Dicle University
Journal of Engineering
https://dergipark.org.tr/tr/pub/dumf
duje.dicle.edu.tr

Derleme Makalesi/Review Article

Akıllı Kontrat Geliştirme Süreçlerinde Yapay Zekanın Rolü: Kapsamlı ve Sistematik Bir Literatür Taraması

The Role of Artificial Intelligence in Smart Contract Development Processes: A Comprehensive and Systematic Literature Review

Buğra Ayan[1*], Mutlu Tahsin Üstündağ[2]
[1] Gazi Üniversitesi, Bilişim Sistemleri, bugraayan.com@gmail.com
ORCID: https://orcid.org/ 0000-0002-4915-4970

[2] Gazi Üniversitesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi, mutlutahsin@gazi.edu.tr
ORCID: https://orcid.org/0000-0001-6198-2819

MAKALE BİLGİLERİ

Makale Geçmişi:

Geliş 8 Aralık 2025
Revizyon 11 Mart 2026
Kabul 31 Mart 2026
Online 28 Temmuz 2026

Anahtar Kelimeler:

Akıllı kontratlar,
Blokzincir,
Güvenlik açığı tespiti,
Kod otomasyonu,
Yapay zekâ

ÖZ

Blokzincir teknolojisinin temelini oluşturan akıllı kontratlar, dijital anlaşmaların otomasyonunda önemli bir dönüşüm potansiyeli taşımaktadır. Ancak bu potansiyel, ciddi güvenlik açıkları, yüksek geliştirme maliyetleri ve performans verimsizlikleri gibi önemli zorluklarla gölgelenmektedir. Bu literatür taraması, yapay zeka (AI) ve makine öğrenmesi (ML) teknolojilerinin, akıllı kontrat yaşam döngüsünün her aşamasında bu zorlukları ele almak için nasıl giderek daha merkezi bir rol oynadığını incelemektedir. 2018-2025 yılları arasını kapsayan 38 hakemli akademik makale, konferans bildirisi ve endüstri raporu analiz edilerek, AI'nın akıllı kontrat geliştirme süreçlerindeki uygulamaları dört ana tematik eksende incelenmiştir: (1) Güvenlik Açığı Tespiti ve Analizi, (2) Kod Üretimi ve Otomasyon, (3) Performans Optimizasyonu ve Gaz Maliyeti Azaltma, ve (4) Ekosistem Entegrasyonu ve Sektörel Uygulamalar. Bulgularımız, özellikle Transformer tabanlı modeller (örneğin, CodeBERT) ve Büyük Dil Modelleri (LLM'ler) gibi ileri AI tekniklerinin, statik analiz araçlarının sınırlamalarını aşarak güvenlik açığı tespitinde yüksek doğruluk oranları sunduğunu ortaya koymaktadır. Ayrıca, LLM'lerin doğal dilden otomatik olarak güvenli ve verimli Solidity kodu üretme yeteneği, geliştirme paradigmalarını temelden değiştirme potansiyeli taşımaktadır. Bu çalışma, mevcut metodolojileri, kullanılan veri setlerini ve elde edilen performans metriklerini karşılaştırmalı tablolarla sunmakta, alandaki temel evrimsel eğilimleri (2018'den günümüze) analiz etmekte, önemli araştırma boşluklarını (örneğin, açıklanabilirlik, proje düzeyinde anlayış) belirlemekte ve gelecekteki araştırma yönelimleri için somut öneriler sunmaktadır. Bu kapsamlı analiz hem akademik araştırmacılar hem de endüstri profesyonelleri için mevcut durumu ve gelecekteki potansiyeli gösteren temel bir kaynak olmayı amaçlamaktadır.

ARTICLE INFO

Article history:

Received 8 December 2025
Received in revised form 11 March 2026
Accepted 31 March 2026
Available online 28 July 2026

Keywords:

Artificial intelligence,
Blockchain,
Code automation,
Smart contracts,
Vulnerability detection

Doi: 10.24012/dumf.1838519

* Sorumlu Yazar

ABSTRACT

Smart contracts, which form the foundation of blockchain technology, hold transformative potential in the automation of digital agreements. However, this potential is overshadowed by significant challenges such as serious security vulnerabilities, high development costs, and performance inefficiencies. This literature review provides an examination of how artificial intelligence (AI) and machine learning (ML) technologies are playing an increasingly central role in addressing these challenges at every stage of the smart contract lifecycle. By analyzing 38 peer-reviewed academic papers, conference proceedings, and industry reports covering 2018-2025 period, the applications of AI in smart contract development processes are investigated along four main thematic axes: (1) Vulnerability Detection and Analysis, (2) Code Generation and Automation, (3) Performance Optimization and Gas Cost Reduction, and (4) Ecosystem Integration and Sectoral Applications. Our findings reveal that advanced AI techniques, particularly Transformer-based models (e.g., CodeBERT) and Large Language Models (LLMs), offer high accuracies in vulnerability detection, surpassing the limitations of traditional static analysis tools. Furthermore, the ability of LLMs to automatically generate secure and efficient Solidity code from natural language holds the potential to fundamentally transform development paradigms. This study presents comparative tables summarizing existing methodologies, datasets, and performance metrics, analyzes key evolutionary trends in the field (from 2018 to the present), identifies major research gaps (e.g., explainability, repository-level understanding), and provides concrete recommendations for future research directions. Through this comprehensive analysis, the study aims to serve as a fundamental resource demonstrating both the current state and future potential of the field for academic researchers and industry professionals alike.

Giriş

Blokzincir teknolojisi, 2008'deki başlangıcından bu yana, merkezi olmayan, değişmez ve şeffaf bir veri kayıt sistemi sunarak finans, tedarik zinciri, sağlık ve daha birçok sektörde dönüştürücü bir güç olarak ortaya çıkmıştır. Bu teknolojinin yenilikçi ve etkili uygulamalarından biri, ilk olarak 1994 yılında kriptograf Nick Szabo tarafından kavramsallaştırılan akıllı kontratlardır [1]. Akıllı kontratlar, bir anlaşmanın veya sözleşmenin şartlarının doğrudan kod satırlarına yazıldığı, kendi kendini yürüten programlardır. Bu dijital sözleşmeler, bir blokzincir üzerinde çalışır ve önceden tanımlanmış koşullar karşılandığında aracı olmadan otomatik olarak yürütülür. Ethereum platformunun 2015 yılında piyasaya sürülmesiyle birlikte, akıllı kontratlar teorik bir kavram olmaktan çıkıp, merkezi olmayan uygulamaların ve merkezi olmayan finansın temel yapı taşları haline gelmiştir.

Ancak, akıllı kontratların getirdiği otomasyon ve güvenilirlik vaadi, doğasında var olan zorluklar ve risklerle birlikte gelir. En başta, akıllı kontratların değişmezliği gelir. Bir kontrat blokzincir üzerine dağıtıldıktan sonra, içindeki kodun değiştirilmesi veya güncellenmesi son derece zordur. Bu özellik, güven ve tutarlılık sağlarken, aynı zamanda kodda keşfedilen bir hatanın veya güvenlik açığının kalıcı olabileceği anlamına gelir. Yaşanan çeşitli güvenlik ihlalleri küçük bir kodlama hatasının bile milyonlarca dolarlık finansal kayba yol açabildiğini göstermiştir. Bu durum, akıllı kontrat güvenliğini ekosistemin en öncelikli sorunu haline getirmiştir.

İkinci olarak, akıllı kontrat geliştirme süreci, Solidity gibi özel programlama dillerinde uzmanlık gerektiren karmaşık ve hataya açık bir süreçtir. Geliştiricilerin hem programlama mantığını hem de blokzincirin kendine özgü kısıtlamalarını (örneğin, gaz maliyetleri, blok limiti) göz önünde bulundurması gerekir. Bu yüksek giriş engeli, geliştirici havuzunu sınırlar ve geliştirme sürecini yavaşlatır.

Üçüncü olarak, özellikle Ethereum gibi halka açık blokzincirlerde performans ve maliyet optimizasyonu önemli bir sorundur. Her işlemin bir maliyeti (gaz ücreti) vardır ve verimsiz yazılmış bir kontrat, kullanıcılar için fahiş maliyetlere yol açarak uygulamanın benimsenmesini engelleyebilir [2, 3]. Bu nedenle, gaz optimizasyonu, geliştirme sürecinin ayrılmaz bir parçasıdır.

Bu zorlukların üstesinden gelmek amacıyla yapay zeka ve makine öğrenmesi teknikleri, son yıllarda giderek artan bir ilgi görmektedir. Yapay zekâ, büyük miktarda kod verisinden karmaşık kalıpları öğrenme, anlamsal ilişkileri anlama ve insan benzeri muhakeme yapma yetenekleriyle, akıllı kontrat yaşam döngüsünün her aşamasını iyileştirme potansiyeli sunmaktadır. Bu literatür taraması, 2018'den 2025'e kadar olan dönemi kapsayan 38 temel çalışmayı analiz ederek, AI'nın akıllı kontrat geliştirme süreçlerindeki uygulamalarını dört ana tematik eksende, (1) Güvenlik Açığı Tespiti ve Analizi, (2) Kod Üretimi ve Otomasyon, (3) Performans Optimizasyonu ve Gaz Maliyeti Azaltma, ve (4) Ekosistem Entegrasyonu ve Sektörel Uygulamalar, kapsamlı bir şekilde incelemeyi amaçlamaktadır. Bu tematik çerçeve doğrultusunda, çalışma aşağıdaki temel araştırma sorularına yanıt aramaktadır:

1. Yapay zeka, akıllı kontratlardaki güvenlik açıklarını tespit etme ve analiz etmede geleneksel yöntemlere kıyasla ne gibi üstünlükler sunmaktadır?

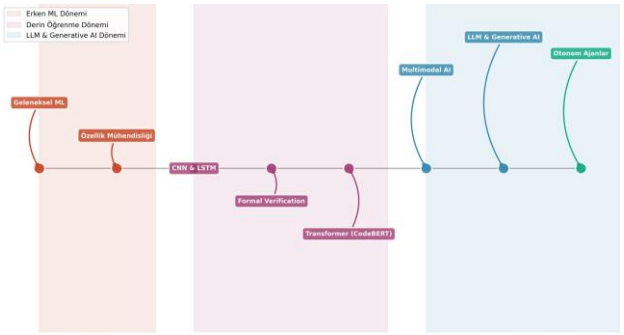
2. Büyük Dil Modelleri (LLM'ler), akıllı kontrat kod üretimini nasıl otomatikleştirmekte ve geliştirme sürecini nasıl dönüştürmektedir?

