

Yazılım Mühendisliği Eğitiminde Üretken Yapay Zekânın Bibliyometrik Analiz Yöntemi ile İncelenmesi

Fatih Çağatay BAZ^{1*}, Şerife KELEŞ²

^{1,2}Osmaniye Korkut Ata Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, Yönetim Bilişim Sistemleri Bölümü, 80000, Osmaniye

¹<https://orcid.org/0000-0002-6398-9851>

²<https://orcid.org/0009-0006-5771-6458>

*Sorumlu yazar: fatihcagataybaz@osmaniye.edu.tr

Araştırma Makalesi

Makale Tarihiçesi:

Geliş tarihi: 15.08.2025

Kabul tarihi: 28.11.2025

Online Yayınlanma: 15.06.2026

Anahtar Kelimeler:

Üretken yapay zekâ
Yazılım mühendisliği eğitimi
Büyük dil modelleri
Bibliyometrik analiz

ÖZ

Bu çalışma, üretken yapay zekâ teknolojilerinin yazılım mühendisliği eğitimine etkilerini bibliyometrik analiz yoluyla incelemektedir. Scopus veri tabanında gerçekleştirilen tarama sonucunda, üretken yapay zekâ ve yazılım mühendisliği eğitimi kesiştiren 134 yayın belirlenmiş, VOSviewer yazılımı kullanılarak anahtar kelime birlikte geçme ve ülkeler arası ortak yazarlık analizi yapılmıştır. Üretken yapay zekâ teknolojilerinin eğitim süreçlerine entegrasyonuna ilişkin mevcut eğilimler, iş birliği kalıpları ve kavramsal kümelenmeleri ortaya koyarak hem literatürdeki boşlukların tespit edilmesi hem de gelecekteki araştırmalar için yol gösterici bir temel oluşturulmasına katkı sağlamak amaçlanmaktadır. Yapılan analiz sonucunda kavramsal olarak dört tema öne çıkmaktadır: (1) büyük dil modelleri ve kod üretimi, (2) müfredat ve öğretim teknolojileri, (3) yazılım süreçleri ve kalite, (4) öğrenci merkezli pedagojik yaklaşımlar. Bulunan veriler 2024 yılının üretken yapay zekâ ve yazılım mühendisliği eğitimi alanındaki akademik yayınların yoğunlaştığı bir yıl olduğunu göstermektedir. Özellikle "computer aided instruction", "software design", "education computing", "generative AI" ve "engineering education" gibi anahtar kavramların ortalama yayın yıllarının 2024'e yakın olması bu temaların literatürde oldukça güncel ve hızla gelişen araştırma konuları olduğunu ortaya koymaktadır. Bu durum üretken yapay zekânın eğitim alanına entegrasyonunun yakın dönemde ivme kazandığını ve bu alandaki akademik ilginin arttığını göstermektedir. Çalışma üretken yapay zekânın yazılım mühendisliği eğitimine etkilerini derinlemesine anlamak ve bu alanda yapılacak yeni araştırmalara yön vermek amacıyla alanyazına önemli katkılar sunmaktadır.

Examination of Generative Artificial Intelligence in Software Engineering Education Using Bibliometric Analysis Method

Research Article

Article History:

Received: 15.08.2025

Accepted: 28.11.2025

Published online: 15.06.2026

Keywords:

Generative artificial intelligence
Software engineering education
Large language models
Bibliometric analysis

ABSTRACT

This study examines the impact of generative artificial intelligence technologies on software engineering education through bibliometric analysis. Through a search of the Scopus database, 134 publications intersecting generative AI and software engineering education were identified, and keyword co-occurrence and cross-country co-authorship analysis were conducted using VOSviewer software. By revealing current trends, patterns of collaboration, and conceptual clusters regarding the integration of generative AI technologies into educational processes, the study aims to contribute to both identifying gaps in the literature and providing a guiding basis for future

research. As a result of the analysis, four conceptual themes emerge: (1) big language models and code generation, (2) curriculum and instructional technologies, (3) software processes and quality, (4) student-centered pedagogical approaches. The data shows that 2024 is a year of intensified academic publications in the field of generative AI and software engineering education. In particular, the fact that the average publication years of key concepts such as "computer aided instruction", "software design", "educational computing", "generative AI" and "engineering education" are close to 2024 reveals that these themes are very current and rapidly developing research topics in the literature. This shows that the integration of generative AI into the field of education has recently gained momentum and academic interest in this field is gradually increasing. This study provides important contributions to better understand the effects of generative AI on software engineering education in depth and to guide future research in this field.

To Cite: Baz FÇ., Keleş Ş. Yazılım Mühendisliği Eğitiminde Üretken Yapay Zekânın Bibliyometrik Analiz Yöntemi ile İncelenmesi. Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi 2026; 9(3): 1328-1359.

1. Giriş

Üretken yapay zekâ, veri analizi ve içerik üretimi alanında devrim niteliğinde bir teknoloji olarak öne çıkmaktadır. Bu sistemler, mevcut verilerden öğrendikleri kalıpları kullanarak yeni ve özgün içerikler oluşturma yeteneğine sahiptir. Örneğin; metin, görüntü, video, ses ve programlama kodu gibi çeşitli alanlarda uygulama bulan üretken yapay zekâ, üretken çekışmeli ağlar ve dönüştürücü modeller gibi gelişmiş algoritmalar aracılığıyla çalışmaktadır. Bu modeller, orijinal veri kümelerini analiz ederek benzer yapıda veya daha geniş kapsamlı çıktılar üretebilmektedir (Banh ve Strobel, 2023). Üretken yapay zekânın en dikkat çekici özelliklerinden biri, büyük dil modellerini kullanarak metinlerdeki anlamsal incelikleri kavrayabilme ve bu bilgiyi yeni içerik üretiminde kullanabilme yeteneğidir. Bu özellik, özellikle sıfır atış öğrenme senaryolarında önem kazanmaktadır. Modelin daha önce karşılaşmadığı örnekler üzerinde bile etkili tahminlerde bulunabilmesi, taslak oluşturma, özetleme ve sözleşme incelemesi gibi karmaşık görevlerde kullanılmasına olanak tanımaktadır (Savelka, 2023). Benzer şekilde, eğitim ortamlarında öğrencilere kişiselleştirilmiş içerik sunma, otomatik geri bildirim sağlama, öğrenme materyalleri oluşturma ve programlama eğitimi kapsamında kod üretme desteği sağlama gibi uygulamalarla katkı sunmaktadır. Bu yetenekler, üretken yapay zekânın iş dünyasından akademik araştırmalara kadar geniş bir yelpazede uygulanabilirliğini artırmaktadır. Derin üretken modellere dayanan üretken yapay zekâ, kullanıcıların basit komutlarıyla çeşitli alanlarda özgün içerik oluşturma kapasitesine sahiptir. Bu teknoloji, metin üretiminden görsel tasarıma, müzik kompozisyonundan yazılım kodlamaya kadar geniş bir yelpazede uygulanabilir. Üretken yapay zekâ sistemleri, büyük veri setleri üzerinde eğitilerek, insan benzeri yaratıcılık ve anlayış sergileyebilen karmaşık algoritmalar kullanmaktadır. Bu sistemler, kullanıcı girdilerine dayalı olarak tutarlı ve bağlamsal olarak uygun içerik üretmekte, içerik üretimi süreçlerini hızlandırmakta ve çeşitli sektörlerde verimliliği artırmaktadır. Ayrıca, üretken yapay zekânın sürekli gelişen doğası, gelecekte daha da sofistike ve özelleştirilmiş içerik üretimi potansiyelini vaat etmektedir.

Üretken yapay zekâ, eğitimde içerik üretimi, tasarım, veri artırımı ve simülasyon senaryoları gibi çeşitli uygulama alanlarında etkili sonuçlar sunmaktadır. Üretken yapay zekâ, eğitim sektöründe geniş bir

uygulama yelpazesine sahiptir. İçerik üretimi alanında, ders materyalleri, öğrenci değerlendirmeleri ve kişiselleştirilmiş öğrenme planlarının oluşturulmasında kullanılabilir. Tasarım açısından, eğitim materyallerinin görsel unsurlarını geliştirmek ve öğrenci katılımını artırmak amacıyla etkileşimli grafikler ve animasyonlar üretilebilir. Veri artırımı, mevcut eğitim verilerini genişleterek daha kapsamlı ve çeşitli öğrenme senaryolarının oluşturulmasına olanak tanır. Simülasyon senaryoları, öğrencilerin gerçek dünya durumlarını güvenli bir ortamda deneyimlemelerine olanak tanıyarak, onların pratik becerilerini geliştirmelerine katkıda bulunmaktadır (Sengar ve ark., 2024; Yusuf ve ark., 2024).

Üretken yapay zekâ teknolojisi, derin öğrenme modellerinin gelişimiyle birlikte yazılım geliştirme alanında devrim yaratma potansiyeline sahiptir. Bu modeller, doğal dil veya modelleme dillerinden kaynak kodu üretme yeteneği ile geliştiricilerin çalışma süreçlerini önemli ölçüde değiştirebilir. CodeBERT ve GraphCodeBERT gibi özel olarak eğitilmiş modeller, programlama kodları üzerinde çalışarak yeni yazılım programları oluşturabilirken, genel amaçlı büyük dil modelleri de giderek artan kod içeren veri setleriyle eğitilerek benzer yetenekler kazanmaktadır. Bu teknolojinin pratik uygulamaları, üretken kod üreten yapay zekâ araçları aracılığıyla belirginleşmektedir. Bu araçlar, programcılara kod üretimi, hata ayıklama, otomatik kod parçaları yazma, fikirleri eyleme dönüştürülebilir betiklere çevirme, otomatik tamamlama işlevleri sunma, birim testleri oluşturma ve yinelenen kodları tespit etme gibi çeşitli yetenekler sağlamaktadır. Bu özellikler, yazılım geliştirme sürecini hızlandırabilir, kod kalitesini artırabilir ve geliştiricilerin daha karmaşık problemlere odaklanmasına olanak tanıyabilir. Ancak bu teknolojinin etik kullanımı ve insan denetiminin önemi de göz ardı edilmemelidir (Banh ve Strobel, 2023).

Üretken yapay zekâ teknolojilerinin eğitim alanındaki uygulamaları, çeşitli derin öğrenme yöntemlerini kullanarak öğrenme süreçlerini geliştirmeyi amaçlamaktadır. Üretken yapay zekâ ile eğitim alanında gerçekleştirilen çalışmalarda, çeşitli derin öğrenme yöntemleri öne çıkmaktadır. Wang (2025), derin bilgi izleme yöntemini kullanarak kişiselleştirilmiş eğitimde öğrenci öğrenme süreçlerinin izlenmesine odaklanmıştır. Alipour ve Gallegos, (2025) eğitimde yapay verilerin üretilmesi ve sosyal verilerle bütünleştirilmesi amacıyla çekişmeli öğrenme ve karşıtlık tabanlı öğrenme yöntemlerini vurgulamıştır. Al-Qaysi ve ark., (2025) eğitimde sürdürülebilirliği ele alırken karşıtlık tabanlı öğrenme ve birleşik öğrenme yaklaşımlarını kullanmışlardır. Gani ve ark., (2025) bloom taksonomisine dayalı otomatik soru sınıflandırmasını geliştirmek için evrişimli sinir ağları ve karşıtlık tabanlı öğrenme yöntemlerinden faydalanmıştır. Soni ve Kapoor (2025), eğitim metnlerinin özetlenmesi ve içerik üretiminde açıklayıcı özetleme tekniğini uygulamıştır. Bu gelişmeler, üretken yapay zekânın eğitim alanında geniş bir uygulama yelpazesine sahip olduğunu ve öğrenme deneyimlerini dönüştürme potansiyeline sahip olduğunu göstermektedir.

Üretken yapay zekâ teknolojileri yazılım mühendisliği eğitiminde öğrencilere kod üretimi, problem çözme ve algoritmik düşünme becerilerini geliştirme konusunda destek sağlayarak, geleneksel eğitim yöntemlerini tamamlayıcı bir rol üstlenmektedir. OpenAI Codex, Tabnine, Amazon CodeWhisperer, Codeium, Replit Ghostwriter ve ChatGPT gibi üretken yapay zekâ kodlama araçları, öğrencilere gerçek

zamanlı kod önerileri sunarak ve kişiselleştirilmiş geri bildirimler sağlayarak öğrenme sürecini hızlandırmakta ve zenginleştirmektedir. Ancak, bu teknolojilerin eğitim ortamlarında kullanımı henüz yeni olduğundan, etkileri ve potansiyel riskleri tam olarak anlaşılamamıştır.

Üretken yapay zekâ sağlık alanında, hastalık teşhisi, ilaç keşfi ve hasta bakımı optimizasyonunda kullanılmaktadır. Finans sektöründe risk analizi, piyasa tahminleri ve kişiselleştirilmiş finansal tavsiyelerde önemli rol oynamaktadır. Hukuk alanında yasal belge analizi ve dava sonuç tahminleri gibi uygulamalarda kullanılmaktadır. Bu geniş kapsamlı uygulamalar, üretken yapay zekânın farklı alanlarda sunduğu potansiyeli ortaya koymaktadır (Sengar ve ark., 2024).

Üretken yapay zekânın yazılım mühendisliği eğitimindeki rolünü ve etkisini sistematik bir şekilde incelemek büyük önem taşımaktadır. Araştırmacılar, bu teknolojilerin hangi pedagojik yaklaşımlarla en etkili şekilde kullanılabileceğini, etik ve akademik dürüstlük konularında nasıl bir denge kurulması gerektiğini ve teknik güvenilirlik açısından ne gibi önlemler alınması gerektiğini araştırmaktadır. Ayrıca, bu alandaki literatürün bibliyometrik analizi, hangi kurumların ve ülkelerin bu konuda öncü olduğunu, araştırmaların hangi kavramsal temalar etrafında yoğunlaştığını ve gelecekte hangi alanlarda daha fazla çalışmaya ihtiyaç duyulacağını ortaya çıkarmak açısından kritik öneme sahiptir. Bu çalışma, tüm bu sorulara yanıt aramakta ve üretken yapay zekânın yazılım mühendisliği eğitimindeki durumunu ve gelecekteki potansiyelini kapsamlı bir şekilde değerlendirmeyi amaçlamaktadır.

