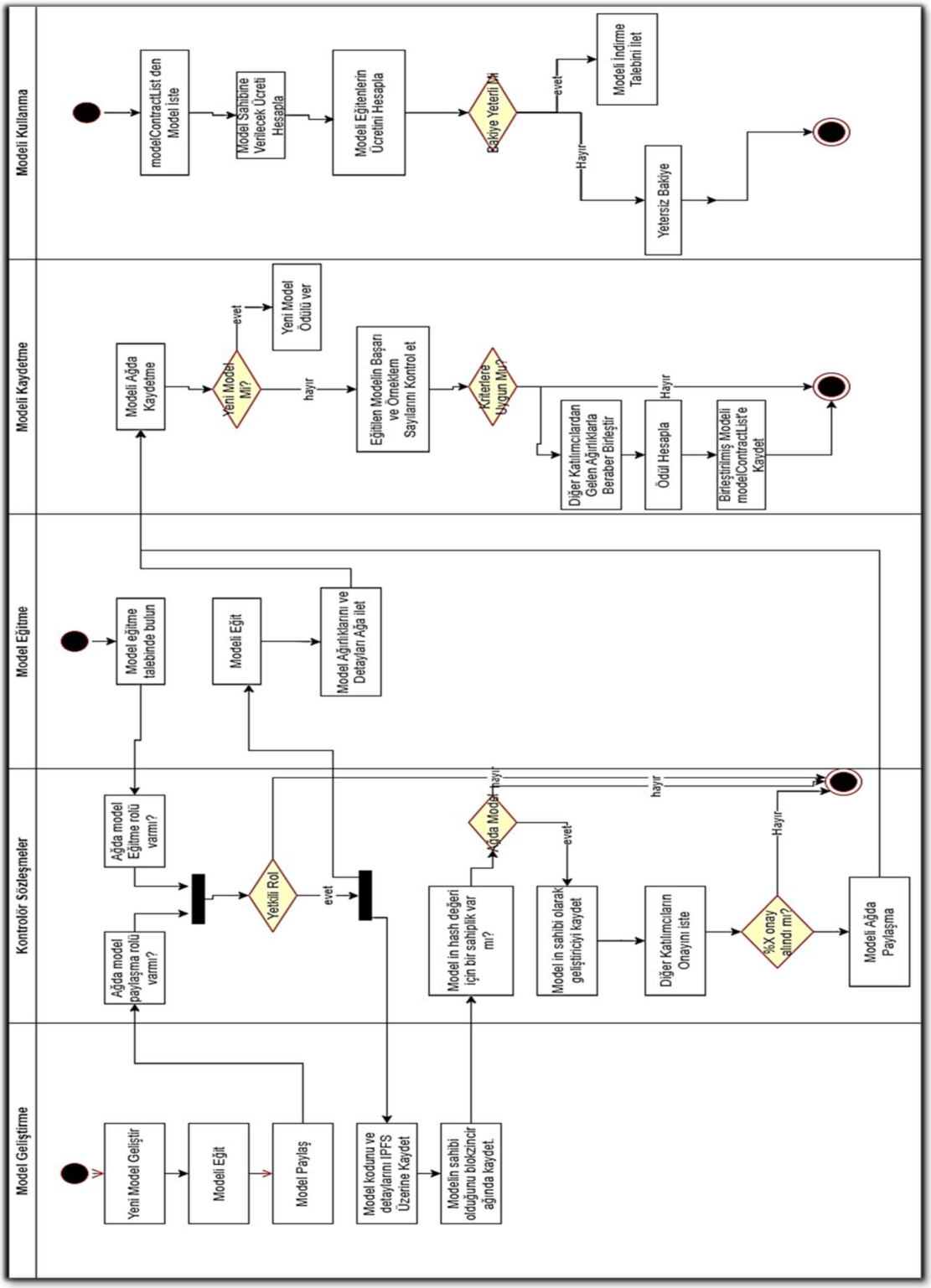
url14:https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html. Erişim[22.05.2022]

url15: https://teknonel.com/tr/web3-nedir/https://blokzincir.bilgem.tubitak.gov.tr/blok-zincir.html adresinden 17.05.2022 tarihinde erişilmiştir.