3. Yapay zeka, akıllı kontratların performansını ve gaz verimliliğini optimize etmek için hangi yöntemleri kullanmaktadır?

4. Alandaki temel metodolojiler, kullanılan veri setleri, performans metrikleri ve evrimsel eğilimler nelerdir?

5. Mevcut araştırmalardaki temel boşluklar nelerdir ve gelecekteki çalışmalar hangi yönlere odaklanmalıdır?

Bu soruları yanıtlamak için, bu tarama, incelenen literatürü dört ana tematik başlık altında yapılandırmaktadır: Güvenlik Açığı Tespiti, Kod Üretimi, Performans Optimizasyonu ve Ekosistem Entegrasyonu. Her bölümde, temel çalışmalar detaylı bir şekilde incelenecek, metodolojileri karşılaştırılacak ve elde edilen sonuçlar sentezlenecektir. Sonuç olarak, bu çalışma, akıllı kontratlar ve yapay zeka kesişimindeki mevcut bilgi birikimini özetleyen ve gelecekteki inovasyonlar için bir yol haritası sunan temel bir kaynak olmayı hedeflemektedir.



Şekil 1. AI ve Akıllı Kontrat Alanındaki Teknolojik Evrim (2018-2025)

Şekil 1'de gösterilen zaman çizelgesi, incelenen literatüre dayanarak AI teknolojilerinin akıllı kontrat alanındaki evrimini üç ana döneme ayırmaktadır: Erken ML Dönemi, Derin Öğrenme Dönemi ve LLM & Generative AI Dönemi. Her dönem, alandaki metodolojik kaymaları ve temel teknolojik gelişmeleri vurgulamaktadır. Alandaki metodolojik yaklaşımın kural tabanlı ve manuel özellik mühendisliğinden, kodun anlamsal yapısını öğrenebilen derin öğrenme modellerine ve nihayetinde üretken yapay zekaya doğru belirgin bir ilerleme kaydetmiştir.

Metodoloji

Araştırma Tasarımı ve Tarama Protokolü

Yapılan çalışma, yapay zekanın akıllı kontrat geliştirme süreçlerindeki rolünü incelemek amacıyla sistematik bir literatür taraması olarak tasarlanmıştır. Tarama süreci, PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) kılavuzunun temel ilkeleri doğrultusunda yürütülmüştür. Araştırma protokolü, aşağıda açıklanan adımları içermektedir.

Veri Tabanları ve Arama Stratejisi

Literatür taraması, Ocak 2018 – Haziran 2025 zaman aralığında, aşağıdaki akademik veri tabanlarında gerçekleştirilmiştir: Scopus, Web of Science (WoS), IEEE Xplore, ACM Digital Library ve Google Scholar. Ayrıca, endüstri uygulamalarına ilişkin güncel bilgilere ulaşmak amacıyla seçici olarak arXiv ön-baskı sunucusu da taranmıştır.

Arama dizgeleri, uluslararası veri tabanlarının İngilizce indeksleme yapısına uygun olarak İngilizce anahtar kelimelerle oluşturulmuş; buna ek olarak, Türkçe literatürü de kapsayabilmek amacıyla temel terimlerin Türkçe karşılıkları arama dizgesine dahil edilmiştir. Kullanılan kombinasyon şu şekildedir: ("smart contract" OR "akıllı kontrat") AND ("artificial intelligence" OR "yapay zeka" OR "machine learning" OR "makine öğrenmesi" OR "deep learning" OR "derin öğrenme" OR "large language model" OR "büyük dil modeli" OR "LLM" OR "natural language processing" OR "doğal dil işleme") AND ("vulnerability detection" OR "güvenlik açığı tespiti" OR "code generation" OR "kod üretimi" OR "gas optimization" OR "gaz optimizasyonu" OR "security analysis" OR "güvenlik analizi" OR "formal verification" OR "formal doğrulama"). Arama, başlık, özet ve anahtar kelime alanlarında uygulanmıştır.

Dahil Etme ve Dışlama Kriterleri

Dahil etme kriterleri: (1) Yapay zeka veya makine öğrenmesi tekniklerini akıllı kontrat geliştirme süreçlerine uygulayan çalışmalar; (2) Hakemli dergilerde, konferanslarda veya tanınmış ön-baskı sunucularında (arXiv) yayımlanmış çalışmalar; (3) 2018–2025 yılları arasında yayımlanmış İngilizce veya Türkçe çalışmalar; (4) Deneysel sonuçlar, karşılaştırmalı analizler veya kapsamlı literatür sentezi sunan çalışmalar.

Dışlama kriterleri: (1) Yalnızca blokzincir genel mimarisine odaklanan, akıllı kontrat geliştirme sürecine doğrudan katkı sunmayan çalışmalar; (2) Tam metni erişilemeyen çalışmalar; (3) Editöre mektup, kısa bildiri veya poster sunumu niteliğindeki çalışmalar; (4) Yapay Zeka bileşeni içermeyen, yalnızca geleneksel statik/dinamik analiz araçlarına odaklanan çalışmalar.

Çalışma Seçim Süreci

Tarama süreci dört aşamada yürütülmüştür. İlk aşamada, belirlenen arama dizgeleri ile veri tabanlarından toplam 247 potansiyel çalışma tespit edilmiştir. İkinci aşamada, mükerrer kayıtların çıkarılmasının ardından 163 benzersiz çalışma elde edilmiştir. Üçüncü aşamada, başlık ve özet incelemesi ile dahil etme/dışlama kriterlerine uymayan çalışmalar elenmiş ve 72 çalışma tam metin incelemesine alınmıştır. Dördüncü ve son aşamada, tam metin değerlendirmesi sonucunda araştırma sorularına doğrudan katkı sağlayan 38 çalışma nihai analize dahil edilmiştir. Bu çalışmalar arasında 18 hakemli akademik makale, 8 konferans bildirisi, 3 endüstri raporu/dokümantasyonu ve 7 arXiv ön baskısı ile 2 diğer kaynak yer almaktadır.

Endüstri araçlarına (ChainGPT, ContractGPT, Workik gibi) ilişkin bilgiler, bu araçların resmi dokümantasyonlarından elde edilmiş olup, bu kaynakların hakemli akademik yayınlar olmadığı ve bu bilgilerin bağımsız olarak doğrulanmadığı okuyucunun dikkatine sunulmaktadır. Bu kaynaklar, alandaki pratik uygulamaların güncel durumunu yansıtmak amacıyla dahil edilmiştir.

Tematik Sınıflandırma

Nihai analize dahil edilen 38 çalışma, araştırma sorularına uygun olarak dört ana tematik eksende sınıflandırılmıştır: (1) Güvenlik Açığı Tespiti ve Analizi, (2) Kod Üretimi ve Otomasyon, (3) Performans Optimizasyonu ve Gaz Maliyeti Azaltma, ve (4) Ekosistem Entegrasyonu ve Sektörel Uygulamalar. Her çalışma, kullanılan AI tekniği, veri seti, performans metrikleri ve temel katkısı açısından analiz edilerek karşılaştırmalı tablolarda sunulmuştur.

Güvenlik Açığı Tespiti ve Analizi

Akıllı kontrat ekosisteminin sürdürülebilirliği ve güvenilirliği, büyük ölçüde bu kontratların güvenliğine bağlıdır. Kodun değişmez doğası, en küçük bir güvenlik açığının bile geri döndürülemez finansal kayıplara yol açmasına neden olabilir. Bu nedenle, güvenlik açığı tespiti, akıllı kontrat geliştirme yaşam döngüsünün en kritik aşamasıdır. Geleneksel olarak bu görev, belirli ve önceden tanımlanmış hata kalıplarını arayan statik ve dinamik analiz araçları (örneğin, Slither, Mythril) tarafından yerine getirilir. Ancak bu araçlar, genellikle yüksek yanlış pozitif ve yanlış negatif oranları üretme, yeni veya daha önce görülmemiş güvenlik açığı türlerini (sıfır gün açıkları) tespit edememe ve kodun anlamsal bütünlüğünü tam olarak kavrayamama gibi önemli sınırlılıklara sahiptir [4]. Yapay zeka, bu sınırlılıkları aşmak için güçlü bir alternatif olarak ortaya çıkmıştır. AI modelleri, büyük kod veri kümelerinden öğrenerek, insan tarafından önceden tanımlanmamış karmaşık kalıpları tanıyabilir ve daha yüksek bir doğrulukla güvenlik açıklarını tespit edebilir.

Makine Öğrenmesi ve Derin Öğrenme Yaklaşımlarının Evrimi

Yapay zeka tabanlı güvenlik açığı tespitindeki ilk çalışmalar, geleneksel makine öğrenmesi modellerine dayanıyordu. Örneğin, Momeni vd. (2019) tarafından yapılan öncü bir çalışma, kod analizcilerinden elde edilen özellikleri kullanarak bir makine öğrenmesi modeli geliştirmiş ve %95 gibi bir doğruluk oranına ulaşmıştır [5]. Bu tür çalışmalar, özellik mühendisliğinin önemini ortaya koymuş ve AI'nın bu alandaki potansiyelini göstermiştir. Benzer şekilde, Crisostomo vd. (2025), Ethereum blokzincir üzerindeki makine öğrenmesi uygulamaları ile akıllı kontrat güvenlik açıkları arasındaki kesişimi kapsamlı bir şekilde inceleyerek bu alandaki temel metodolojileri ortaya koymuştur [6].

Ancak, kodun ham temsilinden doğrudan öğrenebilen derin öğrenme modellerinin ortaya çıkmasıyla gerçekleşmiştir. Derin öğrenme, manuel özellik mühendisliğine olan ihtiyacı ortadan kaldırarak, modellerin kodun sözdizimsel ve anlamsal özelliklerini hiyerarşik bir şekilde öğrenmesine olanak tanır. Huang vd. (2025) tarafından yapılan çalışmada, kontratın makine düzeyindeki kodunu bir görüntüye dönüştürüp bu görüntü üzerinde Evrişimli Sinir Ağları (CNN) kullanmak gibi yaratıcı yaklaşımları içermektedir. Ancak bu yöntemin, kodun yapısal ve anlamsal bilgilerinin bir kısmını kaybetme riski taşımakta olduğu belirtilmiştir [7].