Çoğu mevcut çalışma (Vhatkar ve ark., 2024; Bicen, 2025; Ng ve Ho, 2025) üretken yapay zekâyı genel eğitim bağlamında ele alırken, bu çalışma özel olarak yazılım mühendisliği eğitime odaklanmakta ve pedagojik etkilerin yanı sıra kod üretimi, algoritmik düşünme ve mühendislik müfredatı gibi daha teknik boyutları da kapsamaktadır. Literatürdeki araştırmalar genellikle üretken yapay zekâyı kavramsal düzeyde incelerken, bu çalışmada OpenAI Codex, Tabnine, Amazon CodeWhisperer, Codeium, Replit Ghostwriter ve ChatGPT gibi araçların yazılım mühendisliği öğrencilerine sağladığı doğrudan katkılar analiz edilmekte, böylece uygulamalı boyutta daha güçlü bir perspektif sunulmaktadır. Ayrıca Vhatkar ve ark., (2024) ile Bicen (2025) genel eğitim alanında bibliyometrik analizler gerçekleştirirken, bu çalışma yazılım mühendisliği özelinde anahtar kelime kümeleri, ortak yazarlık ağları ve atıf yapıları üzerinden alanın özgün dinamiklerini ortaya koymakta ve yalnızca mevcut eğilimleri tanımlamakla kalmayıp, müfredat reformu, öğretmen eğitimi ve etik politikalar gibi doğrudan uygulanabilir öneriler de geliştirmektedir.

Bu çalışma kapsamında ele alınan araştırma problemi şu soruya odaklanmaktadır: “Üretken yapay zekâ, yazılım mühendisliği eğitiminde hangi tematik alanlarda konumlanmakta ve kavramsal, pedagojik ve teknik düzeyde nasıl bir dönüşüm oluşturmaktadır?”

2. Literatür Taraması

2.1. Üretken Yapay Zekânın Yazılım Geliştirmedeki Rolü

Üretken yapay zekâ, yazılım geliştirme süreçlerinde verimlilik, yenilikçilik ve güvenliği artıran önemli bir rol üstlenmektedir ve yazılım geliştirme sürecinin tüm aşamalarında devrim niteliğinde değişiklikler

sunmaktadır. Kod üretiminden hata ayıklamaya, test senaryoları oluşturmada dokümantasyona kadar geniş bir yelpazede etkili olan üretken yapay zekâ, geliştiricilerin işlerini hızlandırmakta ve optimize etmektedir. Özellikle, tekrarlayan görevleri otomatikleştirerek zaman kazandırmakta ve insan geliştiricilerin daha karmaşık, yaratıcı problemlere odaklanmalarına olanak tanımaktadır. Üretken yapay zekânın yazılım geliştirmedeki bir diğer önemli katkısı, kod kalitesini ve güvenliğini artırmasıdır. Makine öğrenimi algoritmalarını kullanarak, potansiyel güvenlik açıklarını tespit edebilmekte, kod optimizasyonu önerileri sunabilmekte ve en iyi uygulamaları teşvik edebilmektedir. Ayrıca, farklı programlama dillerinde ve çerçevelerinde geniş bir bilgi tabanına sahip olan Üretken yapay zekâ, geliştiricilere yeni teknolojileri öğrenme ve uygulamada yardımcı olarak, sürekli değişen teknoloji ortamında rekabet güçlerini korumalarını sağlamaktadır.

2.1.1. Kod Oluşturma ve Hata Ayıklama

OpenAI Codex ve GitHub Copilot gibi üretken yapay zekâ modelleri, doğal dil açıklamalarını sözdizimsel olarak doğru ve bağlam farkında kaynak koduna dönüştürerek kod oluşturma süreçlerinde devrim yaratmıştır. Bu araçlar, geliştirme süresini önemli ölçüde azaltmakta, kod kalitesini artırmakta ve özellikle yeni başlayan geliştiriciler için programlamayı daha erişilebilir hale getirmektedir (Alenezi ve Akour, 2025; Saravanan ve ark., 2025). Hata ayıklama bağlamında, üretken yapay zekâ modelleri hataları akılcıca tespit edebilmekte ve geliştiriciler arasında güven ve kullanılabilirliği artıran açıklanabilir yapay zekâ tekniklerini birleştirerek düzeltmeler önerebilmektedir (Saravanan ve ark., 2025).

2.1.2. Otomatik Test

Üretken yapay zekâ, yazılım testini otomatik test senaryosu oluşturma ve hata tanımlama yoluyla geliştirerek test kapsamını ve verimliliğini önemli ölçüde artırmaktadır. Yapay zekâ destekli bu yöntemler, manuel testlerin sınırlamalarını ele alarak daha iyi hata tahmini ve test sonucu analizine olanak tanımakta ve böylece yüksek yazılım kalitesi sağlamaktadır (Bajaj ve Samal, 2023; Hnatushenko ve Pavlenko, 2024). Yapay zekânın testlere entegrasyonu, gelişmiş yazılım güvenilirliği için öngörücü analize de olanak tanımaktadır (Nama, 2024).

2.1.3. Proje Yönetimi ve İş Birliği

Üretken yapay zekânın proje yönetimine, özellikle çevik ortamlarda entegrasyonu, iş akışlarının kolaylaştırılmasını, karar alma süreçlerini ve gelişmiş iş birliğini desteklemektedir. Bu entegrasyon, yaygın proje yönetimi zorluklarının ele alınmasına, üretkenliğin artırılmasına ve hızlı teslimat döngülerinin desteklenmesine katkı sağlamaktadır (Bahi ve ark., 2024; Vergara ve ark., 2025).

2.1.4. Güvenlik Geliştirme

Güvenlik bağlamında, üretken yapay zekâ, siber güvenlik tehditlerini gerçek zamanlı olarak tahmin etme, tespit etme ve azaltma yeteneğiyle web uygulamaları ve kurumsal sistemlerin korunmasına önemli ölçüde katkıda bulunmaktadır. Üretken yapay zekâ modellerinin uyarlanabilir yetenekleri, tehdit tespitini geliştirir ve proaktif güvenlik önlemlerinin uygulanmasına olanak tanır (Tariq, 2024; Vemuri ve ark., 2024). Bu teknikler, veri korumasını sağlamakta ve dijital ortamların gelişen tehditlere karşı bütünlüğünü muhafaza etmektedir.

2.2. Üretken Yapay Zekânın Yazılım Geliştirme Eğitimindeki Rolü

Üretken yapay zekâ, yazılım geliştirme eğitiminde dönüştürücü bir rol oynayarak hem öğretim hem de öğrenme deneyimlerini geliştirir. Üretken yapay zekânın etkisinin belirgin olduğu birkaç temel alan şunlardır: Üretken yapay zekâ, yazılım geliştirme eğitiminde çok yönlü bir araç olarak öne çıkmaktadır. Öğrencilere kişiselleştirilmiş öğrenme deneyimleri sunarak, karmaşık programlama kavramlarını daha anlaşılır hale getirmektedir. Üretken yapay zekâ, öğrencilerin kendi hızlarında ilerlemelerine olanak tanırken, anlık geri bildirimler sağlayarak öğrenme sürecini hızlandırmaktadır. Ayrıca, gerçek dünya senaryolarına dayalı pratik örnekler üreterek, teorik bilgilerin uygulamaya dönüştürülmesini kolaylaştırmaktadır.

Eğitmenler açısından üretken yapay zekâ ders içeriklerinin hazırlanmasından değerlendirme süreçlerine kadar geniş bir yelpazede destek sağlamaktadır. Otomatik kod analizi ve hata tespiti özellikleri, eğitmenlerin öğrenci çalışmalarını daha verimli bir şekilde değerlendirmesine yardımcı olmaktadır. Bunun yanı sıra, üretken yapay zekâ, eğitim materyallerinin sürekli güncellenmesini ve endüstri trendlerine uygun hale getirilmesini sağlayarak, öğrencilerin güncel teknolojilerle tanışmasını mümkün kılmaktadır. Bu sayede, yazılım geliştirme eğitimi daha dinamik, etkileşimli ve sektör ihtiyaçlarına uygun bir yapıya kavuşmaktadır. Üretken yapay zekânın etkisinin yoğun şekilde hissedildiği başlıca alanlar aşağıda sıralanmıştır.

2.2.1. Kodlama ve Hata Ayıklama Yardımı

Üretken yapay zekâ, kod oluşturma ve hata ayıklama gibi kritik geliştirme görevlerinin otomasyonuna katkıda bulunmaktadır. OpenAI'nin Codex'i gibi büyük dil modelleri, doğal dil açıklamalarından sözdizimsel olarak doğru ve bağlam farkında kaynak kodu üreterek geliştirme süresini %45'e kadar azaltmakta ve kod kalitesini artırmaktadır. Bu durum, acemi programcılar için giriş engelini azaltmakta ve gerekçeli kod parçacıkları sağlamak amacıyla açıklanabilir yapay zekâ tekniklerini birleştirerek geliştiriciler arasında güven ve kullanılabilirliği artırmaktadır (Saravanan ve ark., 2025).

2.2.2. Müfredat Geliştirme

Üretken yapay zekâ, çeşitli öğrenci gereksinimlerini karşılayan kişiselleştirilmiş ve uyarlanabilir yollar oluşturarak eğitim müfredatlarının geliştirilmesini yeniden şekillendirmektedir. Bu durum, geleneksel

eğitim sınırlamalarını ele almakta ve uygulamalı öğrenmeyi geliştiren sanal laboratuvarlar ve simülasyonların oluşturulmasını kolaylaştırmaktadır. Sonuç olarak, üretken yapay zekâ, yükseköğretimde daha kişiselleştirilmiş ve verimli eğitim deneyimlerine katkıda bulunmaktadır (Tariq, 2024).

2.2.3. Öz Düzenlemeli Öğrenme

K-12 ortamlarında, üretken yapay zekâ, öğrenci katılımını artıran, anında geri bildirim sunan ve öz değerlendirmeyi teşvik eden araçlar sağlayarak öz düzenlemeli öğrenmenin geliştirilmesine katkıda bulunmaktadır. Bu yaklaşım, öğretimi, öğretmenlerin kolaylaştırıcı olarak hareket ettiği ve bireysel öğrenci ihtiyaçlarını karşılamak amacıyla talimatları kişiselleştirdiği daha insan merkezli bir süreçte dönüştürmektedir (Kong ve Yang, 2024).

2.2.4. Değerlendirme ve Geri Bildirim

Üretken yapay zekâ araçları kodlama alıştırmalarına ve argümanlar oluşturmaya yardımcı olmasını sağlayarak hem biçimlendirici hem de toplu değerlendirmelere katkıda bulunur. Bu araçlar, öğrencilerin yapay zekâ çıktılarının eleştirel bir değerlendirmesini yapmalarına olanak tanır ve böylece ödevleri onlar adına tamamlamadan yapay zekâyı öğrenme süreçlerine entegre eder. Bu yaklaşım, öğrencilerin ihtiyaçları ve sektör beklentileriyle uyumlu olduğundan, öğrencileri gerçek dünyadaki yazılım mühendisliği rollerine hazırlamada önemli bir araç haline gelmektedir (Petrovska ve ark., 2024).

2.2.5. Uyarlanabilir ve Kişiselleştirilmiş Öğrenme

Üretken yapay zekânın entegrasyonu, uyarlanabilir ve kişiselleştirilmiş öğrenme ortamlarının teşvik edilmesine katkıda bulunmaktadır. Bu entegrasyon, öğrenci katılımını artırmakta, test puanlarını iyileştirmekte ve kişiye özel eğitim deneyimleri aracılığıyla beceri gelişimini hızlandırmaktadır. Bu durum, geleneksel eğitim yöntemlerinden, kişiselleştirmenin yanı sıra sosyal-duygusal öğrenmeyi de içeren daha yenilikçi yaklaşımlara geçişi işaret etmektedir (Guettala ve ark., 2024).

2.2.6. Öğretim Çerçevesi

Üretken yapay zekâ, eğitim ortamlarında öğretim çerçevelerinin geliştirilmesine önemli katkılar sunmaktadır. Eğitimcilerin stratejilerini geliştirmelerine ve yapay zekâ ile entegre dersler tasarımlarına olanak tanıyarak, öğrencilerin dikkatini, katılımını, özgüvenini ve memnuniyetini artırmaktadır. Bu tür çerçeveler, eğitimin geleceğe hazırlanması ve yapay zekâ teknolojilerinin öğretim uygulamalarına daha iyi entegre edilmesi açısından kritik bir öneme sahiptir (Kong ve Yang, 2024).

