



Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü Dergisi

<https://dergipark.org.tr/tr/pub/yyufbed>



Araştırma Makalesi

ChatGPT'nin Programlama Kodu Hata Ayıklama Performansının PyCharm Geleneksel Hata Ayıklama Aracı Performansı ile Karşılaştırılması

Fırat KAPAR^{*1}, Halit Eray ÇELİK², Serbest ZİYANAK³

¹Van Yüzüncü Yıl Üniversitesi, Özalp Meslek Yüksekokulu, Bilgisayar Programcılığı Programı, 65800, Van, Türkiye

²Van Yüzüncü Yıl Üniversitesi, İktisadi İdari Bilimler Fakültesi, Ekonometri Bölümü, 65080, Van, Türkiye

³Van Yüzüncü Yıl Üniversitesi, Van Meslek Yüksekokulu, Bilgisayar Programcılığı Programı, 65080, Van, Türkiye

Fırat KAPAR, ORCID No: 0000-0001-9559-710X, Halit Eray ÇELİK, ORCID No: 0000-0001-7490-8124, Serbest ZİYANAK, ORCID No: 0000-0002-9575-8791

*Sorumlu yazar e-posta: firatkapar@yyu.edu.tr

Makale Bilgileri

Geliş: 11.07.2024
Kabul: 14.11.2024
Online Aralık 2024

DOI: [10.53433/yyufbed.1514642](https://doi.org/10.53433/yyufbed.1514642)

Anahtar Kelimeler

ChatGPT,
Hata ayıklama,
PyCharm,
Python

Öz: Bu çalışmanın amacı ChatGPT'nin programlama kodu hata ayıklama yeteneklerini geleneksel hata ayıklama aracı ile karşılaştırarak yeni nesil yapay zeka tabanlı hata ayıklama yaklaşımlarının potansiyelini incelemektir. Bu amaçla, Python programlama dilinde sıkça rastlanan söz dizimi, mantık ve tür uyumsuzluğu hatalarını içeren programlama kodları türetilmiş ve PyCharm ve OpenAI tarafından geliştirilen ChatGPT-4 ile bu programlama kodları analiz edilerek her iki aracın hata ayıklama performansı karşılaştırılmıştır. Çalışma sonucunda ChatGPT-4'ün söz dizimi hatalarında PyCharm ile benzer başarı gösterirken mantık ve tür uyumsuzluğu hatalarını belirlemede PyCharm'a göre çok daha başarılı olduğu saptanmıştır. Bu sonuç doğrultusunda, yapay zekâ destekli araçların, programlama hatalarını etkili bir şekilde tespit etme ve düzeltme konusunda geleneksel yöntemlere kıyasla avantaj sağlayabileceği görülmektedir.

Comparison of ChatGPT's Programming Code Debugging Performance with PyCharm Traditional Debugging Tool Performance

Article Info

Received: 11.07.2024
Accepted: 14.11.2024
Online December 2024

DOI: [10.53433/yyufbed.1514642](https://doi.org/10.53433/yyufbed.1514642)

Keywords

ChatGPT,
Debugging,
PyCharm,
Python

Abstract: This study aims to examine the potential of next-generation AI-based debugging approaches by comparing the programming code debugging capabilities of ChatGPT with a traditional debugging tool. For this purpose, programming codes containing common syntax, logic, and type mismatch errors in Python were generated and analyzed using PyCharm and OpenAI's ChatGPT-4. The study found that while ChatGPT-4 showed similar success to PyCharm in identifying syntax errors, it was significantly more successful than PyCharm in detecting logic and type mismatch errors. These results suggest that AI-supported tools could provide an advantage over traditional methods in effectively detecting and correcting programming errors.