Transformer Modelleri ve Multimodal Yaklaşımlar

Transformer mimarisinin, dil anlama görevlerindeki başarısından hareketle kod alanına uyarlanması bu sürece katkı sağlamıştır. CodeBERT gibi ön-eğitimli modeller, büyük kod veri kümeleri üzerinde eğitilerek, programlama dillerinin hem doğal dil benzeri (sözdizimsel) hem de yapısal (anlamsal) özelliklerini derinlemesine anlamayı öğrenmiştir [8, 9]. Tang vd. (2023) tarafından geliştirilen "Lightning Cat" çözümü, bu yaklaşımın gücünü açıkça ortaya koymaktadır. Bu çalışma, Optimized-CodeBERT, Optimized-LSTM ve Optimized-CNN modellerini karşılaştırmış ve CodeBERT tabanlı modelin, SolidiFI-benchmark veri setinde %93.53'lük bir f1-skoru ile diğerlerini geride bıraktığını göstermiştir [4]. Bu başarıya CodeBERT'in kodun anlamını daha derinlemesine kavraması ve araştırmacıların yalnızca kritik güvenlik açığı içeren kod segmentlerine odaklanarak modelin eğitimindeki gürültüyü azaltması ile ulaşılmıştır.

Güncel teknolojik yaklaşımlar, tek bir veri temsiline güvenmek yerine, multimodal bir yaklaşım benimsemektedir. Bu yaklaşımlar, bir akıllı kontratı birden çok açıdan temsil eden farklı veri türlerini (modaliteleri) birleştirir. Örneğin, ham kaynak kodu, kontrol akış grafiği (CFG), soyut sözdizimi ağacı (AST) ve işlem grafiği gibi farklı temsiller bir arada kullanılır. Jie vd. (2023) tarafından önerilen ve %99.71 gibi olağanüstü bir tespit performansına ulaşan multimodal AI iskeleti, bu yaklaşımın zirvesini temsil etmektedir [10]. Bu çalışma, hem tek bir modalite içindeki özellikleri hem de farklı modaliteler arasındaki ilişkileri birleştiren bir "beyaz kutu" metodolojisi sunmaktadır. Bu, modelin sadece bir kod satırına değil, aynı zamanda bu kodun programın genel akışındaki ve veri bağımlılıklarındaki yerine de bakmasını sağlar. Benzer şekilde, Deng vd. (2023) de derin öğrenme ile multimodal karar füzyonunu birleştiren bir yöntem önermiştir [11].

Formal Doğrulama ile Hibrit Sistemler

Yüksek doğruluk oranlarına rağmen, derin öğrenme modelleri doğaları gereği olasılıksaldır ve %100 garanti sunmazlar. Bu noktada, bir programın belirli bir spesifikasyona göre matematiksel olarak doğru çalıştığını kanıtlayan formal doğrulama devreye girer. Tolmach vd. (2021) ve Murray & Anisi (2019) tarafından yapılan kapsamlı anket çalışmaları, formal metodların akıllı kontrat güvenliğindeki artan önemini ve çeşitliliğini göstermektedir [12, 13].

Araştırma yönelimlerinden en önemlilerinden biri, derin öğrenmenin sezgisel kalıp tanıma gücünü, formal doğrulamanın matematiksel kesinliği ile birleştiren hibrit sistemlerdir. Örneğin, Chen & Li (2024), Ponzi şeması içeren kontratları tespit etmek için derin öğrenmeyi formal doğrulama ile entegre eden bir yöntem geliştirmiştir [14]. Bu yaklaşımda, derin öğrenme modeli potansiyel olarak şüpheli kontratları işaretlerken, formal doğrulama modülü bu kontratların gerçekten de bir Ponzi şemasının mantıksal özelliklerini taşıyıp taşımadığını matematiksel olarak kanıtlar. Benzer şekilde, Guo vd. (2025), akıllı kontrat davranışını Geçiş Sistemleri (TS) ve Hesaplama Ağacı Mantığı (CTL) gibi yöntemlerle modelleyerek doğrulamayı blokzincir seviyesine taşımıştır [15].

Güvenlik Analizinde Üretken Yapay Zeka (Generative AI)

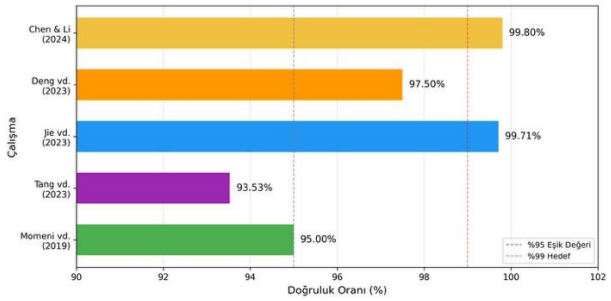
Yapay zeka, güvenlik analizinde sadece savunma amaçlı değil, aynı zamanda saldırı amaçlı da kullanılmaktadır. Güvenlik araştırmacıları, potansiyel güvenlik açıklarını daha etkili bir şekilde bulmak için AI modellerini kullanarak otomatik olarak saldırı kodu üretmektedir. Gervais vd. (2025) tarafından geliştirilen bir AI ajanı, LLM'leri kullanarak akıllı kontratlar için exploit'ler üretmiş ve %63'lük bir başarı oranıyla 8.59 milyon dolara kadar varlık çıkarabilen saldırılar gerçekleştirmiştir [16]. Bu tür çalışmalar, savunma mekanizmalarını test etmek ve potansiyel saldırı vektörlerini proaktif olarak belirlemek için paha biçilmezdir. Benzer şekilde, kötü niyetli geliştirici niyetlerini tespit etmeye yönelik çalışmalar da, CodeBERT ve GPT-4.1 gibi modelleri kullanarak kodun sadece ne yaptığını değil, aynı zamanda neden yaptığını da anlamaya çalışmaktadır [7].

Tablo 1, güvenlik açığı tespiti alanındaki temel çalışmaları yöntem, performans ve katkı boyutlarında karşılaştırmalı biçimde sunmaktadır. Tabloda yer alan yedi çalışma, 2018-2025 yılları arasındaki metodolojik dönüşümü açıkça yansıtmaktadır. Performans açısından değerlendirildiğinde, erken dönem makine öğrenmesi yaklaşımlarının (Momeni vd., 2019) %95 doğruluk oranı elde ettiği, buna karşın derin öğrenme ve Transformer tabanlı modellerin devreye girmesiyle bu oranın belirgin biçimde artış kaydettiği görülmektedir. Nitekim Jie vd. (2023) tarafından önerilen multimodal AI iskeleti %99,71, Chen & Li (2024) tarafından geliştirilen hibrit sistem ise %99,80 doğruluk oranına ulaşmış; bu durum, birden fazla veri temsilini ve yöntemi bir arada kullanan yaklaşımların üstünlüğünü ortaya koymaktadır. Kullanılan veri setleri incelendiğinde, SmartEmbed ve SolidiFI-benchmark gibi standartlaşmış veri setlerinin modeller arası karşılaştırılabilirliği mümkün kıldığı; bununla birlikte bazı çalışmaların belirli bir veri setine dayanmamasının genel geçer değerlendirme yapılmasını güçleştirdiği anlaşılmaktadır. En fazla atıf alan çalışma, formal doğrulama yöntemlerini kapsamlı biçimde ele alan Tolmach vd. (2021) çalışmasıdır (296 atıf). Bu durum, matematiksel kesinlik sunan formal yöntemlerin alandaki temel referans niteliğini koruduğuna işaret etmektedir. Öte yandan, LLM tabanlı saldırı kodu üretimini ele alan Gervais vd. (2025) çalışması yalnızca 3 atıfla temsil edilmekte olup bu, söz konusu yaklaşımın güncelliğini yansıtmaktadır.

Sonuç olarak Tablo 1, alandaki araştırmaların tek modelli ve kural tabanlı yöntemlerden, multimodal ve hibrit mimarilere doğru evrildiğini; güvenlik açığı tespitinde doğruluğun sürekli artmasına karşın model açıklanabilirliği ve gerçek dünya genellemesi gibi sorunların hâlâ öncelikli araştırma konuları arasında yer aldığını göstermektedir.

Tablo 1. Güvenlik Açığı Tespiti Alanındaki Temel Çalışmaların Karşılaştırması

Çalışma	Yıl	Yöntem	Model/Teknik	Performans	Veri Seti	Atıf	Katkı
Jie vd. [10]	2023	Multimodal AI	bi-LSTM, TextCNN, RF	%99.71 Doğruluk	SmartEmbed	56	Kod ve grafikleri birleştiren beyaz kutu metodolojisi, en yüksek performans.
Momeni vd. [5]	2019	Makine Öğrenmesi	Kod Analizcileri	%95 Doğruluk	Ethereum Kontratları	111	Erken dönem başarılı ML uygulaması, özellik mühendisliğinin önemi.
Tang vd. [4]	2023	Derin Öğrenme	Optimized-CodeBERT	%93.53 F1-Skoru	SolidiFI	53	CodeBERT'in üstünlüğünü ve kritik kod segmenti kavramını gösterdi.
Chen & Li [14]	2024	Hibrit Sistem	DL + Formal Verification	%99.80 Doğruluk	Ponzi Şemaları	6	Derin öğrenme ve formal doğrulamayı birleştiren hibrit yaklaşım.
Gervais vd. [16]	2025	Generative AI	LLM Exploit Gen.	%63 Başarı	VERITE	3	Savunmaları test etmek için otomatik istismar kodu üreten AI ajanı.
Tolmach vd. [12]	2021	Survey	Formal Methods	-	-	296	Formal doğrulama alanındaki en kapsamlı ve en çok atıf alan anket.
Deng vd. [11]	2023	Derin Öğrenme	DL + Multimodal Karar Füzyonu	97.50 Doğruluk	-	23	Derin öğrenme ile multimodal karar füzyonunu birleştiren güvenlik açığı tespit yöntemi.



Şekil 2. Güvenlik Açığı Tespiti: Performans Karşılaştırması

Şekil 2'deki grafik, güvenlik açığı tespiti alanındaki yapılan temel çalışmaların doğruluk oranlarını karşılaştırmaktadır. 2019'daki geleneksel ML yaklaşımından (%95, Momeni vd.) 2023'teki multimodal AI yaklaşımına (%99.71, Jie vd.) ve 2024 yılındaki DL yaklaşımına (%99.80, Chen & Li) doğru belirgin bir performans artışının yaşanmış ve özellikle Transformer tabanlı ve multimodal modellerin bu alanda başarı sağlamıştır.