Üretken Yapay Zekâ, kodlama ve eğitim alanlarında sunduğu fırsatların yanı sıra, öğrencilere çeşitli zorluklar da sunmaktadır. Öğrencilerin yapay zekâyı aşırı güvenmesi, eleştirel düşünme ve özgün fikir üretme becerilerini zayıflatabilir. Yapay zekâ tarafından üretilen içeriği yeterince analiz etmeden kullanmak, öğrencilerin temel kavramları ve süreçleri derinlemesine anlamadan yüzeysel bir öğrenme

gerçekleştirmelerine yol açabilir. Bu durum, özellikle yazılım geliştirmede kritik öneme sahip problem çözme becerilerinin gelişimini olumsuz etkileyebilir (Deng ve Joshi, 2024). Bu zorluklar, eğitimcilerin rolünü yeniden şekillendirmekte ve eğitim politikalarında değişiklikler gerektirmektedir. Eğitimciler, öğrencilere sadece teknik becerileri öğretmekle kalmayıp, aynı zamanda yapay zekâ çıktılarını eleştirel bir gözle değerlendirme, doğrulama ve etik kullanım becerilerini de kazandırmalıdır (Aarti, 2024). Bu bağlamda, sorumlu yapay zekâ entegrasyonu ve etik kullanımı vurgulayan eğitim politikalarının geliştirilmesi büyük önem taşımaktadır (Alali ve Wardat, 2024). Öğrencilerin yapay zekâ araçlarını bilinçli ve etkin bir şekilde kullanabilmeleri için, eleştirel düşünme, problem çözme ve etik karar verme becerilerinin geliştirilmesine odaklanan bütüncül bir eğitim yaklaşımı benimsenmesi gerekmektedir. Bu sayede, öğrenciler yapay zekânın sunduğu fırsatlardan yararlanırken, potansiyel dezavantajlarını da en aza indirebileceklerdir.

Üretken yapay zekânın yazılım geliştirme eğitimindeki potansiyel rolü umut verici olmakla birlikte, bu durum etik, akademik dürüstlük ve teknik güvenilirlik konularında birtakım zorlukları da beraberinde getirmektedir. Genel olarak, üretken yapay zekâ, eğitimcilerin yerini alacak bir araç olarak değil, yazılım geliştirmenin öğretilme ve öğrenilme biçiminde devrim yaratabilecek bir araç olarak değerlendirilmektedir. Eğitim kurumları, bu zorlukları dikkatlice ele alarak, daha verimli, akıllı ve erişilebilir öğrenme ortamları yaratmak amacıyla bu teknolojinin dönüştürücü potansiyelinden faydalanabilirler (Sultana, 2024).

2.3. Popüler Üretken Yapay Zekâ Kodlama Araçları

Üretken yapay zekâ araçları, yazılım geliştirme sürecinde devrim niteliğinde değişimler yaratmaktadır. Kod üretimi, hata tespiti ve algoritma optimizasyonu gibi görevleri otomatikleştirerek, geliştiricilerin verimliliğini artırmakta ve yaratıcı süreçlere odaklanmalarını sağlamaktadır. Farklı programlama dillerine uyarlanabilir yapıları sayesinde çok çeşitli alanlarda etkili olan bu araçlar, programcıların yetkinliklerini tamamlayıcı niteliktedir. Aşağıda kod üretebilen bazı popüler üretken yapay zekâ araçlarına dair kısa bir genel bakış sunulmaktadır:

2.3.1. GitHub Copilot

Visual Studio Code ile entegre edilmiş yapay zekâ tabanlı bir araç olan GitHub Copilot, geliştiricilere akıllı kodlama önerileri sunmak amacıyla tasarlanmıştır. Codex adlı dil modeli tarafından desteklenen bu araç, otomatik kod tamamlama ve sentezleme yoluyla geliştirme sürecini önemli ölçüde iyileştiren özellikler sunmaktadır (Sobania ve ark., 2022).

"Yapay zekâ çift programcısı" olarak işlev gören bu araç, kodu otomatik tamamlama, sözdizimini hatırlamaya yardımcı olma ve tuş vuruşlarını azaltarak kodlama görevlerini daha hızlı ve verimli hale getirme yeteneği nedeniyle geliştiriciler tarafından benimsenmiştir (Liang ve ark., 2024).

"Yapay zekâ çift programcısı" olarak adlandırılan ve OpenAI'nin GPT-3'ün bir türevi olan Codex tarafından desteklenen bir araçtır. Bu araç, geliştiricinin sağladığı bağlam doğrultusunda kod

parçacıkları ve tam işlevler önerir. Visual Studio Code gibi entegre geliştirme ortamlarına (IDE) entegre edilmiştir ve özellikle JavaScript ve Python dillerinde yaygın olarak kullanılmaktadır. Copilot, kod satırlarını otomatik olarak tamamlayarak ve yöntemler önererek üretkenliği artırmaktadır; ancak, güvenlik açıkları açısından dikkatli bir inceleme gerektirmektedir (Fu ve ark., 2025; Pearce ve ark., 2025).

2.3.2. OpenAI Codex

OpenAI Codex, doğal dil açıklamalarından kod üretme yeteneğiyle yazılım geliştirme ve eğitim alanlarını dönüştüren üretken bir yapay zekâ modelidir. Bu model, kod oluşturma, hata ayıklama ve optimizasyon gibi yetenekler sunarak üretkenliği önemli ölçüde artırmakta ve yeni başlayan programcılar için giriş engellerini azaltmaktadır. Bu özellikler, Codex'i bir dizi büyük dil modelinin (LLM) parçası haline getirmektedir (Saravanan ve ark., 2025).

GPT-3 ailesinin bir üyesi olarak, GitHub Copilot gibi araçlara güç sağlamaktadır ve doğal dil açıklamalarından kod anlama ve oluşturma konusunda üstün yetenekler sunmaktadır. Birden fazla programlama dilini destekleyen Codex, kod parçacıkları, hata ayıklama ve daha karmaşık programlama görevleri için oldukça kullanışlıdır (Aarti, 2024).

OpenAI Codex, kod oluşturma sürecini daha sezgisel ve erişilebilir hale getirerek yazılım mühendisliği ve eğitimdeki geleneksel uygulamaları dönüştürmektedir. Avantajlarına rağmen, Codex'in kullanımı, potansiyelini en üst düzeye çıkarırken riskleri azaltmak için sürekli araştırma ve etik değerlendirmeler gerektiren zorluklarla birlikte gelir (Macneil ve ark., 2023; Aarti, 2024).

2.3.3. Tabnines

Yazılım geliştirme süreçlerini iyileştirmek amacıyla tasarlanmış, yapay zekâ destekli bir kod asistanıdır. Bu araç, gerçek zamanlı kod önerileri, akıllı kod tamamlama ve kodlama görevlerinin otomasyonu sunarak, yazılan kodun bağlamını ve amacını anlamak için makine öğrenimi modellerinden faydalanır ve geliştiricilere verimli kod oluşturma ve tamamlama özellikleri sağlar (Aarti, 2024).

Tabnine'in temel avantajlarından biri, tekrarlayan görevleri otomatikleştirerek ve kod yazma süresini azaltarak geliştirici verimliliğini artırma yeteneğidir. Bu, genellikle doğru ve bağlamsal olarak alakalı kod önerileri üreterek gerçekleştirilir ve yapay zekâ, kod oluşturma rutin kısımlarını üstlenirken, geliştiricilerin geliştirme süreçlerinin kritik yönlerine daha fazla odaklanmalarını mümkün kılar (Ajiga ve ark., 2024).

Bu yapay zekâ aracı, genel ve özel kod depolarından öğrenerek kod tamamlama ve öneriler sağlar. Geliştiricilerin birden fazla programlama dilinde verimli bir şekilde kod yazmalarına yardımcı olmayı amaçlar ve çeşitli geliştirme ortamlarıyla uyumludur (Corso ve ark., 2024).

2.3.4. ChatGPT

Önemli bir yapay zekâ dil modeli olan ChatGPT, yazılım geliştirmeye çeşitli açılardan katkı sağlayarak çok sayıda programlama dilini desteklemektedir: ChatGPT, Python ve Java gibi çeşitli programlama dillerinde kod oluşturma ve geliştirme yeteneğine sahiptir. Ayrıca, farklı dillerde kodun yeniden yazılmasına, hata ayıklamaya ve sayısal algoritmaların performansını artırmaya yardımcı olmaktadır (Kashefi ve Mukerji, 2023; Ouh ve ark., 2023).

OpenAI tarafından geliştirilen ChatGPT, yalnızca bir kodlama aracı olmasa da kod üretebilir ve kodlamayla ilgili sorgulara yardımcı olabilir. Çeşitli alanlardaki kod oluşturma isteklerini işleyebilir ve hata ayıklama ve kodlama çözümlerini keşfetmeye yardımcı olabilir (Özpolat ve ark., 2023).

Yapay zekâ modeli, kodlama alıştırmalarına çözümler sunarak, alternatif yaklaşımlar önererek ve öz düzenlemeli öğrenme süreçlerini destekleyerek programlama eğitimini geliştirmektedir (Hartley ve ark., 2024; Yang ve ark., 2025).

2.3.5. Amazon CodeWhisperer

Üretken yapay zekânın dönüştürücü etkisinin belirgin olduğu kod üretimi alanında önemli bir yardımcıdır. Tekrarlayan kodlama görevlerini otomatikleştirerek, geliştiricilerin yazılım geliştirmenin daha karmaşık ve yaratıcı yönlerine odaklanmalarını sağlamak suretiyle genel üretkenliği artırır (Ajiga ve ark., 2024).

Ayrıca, aracın makine öğrenimi algoritmaları, önceki projelerden elde edilen geniş veri kümelerini analiz ederek öngörücü analizler sunabilir ve bu da daha bilinçli karar alma ve kaynak tahsisi sağlar. Bu öngörücü yetenek, geliştirme ekiplerinin çevikliğini korumak, değişikliklere hızla uyum sağlamalarını ve proje teslim tarihlerini verimli bir şekilde karşılamalarını sağlamak için kritik öneme sahiptir.

Amazon CodeWhisperer'ın bir diğer önemli avantajı, akıllı hata ayıklama ve otomatik hata tespiti yoluyla kod kalitesini artırmadaki rolüdür. Bu, hataların belirlenmesi ve düzeltilmesi için harcanan süreyi azaltmanın yanı sıra, nihai yazılım ürününün güvenilirliğini de artırır (Alenezi ve Akour, 2025).

2.3.6. Codeium

Akıllı kod önerileri ve eksiksiz kod parçacıkları sunarak kodlama hatalarını azaltmayı ve üretkenliği artırmayı amaçlayan gelişmiş bir gerçek zamanlı kodlama aracıdır. Bu araç, güvenlik ve etik etkilerle ilgili endişeler bulunmasına rağmen, bu görevlerde yardımcı olmak için büyük dil modellerinin (LLM) yeteneklerinden faydalanmaktadır. LLM'ler, doğal dil girdilerinden kod oluşturma, çevirme ve optimize etme konularında güçlü yetenekler sergileyerek yazılım geliştirmeye önemli katkılarda bulunmuştur (Cooper ve ark., 2024; Ni ve ark., 2024). Örneğin, Programcı Asistanı gibi sistemler, genişletilmiş etkileşimleri destekleyen ve böylece devam eden etkileşimlerden bağlamı dahil ederek programcıların üretkenliğini artıran konuşma temelli LLM'ler kullanmaktadır (Ross ve ark., 2023).

2.3.7. Replit Ghostwriter

Replit güçlü bir yapay zekâ aracıdır ve otomatik kod tamamlama ile akıllı öneriler sunarak kodlama deneyimini geliştirmek amacıyla özel olarak tasarlanmıştır. Bu araç, etkileşimli bir kodlama ortamında üretkenliği önemli ölçüde artırabilir ve bu durum, onu özellikle kodlama kavramlarını inceleyen eğitimciler ve öğrenciler için değerli kılmaktadır (Owan ve ark., 2023; Petrovska ve ark., 2024). Eğitim alanında, Ghostwriter gibi üretken yapay zekâ araçları, kişiselleştirilmiş öğrenme deneyimlerine katkıda bulunabilir. Karmaşık programlama kavramlarını anlamak için kritik öneme sahip olan kişiye özel geri bildirim ve gerçek zamanlı yardım sunarak çeşitli öğrenme stillerini desteklerler. Bu araçlar, kodlama sürecini kolaylaştırarak öğrencilerin sözdiziminden ziyade problem çözme ve kodlamanın yaratıcı yönlerine daha fazla odaklanmalarını sağlamaktadır (Imran ve ark., 2023; P, 2024).