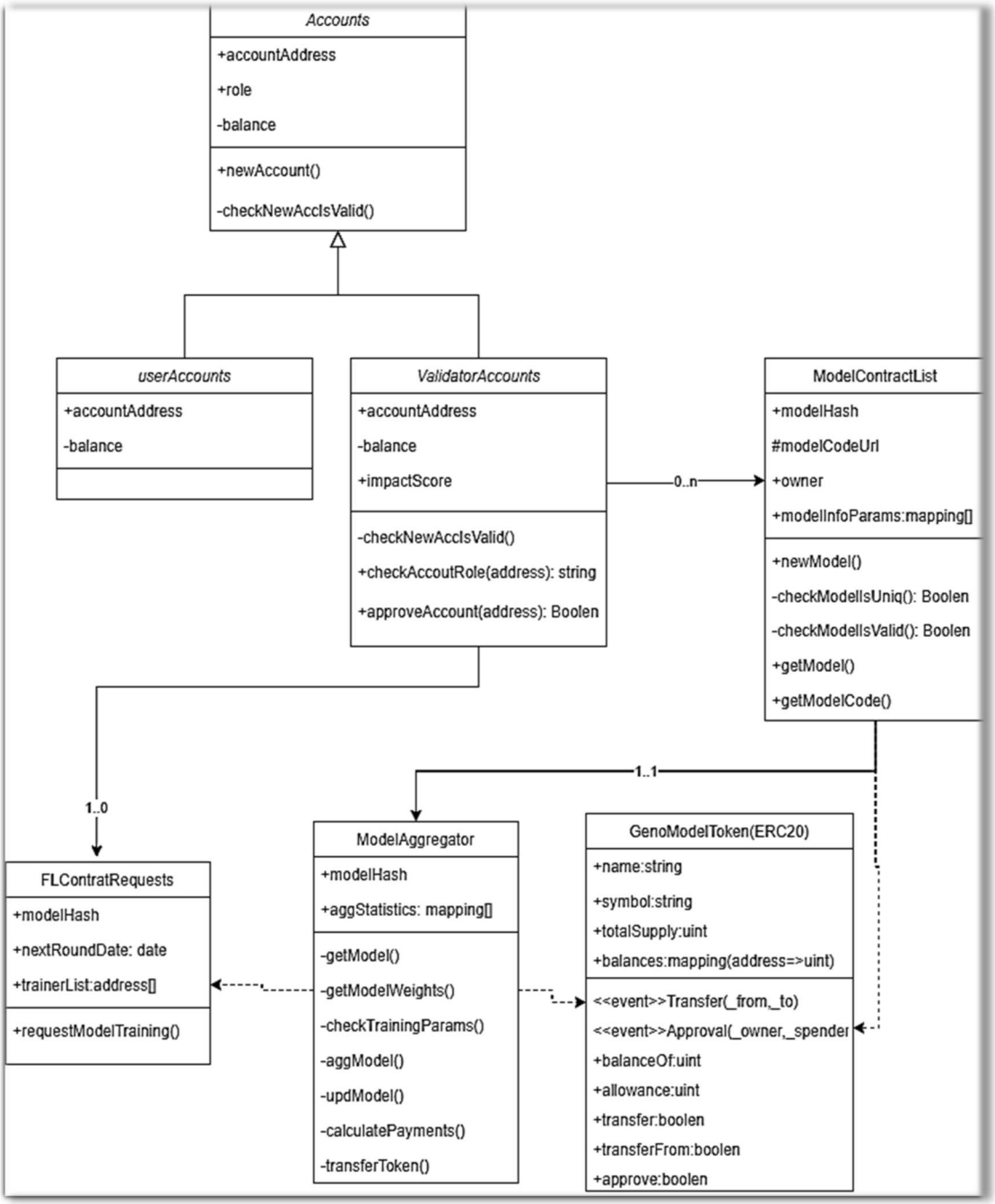
url16: https://bilimgenc.tubitak.gov.tr/dnanin-kimyasal-yapisi adresinden 29.03.2022

EKLER

EK-1. AKTİVİTE DİYAGRAMI



EK-2. ÖNERİLEN MODELİN SINIF DİYAGRAMI



EK-3. SÜREÇ TASARIMI