Akıllı Kontrat Kod Üretimi ve Otomasyon

Akıllı kontrat geliştirme süreci, geleneksel olarak yüksek düzeyde uzmanlık gerektiren, zaman alıcı ve insan hatasına açık bir süreçtir. Geliştiricilerin, Solidity gibi dillere hâkim olmaları, en iyi güvenlik uygulamalarını bilmeleri ve gaz optimizasyonu gibi blokzincire özgü kısıtlamaları sürekli göz önünde bulundurmaları gerekir. Yapay Zeka, özellikle de Büyük Dil Modelleri bu süreci otomatize ederek, geliştirici verimliliğini artırarak ve hatta kodlama bilgisi olmayan paydaşların bile geliştirme sürecine dahil olmasını sağlayarak bir paradigma kayması yaratma potansiyeline sahiptir.

Büyük Dil Modelleri (LLM) ile Kod Üretimi

LLM'lerin en dönüştürücü uygulaması, doğal dil tanımlarından, yasal metinlerden veya tanımlamalardan doğrudan işlevsel ve güvenli akıllı kontrat kodu üretme yeteneğidir. Bu, geliştirme döngüsünü haftalardan veya günlerden saatlere veya dakikalara indirme potansiyeli taşır. Bu alandaki araştırmalar, LLM'lerin bu görevdeki yeteneklerini ve sınırlılıklarını değerlendirmeye odaklanmıştır.

Bu değerlendirmeyi karşılaştırılabilir hale getirmek için benchmarklar önemli bir rol oynamaktadır. Peng vd. (2025) tarafından geliştirilen SolEval, LLM'lerin Solidity kodu üretme yeteneklerini değerlendirmek için tasarlanmış ilk proje düzeyindeki benchmarktır [17]. Tekil fonksiyonlar yerine bütün bir proje bağlamını dikkate alan SolEval, modellerin sadece doğru kod üretip üretmediğini değil, aynı zamanda bu kodun projenin diğer bileşenleriyle ne kadar uyumlu olduğunu da ölçer. Bu, LLM'lerin gerçek dünya geliştirme senaryolarındaki pratikliğini değerlendirmek için önemli bir adımdır.

Araştırmacılar, LLM'leri kullanarak kod üretmek için çeşitli metodolojiler önermiştir. De Vito vd. (2025), doğal dil girdilerinden Solidity kodu üretmek için üç farklı strateji sunmuştur: (1) Doğrudan LLM'e Girdi Verme, (2) Girdiyi Örneklerle Zenginleştirme ve (3) Üretilen Kodu Geri Besleme Döngüsüyle İyileştirme [18]. Bu tür çalışmalar, LLM'lerle etkileşimin kalitesinin, üretilen kodun kalitesini doğrudan etkilediğini göstermektedir. Benzer şekilde, Napoli vd. (2025), LLM'lerin geleneksel yasal anlaşma metinlerini

analiz ederek bunları otomatik olarak yürütülebilir akıllı kontratlara dönüştürme potansiyelini araştırmıştır [19]. Bu, hukuk ve teknoloji arasındaki boşluğu doldurma potansiyeli taşıyan önemli bir araştırma yönüdür.

Bu alandaki genel eğilim, LLM'lerin sadece birer "kod yazıcı" olarak değil, aynı zamanda birer "geliştirme ortağı" olarak konumlandırılmasıdır. He vd. (2024) tarafından yapılan kapsamlı literatür taraması, LLM'lerin blokzincir güvenliğindeki rolünü incelerken, kod üretiminin bu ekosistemin ayrılmaz bir parçası olduğunu vurgulamaktadır [20]. Khalid & Brown (2025) ise üretken AI araçlarının genel yeteneklerini ve geliştiriciler için sunduğu perspektifleri değerlendirmiştir [21].

Endüstri Araçları ve Platformlar

Akademik araştırmaların yanı sıra, endüstri de geliştiricilerin kullanımına yönelik pratik araçlar ve platformlar sunarak bu alanda hızla ilerlemektedir. Bu araçlar, LLM'lerin gücünü soyutlayarak, geliştiricilerin karmaşık AI modelleriyle doğrudan uğraşmadan kod üretme ve analiz etme yeteneklerinden faydalanmasını sağlar.

• ChainGPT AI Smart-Contract Generator: Bu platform, kullanıcıların doğal dil girdileriyle akıllı kontratlar oluşturmasına olanak tanır. Modelin, en iyi güvenlik uygulamaları, standart kütüphaneler ve standartlaştırma örnekleri gibi geniş bir bilgi tabanıyla beslendiği belirtilmektedir [22].

• ContractGPT: HackQuest platformu tarafından sunulan bu araç, CodeBERT modeli üzerine inşa edilmiştir ve geliştiricilerin merkezi olmayan uygulamalar için güvenli ve hızlı bir şekilde akıllı kontratlar oluşturmasına yardımcı olmayı amaçlar [23].

• Workik Smart Contract Generator: Bu araç, sadece kod üretmekle kalmaz, aynı zamanda mevcut kodu yeniden düzenleme, gaz kullanımını optimize etme, güvenlik açıkları için hata ayıklama ve Chainlink gibi harici veri sağlayıcılara entegrasyon gibi bir dizi ek özellik sunar [24].

Bu araçlar, AI tabanlı akıllı kontrat geliştirmenin artık teorik bir olasılık olmadığını, geliştiricilerin günlük iş akışlarına entegre edebilecekleri pratik bir gerçeklik olduğunu göstermektedir. Bununla birlikte, bu endüstri araçlarına ilişkin bilgilerin büyük ölçüde araçların kendi resmi dokümantasyonlarından elde edildiği ve bağımsız hakemli akademik çalışmalarla henüz kapsamlı bir şekilde doğrulanmadığı belirtilmelidir. Bu nedenle, söz konusu araçların performansına ve yeteneklerine dair iddiaların, gelecekte bağımsız deneysel değerlendirmelerle desteklenmesi gerekmektedir.

Kod Otomasyonu ve Veri Artırımı

Yapay zekanın otomasyon yetenekleri, tam kod üretiminin ötesine geçmektedir. Geliştirme sürecindeki diğer tekrarlayan ve zaman alıcı görevler de AI tarafından otomatikleştirilebilir. Örneğin, bir çalışma, LLM'lerin ve bağlam içi öğrenmenin, akıllı kontratlar için otomatik olarak açıklayıcı yorumlar üretmek için nasıl kullanılabileceğini göstermiştir [26]. Bu yaklaşım kodun okunabilirliğini, anlaşılırlığını ve uzun vadeli sürdürülebilirliğini artırır.

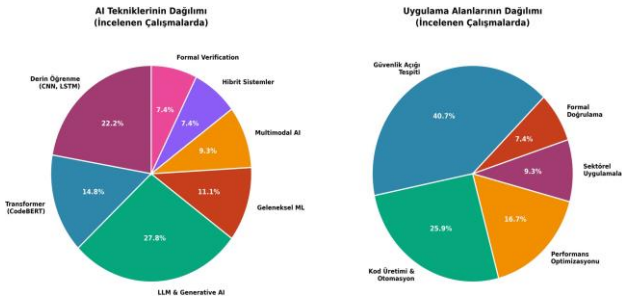
AI'nın bir diğer güçlü uygulaması ise veri artırımı alanındadır. Güvenlik açığı tespiti gibi denetimli öğrenme görevleri için yüksek kaliteli ve büyük miktarda etiketli veriye ihtiyaç vardır. CGGNet gibi derleyici kılavuzlu üretim ağları, bu sorunu çözmek için tasarlanmıştır [27]. Bu tür bir sistem, mevcut binlerce geçerli akıllı kontratı analiz ederek ve derleyici düzeyinde manipülasyonlar yaparak, dilin kurallarına uygun milyonlarca yeni, benzersiz ve geçerli sentetik akıllı kontrat üretebilir. Bu sentetik veriler, daha sonra güvenlik açığı tespit modellerini eğitmek için kullanılarak modellerin daha sağlam ve genelleştirilebilir olması sağlanır.

Tablo 2. Kod Üretimi ve Otomasyon Alanındaki Temel Yaklaşımlar

Çalışma/Araç	Yıl	Model/Teknik	Odak Alanı	Temel Katkı
Peng vd. (SolEval) [17]	2025	Çeşitli LLM'ler	Kod Üretimi Benchmark	Solidity için ilk proje düzeyinde benchmark oluşturdu.
De Vito vd. [18]	2025	LLM	Doğal Dil → Solidity	Kod üretimi için Naive, Augmented ve Enhanced gibi farklı stratejiler önerdi.
Napoli vd. [19]	2025	LLM	Yasal Metin → Kontrat	Yasal anlaşmalardan otomatik olarak akıllı kontrat üretme potansiyelini araştırdı.
CGGNet [26]	2024	Derleyici Kılavuzlu Ağ	Veri Artırımı	Milyonlarca sentetik ve geçerli akıllı kontrat üreterek eğitim verisi sorununu çözdü.
Chain-GPT [22, 24]	2025	İnce Ayarlı LLM	Endüstri Platformu	Geliştiricilere yönelik doğal dilden kod üreten pratik bir araç sundu.

ContractGPT [23]	2025	CodeBERT	Endüstri Platformu	Güvenli ve hızlı kontrat geliştirmeye odaklanan bir AI aracı sağladı.