Üretken Yapay Zekâ Kodlama Araçları üzerine yapılan akademik araştırmalar, bu araçların yazılım mühendisliği, programlama eğitimi, eğitim ve güvenlik boyutlarındaki etkilerini çeşitli perspektiflerden incelemektedir. ChatGPT'nin yazılım mühendisliği eğitiminde kullanımı üzerine yapılan araştırmalar, bu teknolojinin hem pedagojik hem de pratik açılardan dönüştürücü bir potansiyele sahip olduğunu göstermektedir. Liu (2024), büyük dil modellerinin ders içi uygulamalara entegre edilmesinin öğrencilerin problem çözme süreçlerini kolaylaştırdığını ve yazılım geliştirme becerilerini desteklediğini vurgulamaktadır. Benzer şekilde, Daun ve Brings (2023) ChatGPT'nin mühendislik eğitiminde paradigma değişimine yol açabileceğini, öğretim üyelerinin rolünü yeniden tanımladığını ve öğrenme deneyimini kişiselleştirdiğini belirtmektedir. Li ve ark., (2023) ChatGPT'nin öğrencilere gerçek zamanlı geri bildirim sağlayarak yazılım mühendisliği derslerinde etkileşimi artırdığını ortaya koyarken, Abdelfattah ve ark., (2023) bu teknolojinin gelecekteki eğitim yaklaşımlarına yönelik bir yol haritası sunduğunu öne sürmektedir. Bu çalışmalar, ChatGPT'nin yazılım mühendisliği eğitiminde yalnızca teknik destek sağlayan bir araç değil, aynı zamanda pedagojik yeniliklerin de katalizörü olabileceğini göstermektedir (Nizamudeen ve ark., 2024). Hollanda'daki bir üniversitede gerçekleştirdikleri araştırmada, öğrencilerle yapılan görüşmeler ve anketler aracılığıyla GitHub Copilot ve ChatGPT'nin programlama eğitimindeki etkilerini incelemiştir. Bulgular, öğrencilerin bu araçlara genel olarak olumlu yaklaştığını ve birçok üniversite düzeyindeki programlama ödevinin kısmen ya da tamamen bu araçlarla çözülebildiğini göstermektedir. Çalışma, bu durumun öğrenmeyi desteklediğini, ancak öğretim üyelerinin ödev ve içerik tasarımlarını yeniden gözden geçirmesi gerektiğini vurgulamaktadır (Araujo ve Hayajneh, 2025). Amazon CodeWhisperer tarafından üretilen kodun güvenlik açısından değerlendirilmesini gerçekleştirmiştir. Çalışmada, üretilen kodların %85'inin güvenlik zafiyetleri içerdiği ve CodeWhisperer'in güvenlik tarama aracının bu zafiyetlerin yalnızca %62'sini tespit edebildiği ortaya konulmuştur. Bu bulgular, kod üretim araçlarının yazılım geliştirmede verimlilik sağlarken aynı zamanda ciddi güvenlik riskleri de barındırabileceğini göstermektedir. Ovi ve ark., (2024) büyük dil modellerine dayalı programlama asistanlarını (ChatGPT, GitHub Copilot, Codeium) kod üretimi, hata ayıklama ve optimizasyon bağlamında karşılaştırmalı olarak değerlendirmiş ve başarı, çalışma süresi, bellek kullanımı ve hata yönetimi gibi metrikleri incelemiştir. Bulguları, bu

araçların özellikle düşük/orta karmaşıklıkta görevlerde anlamlı kazanımlar sağladığını; GitHub Copilot'un genel performansta öne çıktığını, ChatGPT'nin bellek verimliliği ve hata ayıklama süreçlerinde güçlü olduğunu; Codeium'un ise yüksek karmaşıklıkta daha fazla zorlandığını göstermektedir. Bununla birlikte, ileri düzey görevlerde tüm araçların sınırlılıkları bulunduğundan, büyük dil modellerinin eğitim ve yazılım geliştirme iş akışlarında tamamlayıcı araçlar olarak, sıkı test ve doğrulama süreçleriyle birlikte kullanılmasının önemi vurgulanmaktadır. Ovi ve ark., (2024) ile Chang ve ark., (2025) sistematik derlemeleri, Codex'in hız, hata azaltma ve çoklu dil desteği gibi güçlü yönlerini ortaya koyarken, etik kaygılar ve bağımlılık risklerine dikkat çekmiştir. Godoy ve ark., (2023) Codex'in HPC ortamlarında temel kod üretiminde başarılı olduğunu, ancak karmaşık optimizasyonlarda yetersiz kaldığını belirtmiştir. Eğitim alanında Finnie-Ansley ve ark., (2023), Codex'in öğretme sürecini kolaylaştırmasına rağmen akademik dürüstlük ve sınav güvenliği açısından riskler doğurduğunu vurgulamıştır. Prenner ve ark., (2022) ise, Codex'in yaygın hata türlerini düzeltebildiğini, ancak karmaşık mantık hatalarında sınırlı kaldığını göstermiştir.

Sonuç olarak, literatürdeki çalışmalar üretken yapay zekâ kodlama araçlarının eğitimden yazılım geliştirmeye, hata ayıklamadan güvenliğe kadar geniş bir yelpazede önemli katkılar sunduğunu, ancak etik, güvenlik ve performans sınırlarının bu teknolojilerin dikkatli kullanılmasını gerektirdiğini ortaya koymaktadır.

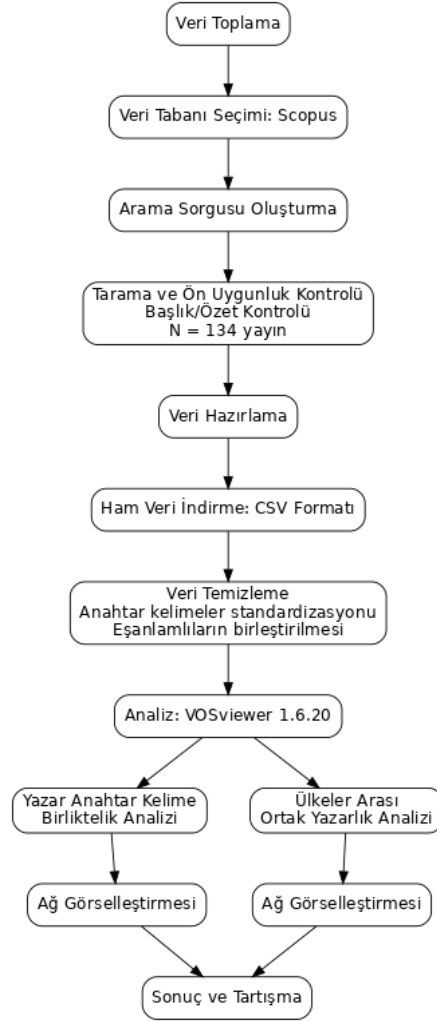
3. Materyal ve Metot

Bu çalışmada, üretken yapay zekânın yazılım mühendisliği eğitimi bağlamında kullanımını analiz etmek amacıyla kapsamlı bir literatür taraması gerçekleştirilmiştir. Veri kaynağı olarak Scopus veri tabanı tercih edilmiştir; Scopus, gelişmiş analiz araçları, bibliyometrik göstergeler ve kullanıcı dostu bir arayüz sunarak araştırmacılara destek sağlamaktadır. Kapsamlı literatür taramaları, bibliyometrik analizler ve bilimsel ölçüm çalışmaları yürütmek için önemli bir araçtır (Schotten ve ark., 2017). Literatür taramasında, alanla doğrudan ilişkili çalışmaları belirlemek amacıyla kapsamlı bir arama sorgusu kullanılmıştır. Arama ifadesi olarak (TITLE (genAI) OR TITLE (generative ai) OR TITLE-ABS-KEY (generative artificial intelligent) OR TITLE (chatGPT) OR TITLE (LLM) OR TITLE (AI programming tutor) AND TITLE-ABS-KEY (software engineering education)) sorgusu kullanılmıştır. Arama 29 Temmuz 2025 tarihinde gerçekleştirilmiş olup, bu sorgu, hem başlık hem de özet/anahtar kelime alanlarında üretken yapay zekâ ile yazılım mühendisliği eğitimi kesiştiren çalışmaları tespit etmeye yönelik olarak özel biçimde yapılandırılmıştır. Bu çalışmanın odak noktası olarak "software engineering education" alanının seçilmesinin temel nedeni, yazılım geliştirme sürecinin yalnızca programlama becerilerinden ibaret olmaması, aynı zamanda sistematik düşünme, mimari tasarım, yazılım doğrulama ve sürdürme gibi çok boyutlu bilişsel ve teknik yeterlilikler içermesidir. Yazılım mühendisliği eğitimi, üretken yapay zekâ araçlarının bu karmaşık süreçlerde nasıl bir öğrenme destekleyicisi rolü üstlendiğini analiz etmek açısından zengin bir bağlam sunmaktadır. Ayrıca bu alan, yapay zekâ destekli öğrenme ortamlarının pedagojik etkilerinin ve öğrencilerin problem çözme,

algoritmik düşünme ve yazılım kalitesi üretme gibi yetkinlikler üzerindeki katkılarının daha somut gözlemlenebileceği bir zemin sunmaktadır. Bu detaylı arama sonucunda, çalışmanın kapsamına uygun olarak 134 akademik yayın tespit edilmiştir. İlgili literatürde üretken yapay zekâ ve yazılım mühendisliği eğitimi kesişiminde gerçekleştirilen çalışmaların sayıca sınırlı olması nedeniyle, araştırma makalesi, konferans bildirisi, konferans değerlendirmesi, kitap bölümü ve derleme makalesi türlerindeki yayınların tamamı herhangi bir dışlama yapılmaksızın analize dahil edilmiştir. Bu sayede analiz hem güncel hem de temsili bir yayın kümesine dayanarak gerçekleştirilmiştir.

Elde edilen veriler, bibliyometrik analiz için özel olarak tasarlanmış ve özellikle bibliyometrik haritaların oluşturulması ile görselleştirilmesi açısından önemli avantajlar sunan VOSviewer yazılımı (sürüm 1.6.20) kullanılarak incelenmiştir. Grafiksel gösterime odaklanması sayesinde büyük ölçekli bibliyometrik ağların yorumlanmasını kolaylaştıran VOSviewer, bibliyometrik yazılımlar arasında önemli bir konuma sahiptir (Eck ve Waltman, 2009). Bu yazılım, akademik yayınlar arasındaki ilişkileri görselleştirme ve analiz etme konusunda güçlü özelliklere sahiptir. Bibliyometrik analiz, yayınların atıf ağlarını, yazar iş birliklerini, anahtar kelime eğilimlerini ve araştırma alanındaki tematik kümelenmeleri ortaya çıkarmak için kullanılmıştır. Bu analiz yöntemi, üretken yapay zekâ ve yazılım eğitimi alanlarındaki araştırma trendlerini, etkili çalışmaları ve potansiyel araştırma boşluklarını belirlemeye yardımcı olacaktır.

Bu çalışmada, veri toplama, hazırlama, analiz ve tartışma aşamaları sistematik bir şekilde yürütülmüştür. Scopus veri tabanından elde edilen 134 yayın csv formatında indirilmiş ve anahtar kelime standardizasyonu yapıldıktan sonra VOSviewer 1.6.20 kullanılarak yazar anahtar kelime ve ülkeler arası ortak yazarlık analizleri gerçekleştirilmiştir. Bulgular, ağ görselleştirmeleri ile desteklenmiş ve tartışılmıştır. Sürecin tüm adımları Şekil 1'deki akış şemasında özetlenmektedir.



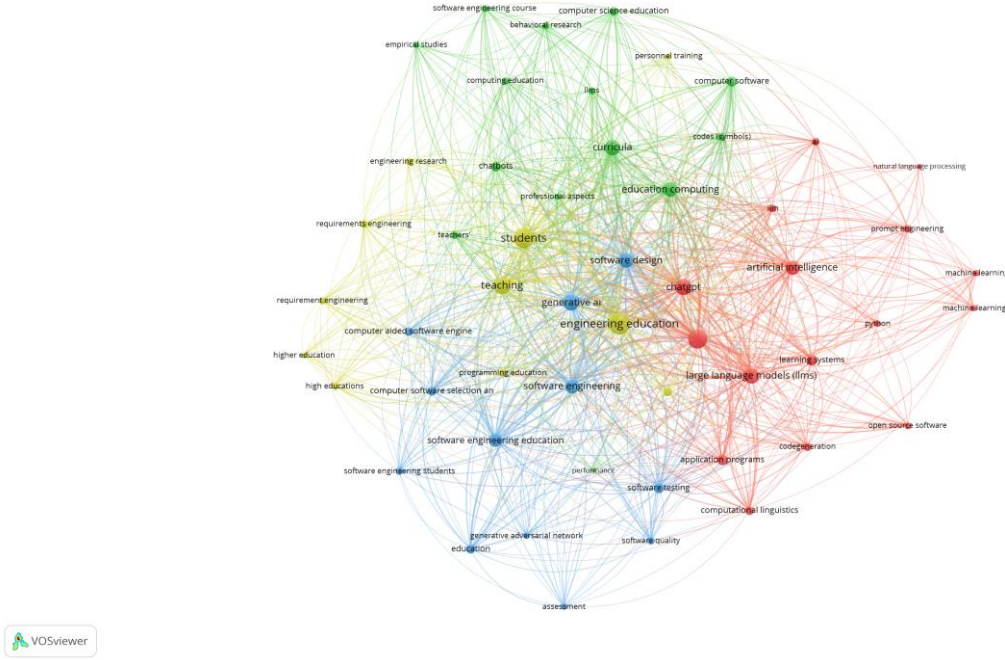
Şekil 1. Çalışmanın akış şeması

4. Bulgular

4.1. Yazar Anahtar Kelime Birliktelik Analizi

Yazar Anahtar Kelime Birliktelik Analizi Ağ Görselleştirmesi, bilimsel yayınlardaki anahtar kelimelerin kullanım örüntülerini ve yazarlar arasındaki ilişkileri incelemek amacıyla kullanılan etkili bir analitik yöntemdir. Bu analizde belirlenen beş eşik değeri, araştırma alanındaki önemli kavramları ve eğilimleri daha belirgin bir şekilde ortaya koymak için stratejik bir seçimdir. Bu eşik, yalnızca birden fazla yayında veya yazar tarafından kullanılan anahtar kelimelerin analize dâhil edilmesini sağlayarak, araştırma alanının daha kapsamlı ve güvenilir bir temsili sunar. Analizde tespit edilen 53 anahtar kelime, araştırma alanındaki temel kavramları ve güncel konuları yansıtmaktadır. Bu kelimelerin seçimi, alandaki bilgi birikiminin ve araştırma odaklarının bir haritasını çıkarmaya yardımcı olur. Ayrıca, bu yaklaşım, farklı yazarlar ve çalışmalar arasındaki bağlantıları göstererek, disiplinler arası iş birliği fırsatlarını ve araştırma boşluklarını belirlemeye olanak tanır. Sonuç olarak, bu analiz yöntemi, araştırma alanının evrimini anlamak ve gelecekteki araştırma yönelimlerini tahmin etmek için değerli bir araç sunmaktadır. Analiz sonucunda elde edilen ağ görselleştirmesi Şekil 2'de sunulmuştur. Bu ağda, düğüm boyutları ilgili anahtar kelimenin literatürdeki kullanım sıklığını temsil etmektedir; renkler, VOSviewer

tarafından belirlenen kümeleri göstermektedir; düğümler arasındaki çizgilerin kalınlığı ise anahtar kelimeler arasındaki toplam bağlantı gücüne karşılık gelmektedir. Böylece, Şekil 2 yalnızca kavramsal kümelenmeyi nitel olarak göstermemekte, aynı zamanda anahtar kelimelerin literatürdeki göreceli önemini ve birbirleriyle olan bağlantı yoğunluğunu nicel olarak da yansıtmaktadır. Bu yapı, üretken yapay zekâ ve yazılım mühendisliği eğitimi alanında hangi temaların merkezî konumda olduğunu ve hangi kavramların ikincil düzeyde ilişki kurduğunu sayısal olarak inceleme olanağı sunmaktadır.