1. Giriş

Bilgisayar kodlarında var olabilen hatalar yazılım geliştirmede zorlayıcı olmakta ve bulunup düzeltilmeleri zaman ve çaba gerektirmektedir. Bilgisayar kodundaki hatalar, küçük yazılım çökmelerinden güvenlik açıklarına ve veri kaybına kadar uzanan önemli sorunlara neden olabilmektedir. Kodda hata ayıklama, yani bu hataları bulma ve düzeltme süreci, yazılım geliştirmenin kritik bir yönü

olmakla birlikte zaman alıcı ve karmaşık bir görev olabilmektedir (Wong ve ark., 2017). Bu amaçla, geleneksel hata ayıklama araçları, Entegre Geliştirme Ortam (Integrated Development Environment – IDE)'leri, hata ayıklayıcılar ve kod analizörleri uzun süredir geliştiriciler tarafından hataları tanımlama ve çözme konusunda kullanılmaktadır. Bu araçlar, hata ayıklama sürecini kolaylaştırmak için kesme noktaları, adım adım yürütme, değişken inceleme ve hata tespiti gibi özellikler sunmaktadır. Syntax, mantıksal ve anlamsal hataları tanımlayıp düzeltebilme yeteneği, geliştiricilere kodlarını daha iyi anlama ve hataları etkili bir şekilde çözme konusunda yardımcı olabilmektedir. Modelin ya da kodun analiz edilmesi, potansiyel sorunların belirlenmesi, olası çözümlerin sıralanması ve düzeltmelerin önerilmesi süreci, geliştiricilere hem zaman kazandırmakta hem de kod kalitesini artırabilmektedir (Haque & Li, 2023). Ancak, yazılım mühendislerinin hataları en aza indirme ve kod kalitesini iyileştirme çabalarına rağmen hatalar, yazılım geliştirme sürecinin hala kaçınılmaz bir parçasıdır (Wong ve ark., 2017). Geçmişten günümüze gelen ve yaygın kullanılan yöntemlerin yanında OpenAI tarafından geliştirilen, doğal dil işleme ve anlama yetenekleri ile programlama dillerindeki kod hatalarını tespit etme ve çözüm önerileri sunma kapasitesine sahip bir derin öğrenme modeli olan Chat Generative Pretrained Transformer (ChatGPT) de farklı bir yaklaşım sunmaktadır (Surameery & Shakor, 2023). OpenAI tarafından geliştirilen ve bir dil modeli olan ChatGPT, akademide bir dönüm noktası olarak kabul edilmektedir (Quintans-Júnior et al., 2023). Çeşitli türlerde yüksek kaliteli, özgün içerik üretme yeteneği ile istatistiksel analiz ve kodlamadaki potansiyeli, onu değerli bir araç haline getirmektedir. Ancak, empati eksikliği ve olası önyargılar gibi sınırlamaları göz önünde bulundurulmalıdır (Biswas, 2023a). Ticari ve akademik alanlarda, ChatGPT'nin gerçek zamanlı metin üretim yetenekleri büyük ilgi görmüştür (Gupta et al., 2023). Özellikle sosyal medyada müşteri hizmetleri otomasyonu ve içerik üretimi gibi uygulamaları dikkat çekicidir. Ayrıca, makine öğrenimi ve derin öğrenme algoritmalarının kullanımı, insan-bilgisayar etkileşimlerini devrim niteliğinde değiştirmiştir (Hadi et al., 2023). Rahman & Watanobe (2023), ChatGPT'nin programlama ve hata ayıklama yeteneklerinin Codex ve CoCoNut gibi ileri modellerle rekabet edebilecek düzeyde olduğunu göstermişlerdir. Ayrıca, ChatGPT'nin eğitimde, özellikle programlama öğreniminde hem eğiticiler hem de öğrenciler için yardımcı olacağı da aşikardır (Rahman & Watanobe, 2023). ChatGPT, algoritmik ve kütüphane ile ilgili görevleri tamamlama konusunda yazılım uzmanlarının karşılaştıkları hataların çözümlerinin yer aldığı ve bir soru cevap web sitesi olan Stack Overflow'u önemli ölçüde geride bırakmaktadır, ancak hata ayıklama görevlerinde Stack Overflow daha iyidir. Algoritmik zorluklarda ise ChatGPT grubu, Stack Overflow grubundan daha hızlıdır (Liu ve ark., 2023).