Tablo 2, kod üretimi ve otomasyon alanındaki temel yaklaşımları model türü, odak alanı ve katkı boyutlarında karşılaştırmalı biçimde sunmaktadır. 2024-2025 yıllarına ait olan çalışmalar alandaki güncel yönelimleri yansıtmaktadır. Odak alanları incelendiğinde, çalışmaların doğal dilden kod üretimi, yasal metinlerin akıllı kontratlara dönüştürülmesi, veri artırımı ve endüstri platformları gibi birbirini tamamlayan farklı problemlere yöneldiği görülmektedir. Bu çeşitlilik, kod üretimi ve otomasyon araştırmalarının yalnızca teknik doğrulukla sınırlı kalmayıp geliştirme sürecinin daha geniş bir perspektiften ele alındığına işaret etmektedir. Kullanılan modeller açısından değerlendirildiğinde, LLM tabanlı yaklaşımların baskın olduğu dikkat çekmektedir. Bununla birlikte CGGNet'in derleyici kılavuzlu bir ağ mimarisi benimsemesi, veri artırımı problemine farklı bir çözüm yolu sunması bakımından dikkat çekicidir. Milyonlarca sentetik ve geçerli akıllı kontrat üretebilen bu yaklaşımın, etiketli veri kıtlığı sorununu ele alması açısından alanda önemli bir boşluğu doldurduğu anlaşılmaktadır. Akademik çalışmalar ile endüstri araçları arasındaki denge de tablonun öne çıkan bir özelliğidir. Peng vd., De Vito vd. ve Napoli vd. gibi akademik çalışmalar metodolojik katkılara odaklanırken; Chain-GPT ve ContractGPT gibi endüstri platformlarının bu yöntemleri geliştiricilerin kullanımına sunduğu gözükmektedir. Bu durum, alandaki araştırma bulgularının görece hızlı biçimde pratiğe aktarılabildiğine işaret etmektedir. Kod üretimi ve otomasyon alanının hem metodolojik hem de uygulama boyutunda hızla olgunlaştığı; bununla birlikte üretilen kodun güvenliği, model genellenebilirliği ve bağımsız doğrulama gibi konuların henüz yeterince ele alınmadığı anlaşılmaktadır.



Şekil 3. AI Tekniklerinin ve Uygulama Alanlarının Dağılımı

Şekil 3, incelenen 38 çalışmanın iki farklı boyuttaki dağılımını grafiklerle sunmaktadır. Sol grafik, kullanılan AI tekniklerinin oransal dağılımını göstermekte olup, Derin Öğrenme (%33.3) ve LLM & Generative AI (%27.4) yaklaşımlarının toplamda çalışmaların %60'ından fazlasını oluşturduğu görülmektedir. Sağdaki grafikte ise Güvenlik Açığı Tespiti (%40.7) araştırmaların uygulama alanlarına göre dağılımı en yoğun çalışılan alan olarak öne çıkmaktadır. Alandaki araştırma çabalarının ağırlıklı olarak güvenlik odaklı olduğu ve ileri düzey AI tekniklerinin (Derin Öğrenme, LLM) geleneksel yöntemlere kıyasla baskın hale geldiği gözükmektedir.

Performans Optimizasyonu ve Gaz Maliyeti Azaltma

Akıllı kontratların pratik uygulanabilirliği, sadece güvenliklerine değil, aynı zamanda verimliliklerine de bağlıdır. Özellikle Ethereum gibi halka açık ve yoğun kullanılan blokzincir ağlarında, her bir hesaplama adımı ve depolama işlemi, "gaz" olarak adlandırılan bir birimle ölçülen bir maliyete sahiptir. Bu gaz maliyeti, ağın o anki yoğunluğuna göre değişen bir gaz fiyatı ile çarpılarak kullanıcının ödeyeceği işlem ücretini belirler. Verimsiz yazılmış bir akıllı kontrat, basit bir işlem için bile fahiş maliyetlere yol açabilir, bu da uygulamanın benimsenmesinin önünde büyük bir engel teşkil eder. Bu nedenle, gaz optimizasyonu, akıllı kontrat geliştirme sürecinin en önemli ve zorlu yönlerinden biridir. Yapay zeka, bu sorunu iki temel stratejiyle ele almaktadır: (1) İşlem ücretlerini tahmin ederek ve zamanlamayı optimize ederek maliyetleri düşürmek ve (2) Kodun kendisini analiz ederek ve yeniden düzenleyerek gaz tüketimini azaltmak.

Gaz Fiyatı Tahmini ve İşlem Zamanlaması

Gaz fiyatları, ağ talebine göre dinamik olarak dalgalanır. Bir işlemi en düşük maliyetle gerçekleştirmek için doğru anı yakalamak, kullanıcılar için önemli bir tasarruf sağlayabilir. Makine öğrenmesi modelleri, geçmiş işlem verilerini, bekleyen işlem havuzunun durumunu, blok doluluk oranlarını ve diğer ağ metriklerini analiz ederek gelecekteki gaz fiyatlarını tahmin etme yeteneğine sahiptir.

Bu alandaki önemli çalışmalardan biri, Laurent vd. (2022) tarafından yapılmıştır. Bu çalışma, bir işlemin belirli bir gaz fiyatıyla ne kadar sürede onaylanacağını tahmin etmek için bir makine öğrenmesi yaklaşımı önermiştir ve bu alanda 62 atıf alarak temel referanslardan biri haline gelmiştir [27]. Benzer şekilde, Mars vd. (2021), kullanıcılara o anki koşullar için en uygun gaz fiyatını öneren bir tavsiye sistemi geliştirmiştir [28]. Bu tür öngörüsel sistemler, cüzdan yazılımlarına entegre edilerek, kullanıcıların "gaz savaşı" olarak bilinen yüksek maliyetli dönemlerden kaçınmasına ve işlemlerini daha ekonomik bir şekilde zamanlamasına olanak tanır. Yakın tarihli bir çalışma ise, bu tahmin modellerini daha da ileri taşıyarak, gaz ücreti dalgalanmalarını tahmin etmek ve işlem gönderimini stratejik olarak optimize etmek için tahminsel ve uyarlanabilir bir makine öğrenmesi metodolojisi sunmuştur [29].

Otomatik Kod Optimizasyonu ve Patern Tespiti

Gaz maliyetini düşürmenin daha temel bir yolu, akıllı kontrat kodunun kendisini daha verimli hale getirmektir. Geliştiriciler genellikle manuel olarak bir dizi en iyi

uygulamayı takip etmeye çalışır; örneğin, depolama yerine bellek kullanmak, gereksiz hesaplamalardan kaçınmak veya verimli veri türleri seçmek gibi. Ancak bu süreç hem uzmanlık gerektirir hem de hataya açıktır. Yapay zeka, bu optimizasyon sürecini otomatikleştirmek için güçlü yetenekler sunar.

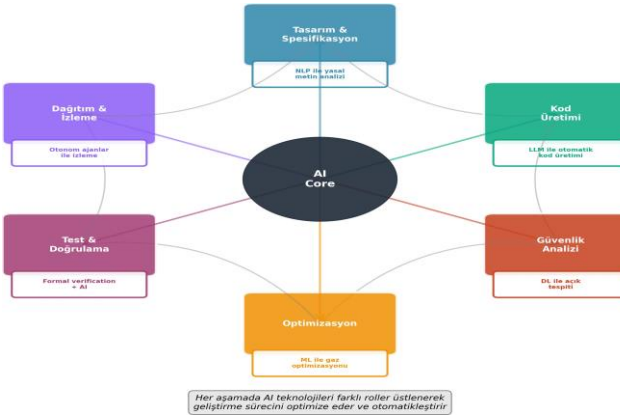
AI modelleri, binlerce mevcut akıllı kontratı analiz ederek, gereksiz yere yüksek gaz maliyetine neden olan yaygın kodlama kalıplarını tespit etmeyi öğrenebilir. Li vd. (2023) tarafından yapılan çalışma, bu tür gaz-maliyet kalıpların otomatik olarak tespit edilmesine odaklanmıştır [2]. Bir model bu kalıpları öğrendikten sonra, yeni bir kontrat kodunu analiz ederek geliştiriciye optimizasyon önerilerinde bulunabilir. Bu öneriler, "Bu döngüde SSTORE işlem kodunu kullanmak yerine, verileri bir bellek dizisinde toplayıp döngü sonunda tek bir SSTORE ile yazmak gaz maliyetini %X oranında azaltabilir" gibi somut tavsiyeler içerebilir. Bu alandaki temel bilgiler, Khanzadeh vd. (2023) tarafından belgelenen yaklaşık 28 gaz verimli kalıp [30] veya daha genel tasarım kalıpları [3, 31] gibi çalışmalarda özetlenmiştir.

Daha da ileri bir yaklaşım, sadece tespit ve öneri ile kalmayıp, optimizasyonu otomatik olarak uygulayan sistemlerdir. GasAgent (2025), bu alandaki en yenilikçi yaklaşımlardan biridir. Bu çalışma, LLM'leri kullanan bir çoklu ajan mimarisi sunarak, akıllı kontratlarda gaz optimizasyonunu tamamen otomatikleştirmeyi amaçlamaktadır [32]. Bu sistemde, farklı AI ajanları kodun farklı yönlerini analiz eder, optimizasyon stratejileri önerir ve bu stratejileri uygulayarak kodun yeni, daha verimli bir versiyonunu üretir. Benzer şekilde, üretken AI (Generative AI) da bu alanda kullanılmaktadır. Misbah vd. (2025), akıllı kontrat üretimini daha dinamik hale getiren ve güvenliğin yanı sıra ölçeklenebilirliği de gözeten üretken bir AI yaklaşımı önermiştir [33]. Bu tür sistemler, geliştiricinin yükünü önemli ölçüde azaltarak, en başından itibaren optimize edilmiş kontratların üretilmesini sağlar.

Tablo 3. Performans Optimizasyonu ve Gaz Maliyeti Azaltma Yaklaşımları

Yaklaşım	Amaç	Kullanılan Teknik	Örnek Çalışma /Araç	Temel Katkı
Fiyat Tahmini	İşlem ücretlerini düşürmek	Zaman Serisi Analizi, Regresyon (ML)	Laurent vd. [27], Mars vd. [28]	Kullanıcıların işlemlerini en ucuz zamanda yapmalarını sağlar.
Patern Tespiti	Verimsiz kodu belirlemek	Sınıflandırma, Kalıp Tanıma (ML/DL)	Li vd. [2]	Geliştiricilere kodlarındaki gaz-pahalı anti-pattern'leri gösterir.
Otomatik Optimizasyon	Kodu otomatik yeniden yazmak	Generative AI, Çok-Agent LLM'ler	GasAgent [32], Misbah vd. [33]	Optimizasyon sürecini tamamen otomatize ederek geliştirici yükünü azaltır.
	Tasarım Kalıbı Dok.	En iyi uygulamaları özetlemek	Manuel/ AI Destekli Analiz	Khanzadeh vd. [30]