Şekil 2. Yazar anahtar kelime birliktelik analizi ağ görselleştirmesi

Şekil 2 yazılım mühendisliği eğitimi bağlamında üretken yapay zekâ, odaklı çalışmaların yazar anahtar kelimeleri temelinde görselleştirildiği bir kavramsal birliktelik ağı sunmaktadır. Ağın merkezinde yer alan "engineering education", tematik kümelerin kesişim noktasında konumlanarak çalışmalarda en yaygın ve bağlayıcı kavram olduğunu göstermektedir. Özellikle "generative ai", "chatgpt" ve "large language models (llms)" gibi terimlerin, yüksek merkezîyetleri ve çok sayıda kavramla olan bağlantıları sayesinde ağın odak noktalarından biri haline geldiği görülmektedir. Bu durum, üretken yapay zekânın yazılım eğitimi üzerindeki dönüştürücü etkisini açıkça yansıtmaktadır. "artificial intelligence", "machine learning", "natural language processing", "prompt engineering" ve "python" gibi kavramları içeren kırmızı küme, üretken yapay zekânın teknik bileşenlerini öne çıkarırken; "software engineering", "software engineering education", "software quality", "software testing", "application programs" ve "assessment" gibi kavramların yer aldığı mavi küme, yazılım mühendisliği eğitiminin geleneksel yapılarına odaklanmaktadır. Bu bağlamda, "software engineering education" terimi hem yazılım mühendisliği pedagojisini hem de eğitimde kalite, değerlendirme ve öğrencilerin yazılım geliştirme yetkinliklerini artırma çabalarını yansıtan merkezi bir kavram olarak öne çıkmaktadır. "curricula", "education computing", "students", "teachers", "computer science education" ve "behavioral research"

gibi kavramlarla temsil edilen yeşil küme, pedagojik yaklaşımlar ve eğitim içeriği tasarımıyla ilişkilidir. "higher education", "requirement engineering", "empirical studies", "teaching" ve "professional aspects" gibi kavramlardan oluşan sarı küme ise yükseköğretim düzeyindeki eğitim ortamlarını ve gereksinim mühendisliğini kapsamaktadır. Bu kümeler arasında yer alan "students", "teaching", "software engineering education" ve "curricula" gibi kavramlar, kümeler arası güçlü tematik bağlar kurmakta ve multidisipliner etkileşimi ortaya koymaktadır. Özellikle "generative ai" ve "chatgpt" gibi terimlerin son yıllarda yoğun biçimde literatüre girmesi, bu teknolojilerin hem teknik hem pedagojik alanlarda yaygınlaştığını göstermektedir. Bu analiz, yazılım mühendisliği eğitiminde üretken yapay zekânın yalnızca içerik değil, aynı zamanda öğretim yöntemleri ve değerlendirme süreçleri düzeyinde de kapsamlı bir paradigma değişimi başlattığını ortaya koymaktadır.

4.1.1. Küme 1 (Kırmızı): Üretken Yapay Zekâ ve Kodlama Teknolojileri

Bu çalışma, yazılım mühendisliği eğitimi bağlamında üretken yapay zekâ uygulamalarının, özellikle büyük dil modelleri ve otomatik kod üretimi süreçleriyle nasıl entegre edildiğini incelemektedir. Çalışmada öne çıkan anahtar kelimeler, yapay zekâ temelli öğrenme ortamlarının, doğal dil işleme teknolojilerinin ve kodlama araçlarının eğitimde üstlendiği yeni rolleri ortaya koymaktadır.

4.1.1.1. Kavramsal Odağın Tematik Başlıkları

4.1.1.1.2. Büyük Dil Modelleri ve Kod Üretimi

"Language model", "large language models (LLMs)" ve "LLM" gibi anahtar kelimeler, yazılım eğitiminde LLM tabanlı içerik üretiminin yaygınlaştığını göstermektedir. Bu anahtar kelimeler, öğretim materyallerinin otomatikleştirilmesi ve içerik oluşturmanın makine destekli hale gelmesini temsil eder. Aynı şekilde, "codegeneration" ve "application programs" anahtar kelimeleri de LLM'lerin otomatik kod üretiminde aktif olarak kullanıldığını vurgulamaktadır.

4.1.1.1.3. Yapay Zekâ Tabanlı Öğrenme Sistemleri

"learning systems", "ai" ve "artificial intelligence" anahtar kelimeleri, eğitimde yapay zekâ destekli sistemlerin öğrenme süreçlerine entegre edildiğini göstermektedir. Bu bağlamda, öğrenciler yalnızca bilgi tüketicisi değil, yapay zekâ ile öğrenme sürecinin aktif birer bileşeni haline gelmektedir.

4.1.1.1.4. Kodlama ve Programlama Araçları

"Python", "açık kaynak yazılım", "makine öğrenimi" ve "makine-öğrenimi" gibi anahtar kelimeler, yazılım eğitimi bağlamında en yaygın kullanılan programlama dillerini ve teknolojilerini temsil etmektedir. Açık kaynak yazılım ortamları, üretken yapay zekâ destekli kod geliştirme süreçlerini daha erişilebilir ve yaygın hale getirmektedir.

4.1.1.1.5. Doğal Dil İşleme (NLP) Tabanlı Etkileşim

"ChatGPT", "doğal dil işleme" ve "hesaplamalı dilbilim" anahtar kelimeleri, öğrencilerin doğal dil girdileriyle yazılım geliştirebileceği bir öğrenme modelinin ortaya çıktığını göstermektedir. Bu durum, özellikle kodlama eğitiminin erişilebilirliğini ve etkileşimli yapısını artırmaktadır.

Kümedeki anahtar kelimelerin ortalama yayın yıllarının 2024'e yakın olması, bu temaların literatürde oldukça yeni ve hızla yükselen konular olduğunu göstermektedir. Örneğin, "language model" (2024.16), "codegeneration" (2024.37) ve "open source software" (2025) gibi anahtar kelimeler bu yükselişi desteklerken, "chatgpt" (2023.96) erken benimsenen bir araç olarak öne çıkmaktadır. Bu bağlamda küme, üretken yapay zekânın yazılım eğitiminde yalnızca manuel kodlama becerilerini değil, aynı zamanda doğal dil işleme temelli araçlarla etkileşimli öğrenmeyi desteklediğini ortaya koymaktadır. Öğrenciler artık yalnızca geliştirici değil, "prompt engineering" aracılığıyla LLM'leri yönlendiren aktif öğrenenler haline gelmektedir. Disiplinlerarası yapısı sayesinde, bilgisayar bilimi, yapay zekâ, dilbilim ve eğitim teknolojilerini birleştiren bu yaklaşım, yazılım mühendisliği eğitiminin pedagojik ve teknolojik dönüşümünde belirleyici bir rol üstlenmektedir.

4.1.2. Küme 2 (Yeşil): Bilişim Eğitimi, Müfredat ve Öğretim Yaklaşımları

Küme 2, yazılım mühendisliği eğitiminin pedagojik ve içerik temelli boyutlarına odaklanan çalışmaları içermektedir. Özellikle bilişim eğitimi, müfredat geliştirme, öğretim yöntemleri, öğrenci performansı ve ampirik araştırmalar bu kümenin temel araştırma eksenlerini oluşturmaktadır. Anahtar kelimelerin yerleşimi ve bağlantı gücü, bu alanların yazılım eğitimi literatüründe giderek daha fazla önem kazandığını göstermektedir.

4.1.2.1. Kavramsal Odağın Tematik Başlıkları

4.1.2.1.1. Müfredat Geliştirme ve Bilişim Eğitimi

"curricula" (müfredat), 244 bağlantı gücü ve 32 kez kullanım sıklığıyla kümenin en merkezi anahtar kelimesidir. Bu durum, yazılım mühendisliği eğitiminde ders içeriklerinin nasıl yapılandırıldığını ele alan çalışmaların yoğunluğunu göstermektedir. Benzer şekilde, "education computing", "computing education" ve "computer science education" gibi anahtar kelimeler de bilgisayar destekli eğitimin ve bilişim pedagojisinin sistematik şekilde ele alındığını yansıtmaktadır.

4.1.2.1.2. Eğitsel Araçlar ve Öğrenme Teknolojileri

"chatbots", özellikle üretken yapay zekâ temelli araçların (örneğin ChatGPT) müfredat içinde nasıl konumlandığını gösterirken, "LLMs" anahtar kelimesinin bu kümede yer alması, üretken yapay zekânın yalnızca teknik değil, aynı zamanda pedagojik bir araç olarak da değerlendirildiğini ortaya koymaktadır. "computer software" ve "codes (symbols)" gibi anahtar kelimeler, eğitimde kullanılan yazılımsal içeriklerin ve sembolik temsillerin öğretimdeki rolünü göstermektedir.

4.1.2.1.3. Öğretmen Roller ve Öğrenme Dinamikleri

"teachers'", "professional aspects", "performance" ve "behavioral research" gibi anahtar kelimeler hem öğretmenlerin eğitsel katkılarını hem de öğrencilerin performanslarını değerlendiren davranışsal ve profesyonel araştırmaları temsil etmektedir. "software engineering course" ve "empirical studies" anahtar kelimeleri, yazılım mühendisliği derslerine dair ampirik gözlem ve veri temelli analizlerin yapıldığını göstermektedir.

Kümedeki anahtar kelimelerin ortalama yayın yıllarının genellikle 2024 civarında yoğunlaşması, bu alanların literatürde yeni ve hızla gelişen temalar olduğunu göstermektedir. Özellikle "performance" (2023.8) ve "chatbots" (2024) gibi uygulamalı anahtar kelimelerin güncelliği, teknolojinin eğitsel bağlamda hızla benimsenmeye başladığını ortaya koymaktadır. Bu bağlamda küme, yazılım mühendisliği eğitiminin teknolojik yönünden ziyade öğretim tasarımı, öğrenme süreçleri ve eğitim çıktıları gibi pedagojik boyutlara odaklanmaktadır. Ayrıca bilgisayar bilimi, eğitim bilimleri ve psikoloji gibi farklı disiplinleri bir araya getirmesiyle disiplinlerarası bir yaklaşımı temsil etmektedir. "chatbots" ve "LLMs" gibi anahtar kelimelerin öğretim ortamlarında aktif olarak kullanılması, bu kümenin uygulamalı eğitim teknolojilerine yönelik ilgisini yansıtırken; öğretmen rolleri, öğrenci performansı ve davranışsal araştırmalar gibi sosyal içerikli konuların yüksek atıf alması, bu çalışmaların literatürde etkili ve görünür olduğunu göstermektedir.

4.1.3. Küme 3 (Mavi): Yazılım Mühendisliği Eğitimi ve Üretken Yapay Zekâ Tabanlı Yazılım Geliştirme Küme 3, yazılım mühendisliği eğitimi ile üretken yapay zekâ uygulamalarının yazılım geliştirme süreçlerine entegrasyonuna odaklanan çalışmaları kapsamaktadır. Bu küme hem geleneksel yazılım mühendisliği bileşenlerini hem de üretken yapay zekâ destekli araçların eğitim ve yazılım kalitesi üzerindeki etkilerini bir araya getirmektedir.

4.1.3.1. Kavramsal Odağın Tematik Başlıkları

4.1.3.1.1. Üretken Yapay Zekâ ve Yeni Nesil Yazılım Geliştirme

"generative ai" (293 toplam bağlantı gücü) kümenin en merkezi anahtar kelimesidir ve yazılım eğitimi ile üretken yapay zekâ entegrasyonunu temsil eder. "generative adversarial networks" (37 toplam bağlantı gücü) derin öğrenme temelli üretken modellerin yazılım süreçlerinde değerlendirilmesini temsil eder. Bu anahtar kelimeler, üretken yapay zekânın sadece kod üretimi değil, aynı zamanda yazılım tasarımı ve kalite değerlendirme gibi alanlarda da aktif biçimde kullanıldığını göstermektedir.

4.1.3.1.2. Yazılım Tasarımı ve Mühendisliği Temelleri

"software design" (240 toplam bağlantı gücü) yazılım geliştirme sürecinin ilk aşamasındaki tasarım anahtar kelimesini temsil eder. "software engineering" (214 toplam bağlantı gücü) disiplinin genel çerçevesini ve mühendislik yaklaşımını tanımlar. "Software engineering education" (174 toplam bağlantı gücü) yazılım mühendisliği öğretim süreçlerini ve müfredat yapılarını incelemektedir.

Kalite, Test ve Değerlendirme: "software testing" (96 toplam bağlantı gücü), "software quality" (51 toplam bağlantı gücü), "assessment" (36 toplam bağlantı gücü) bu anahtar kelimeler, yazılım geliştirme çıktılarının doğruluk, güvenilirlik ve kalite açısından değerlendirilmesini içermektedir.

4.1.3.1.3. Eğitim Uygulamaları ve Öğrenci Deneyimi

"education" (82 toplam bağlantı gücü), "software engineering students" (51 toplam bağlantı gücü) bu anahtar kelimeler, üretken yapay zekâ destekli araçların öğrenci başarısı ve öğrenme deneyimi üzerindeki etkilerini işaret etmektedir. "computer software selection and evaluation" (79 toplam bağlantı gücü), "computer aided software engineering" (71 toplam bağlantı gücü) yazılım seçim kriterleri ve yazılım geliştirme süreçlerinde bilgisayar destekli araçların önemini vurgulamaktadır.