ChatGPT'nin yazılım optimizasyonu ve algoritma programlamada da kullanılabilirliği üzerine yapılan çalışmalar, modelin 3D baskıda G-Code optimizasyonu ve çeşitli yapay zekâ görevleri için Büyük Dil Modelleri (Large Language Models - LLM)'lerin kontrolör olarak kullanılması gibi yenilikçi yaklaşımlar sunabileceğini göstermiştir. Bu çalışmalar, ChatGPT'nin çok yönlülüğünü ve farklı yazılım geliştirme zorluklarını çözme potansiyelini vurgulamaktadır (Jaber ve ark., 2023). ChatGPT, bilgisayar programlama için kod tamamlama, düzeltme, tahmin, hata düzeltme, optimizasyon, belge oluşturma, sohbet robotu geliştirme, metinden koda dönüştürme dahil olmak üzere çok çeşitli yetenekler sunmaktadır. Açıklamalar, örnekler ve rehberlik sağlama yeteneği ile ChatGPT, teknik destek için değerli bir kaynaktır ve programlamayla ilgili yetenekleri, geliştiriciler ve kuruluşlar için verimliliği ve doğruluğu artırmaktadır (Biswas, 2023b). Surameery & Shakor (2023), Rahman & Watanobe (2023), Jaber ve ark. (2023), vb. çalışmalarında elde edilen bulgulara göre ChatGPT'nin hata ayıklama süreçlerinde önemli bir rol oynaması beklenmektedir. Ouh ve ark. (2023), ChatGPT, yüksek okunabilirlik ve iyi yapılandırılmış organizasyon ile doğru Java programlama çözümleri üretebildiğini ve bunun yanında öğrencilerin programlama zorluklarını aşmalarına ve alternatif problem çözme yaklaşımlarını keşfetmelerine yardımcı olabilecek değerli bir araç olma potansiyeline sahip olduğunu bildirmişlerdir. Sobania ve ark. (2023), ChatGPT'nin hata düzeltme performansı, CoCoNut ve Codex gibi derin öğrenme yaklaşımlarıyla rekabet etmekte olup, standart program onarım yaklaşımlarını önemli ölçüde geride bıraktığını bildirmişlerdir. Dan'azumi ve ark. (2023), ChatGPT'nin programlama hatalarını tanımlama, tahmin etme, açıklama ve çözmedeki etkinliğini araştırdıkları çalışmalarının bulguları ChatGPT'nin kodu analiz edebildiğini ve anlayabildiğini ortaya koymuştur. Ayrıca, çalışmalarının sonuçları, ChatGPT'nin hata ayıklama sürecini daha verimli hale getirme potansiyeline sahip olduğunu ve farklı deneyim seviyelerine sahip geliştiriciler için daha uygun hale getirdiğini bildirmişlerdir. Literatür incelendiğinde, geleneksel hata ayıklama araçları ile ChatGPT'nin hata ayıklama performanslarının karşılaştırıldığı çalışmaların çok sınırlı sayıda olduğu görülmektedir. Bu

nedenle, ilgili literatüre katkıda bulunmak amacıyla, “söz dizimi hataları”, “mantık hataları” ve “tür uyumsuzluğu hataları” başlıkları altında yer alan kodlama hatalarının tespitinde Python programlama dili için geliştirilmiş ve geleneksel bir hata ayıklama aracı olan PyCharm ve ChatGPT-4’ün performansları karşılaştırılmıştır. Bu çalışma sonucunda ChatGPT-4’ün programlama kodu hata ayıklama performansının farklı başlıklar altında klasik yöntemlerden olumlu anlamda farklılaştığı saptanmıştır. Dolayısıyla, bu çalışmada elde edilen bulguların ilgili literatüre katkıda bulunacağı ve gelecek çalışmaların yönlendirilmesinde fayda sağlayacağı düşünülmektedir. Bu çalışmanın kalan kısmı üç ana başlıktan oluşmaktadır. Materyal ve Yöntem bölümünde bu çalışmada kullanılan veri seti tanıtılmış ve veri setinin analizinde kullanılan yöntem kısaca belirtilmiştir. Daha sonra Uygulama ve Bulgular bölümünde veri setinin analizi gerçekleştirilmiş ve analiz bulguları sunulmuştur. Sonuç ve Tartışma bölümünde ise bu çalışmada elde edilen bulgular yorumlanarak çalışma sonlandırılmıştır.