Tablo 3, performans optimizasyonu ve gaz maliyeti azaltma alanındaki temel yaklaşımları amaç, kullanılan teknik ve katkı boyutlarında karşılaştırmalı biçimde ele almaktadır. Tabloda yer alan dört yaklaşım, gaz optimizasyonu probleminin farklı boyutlarına yönelik çeşitli çözüm yollarını yansıtmaktadır. Yaklaşımlar incelendiğinde, fiyat tahmini, patern tespiti ve otomatik optimizasyon olmak üzere birbirini tamamlayan üç temel stratejinin benimsendiği anlaşılmaktadır. Fiyat tahminine yönelik çalışmaların (Laurent vd. [27]; Mars vd. [28]) zaman serisi analizi ve regresyon gibi geleneksel makine öğrenmesi yöntemlerine dayandığı, bu yöntemlerin kullanıcıların işlemlerini en düşük maliyetle gerçekleştirebileceği zaman dilimlerini belirlemeye olanak tanıdığı görülmektedir. Patern tespiti yaklaşımının (Li vd. [2]) ise sınıflandırma ve kalıp tanıma teknikleri aracılığıyla kodun kendisindeki verimsizlikleri belirlemeye odaklandığı anlaşılmaktadır. Bu yaklaşımın, geliştiricilere doğrudan geri bildirim sunması bakımından pratik bir değer taşıdığı değerlendirilmektedir. Otomatik optimizasyon yaklaşımı (GasAgent [32]; Misbah vd. [33]) ise yalnızca tespit ve öneriyle sınırlı kalmayıp kodu doğrudan yeniden düzenleyen çok-ajan LLM mimarileri aracılığıyla optimizasyon sürecini bütünüyle otomatize etmeyi amaçlaması bakımından tablodaki en ileri yöntem olarak öne çıkmaktadır. Tasarım kalıbı dokümantasyonu ise Khanzadeh vd. [30] tarafından derlenen yaklaşık 28 gaz verimli kalıbı kapsayan rehber niteliğindeki çalışmasıyla, geliştirme sürecinde manuel ve yapay zeka destekli analizin bir arada kullanılmasına zemin hazırlayan tamamlayıcı bir kaynak işlevi görmektedir. Sonuç olarak Tablo 3, gaz optimizasyonu araştırmalarının tahmin odaklı yaklaşımlardan kod düzeyinde otomatik müdahaleye doğru ilerlediğini ortaya koymaktadır. Bununla birlikte, önerilen yöntemlerin gerçek dünya koşullarındaki etkinliğini ölçen kapsamlı deneysel değerlendirmelerin literatürde hâlâ sınırlı kaldığı anlaşılmaktadır.



Şekil 4. AI'nın Akıllı Kontrat Yaşam Döngüsündeki Rolü

Şekil 4'deki diyagram, yapay zekanın akıllı kontrat geliştirme yaşam döngüsünün altı ana aşamasına nasıl entegre olduğunu göstermektedir. AI Core bileşeni, altı aşamayla (Tasarım ve Spesifikasyon, Kod Üretimi, Güvenlik Analizi, Test ve Doğrulama, Dağıtım ve İzleme, Optimizasyon) bağlantılıdır. Yapay zekanın artık geliştirme sürecinin yalnızca tek bir aşamasına değil, tüm yaşam döngüsüne nüfuz eden bütünleşik bir bileşen haline geldiği görünmektedir.

Ekosistem Entegrasyonu ve Sektörel Uygulamalar

Yapay zekanın akıllı kontrat geliştirme süreçlerine etkisi, sadece kod düzeyindeki iyileştirmelerle sınırlı değildir. Bu iki teknolojinin birleşimi, aynı zamanda daha geniş bir ekosistem içinde yeni yetenekler ve sektörel çözümler ortaya çıkarmaktadır. Bu bölümde, AI ve akıllı kontratların birlikte çalışarak belirli endüstriyel sorunları nasıl çözdüğü ve daha karmaşık sistemler oluşturmak için nasıl entegre edildiği incelenmektedir.

Sektörel Uygulamalar

AI ve akıllı kontratların bir arada kullanımı, çeşitli sektörlerde yenilikçi uygulamalara yol açmıştır:

•Akıllı Üretim ve Tedarik Zinciri: Manimuthu vd. (2022) tarafından yapılan ve 91 atıf alan çalışma, Federe Öğrenme tabanlı yapay zeka ile akıllı kontratları birleştirerek bir otomobil montaj modeli geliştirmiştir [34]. Bu modelde, AI karar verme süreçlerini yönetirken, akıllı kontratlar bu kararların güvenli ve otomatik bir şekilde uygulanmasını sağlar. Bu, merkezi olmayan bir ortamda üretim süreçlerinin nasıl otonom hale getirilebileceğine dair örnek teşkil etmektedir.

•Finans ve İslami Finans: Rahim vd. (2018) çalışmasında, yapay zeka ve akıllı kontratların İslami finans prensiplerine uygunluğunu ve bu alandaki potansiyel uygulamalarını incelemiştir [35]. AI'nın analitik gücü, faizsiz finansman modellerinin risk analizini yaparken, akıllı kontratlar bu anlaşmaların şeffaf ve İslami hukuka uygun bir şekilde yürütülmesini garanti edebilir.

•Sağlık ve Eğitim: Ouf (2025), blokzincir ve derin öğrenmeyi birleştiren bir akıllı öğrenme sistemi için bir çerçeve önermiştir [36]. Bu sistemde, öğrenci verileri blokzincir üzerinde güvenli bir şekilde saklanırken, AI modelleri kişiselleştirilmiş öğrenme yolları sunar ve akıllı kontratlar başarıları otomatik olarak onaylayarak sertifikalar düzenler.

Gizlilik Koruma ve Güvenlik Entegrasyonu

Veri gizliliği hem AI hem de blokzincir için temel bir gerekliliktir. Deebak & Fadi (2021) tarafından yapılan ve 120 atıf alan çalışma, blokzincir ve yapay zekayı kullanarak akıllı kontratlarda gizliliği koruyan siber risk ölçüm mekanizmaları geliştirmiştir [37]. Bu tür sistemlerde, AI modelleri hassas veriler üzerinde analiz yaparken, blokzincir bu verilerin bütünlüğünü ve erişim kontrolünü sağlar, akıllı kontratlar ise önceden tanımlanmış gizlilik politikalarını otomatik olarak uygular.

LLM ve Blokzincir Ekosisteminin Bağlantısı

Daha geniş bir perspektiften bakıldığında, araştırmacılar LLM'leri sadece birer geliştirme aracı olarak değil, aynı zamanda blokzincir ekosisteminin bir parçası olarak entegre etmeye çalışmaktadır. C-LLM (2024) gibi çerçeveler, LLM'lerin doğrudan zincir üstü verilere erişmesini, bu verileri anlamlandırmasını ve hatta bu verilere dayanarak işlem başlatmasını sağlayacak evrensel bir entegrasyon katmanı önermektedir [38]. Bu, bir LLM'in bir DeFi protokolündeki anormal aktiviteleri tespit edip, bir akıllı kontrat aracılığıyla koruyucu bir işlemi tetikleyebileceği otonom sistemlerin önünü açabilir.

Tartışma ve Karşılaştırmalı Analiz

Bu literatür taraması, yapay zekanın akıllı kontrat geliştirme süreçlerindeki rolünü incelenen çalışmalar çerçevesinde ele almaktadır. İncelenen çalışmalar, AI'nın sadece mevcut sorunlara çözüm bulmakla kalmayıp, aynı zamanda daha önce mümkün olmayan yeni yetenekler ve uygulama alanları yarattığını göstermektedir. Bu bölümde, elde edilen bulgular sentezlenerek, alandaki temel eğilimler, metodolojik karşılaştırmalar ve genel çıkarımlar tartışılmaktadır.

Metodolojik Evrim ve Performans Karşılaştırması

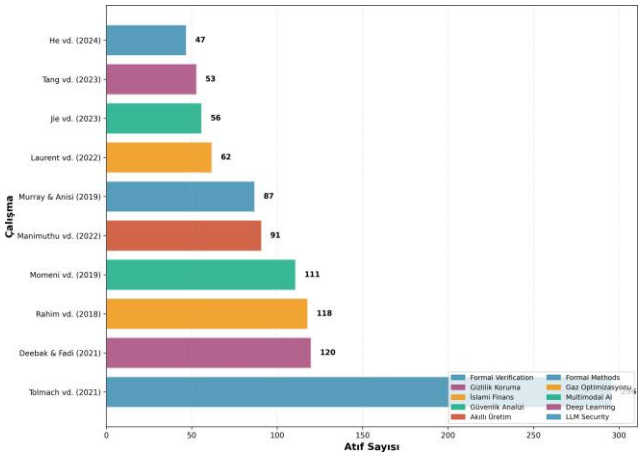
İncelenen çalışmaların sentezi, her dönemin kendine özgü metodolojik katkılarını ve sınırlılıklarını ortaya koymaktadır. 2018-2019 yıllarına ait ilk çalışmaların geleneksel makine öğrenmesi modellerine ve manuel özellik mühendisliğine dayandığı görülmektedir [5]. Bu çalışmaların AI'nın uygulanabilirliğini göstermekle birlikte anlamsal anlayış açısından belirli kısıtlar barındırdığı anlaşılmaktadır. 2020-2023 döneminde ise derin öğrenmenin, özellikle CodeBERT gibi Transformer tabanlı modellerin, alanda giderek daha fazla benimsenmekte olduğu dikkat çekmektedir [4, 8, 9]. Bu modellerin kodun karmaşık yapısını daha kapsamlı biçimde ele alarak güvenlik açığı tespitinde %90'ların üzerinde F1-skorlarına ulaştığı görülmektedir. Söz konusu dönemin en ileri noktasını ise farklı veri

temsillerini bir arada kullanarak %99'un üzerinde doğruluk oranları elde eden multimodal yaklaşımların oluşturduğu anlaşılmaktadır [10]. 2024-2025 döneminde odağın yalnızca tespitten ziyade kod üretimi [17, 18], otomatik optimizasyon [32] ve otomatik exploit üretimi [16] gibi daha karmaşık ve otonom görevlere doğru kaydığı gözlemlenmektedir.