Küme 3' teki anahtar kelimelerin ortalama yayın yılının 2024 civarında yoğunlaşması, bu alanların güncel ve hızla yükselen araştırma temaları olduğunu ortaya koymaktadır. Özellikle "generative ai" (2024.30), "software design" (2024.42) ve "computer software selection and evaluation" (2024.3) gibi anahtar kelimelerin zaman dağılımı, üretken yapay zekâ ile yazılım mühendisliği entegrasyonunun yakın dönemde literatürde odak haline geldiğini göstermektedir. Bu bağlamda küme, geleneksel yazılım mühendisliği konularını (tasarım, test, kalite) içerirken; üretken yapay zekâ, generative adversarial networks ve değerlendirme araçları gibi yenilikçi teknolojileri de eğitim süreçlerine entegre etmektedir. Yazılım mühendisliği eğitimi, öğrenci başarısı, yazılım seçimi ve değerlendirme gibi pedagojik unsurlar bu kümede güçlü bir şekilde temsil edilirken; üretken yapay zekânın kod üretiminin ötesine geçerek tasarım ve kalite süreçlerine de nüfuz ettiği görülmektedir. Ayrıca yüksek bağlantı gücü ve yayın yılı ortalaması, bu anahtar kelimelerin hem teorik tartışmalarda hem de uygulamalı araştırmalarda yoğun ilgi gördüğünü göstermektedir.

4.1.4. Küme 4 (Sarı): Mühendislik Eğitimi, Öğrenci Odaklı Pedagoji ve Öğretim Yöntemleri

Küme 4, mühendislik eğitiminin temelleri, öğretim stratejileri ve öğrenci merkezli yaklaşımlar üzerine odaklanan çalışmaları içermektedir. Bu küme, özellikle mühendislik eğitimi bağlamında öğretim faaliyetleri, öğrenci deneyimi, programlama eğitimi ve gereksinim mühendisliği gibi konuları incelemektedir.

4.1.4.1. Temel Temalar ve Kavramsal Yapı

4.1.4.1.1. Mühendislik Eğitiminin Temel Anahtar Kelimeleri

"Engineering education" (562 toplam bağlantı gücü) kümenin en merkezi ve yoğun kullanılan anahtar kelimesidir. Yazılım ve genel mühendislik alanlarındaki eğitim uygulamalarının teorik çerçevesini oluşturur. "engineering research" (42 toplam bağlantı gücü) mühendislik eğitimi bağlamında yapılan bilimsel araştırmaları temsil eder. "higher education" (28 toplam bağlantı gücü), "high educations" (47 toplam bağlantı gücü) mühendislik eğitiminin üniversite düzeyindeki uygulamalarını ifade eder.

4.1.4.1.2. Öğrenci ve Öğretim Süreçleri

"students" (toplam bağlantı gücü) öğrenci merkezli öğrenme ve değerlendirme süreçlerini temsil eden ikinci en güçlü düğümdür. "teaching" (318 TLS, 44 kullanım, 2024.45) pedagojik uygulamalar ve öğretim stratejileri ile ilişkilidir. "personnel training" (25 toplam bağlantı gücü) öğrencilerin gelecekteki profesyonel rollere hazırlanmasına yönelik eğitim süreçlerini yansıtır.

4.1.4.1.3. Bilgisayar Destekli Eğitim ve Programlama

"computer aided instruction" (61 toplam bağlantı gücü) mühendislik eğitiminde bu yöntemlerin önemini vurgulamaktadır. "programming education" (53 toplam bağlantı gücü) ise programlama derslerinin öğretiminde kullanılan yöntem ve içeriklere odaklanan anahtar kelimedir.

4.1.4.1.4. Gereksinim Mühendisliği Eğitimi

"requirements engineering" (55 toplam bağlantı gücü) ve "requirement engineering" (39 toplam bağlantı gücü) terimleri, yazılım geliştirme sürecinin başlangıcında yer alan gereksinim toplama ve analiz süreçlerinin öğretimini kapsamaktadır. Yazım farklılıklarına rağmen, içerik açısından benzerlik gösteren bu anahtar kelimeler, terminolojik ayrışmalara rağmen ortak bir tema çerçevesinde değerlendirilmelidir. Küme 4'teki anahtar kelimelerin büyük çoğunluğunun ortalama yayın yılının 2024 civarında yoğunlaşması, bu konuların literatürde güncel ve hızla gelişmekte olan temalar olduğunu ortaya koymaktadır. Özellikle "computer aided instruction" (2024.57), "teaching" (2024.45), "students" (2024.27) ve "engineering education" (2024.23) gibi anahtar kelimelerin zaman dağılımı, mühendislik eğitiminin içerik ve yöntemlerinin 2020'li yıllarda dijital pedagojik yaklaşımlar doğrultusunda yeniden şekillendiğini göstermektedir. Bu bağlamda, "students" ve "teaching" anahtar kelimelerinin yüksek bağlantı gücü, eğitimde öğrenci ihtiyaçlarına duyarlı, katılımcı ve aktif öğrenmeyi teşvik eden pedagojik yaklaşımların öne çıktığını göstermektedir. "computer aided instruction" ve "programming education" gibi dijital eğitim odaklı anahtar kelimeler, mühendislik öğretiminde teknoloji destekli öğretim stratejilerinin önemini vurgularken; "requirements engineering" gibi teknik içerikli anahtar kelimeler, programların analitik düşünme, planlama ve sistem geliştirme odaklı yapısını ortaya koymaktadır. Genel olarak bu küme, eğitim, öğretim, araştırma ve teknik uzmanlık alanlarını bir araya getirerek mühendislik pedagojisinin hem kuramsal hem de uygulamalı boyutlarını bütüncül biçimde yansıtmaktadır.

Üretken yapay zekâ ve yazılım mühendisliği eğitimi literatüründe öne çıkan temaların daha ayrıntılı bir şekilde incelenebilmesi amacıyla, anahtar kelimelerin toplam bağlantı gücüne göre sıralandığı ilk 10 kavrama odaklanılmıştır. Tablo 1'de bu anahtar kelimelere ait küme bilgileri, bağlantı sayıları, toplam bağlantı gücü, görülme sıklıkları, ortalama yayın yılı ve ortalama atıf sayısı gibi bibliyometrik ölçütler yer almaktadır. Bu veriler, ilgili kavramların literatürdeki önemini, güncelliğini ve etki düzeyini ortaya koymaktadır.

Tablo 1. Yazar anahtar kelime birliktelik analizi toplam bağlantı gücüne göre ilk 10 anahtar kelime

Anahtar Kelime	Küme	Bağlantı Sayısı	Toplam Bağlantı Gücü	Kullanım Sıklığı	Ortalama Yılı	Yayın Ort. Alıntılar
engineering education	4	51	562	83	2024,2289	5,2169
students	4	49	468	67	2024,2687	5,1045
language model	1	50	411	56	2024,1607	7,6071
teaching	4	49	318	44	2024,4545	6,1136
generative ai	3	50	293	46	2024,3043	7,6957
chatgpt	1	48	278	53	2023,9623	13,6226
large language models (llms)	1	46	257	33	2024,0303	16,5152
curricula	2	43	244	32	2024,4688	3,0625
software design	3	50	240	35	2024,4286	2,0000
education computing	2	40	222	26	2024,3846	5,6538

Tablo 1’de yer alan anahtar kelimelerin toplam bağlantı gücüne göre yapılan sıralama, üretken yapay zekâ ve yazılım mühendisliği eğitimi alanında öne çıkan araştırma odaklarını ortaya koymaktadır. "Engineering education" (562 toplam bağlantı gücü) en yüksek değeri ile alandaki temel yapıyı temsil ederken, onu "students" (468) ve "language model" (411) anahtar kelimeleri takip etmektedir. Bu durum, mühendislik eğitiminin yapısal temelleri ve öğrenci merkezli yaklaşımlar ile büyük dil modellerinin entegrasyonunun literatürde yoğun biçimde tartışıldığını göstermektedir. "Generative AI" (293), "ChatGPT" (278) ve "large language models (LLMs)" (257) gibi üretken yapay zekâyâ yönelik anahtar kelimeler, bu teknolojilerin eğitim ortamlarında giderek daha fazla benimsendiğini yansıtırken; bu kavramların yüksek ortalama atıf değerleri (özellikle LLMs: 16,5; ChatGPT: 13,6) bu çalışmaların etkisini ortaya koymaktadır. Öte yandan, "curricula" (244), "software design" (240) ve "education computing" (222) gibi anahtar kelimeler, müfredat tasarımı, yazılım geliştirme süreci ve bilişim destekli eğitimin de güncel ve bütüncül biçimde değerlendirildiğini göstermektedir. Ortalama yayın yıllarının neredeyse tamamının 2024’e yakın olması, bu anahtar kelimelere dayalı araştırmaların oldukça güncel olduğunu ve hızlı bir şekilde gelişen bir akademik alanı temsil ettiğini ortaya koymaktadır.

Avrupa ve Kuzey Amerika'daki güçlü araştırma ağlarını temsil etmektedir. Asya ülkeleri (Çin, Güney Kore ve Hindistan) ise daha çok ABD ile ikili ilişkiler kurmakta, Avrupa ile daha zayıf bağlantılar göstermektedir. Sonuç olarak, Slovenya ve İspanya gibi ülkeler ağda daha çevresel konumda yer almakta, merkezi araştırma ağlarında daha düşük temsile sahiptir. Tablo 2, üretken yapay zekâ ve yazılım mühendisliği eğitimi alanında araştırma yapan ülkelerin karşılaştırmalı bir analizini sunmaktadır.

Tablo 2. Ülkeler arası ortak yazarlık analizi toplam bağlantı gücüne göre ülkeler

Ülke	Küme	Bağlantı	Toplam Bağlantı Gücü	Belge Sayısı	Atıf	Ort. Yayın Yılı
united states	6	9	11	33	430	2024,1818
finland	2	5	8	6	49	2024,1667
germany	5	5	7	16	220	2024
canada	2	4	5	4	8	2024,75
norway	1	5	5	3	135	2023,3333
china	3	3	3	11	17	2024,5455
india	1	3	3	5	131	2024
united arab emirates	1	3	3	2	136	2023
australia	3	2	2	4	89	2024,25
south korea	4	2	2	2	36	2021
taiwan	4	1	1	1	27	2018
slovenia	6	1	1	1	35	2024
spain	5	1	1	5	27	2024

Üretken yapay zekâ ve yazılım mühendisliği eğitimi alanındaki çalışmaların yıllara göre dağılımı incelendiğinde, en güncel yayınlara sahip ülkelerin sırasıyla Kanada (2024.75), Çin (2024.55), ABD (2024.18) ve Finlandiya (2024.17) olduğu tespit edilmektedir. Bu ülkeler, literatürdeki güncel eğilimleri belirleme ve yenilikçi eğitim uygulamalarını şekillendirme açısından öncü bir konumda bulunmaktadır. Almanya, Hindistan, Slovenya ve İspanya gibi ülkeler ise 2024 yılı civarındaki ortalama yayın yılı ile bu alana yönelik yakın dönemde artan araştırma ilgisini yansıtmaktadır. Buna karşılık, Birleşik Arap Emirlikleri (2023), Norveç (2023.33), Güney Kore (2021) ve özellikle Tayvan (2018) gibi ülkeler, bu alandaki çalışmalarını daha önceki yıllarda gerçekleştirmiş ancak güncelliğini sürdürmemiştir. Genel olarak bu analiz, üretken yapay zekâ uygulamalarının eğitim teknolojileri ile birleştiği güncel akademik

üretimin, başta ABD, Kanada, Çin ve Finlandiya olmak üzere belirli merkezlerde yoğunlaştığını ve bu ülkelerin hem bilimsel öncülük hem de uygulayıcı rol üstlendiğini ortaya koymaktadır.

Yapılan ortak yazarlık analizi, üretken yapay zekâ ve yazılım mühendisliği eğitimi alanındaki bilimsel üretimin coğrafi dağılımını ve uluslararası iş birliği ağlarını ortaya koymaktadır. Ağ yapısında en merkezi ve en büyük düğüm olarak öne çıkan Amerika Birleşik Devletleri (ABD), bu alanda en fazla dokümana (33) ve en yüksek atıf sayısına (430) sahip ülke konumundadır. ABD'nin, Finlandiya, Kanada, Almanya, Hindistan, Norveç, Çin ve Güney Kore gibi çok sayıda ülkeyle kurduğu yoğun iş birlikleri, bu alandaki araştırmalarda merkezi bir rol üstlendiğini göstermektedir. Finlandiya, görece az sayıda yayın (6) üretmiş olsa da yüksek toplam bağlantı gücü (8) ve güncel yayın yılı ortalaması (2024.17) ile dikkat çekerken; Almanya, 16 yayın ve 220 atıfla, Avrupa'daki araştırma ağlarının merkezinde yer almaktadır. Çin ve Avustralya hem ABD ile hem de Avrupa ülkeleriyle ilişkili görünüp Asya-Pasifik eksenli üretim potansiyelini yansıtmaktadır. Hindistan, Norveç ve Birleşik Arap Emirlikleri, sınırlı sayıda yayın üretmelerine rağmen yüksek atıf sayılarıyla (sırasıyla 131, 135 ve 136) bu çalışmaların etki gücünün yüksek olduğunu göstermektedir. Ayrıca, Slovenya, Güney Kore, Tayvan ve İspanya gibi ülkeler, çevresel konumda yer almakta ve genellikle ABD ile bireysel iş birlikleri aracılığıyla ağa dâhil olmaktadır. Özellikle Tayvan'ın ortalama yayın yılı 2018 olup, bu durum, ülkenin üretken yapay zekâ bağlamındaki çalışmalarının güncellik düzeyinin düşük olduğunu göstermektedir. Kümelenme yapısına bakıldığında; ABD ve Slovenya Küme 6'da Finlandiya ve Kanada Küme 2'de, Almanya ve İspanya Küme 5'te, Çin ve Avustralya Küme 3'te yer alırken; Hindistan, Norveç ve BAE gibi ülkeler Küme 1'de konumlanmıştır. Bu durum, üretken yapay zekâ uygulamalarının eğitimde kullanımı bağlamında çok merkezli ve kıtalar arası bir araştırma yapısının oluştuğunu göstermektedir.