2. Materyal ve Yöntem

Bilgisayar kodundaki hatalar, küçük yazılım çökmelerinden güvenlik açıklarına ve veri kaybına kadar uzanan önemli sorunlara neden olabilmektedir. Kodda hata ayıklama, yani bu hataları bulma ve düzeltme süreci, yazılım geliştirmenin kritik bir yönüdür ve zaman alıcı ve karmaşık bir görev olabilir. Bu çalışmada, programlama kodlarında yer alan hataların tespiti için Python programlama dili ile uygulama geliştirme, hata ayıklama ve hataların yönetimini de sağlayabilen PyCharm 2023.3.4 (Professional Edition) sürümü ve OpenAI tarafından geliştirilen, doğal dil işleme ve anlama yetenekleri ile programlama dillerindeki kod hatalarını tespit etme ve çözüm önerileri sunma kapasitesine sahip bir derin öğrenme modeli olan ChatGPT-4 sürümü kullanılmıştır. Python programlama diline ait resmi dokümantasyon incelendiğinde, bu programlama dilinde oluşabilecek hata türlerinin temelde 3 kategoride toplanabileceği görülmektedir. Bu nedenle, bu çalışma kapsamında, Python programlama dilinde basit söz dizimi hatalarından karmaşık mantık hatalarına kadar Çizelge 1’de gösterildiği gibi 3 farklı kategoride çeşitli hataları içeren kod örneklerinin hazırlanması ve söz konusu hataların ayıklanması amaçlanmıştır.

Çizelge 1. Gerekli durumlarda tek sütun olarak eklenecek çizelge formatı ve alt satırdan devam eden

Hata Türü	Açıklama
Söz dizimi hataları	Programın çalıştırılmasını engelleyen, dilin kurallarına uyulmamasından kaynaklanan hatalar.
Mantık hataları	Programın yanlış sonuçlar üretmesine veya beklenmeyen davranışlarda bulunmasına neden olan hatalar.
Tür uyumsuzluğu hataları	Farklı veri türlerinin birbiriyle uyumsuz şekilde kullanılmasından kaynaklanan hatalar.

Veri seti olarak kullanılan Python kodlarının her biri gerçek bir problemin çözümünü sağlayan kodlardır. Bu kodlar herkese açık olarak yayınlanan çeşitli web sayfalarından toplandıktan sonra bu çalışmada kullanılabilmesi için bir Python yazılım uzmanı olan makalenin yazarları tarafından bilinçli bir şekilde hatalar eklenmiştir. Bu hataların sınıflandırılması makalenin yazarları tarafından yapılmıştır. Literatüre bakıldığında bu tür çalışmalar için kullanılan veri setlerinde 100 ile 160 arasında kod parçasının kullanıldığı görülmektedir; detaylı bilgi için [Surameery & Shakor \(2023\)](#) çalışması ve bu çalışmada verilen ilgili kaynaklar incelenebilir. Bu amaçla internetten toplanan farklı zorluk ve karmaşık seviyesine sahip 150 adet Python kodu, "söz dizimi hatası", "mantık hatası" ve "tür uyumsuzluğu hatası" olmak üzere üç kategoriye ayrılarak uygun hatalar ilgili kodlara eklenmiş ve uygulama için hazır hale getirilmiştir. Hatalı kodlarından oluşan bu veri seti PyBuggy olarak isimlendirilmiş ve <https://github.com/byfirat/PyBuggy/tree/master> adresinde herkesin kullanımına açık olarak paylaşılmıştır. İlgili kodların analizi sonrasında "0-İlgili hata saptanamadı" ve "1-İlgili hata saptandı" değerlerinden oluşan veri setleri "söz dizimi hatası", "mantık hatası" ve "tür uyumsuzluğu hatası" grupları için oluşturulmuş ve bu veri setleri istatistiksel analize tabi tutulmuştur. İlgili değişkenlerin ikili ölçüm düzeyi (0 ve 1) ile sınıflayıcı ölçeğe sahip olması nedeniyle istatistiksel analizde Cochran’ın Q testi kullanılmıştır. Burada, PyCharm ve ChatGPT-4 gibi iki denetleyicinin olması nedeniyle Cochran’ın Q testi ile McNemar testinin denk sonuçlar vereceği de aşikardır. Bu iki teste ilişkin ayrıntılı bilgiler ve

iki denetleyici olması durumunda bu iki testin denk olduklarının ispatı [Conover \(1999\)](#)'da verilmektedir. Cochran'ın Q testine ilişkin test istatistiği

$$T = c(c - 1) \frac{\sum_{j=1}^c (C_j - \frac{N}{c})^2}{\sum_{i=1}^r R_i (c - R_i)} \quad (1)$$