Tablo 4. AI ve Akıllı Kontrat Alanındaki Teknolojik Evrim (2018-2025)

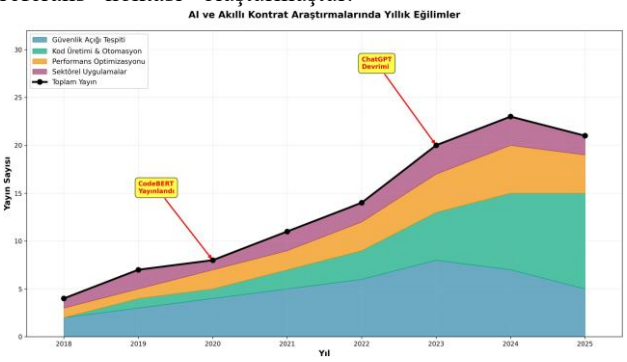
Dönem	Öne Çıkan Gelişmeler	Temel Teknolojiler	Önemli Çalışmalar
2018-2019	Erken ML Uygulamaları	Geleneksel ML, Statik Analiz	Momeni vd. [5], Murray & Anisi [13], Rahim vd. [36]
2020-2023	Derin Öğrenmenin Yükselişi	CNN, LSTM, CodeBERT, Multimodal	Deebak & Fadi [38], Tolmach vd. [12], Tang vd. [4], Jie vd. [10]
2024-2025	LLM ve Üretken AI Devrimi	GPT, Llama2, Üretken AI, Çok-Agent	Misbah vd. [34], Khalid & Brown [21]

Tablo 4, AI ve akıllı kontrat alanındaki teknolojik gelişimi 2018-2025 yılları arasında üç dönem halinde ele alarak her dönemin öne çıkan gelişmelerini, temel teknolojilerini ve ilgili çalışmalarını karşılaştırmalı biçimde sunmaktadır. İlk dönem olan 2018-2019 yıllarında, geleneksel makine öğrenmesi yöntemleri ve statik analiz tekniklerinin benimsenmesiyle alanda ilk uygulamaların hayata geçirildiği görülmektedir. Momeni vd. [5], Murray & Anisi [13] ve Rahim vd. [35] tarafından yürütülen çalışmaların bu dönemin temel referans noktalarını oluşturduğu anlaşılmaktadır. Söz konusu çalışmaların yapay zekanın akıllı kontrat geliştirme süreçlerine uygulanabilirliğini ortaya koyduğu, ancak anlamsal anlayış ve genellenebilirlik açısından belirli kısıtlar barındırdığı değerlendirilmektedir. 2020-2023 döneminde CNN, LSTM ve CodeBERT gibi derin öğrenme mimarilerinin ve multimodal yaklaşımların ön plana çıktığı izlenmektedir. Deebak & Fadi [37], Tolmach vd. [12], Tang vd. [4] ve Jie vd. [10] tarafından yürütülen çalışmaların bu dönemi şekillendirdiği görülmektedir. Transformer tabanlı modellerin kodun sözdizimsel ve anlamsal özelliklerini kavrayarak güvenlik açığı tespitinde performans artışı sağladığı anlaşılmaktadır. 2024-2025 döneminde ise GPT ve Llama2 gibi büyük dil modellerinin ve çok-ajan sistemlerin alanda giderek daha fazla benimsenmekte olduğu dikkat çekmektedir. Misbah vd. [34] ve Khalid & Brown [21] tarafından yürütülen çalışmaların bu dönemin öne çıkan örnekleri arasında yer aldığı görülmektedir. Odağın yalnızca güvenlik açığı tespitinden kod üretimi, otomatik optimizasyon ve otonom ajan sistemleri gibi daha kapsamlı görevlere doğru kaydığı anlaşılmaktadır. Alandaki metodolojik yaklaşımın kural tabanlı ve manuel yöntemlerden üretken yapay zekaya doğru kesintisiz bir ilerleme kaydettiğini ortaya koymaktadır. Her dönemin bir öncekinin sınırlılıklarını aşmaya yönelik yeni yöntemler geliştirdiği, bununla birlikte açıklanabilirlik ve model güvenilirliği gibi sorunların hâlâ güncelliğini koruduğu değerlendirilmektedir.



Şekil 5. En Çok Atıf Alan Çalışmalar

Şekil 5'deki grafik, incelenen literatürdeki en çok atıf alan çalışmaları atıf sayılarına göre sıralayarak sunmaktadır. En yüksek atıf sayısına sahip çalışmalar ağırlıklı olarak formal doğrulama ve gizlilik koruma gibi temel altyapı konularına odaklanmıştır [12, 37]. Bu alanın olgunlaşma sürecinde güvenlik ve güvenilirlik temelli çalışmaların referans noktası oluşturmuştur.



Şekil 6. AI ve Akıllı Kontrat Araştırmalarında Yıllık Eğilimler

Şekil 6'da gösterilen grafik, yıllara göre yayınlanan makale sayısındaki eğilimleri ve bu yayınların ana temalara göre dağılımını göstermektedir. Özellikle 2023 sonrası, Kod Üretimi ve Otomasyon konusundaki araştırmalarda belirgin bir artış gözlemlenmekte olup bu eğilim, LLM tabanlı yaklaşımların alanda giderek daha fazla benimsenmesini yansıtmaktadır.

Bu gelişim, bir doğruluk ve açıklanabilirlik ödünleşmesini de beraberinde getirmiştir. Geleneksel ML modelleri daha yorumlanabilirken, en yüksek performansı sergileyen derin öğrenme ve LLM modelleri genellikle "kara kutu" olarak çalışmaktadır. Bir modelin neden belirli bir kararı verdiği anlaşılamadığında, özellikle yüksek riskli finansal uygulamalarda bu modellere güvenmek zorlaşmaktadır. Bu nedenle, hibrit sistemler [14] ve beyaz kutu metodolojileri [10] gibi yaklaşımlar, bu ödünleşmeyi ele almak için uygulanabilir bir yöntem olarak görülmektedir.

Veri Setleri ve Benchmark'ların Rolü

Herhangi bir AI alanında olduğu gibi, bu alandaki ilerleme de büyük ölçüde yüksek kaliteli veri setlerinin ve standartlaştırılmış benchmark'ların varlığına bağlıdır. SolidiFI-benchmark [4] ve SmartEmbed [10] gibi etiketlenmiş veri setleri, güvenlik açığı tespit modellerinin karşılaştırılabilir bir şekilde değerlendirilmesini sağlamıştır. Ancak, bu veri setleri genellikle belirli güvenlik açığı türlerine odaklanmakta ve gerçek dünyadaki kontratların çeşitliliğini tam olarak yansıtmayabilmektedir. Bu noktada, CGGNet gibi sentetik veri artırımı teknikleri, daha çeşitli ve kapsamlı eğitim verileri oluşturarak bu boşluğu doldurmaya yardımcı olmaktadır [26]. Kod üretimi alanında ise, SolEval gibi proje düzeyindeki benchmarklar, modellerin pratik uygulanabilirliğini ölçmek için kritik bir rol oynamaktadır [17]. Gelecekte, daha çeşitli, daha büyük ölçekli ve daha gerçekçi veri setlerinin ve benchmark'ların geliştirilmesi, alanın ilerlemesinde bir gereklilik olarak öne çıkmaktadır.

Savunma ve Saldırı Arasındaki Dinamik

Yapay zekanın hem güvenlik açığı tespiti hem de saldırı için kullanılması, alanda bir tür "silahlanma yarışı" dinamiği yaratmaktadır. AI tabanlı saldırı araçlarının [16] geliştirilmesi, savunma mekanizmalarının sürekli olarak güncellenmesi ve daha karmaşık hale getirilmesi gerektiğini göstermektedir. Bu dinamik, bir yandan riskleri artırırken, diğer yandan da daha sağlam ve dayanıklı sistemlerin geliştirilmesini teşvik etmektedir. Gelecekte, yapay zekâ tabanlı "kırmızı takım" (saldırı simülasyonu) ve "mavi takım" (savunma) araçlarının, geliştirme yaşam döngüsünün standart bir parçası haline gelmesi beklenmektedir.

Sonuç ve Gelecek Araştırma Yönelimleri

Bu literatür taraması, yapay zekanın akıllı kontrat geliştirme yaşam döngüsünün her aşamasına giderek daha fazla nüfuz ettiğini ve bu alanda kapsamlı değişiklikler getirdiğini göstermektedir. İncelenen literatürün bütünü, AI'nın artık sadece teorik bir olasılık değil, akıllı kontratların daha güvenli, verimli ve erişilebilir hale gelmesini sağlayan pratik bir araç olduğunu kanıtlamaktadır. Güvenlik açığı tespitinde %99'lara varan doğruluk oranları sunan multimodal derin öğrenme modellerinden [10], doğal dil tanımlarından otonom bir şekilde kod üreten büyük dil modellerine kadar, yapay zeka geliştiricilerin karşılaştığı en temel zorluklara yenilikçi çözümler sunmaktadır.

Çalışmamız, alandaki teknolojik evrimin üç belirgin dönem boyunca izlediği yörüngenin, yalnızca mevcut yöntemlerin iyileştirilmesiyle sınırlı kalmadığını, aynı zamanda otomatik istismar kodu üretimi veya yasal metinlerden doğrudan kontrat oluşturma gibi daha önce mümkün olmayan tamamen yeni yetenek kategorileri ortaya çıkardığını göstermektedir." Bununla birlikte, bu hızlı ilerleme, açıklanabilirlik, veri kalitesi, karmaşık proje düzeyinde anlayış ve yeni saldırı vektörlerine karşı dayanıklılık gibi önemli zorlukları da beraberinde getirmektedir.

Gelecek Araştırma Yönelimleri

Bu zorluklar ve mevcut araştırma boşlukları ışığında, gelecekteki çalışmaların aşağıdaki alanlara odaklanması beklenmektedir:

Açıklanabilir AI (XAI) Entegrasyonu: Güvenlik açığı tespit modellerinin karar süreçlerini şeffaf hale getirmek, bu araçlara olan güveni artırmak için bir gerekliliktir. Gelecekteki çalışmalar, bir modelin neden belirli bir kod parçasını "güvenlik açığı" olarak işaretlediğini insan tarafından anlaşılabilir terimlerle açıklayan XAI tekniklerini entegre etmelidir.

Otonom Ajanlar ve Çoklu Ajan Sistemleri: GasAgent [32] gibi çalışmaların öncülük ettiği gibi, gelecekteki sistemler tek bir görev yerine tüm geliştirme sürecini yöneten otonom ajanlardan oluşabilir. Bir "analist" ajan güvenlik açıklarını ararken, bir "geliştirici" ajan kodu yazar, bir "optimize edici" ajan gaz maliyetini düşürür ve bir "test" ajanı üretilen kodu doğrular. Bu tür çoklu ajan yapısındaki sistemler, geliştirme sürecini büyük ölçüde otonom hale getirebilir.