5. Sonuç

Bu çalışma, üretken yapay zekâ teknolojilerinin yazılım mühendisliği eğitimi üzerindeki etkilerini kavramsal ve uluslararası düzeyde bibliyometrik yöntemlerle inceleyerek, bu alanın geçirdiği dönüşümün boyutlarını ortaya koymaktadır. Elde edilen bulgular, üretken yapay zekânın yalnızca teknik bir yenilik değil, aynı zamanda pedagojik bir paradigma değişiminin taşıyıcısı olduğunu açıkça göstermektedir.

Literatürdeki önceki bibliyometrik genel eğitim alanına odaklanarak geniş bir perspektif sunarken, bu çalışma yazılım mühendisliği disiplinine özgü olarak daha derinlemesine ve tematik bir inceleme gerçekleştirmektedir. Özellikle, yazılım mühendisliğinde kullanılan teknik uygulama alanları ve pedagojik dönüşüm süreçleri birlikte ele alınarak, alanın güncel eğilimleri ve gelişim dinamikleri kapsamlı biçimde ortaya konmaktadır çalışmalar (Vhatkar ve ark., 2024; Bicen, 2025). Anahtar kelime analizinde, “engineering education” (562 toplam bağlantı gücü), “students” (468) ve “language model” (411) gibi kavramların yüksek bağlantı gücüyle ön plana çıkması, mühendislik eğitiminin yapısal temelleri ile öğrenci merkezli öğrenmenin, büyük dil modelleriyle güçlü biçimde bütünleştiğini ortaya koymaktadır. “Generative AI”, “ChatGPT” ve “LLMs” gibi kavramlar ise hem teknolojik yenilikleri

hem de pedagojik uygulamaları temsil eden merkezi odaklar haline gelmiştir. Bu kavramların yüksek ortalama atıf sayıları, bu alandaki çalışmaların akademik etkisinin güçlü olduğunu göstermektedir.

Öne çıkan diğer anahtar kelimeler arasında “curricula”, “software design”, “education computing” ve “assessment” gibi kavramlar yer almakta olup, bunlar müfredat geliştirme, yazılım süreçlerinin eğitimi, teknoloji destekli öğretim ve değerlendirme boyutlarının üretken yapay zekâ ile nasıl dönüştüğünü yansıtmaktadır. Bu bağlamda, üretken yapay zekâ yalnızca kod üretimini değil, aynı zamanda öğrenme tasarımı, içerik yapısı ve öğrenme çıktılarının değerlendirilmesini de dönüştürmektedir.

Kavramsal kümeler incelendiğinde, üretken yapay zekânın eğitim üzerindeki etkileri dört temel ekseninde toplanmaktadır: (1) büyük dil modelleri ve otomatik kod üretimi, (2) öğretim teknolojileri ve müfredat tasarımı, (3) yazılım mühendisliği süreçleri ve kalite değerlendirme, (4) öğrenci odaklı öğretim yöntemleri ve mühendislik pedagojisi. Bu temalar, öğrencilerin artık yalnızca bilgi tüketicisi değil, OpenAI Codex, Tabnine, Amazon CodeWhisperer, Codeium, Replit Ghostwriter ve ChatGPT gibi araçlar aracılığıyla üretken, etkileşimli öğrenenler haline geldiğini; öğretmenlerin ise rehber, tasarımcı ve yapay zekâ entegratörüne dönüştüğünü göstermektedir.

Öte yandan, öğretmenlerin rolü yalnızca bilgi aktarıcısı olmaktan çıkmış; öğrenme sürecinin tasarımcısı, kolaylaştırıcısı ve etik rehberi haline gelmiştir. Öğretmenler, teknolojiyi pedagojik bağlamda entegre eden, kişiselleştirilmiş öğrenme yolları sunan, öğrenci verilerine dayalı geribildirim sağlayan ve öğrencilerin etik dijital vatandaşlık bilincini geliştiren rehberler olarak konumlanmaktadır. Müfredatlar artık sabit yapılar olmaktan çıkmış, öğretmenler tarafından dinamik ve modüler biçimde yapılandırılan yapılara dönüşmektedir. Bu dönüşüm yaratıcılığı teşvik eden, iş birliğine dayalı ve veri destekli öğrenmeyi önceliklendiren bir eğitim anlayışını da beraberinde getirmektedir. Öğrenciler dijital araçlarla aktif iletişim kurarken, öğretmenler bu iletişimi anlamlı öğrenmeye yönlendiren kolaylaştırıcılar rolünü üstlenmektedir. Sonuç olarak, üretken yapay zekâ ile desteklenen eğitim ortamlarında hem öğrenci hem de öğretmen rolleri daha katılımcı, uyarlanabilir ve insan merkezli bir yapıya evrilmektedir.

Ülkeler arası ortak yazarlık analizi, Amerika Birleşik Devletleri'nin merkezde yer aldığı çok merkezli ve kıtalar arası bir bilimsel iş birliği ağına işaret etmektedir. Kanada, Çin, Finlandiya ve Almanya gibi ülkeler, güncel ve etkili yayınlarıyla dikkat çekerken; Hindistan, Norveç ve Birleşik Arap Emirlikleri gibi ülkeler, sınırlı sayıda yayınlara yüksek atıf alarak nitelikli katkılar sunmuştur. Bu durum, üretken yapay zekâ destekli eğitim araştırmalarının yalnızca teknik açıdan değil, aynı zamanda kurumsal ve coğrafi düzeyde de yayıldığını göstermektedir.

Bulgular, yazılım mühendisliği eğitiminde geleneksel sabit müfredatların yerine, üretken yapay zekâ teknolojileriyle uyumlu, daha esnek ve dinamik eğitim yapılarına geçilmesi gerektiğini ortaya koymaktadır. Bu dönüşüm, hızla gelişen teknoloji dünyasında öğrencilerin güncel bilgi ve becerilerle donatılmasını sağlayacaktır. Yeni nesil eğitim modelinde, üniversitelerin uygulamalı yapay zekâ entegrasyonunu aktif olarak teşvik etmesi ve öğretmenlerin rolünün yeniden tanımlanması önem arz etmektedir.

Bu bağlamda, eğitim politikalarının kapsamlı bir şekilde yeniden değerlendirilmesi gerekmektedir. Öğretmenlerin yalnızca bilgi aktarıcı rolünden çıkarak, öğrencilere rehberlik eden ve eğitim süreçlerini tasarlayan bir role bürünmeleri amacıyla mesleki gelişim programları oluşturulmalıdır. Ayrıca, öğrencilerin yapay zekâ teknolojileriyle etkileşiminde etik sorumlulukların, telif haklarının ve akademik dürüstlüğün önemini vurgulayan güçlü dijital vatandaşlık politikaları geliştirilmelidir. Bu yaklaşım, öğrencilerin teknolojik gelişmelere uyum sağlamalarını desteklerken, aynı zamanda etik ve yasal çerçevede hareket etmelerini teşvik edecektir.

Tüm bu çıkarımların yanı sıra, araştırmanın metodolojik sınırlarının da dikkate alınması gerekmektedir. Veri kaynağının yalnızca Scopus ile sınırlı olması, elde edilen sonuçların kapsamını ve çeşitliliğini etkileyebilecek bir sınırlılık teşkil etmektedir. Scopus kapsamlı bir indeks olmasına rağmen, hızla gelişen ve disiplinlerarası özellik gösteren üretken yapay zekâ alanındaki bazı yayınlar bu veri tabanında yer almayabilir. Ayrıca, literatürün dinamizmi göz önünde bulundurulduğunda, bu araştırma mevcut durumun anlık bir görünümünü sunmakta olup, gelecekte yapılacak çalışmaların yeni bulgularla bu sonuçları genişletmesi ve güncellemesi beklenmektedir. Bu kapsamda, ilerleyen araştırmalarda birden fazla veri tabanının birlikte kullanılması, literatürün daha bütüncül biçimde temsil edilmesine katkı sağlayacaktır.

Gelecek araştırmaların, üretken yapay zekâ destekli öğrenmenin yalnızca kısa vadeli değil, aynı zamanda uzun vadeli bilgi kalıcılığı, kariyer gelişimi ve mesleki başarı üzerindeki etkilerini incelemesi gerekmektedir. Ayrıca, öğrencilerin yapay zekâyı bağımlılığı, telif hakları ve yanlış bilgi üretimi gibi etik ve sosyo-teknik boyutları derinlemesine ele alması önem arz etmektedir. Bunun yanı sıra, üniversite odaklı çalışmaların ötesine geçilerek ortaöğretim, mesleki eğitim ve disiplinler arası programlardaki uygulamalara da yönelmesi gerekmektedir.

Sonuç olarak, üretken yapay zekâ teknolojileri, yazılım mühendisliği eğitiminde teknik içerikten pedagojik yapıya kadar kapsamlı bir dönüşüm başlatmıştır. Bu yeni eğitim modeli, öğrencilerin aktif katılımcı ve üretken bireyler olarak konumlandığı, öğretmenlerin ise yapay zekâyı entegre eden rehber roller üstlendiği bir yapıyı teşvik ederek, öğrenme süreçlerini daha erişilebilir, kişiselleştirilmiş ve etkileşimli hale getirmektedir. Bu bağlamda, üretken yapay zekâ yalnızca bir araç değil, aynı zamanda eğitimde yeni bir çağın öncüsüdür. Gelecek çalışmaların, bu dönüşümün uzun vadeli etkilerine, etik boyutlarına ve farklı öğrenme düzeylerindeki uygulamalarına odaklanması beklenmektedir.

Bu çalışma, üretken yapay zekânın yazılım mühendisliği eğitiminde hangi tematik alanlarda yer aldığını ve kavramsal, pedagojik ve teknik düzeyde nasıl bir dönüşüm sağladığını açıkça ortaya koyarak araştırma sorusunu bütüncül bir şekilde yanıtlamaktadır.

Çıkar Çatışması Beyanı: Makalenin yazarları, çalışma kapsamında herhangi bir kişisel ve finansal çıkar çatışması olmadığını beyan eder.

Araştırmacıların Katkı Oranı Beyan Özeti: Yazarlar çalışmaya eşit oranda katkı sağlamıştır.

Kaynakça

- Aarti. Generative AI in software development: An overview and evaluation of modern coding tools. *International Journal for Multidisciplinary Research* 2024; 6(3). <https://doi.org/10.36948/ijfmr.2024.v06i03.23271>
- Abdelfattah AM., Ali NA., Elaziz MA., Ammar HH. Roadmap for software engineering education using ChatGPT. 2023 International Conference on Artificial Intelligence Science and Applications in Industry and Society (CAISAIS), 03-05 September 2023, 1–6, Galala, Egypt. <https://doi.org/10.1109/CAISAIS59399.2023.10270477>
- Ajiga D., Folorunsho S., Okeleke P., Ezeigweneme C. Enhancing software development practices with AI insights in high-tech companies. *Computer Science & IT Research Journal* 2024; 5(8): 1897–1919. <https://doi.org/10.51594/csitjr.v5i8.1450>
- Alali R., Wardat Y. Opportunities and challenges of integrating generative artificial intelligence in education. *International Journal of Religion* 2024; 5(7): 784–793. <https://doi.org/10.61707/8y29gv34>
- Al-Qaysi N., Al-Emran M., Al-Sharafi MA., Yaseen ZM., Mahmoud MA., Ahmad A. Generative AI and educational sustainability: Examining the role of knowledge management factors and AI attributes using a deep learning-based hybrid SEM-ANN approach. *Computer Standards & Interfaces* 2025; 93: 103964. <https://doi.org/10.1016/j.csi.2024.103964>
- Alenezi M., Akour M. AI-Driven innovations in software engineering: A Review of Current Practices and Future Directions. *Applied Sciences* 2025; 15(3): 1344. <https://doi.org/10.3390/app15031344>
- Alipour P., Gallegos, E. Leveraging generative AI synthetic and social media data for content generalizability to overcome data constraints in vision deep learning. *Artif Intell Rev* 2025; 58: 145. <https://doi.org/10.1007/s10462-025-11137-6>
- Araujo GF., Hayajneh T. Protecting the Whisper: A security assessment of Amazon CodeWhisperer's generated code. In: Stahlbock, R., Arabnia, H.R. (eds) *Data Science. CSCE 2024. Communications in Computer and Information Science*, vol 2253. Springer, Cham. https://doi.org/10.1007/978-3-031-85856-7_36
- Bahi A., Gharib J., Gahi Y. Integrating generative ai for advancing agile software development and mitigating project management challenges. *International Journal of Advanced Computer Science and Applications* 2024; 15(3). <https://doi.org/10.14569/ijacsa.2024.0150306>
- Bajaj Y., Samal MK. Accelerating software quality: unleashing the power of generative ai for automated test-case generation and bug identification. *International Journal for Research in Applied Science and Engineering Technology* 2023; 11(7): 345–350. <https://doi.org/10.22214/ijraset.2023.54628>
- Banh L., Strobel G. Generative artificial intelligence. *Electronic Markets* 2023; 33: 63. <https://doi.org/10.1007/s12525-023-00680-1>
- Bicen H., Generative AI in aducation: A bibliometric analysis. *International Conference on Virtual Learning*, 2025, 25-33. <https://doi.org/10.58503/icvl-v20y202502>