şeklinde tanımlanmaktadır. Burada c denetleyici sayısını, r satır sayısını (örneklemdeki birim sayısı), N satır ve sütun genel toplamını, C_j j . sütun toplamını ve R_i ise i . satır toplamını göstermektedir. Ayrıca T istatistiği $(c - 1)$ serbestlik derecesine sahip ki-kare (chi-square - $X_{(c-1)}^2$) dağılımına sahiptir. Bu çalışmada, iki denetleyici olduğu dikkate alındığında hesaplanan T istatistiğine ilişkin 0.05 anlamlılık düzeyindeki tablo değeri 3,84 olmaktadır. Başka bir ifade ile $P(T \leq 3.84) = 0.05$ yani $X_{1,0.05}^2 = 3.84$ 'tür. Genel olarak, Cochran'ın Q testine ilişkin yokluk hipotezi " H_0 : Denetleyiciler arasında anlamlı bir farklılık olduğuna yönelik ilgili anlamlılık düzeyinde yeterli istatistiksel kanı yoktur." ve karşı/alternatif hipotez ise " H_1 : Denetleyiciler arasında anlamlı bir farklılık olduğuna yönelik ilgili anlamlılık düzeyinde yeterli istatistiksel kanıt mevcuttur." şeklinde ifade edilmektedir.

3. Uygulama ve Bulgular

Bu çalışmada ChatGPT-4 ve PyCharm araçlarının hata ayıklama performanslarının karşılaştırılması amaçlanmıştır. Bu amaçla Materyal ve Yöntem 'de belirtildiği gibi hataları içeren kodlar PyCharm ile çalıştırılmış ve hatanın tespit edilip edilmediği tek tek not edilmiştir ("0-İlgili hata saptanamadı" ve "1-İlgili hata saptandı"). ChatGPT-4'e ise, "Verilen Python kod parçasında söz dizimi hatası, mantık hatası veya tür uyumsuzluğu hatalarından en az biri var ise, hatanın nedenini göster" şeklinde bir Prompt (İstem) vasıtasıyla test edilmiş ve hatalı kodların kaç tanesini hatalı olarak işaretlediği kaç tanesinde ise hata bulunmadığı not alınmıştır ("0-İlgili hata saptanamadı" ve "1-İlgili hata saptandı"). Bu şekilde veri setini oluşturan bütün kodlar tek tek analiz edilmiş ve ilgili çıktılar alınarak "söz dizimi hatası", "mantık hatası" ve "tür uyumsuzluğu hatası" grupları için veri setleri oluşturulmuştur. Analiz sonucunda elde edilen bulgulara ilişkin frekanslar Çizelge 2' de verilmiştir. PyCharm ve ChatGPT-4 kullanılarak hazırlanan hata senaryolarının ayıklanma sürecinin yürütülmesi, her bir aracın hata tespit sayılarının karşılaştırılması şeklinde gerçekleştirilmiştir. Elde edilen verilerin karşılaştırmalı analizi yapılarak, söz konusu araçların Python programlama kodu hata ayıklama performansları değerlendirilmiştir.

Çizelge 2. ChatGPT-4 ve PyCharm araçlarının hata ayıklama performans sonuçları

Hata Türü	ChatGPT-4	PyCharm
Söz dizimi hataları	49/50	50/50
Mantık hataları	45/50	06/50
Tür uyumsuzluğu hataları	50/50	21/50