Gerçek Zamanlı ve Dinamik Analiz: Mevcut yaklaşımların çoğu statik kod analizine dayanmaktadır. Blokzincirin dinamik doğasını ve kontratlar arası karmaşık etkileşimleri daha iyi yakalamak için, işlem akışını ve ağ durumunu gerçek zamanlı olarak analiz eden dinamik analiz yöntemleriyle AI'nın birleştirilmesi gerekmektedir.

On-Chain AI ve Otonom Kontratlar: LLM'lerin C-LLM [38] gibi çerçeveler aracılığıyla doğrudan zincir üstü verilere erişmesi, kendi kendini yöneten ve piyasa koşullarına veya güvenlik tehditlerine dinamik olarak yanıt veren "canlı" akıllı kontratların önünü açacaktır. Bu otonom kontratlar, merkezi olmayan otonom organizasyonların (DAO'lar) yönetimini ve işleyişini temelden değiştirebilir.

Etik ve Güvenlik Hizalaması: AI modellerinin, özellikle de kod üreten LLM'lerin, sadece işlevsel değil, aynı zamanda etik ve güvenli kod üretmesini sağlamak çok önemlidir. AI "hizalama" araştırmalarının, üretilen kontratların istenmeyen sonuçlara yol açmamasını veya kötüye kullanılabilir arka kapılar içermemesini garanti edecek şekilde bu alana uygulanması gerekmektedir.

Sonuç olarak, yapay zeka ve akıllı kontratların kesişimi hem akademik araştırma hem de pratik uygulama açısından gelişmeye açık bir alan olmayı sürdürmektedir. İncelenen literatür, bu iki teknolojinin bütünleşik kullanımının güvenlik, verimlilik ve otomasyon boyutlarında somut katkılar sağladığını göstermekte; bununla birlikte açıklanabilirlik, veri kalitesi ve güvenlik hizalaması gibi sorunların çözüme kavuşturulmasının alanın olgunlaşması bakımından gerekli olduğuna işaret etmektedir.

Kaynaklar

[1] N. Szabo, "Smart contracts: Building blocks for digital markets," *EXTROPY: The Journal of Transhumanist Thought*, vol. 18, no. 2, p. 28, 1996.

[2] J. Li, Z. Zhao, Z. Su, and W. Meng, "Gas-expensive patterns detection to optimize smart contracts," *Applied Soft Computing*, vol. 145, p. 110542, 2023.

[3] A. Laurent, L. Brotcorne, and B. Fortz, "Transaction fees optimization in the Ethereum blockchain," *Blockchain: Research and Applications*, vol. 3, no. 3, p. 100074, 2022.

[4] X. Tang, Y. Du, A. Lai, Z. Zhang, and L. Shi, "Deep learning-based solution for smart contract vulnerabilities detection," *Scientific Reports*, vol. 13, no. 1, p. 20106, 2023.

[5] P. Momeni, Y. Wang, and R. Samavi, "Machine learning model for smart contracts security analysis," in *Proc. 17th Int. Conf. Privacy, Security Trust (PST)*, 2019, pp. 1–6.

[6] J. Crisostomo, F. Bacao, and V. Lobo, "Machine learning methods for detecting smart contracts vulnerabilities within Ethereum blockchain—A review," *Expert Systems With Applications*, vol. 268, p. 126353, 2025.

[7] Y. Huang, J. Li, S. Fang, Y. Li, P. Yang, and B. Hu, "Smart contract intent detection with pre-trained programming language model," *arXiv preprint arXiv:2508.20086*, 2025.

[8] Z. Feng, D. Guo, D. Tang, N. Duan, X. Feng, M. Gong, L. Shou, B. Qin, T. Liu, D. Jiang, and S. Zhou, "CodeBERT: A pre-trained model for programming and natural languages," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 1536–1547.

[9] P. Xu, X. Zhu, and D. A. Clifton, "Multimodal learning with transformers: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 10, pp. 12113–12132, 2023.

[10] W. Jie, Q. Chen, J. Wang, A. S. V. Koe, J. Li, P. Huang, Y. Wu, and Y. Wang, "A novel extended multimodal AI framework towards vulnerability detection in smart contracts," *Information Sciences*, vol. 636, p. 118907, 2023.

[11] W. Deng, H. Wei, T. Huang, C. Cao, Y. Peng, and X. Hu, "Smart contract vulnerability detection based on deep learning and multimodal decision fusion," *Sensors*, vol. 23, no. 16, p. 7246, 2023.

[12] P. Tolmach, Y. Li, S.-W. Lin, Y. Liu, and Z. Li, "A survey of smart contract formal specification and verification," *ACM Computing Surveys*, vol. 54, no. 7, pp. 1–38, 2021.

[13] Y. Murray and D. A. Anisi, "Survey of formal verification methods for smart contracts on blockchain," in *Proc. 10th IFIP Int. Conf. New Technologies, Mobility Security (NTMS)*, 2019, pp. 1–6.

[14] S. Chen and F. Li, "Ponzi scheme detection in smart contracts using the integration of deep learning and formal verification," *IET Blockchain*, vol. 4, no. 2, pp. 185–196, 2024.

[15] Z. Guo, "Blockchain-enhanced smart contracts for formal verification of IoT access control mechanisms," *Alexandria Engineering Journal*, vol. 118, pp. 315–324, 2025.

[16] A. Gervais and L. Zhou, "AI agent smart contract exploit generation," *arXiv preprint arXiv:2507.05558*, 2025.

[17] Z. Peng, X. Yin, R. Qian, P. Lin, Y. Liu, H. Zhang, C. Ying, and Y. Luo, "SolEval: Benchmarking large language models for repository-level Solidity code generation," *arXiv preprint arXiv:2502.18793*, 2025.

[18] G. De Vito, D. D'Amici, F. Izzo, F. Ferrucci, and D. Di Nucci, "LLM-based generation of Solidity smart contracts from system requirements in natural language: The AstraKode case," in *Proc. IEEE Int. Conf. Softw. Analysis, Evolution Reengineering (SANER)*, 2025, pp. 170–180.

[19] E. A. Napoli, N. Romani, V. Gatteschi, and C. Schifanella, "Light and shadows of smart contract development with LLMs," *Expert Systems With Applications*, p. 129011, 2025.

[20] Z. He, Z. Li, S. Yang, H. Ye, A. Qiao, X. Zhang, X. Luo, and T. Chen, "Large language models for blockchain security: A systematic literature review," *arXiv preprint arXiv:2403.14280*, 2024.

[21] S. Khalid and C. Brown, "Evaluating capabilities and perspectives of generative AI tools in smart contract development," in *Proc. 7th ACM Int. Symp. Blockchain Secure Critical Infrastructure*, 2025, pp. 1–12.

[22] "ChainGPT AI Smart-Contract Generator," ChainGPT Documentation, 2025. [Online]. Available: https://docs.chaingpt.org/ai-tools-and-applications/ai-smart-contract-generator

[23] "ContractGPT: AI Tool Fine-Tuned by CodeBERT for Smart Contract Development," HackQuest Project, 2025. [Online]. Available: https://www.hackquest.io/es/projects/contractgpt

[24] "Workik Smart Contract Generator," Workik, 2025. [Online]. Available: https://workik.com/smart-contract-generator

[25] J. Zhao, X. Chen, G. Yang, and Y. Shen, "Automatic smart contract comment generation via large language models and in-context learning," *Information and Software Technology*, vol. 168, p. 107405, 2024.

[26] S.-J. Hwang, S. H. Ju, and Y.-H. Choi, "CGGNet: Compiler-guided generation network for smart contract data augmentation," *IEEE Access*, vol. 12,
pp. 97515–97532, 2024.
A. Laurent, L. Brotcorne, and B. Fortz, "Transaction fees optimization in the Ethereum blockchain," *Blockchain: Research and Applications*, vol. 3, no. 3, p. 100074, 2022.

[27] "Optimizing Ethereum Gas Fees Using Machine Learning," ResearchGate, 2025. [Online]. Available:
https://www.researchgate.net/publication/388791
2
91_Optimizing_Ethereum_Gas_Fees_Using_Ma
ch
ine_Learning_A_Predictive_and_Adaptive_Appr
o ach

[28] S. Khanzadeh, N. Samreen, and M. H. Alalfi, "Optimizing gas consumption in Ethereum smart contracts: Best practices and techniques," in Proc. IEEE 23rd Int. Conf. Softw. Quality, Reliability, Security Companion (QRS-C), 2023, pp. 300–309.

[29] Q.-P. Kong, Z.-Y. Wang, Y. Huang, X.-P. Chen, X.-C. Zhou, Z.-B. Zheng, and G. Huang, "Characterizing and detecting gas-inefficient patterns in smart contracts," J. Comput. Sci. Technol., vol. 37, no. 1, pp. 67–82, 2022.

[30] J. Zheng, Z. Peng, Y. Liu, J. Wang, Y. Liao, W. Dong, and X. He, "GasAgent: A multi-agent framework for automated gas optimization in smart contracts," arXiv preprint arXiv:2507.15761, 2025.

[31] S. Misbah, M. F. Shahid, S. Siddiqui, T. J. S. Khanzada, R. B. Ashari, Z. Ullah, and M. Jamjoom, "Generative AI-driven smart contract optimization for secure and scalable smart city services," Smart Cities, vol. 8, no. 4, p. 118, 2025.

[32] A. Manimuthu, V. G. Venkatesh, Y. Shi, V. R. Sreedharan, and S. C. L. Koh, "Design and development of automobile assembly model using federated artificial intelligence with

smart contract," Int. J. Production Research, vol. 60, no. 1, pp. 111–135, 2022.

[33] S. M. Rahim, Z. Z. Mohamad, J. A. Bakar, F. H. Mohsin, and N. M. Isa, "Artificial intelligence, smart contract and Islamic finance," Asian Social Science, vol. 14, no. 2, p. 145, 2018.

[34] [S. Ouf, S. Ahmed, and Y. Helmy, "A blockchain based deep learning framework for a smart learning environment," Scientific Reports, vol. 15, no. 1, p. 19519, 2025.

[35] B. D. Deebak and F. AL-Turjman, "Privacy-preserving in smart contracts using blockchain and artificial intelligence for cyber risk measurements," J. Inf. Security Appl., vol. 58, p. 102749, 2021.

[36] X. Zeng, Y. Xian, D. Xuan, D. Yang, C. Li, J. Chen, P. Fan, and P. Liu, "Connecting large language models with blockchain: Making smart contracts smarter," ACM Trans. Web, vol. 20, no. 1, pp. 1–21, 2026.