- Chang CI., Choi WC., Choi IC. A systematic literature review of the opportunities and advantages for AIGC (OpenAI ChatGPT, Copilot, Codex) in programming course. In 2024 the 7th International Conference on Big Data and Education (ICBDE) (ICBDE 2024), September 24–26, 2024, Oxford, United Kingdom. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3704289.3704301>
- Cooper N., Qiao H., Lecomte N., Clark AT., Ellison, AM. Harnessing large language models for coding, teaching and inclusion to empower research in ecology and evolution. *Methods in Ecology and Evolution* 2024; 5(10): 1757–1763. <https://doi.org/10.1111/2041-210x.14325>
- Corso V., Mariani L., Micucci D., Riganelli, O. Assessing ai-based code assistants in method generation tasks. In 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion '24), April 14–20 2024, 380 – 381; Lisbon, Portugal . <https://doi.org/10.1145/3639478.3643122>
- Daun M., Brings J. How ChatGPT will change software engineering education. *ITiCSE 2023: Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education* 2023; 1(1): 110–116, Turku, Finland. <https://doi.org/10.1145/3587102.3588815>
- Deng XN., Joshi KD. Promoting ethical use of generative ai in education. *ACM SIGMIS Database: The DATABASE for Advances in Information Systems* 2024; 55(3): 6–11. <https://doi.org/10.1145/3685235.3685237>
- Finnie-Ansley J., Denny P., Luxton-Reilly A., Santos EA., Prather J., Becker BA .My AI wants to know if this will be on the exam: Testing OpenAI's Codex on CS2 Programming Exercises. In *Australasian Computing Education Conference (ACE '23)*, January 30-February 3, 2023, 97-104, Melbourne, VIC, Australia. ACM, New York, NY, USA. <https://doi.org/10.1145/3576123.3576134>
- Gani MO., Ayyasamy RK., Alhashmi SM., Alam KS., Sangodiah A., Khaleduzzman K., Ponnusamy C. Towards enhanced assessment question classification: a study using machine learning, deep learning, and generative AI. *Connection Science* 2025; 37(1): 2445–2449. <https://doi.org/10.1080/09540091.2024.2445249>
- Godoy W., Valero-Lara P., Teranishi K., Balaprakash P., Vetter J. Evaluation of OpenAI Codex for HPC parallel programming models kernel generation. In *Proceedings of the 52nd International Conference on Parallel Processing Workshops (ICPP Workshops '23)*. Association for Computing Machinery, 2023, 136 – 144, New York, NY, USA. <https://doi.org/10.1145/3605731.3605886>
- Guettala M., Harous S., Kazar O., Bouekkache S. Generative artificial intelligence in education: Advancing Adaptive and Personalized Learning. *Acta Informatica Pragensia* 2024; 13(3): 460–489. <https://doi.org/10.18267/j.aip.235>
- Fu Y., Li Z., Yu J., Shahin M., Liang P., Tahir A., Chen J. Security weaknesses of copilot-generated code in github projects: An empirical study. *ACM Transactions on Software Engineering and Methodology* 2025. <https://doi.org/10.1145/3716848>

- Hartley K., Hayak M., Ko UH. Artificial intelligence supporting independent student learning: an evaluative case study of chatGPT and learning to code. *Education Sciences* 2024; 14(2): 120. <https://doi.org/10.3390/educsci14020120>
- Hnatushenko VV., Pavlenko IV. The use of generative artificial intelligence in software testing. *System Technologies* 2024; 2(151): 113–123. <https://doi.org/10.34185/1562-9945-2-151-2024-10>
- Imran M., Amaliah N., Sage M., Rampen, R., Syam N., Room, F. The feasibility of artificial intelligences (ai) in speaking skill: lecturers' perceptions. *IJOLEH: International Journal of Education and Humanities* 2023; 2(2): 135–144. <https://doi.org/10.56314/ijoleh.v2i2.172>
- Jayavardhini P., Mahalakame RM; Srinivetha P; Eugene Berna I. AI Study Partner: Development of an LLM and GEN AI-enhanced study assistant tool. *Interantional Journal of Scientific Research in Engineering and Management* 2024; 08(03): 1–5. <https://doi.org/10.55041/ijrsrem29626>
- Kashefi A., Mukerji T. ChatGPT for programming numerical methods. *Journal of Machine Learning for Modeling and Computing* 2023; 4(2): 1–74. <https://doi.org/10.1615/jmachlearnmodelcomput.2023048492>
- Kong SC., Yang Y. A human-centered learning and teaching framework using generative artificial intelligence for self-regulated learning development through domain knowledge learning in k–12 settings. *IEEE Transactions on Learning Technologies* 2024; 17: 1562–1573. <https://doi.org/10.1109/tlt.2024.3392830>
- Liang JT., Myers BA., Yang C. A large-scale survey on the usability of ai programming assistants: successes and challenges. In *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE 2024)*, April 14–20 2024, 1–13, Lisbon, Portugal. <https://doi.org/10.1145/3597503.3608128>
- Li Y., Xu J., Zhu Y., Liu H., Liu P. The impact of ChatGPT on software engineering sducation: A quick peek," 2023 10th International Conference on Dependable Systems and Their Applications (DSA), 10-11 August 2023, 595-596, Tokyo, Japan. doi: 10.1109/DSA59317.2023.00087.
- Liu Y. Integrating conversational large language models into student learning: A case study of ChatGPT in software engineering education. In *2024 IEEE Frontiers in Education Conference (FIE)*, 2024, 1-8, Washington, DC, USA. <https://doi.org/10.1109/FIE61694.2024.10893046>
- Macneil S., Sarsa S., Kim J., Hellas A., Denny P., Leinonen J., Tran A., Bernstein S. Experiences from using code explanations generated by large language models in a web software development e-book. *54th ACM Technical Symposium on Computer Science Education (SIGCSE 2023)*, March 15-18 2023, 931–937, Toronto ON Canada. <https://doi.org/10.1145/3545945.3569785>
- Nama P. Integrating ai in testing automation: enhancing test coverage and predictive analysis for improved software quality. *World Journal of Advanced Engineering Technology and Sciences* 2024; 13(1): 769–782. <https://doi.org/10.30574/wjaets.2024.13.1.0486>

- Ng SL., Ho, CC. Generative AI in education: mapping the research landscape through bibliometric analysis. *Information* 2025, 16, 657. <https://doi.org/10.3390/info16080657>
- Ni A., Feng T., Xiong C., Yavuz S., Yin S., Yin P., Liu Y., Joty S., Zhou Y., Zhao Y., Radev D., Shen R., Riddell M., Cohan A. L2CEval: Evaluating language-to-code generation capabilities of large language models. *Transactions of the Association for Computational Linguistics* 2024; 12: 1311–1329. https://doi.org/10.1162/tacl_a_00705
- Nizamudeen F., Gatti L., Bouali N., Ahmed F. Investigating the impact of code generation tools (ChatGPT & Github CoPilot) on programming education. In *Proceedings of the 16th International Conference on Computer Supported Education - Volume 2: CSEDU*; May 2-4, 2024, in Angers, France. DOI: 10.5220/0012628000003693
- Ouh EL., Wlodkowski S., Shim KJ., Gan, BKS. ChatGPT, can you generate solutions for my coding exercises? An evaluation on its effectiveness in an undergraduate java programming course. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V.1 (ITiCSE 2023)*, July 8–12 2023, 54–60, Turku, Finland. <https://doi.org/10.1145/3587102.3588794>
- Ovi MSI., Anjum N., Bithe TH., Rahman MM., Smrity MSA. Benchmarking ChatGPT, Codeium, and GitHub Copilot: A comparative study of AI-driven programming and debugging assistants, 2024 27th International Conference on Computer and Information Technology (ICCIT), 20-22 December 2024, 1546-1551, Cox's Bazar, Bangladesh, 2024. doi: 10.1109/ICCIT64611.2024.11021727.
- Owan VJ., Etta EO., Idika DO., Abang KB., Bassey BA. Exploring the potential of artificial intelligence tools in educational measurement and assessment. *Eurasia Journal of Mathematics, Science and Technology Education* 2023; 19(8): em2307. <https://doi.org/10.29333/ejmste/13428>
- Özpolat Z., Yıldırım Ö., Karabatak M. Artificial intelligence-based tools in software development processes: application of ChatGPT. *European Journal of Technic* 2023; 13(2): 229–240. <https://doi.org/10.36222/ejt.1330631>
- Pearce H., Dolan-Gavitt B., Tan B., Ahmad B., Karri R. Asleep at the keyboard? Assessing the security of github copilot's code contributions. *Communications of the ACM* 2025; 68(2), 96–105. <https://doi.org/10.1145/3610721>
- Petrovska O., Clift L., Pearsall R., Moller F. Incorporating generative ai into software development education. In *Computing Education Practice (CEP '24)*, January 05, 2024, 37–40, Durham, United Kingdom. <https://doi.org/10.1145/3633053.3633057>
- Prenner JA., Babii H., Robbes R. Can OpenAI's codex fix bugs? an evaluation on QuixBugs. In *Proceedings of the Third International Workshop on Automated Program Repair (APR '22)*. Association for Computing Machinery, 2022, 69-75, New York, NY, USA. <https://doi.org/10.1145/3524459.3527351>

- Ross SI., Weisz JD., Houde S., Martinez F., Muller M. The programmer's assistant: Conversational interaction with a large language model for software development. In 28th International Conference on Intelligent User Interfaces (IUI '23), March 27–31 2023, 491–514, Sydney, NSW, Australia. <https://doi.org/10.1145/3581641.3584037>
- Saravanan V., Rambabu C., Seetha A., Ravi S., Kanth T., Kavitha S. Generative ai in software engineering: revolutionizing code generation and debugging. *International Journal of Computational and Experimental Science and Engineering* 2025; 11(2). <https://doi.org/10.22399/ijcesen.1718>
- Savelka J. Unlocking practical applications in legal domain: evaluation of gpt for zero-shot semantic annotation of legal texts. In Nineteenth International Conference on Artificial Intelligence and Law (ICAIL 2023), June 19–23 2023, 447 – 451, Braga, Portugal. <https://doi.org/10.1145/3594536.3595161>
- Sengar SS., Hasan AB., Kumar S., Carroll F. Generative artificial intelligence: a systematic review and applications. *Multimedia Tools and Applications* 2024; 84: 23661–23700. <https://doi.org/10.1007/s11042-024-20016-1>
- Schotten M., El Aisati, M., Meester WJN., Ross CA., Steiging S. A brief history of scopus: the world's largest abstract and citation database of scientific literature. 1st Edition. Auerbach Publications. 2017. <https://doi.org/10.1201/9781315155890-3>
- Sobania D., Rothlauf F., Briesch M. Choose your programming copilot. GECCO '22: Genetic and Evolutionary Computation Conference , July 9 - 13 2022, 1019–1027, Boston Massachusetts. <https://doi.org/10.1145/3512290.3528700>
- Soni S., Kapoor P. Exploring deep learning and generative AI techniques in automatic text summarization," 2025 12th International Conference on Computing for Sustainable Global Development (INDIACom), 2025, 1-7, Delhi, India, <https://doi.org/10.23919/INDIACom66777.2025.11115475>.
- Sultana S. AI in higher education: ai ethics, quality assurance, and academic integrity. In M. Rahman, G. Koolmon, S. Shaikh, & A. Munna (Eds.), *Optimizing Leadership and Governance in Higher Education* 2025; 111-136. IGI Global Scientific Publishing. <https://doi.org/10.4018/979-8-3693-3534-5.ch006>
- Tariq MU. Harnessing generative AI for enhanced web application security. In I. Shah & N. Jhanjhi (Eds.), *Generative AI for Web Engineering Models* 2025; 161-192. IGI Global Scientific Publishing. <https://doi.org/10.4018/979-8-3693-3703-5.ch008>
- Van Eck NJ., Waltman L. Software survey: VOSviewer, a computer program for bibliometric mapping. *Scientometrics* 2009; 84(2): 523–538. <https://doi.org/10.1007/s11192-009-0146-3>
- Vhatkar A. Pawar V. Chavan P. Generative AI in education: A bibliometric and thematic analysis. 2024 8th International Conference on Computing, Communication, Control and Automation

- (ICCUBEA), Aug 23-24 2024, 1-6 ,Pune, India.
<https://doi.org/10.1109/ICCUBEA61740.2024.10774819>
- Vemuri N., Thaneeru N., Tatikonda V. Adaptive generative ai for dynamic cybersecurity threat detection in enterprises. *International Journal of Science and Research Archive* 2024; 11(1): 2259–2265.
<https://doi.org/10.30574/ijsra.2024.11.1.0313>
- Vergara D., Del Bosque A., Lampropoulos G., Fernández-Arias P. Trends and applications of artificial intelligence in project management. *Electronics* 2025; 14(4): 800.
<https://doi.org/10.3390/electronics14040800>
- Wang X. Generative AI based on deep knowledge tracing for academic Insight and Intelligent learning strategy optimization. In 2025 International Conference on Big Data and Informatization Education (ICBDIE 2025), February 21–23, 2025, Suzhou, China. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3729605.3729638>
- Yang TC., Hsu YC., Wu, JY. The effectiveness of ChatGPT in assisting high school students in programming learning: evidence from a quasi-experimental research. *Interactive Learning Environments* 2025; 1–18. <https://doi.org/10.1080/10494820.2025.2450659>
- Yusuf A., Pervin N., Román-González M. Generative AI and the future of higher education: a threat to academic integrity or reformation? Evidence from multicultural perspectives. *International Journal of Educational Technology in Higher Education* 2024; 21: 21.
<https://doi.org/10.1186/s41239-024-00453-6>