Çizelge 2 incelendiğinde "söz dizimi hataları" grubunda ChatGPT-4 ve PyCharm'ın benzer frekanslara sahip olduğu fakat "mantık hataları" ve "tür uyumsuzluğu hataları" gruplarında farklı frekans değerlerine sahip olduğu görülmektedir. Bu farklılığın istatistiksel olarak ortaya konulması ve 0.95 güvenirlilikle genel yorumlanabilmesi için Cochran 'ın Q testi yapılmış ve test sonucunda, "söz dizimi hataları", "mantık hataları" ve "tür uyumsuzluğu hataları" gruplarına ait test istatistiği değerleri sırasıyla 1.00, 35.37 ve 29.00 olarak hesaplanmıştır. İlgili test istatistiği değerleri ve $X_{1,0.05}^2 = 3.84$ tablo değeri dikkate alındığında "söz dizimi hataları" grubu açısından ChatGPT-4 ve PyCharm arasında 0.05 anlam düzeyinde istatistiksel olarak anlamlı bir farklılığın olmadığı fakat "mantık hataları" ve "tür uyumsuzluğu hataları" grupları açısından istatistiksel olarak anlamlı bir farklılığın olduğu görülmektedir.

Bu sonuçlar, ChatGPT-4'ün özellikle mantık hataları ve tür uyumsuzluğu hatalarını tespit etmede PyCharm'a göre çok daha üstün bir performans sergilediğini göstermektedir. Söz dizimi hataları açısından her iki araç da yüksek başarı oranlarına ulaşmıştır, ancak ChatGPT-4'ün mantık ve tür uyumsuzluğu hatalarında gösterdiği yüksek performans, onun programlama dillerindeki hata ayıklama

süreçlerinde etkili bir araç olabileceğini işaret etmektedir. Bu bulgular, ChatGPT-4'ün, programlama kodu hata ayıklama konusunda geleneksel hata ayıklama aracı olan PyCharm'a kıyasla önemli bir alternatif sunabileceğini ve geliştiricilere kodlarını daha iyi anlama ve hataları etkili bir şekilde çözme konusunda yardımcı olabileceğini göstermektedir.

4. Tartışma ve Sonuç

Çalışmamız, ChatGPT-4'ün özellikle mantık ve tür uyumsuzluğu hatalarını tespit etme konusunda PyCharm'a göre üstün bir performans sergilediğini göstermiştir. Bu sonuçlar, doğal dil işleme teknolojilerinin ve özellikle ChatGPT'nin, geleneksel hata ayıklama araçları ile rekabet edebilecek düzeyde olduğunu ve hatta bazı alanlarda bu araçlardan daha etkili olabileceğini ortaya koymaktadır. ChatGPT-4'ün yüksek başarımı, dil modellerinin, programlama dilleri ve hatalarının anlaşılması konusunda gelişmiş yeteneklere sahip olduğunu göstermektedir. Bu durumun, özellikle yazılım geliştirme sürecinde hata tespiti ve çözümü konularında, geliştiricilere zaman kazandırabileceği ve kod kalitesini artırabileceği düşünülmektedir. Öte yandan, PyCharm gibi geleneksel hata ayıklama araçlarının söz dizimi hatalarını tespit etme konusunda çok etkili olduğu, ancak mantık ve tür uyumsuzluğu gibi daha karmaşık hata türlerinde zayıf kaldığı görülmektedir. Bu bulgu, geleneksel araçların bu tür hataları tespit etmek için daha fazla geliştirilmesi gerektiğini göstermektedir.

Çalışmanın bulguları, yazılım geliştirme ve hata ayıklama süreçlerinde, ChatGPT gibi yapay zekâ tabanlı araçların potansiyel kullanımının önemini vurgulamaktadır. Bu tür araçların, geleneksel hata ayıklama yöntemlerine alternatif veya tamamlayıcı olarak kullanılmasının yazılım geliştirme sürecinin verimliliğini ve etkinliğini artırabileceği düşünülmektedir. Ancak, bu teknolojilerin etkin bir şekilde kullanılabilmesi için, onların yetenekleri, limitleri ve kullanım senaryoları üzerine daha fazla araştırma yapılması önem taşımaktadır. Bu çalışma kısıtlı bir veri seti üzerinde gerçekleştirilmiştir. Bu veri seti, çeşitli hata türlerini kapsayıp geniş bir yelpazeyi temsil etse de ileriki çalışmalarda daha farklı programlama dillerinde benzer bir çalışma yapılmasının faydalı olacağı düşünülmektedir. Ayrıca, farklı türde ve karmaşıklıkta hataları içeren daha geniş veri setleri üzerinde çalışmak, sonuçların kapsayıcılığını da artıracaktır. ChatGPT ve PyCharm gibi araçlar sürekli gelişmektedir. Çalışmanın yapıldığı dönemdeki sürümler üzerinden elde edilen sonuçlar, bu araçların gelecek sürümleri için mutlak geçerlilik taşımamaktadır. Özellikle yapay zekâ alanındaki hızlı ilerlemeler, bu araçların yeteneklerini kısa sürede değiştirebilmektedir.

Kaynakça

- Biswas, S. (2023a). Role of chatgpt in computer programming. *Mesopotamian Journal of Computer Science*, 2023, 8-16. <https://doi.org/10.58496/MJCSC/2023/002>
- Biswas, S. (2023b). The function of chat gpt in social media: According to chat gpt. SSSN. <http://dx.doi.org/10.2139/ssrn.4405389>
- Conover, W. J. (1999). *Practical nonparametric statistics* (pp. 196-201). John Wiley & Sons.
- Dan'azumi, H., Mohammed, Y. B., & Badara, M. S. (2024). Exploring the use of ChatGPT for resolving programming bugs. *International Journal of Advanced Multidisciplinary Research*, 11(3), 34-44.
- Gupta, B., Mufti, T., Sohail, S. S., & Madsen, D. Ø. (2023). Chatgpt: A brief narrative review. *Cogent Business & Management*, 10(3), 2275851. <https://doi.org/10.1080/23311975.2023.2275851>
- Hadi, M. A., Abdulredha, M. N., & Hasan, E. (2023). Introduction to chatgpt: A new revolution of artificial intelligence with machine learning algorithms and cybersecurity. *Science Archives*, 4, 276-285. <https://doi.org/10.47587/SA.2023.4406>
- Haque, M. A., & Li, S. (2023). The potential use of chatgpt for debugging and bug fixing. *EAI Endorsed Transactions on AI and Robotics*, 2(1), e4. <https://doi.org/10.4108/airo.v2i1.3276>
- Jaber, M. A., Beganovic, A., & Abd Almisreb, A. (2023). Methods and applications of chatgpt in software development: A literature review. *Southeast Europe Journal of Soft Computing*, 12(1), 08-12.
- Liu, J., Tang, X., Li, L., Chen, P., & Liu, Y. (2023, October). *Chatgpt vs. stack overflow: An exploratory comparison of programming assistance tools*. In 2023 IEEE 23rd International Conference on

- Software Quality, Reliability, and Security Companion (QRS-C) (pp. 364-373). <https://doi.org/10.1109/QRS-C60940.2023.00105>
- Ouh, E. L., Gan, B. K. S., Jin Shim, K., & Wlodkowski, S. (2023, June). *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 (pp. 54-60). <https://doi.org/10.1145/3587102.3588794>
- Quintans-Júnior, L. J., Gurgel, R. Q., Araújo, A. A. D. S., Correia, D., & Martins-Filho, P. R. (2023). Chatgpt: the new panacea of the academic world. *Revista da Sociedade Brasileira de Medicina Tropical*, 56, e0060-2023. <https://doi.org/10.1590/0037-8682-0060-2023>
- Rahman, M. M., & Watanobe, Y. (2023). Chatgpt for education and research: opportunities, threats, and strategies. *Applied Sciences*, 13(9), 5783. <https://doi.org/10.3390/app13095783>
- Sobania, D., Briesch, M., Hanna, C., & Petke, J. (2023, May). *An analysis of the automatic bug fixing performance of chatgpt*. In 2023 IEEE/ACM International Workshop on Automated Program Repair (APR) (pp. 23-30). <https://doi.org/10.1109/APR59189.2023.00012>
- Surameery, N. M. S., & Shakor, M. Y. (2023). Use chat gpt to solve programming bugs. *International Journal of Information Technology & Computer Engineering (IJITC)*, 3(01), 17-22. <https://doi.org/10.55529/ijitc.31.17.22>
- Wong, W. E., Li, X., & Laplante, P. A. (2017). Be more familiar with our enemies and pave the way forward: a review of the roles bugs played in software failures. *Journal of Systems and Software*, 133, 68-94. <https://doi.org/10.1016/j.jss.2017.06.069>