

| | | |
|--------------------|---|-----------------------------------|
| Üniversite | : | T.C. İstanbul Kültür Üniversitesi |
| Enstitü | : | Lisansüstü Eğitim Enstitüsü |
| Anabilim Dalı | : | Bilgisayar Mühendisliği |
| Program | : | Bilgisayar Mühendisliği |
| Tez Danışmanı | : | Doç. Dr. Akhan AKBULUT |
| Tez Türü ve Tarihi | : | Yüksek Lisans – Ocak 2023 |

ÖZET

PLATFORMLAR ARASI KOD KLON TESPİTİ

Günümüzün taleplerine ayak uydurmak için yazılım mimarisi sürekli olarak geliştirilmektedir. En modern mimari yöntemlerden biri olmasına rağmen, mikroservis mimarisini uygulamanın zorlukları vardır. Mikroservis mimarisinin çok dilli yapısı organizasyon kolaylığı sağlarken, kod klonlarının algılanmasını da zorlaştırır. Kod klonları, bir yazılım sistemindeki, sistemi korumak için gereken bakım zamanını ve ihtiyaç duyduğu kaynağı artıran, yinelenen kod parçalarını ifade eder. Yazılım geliştirme için kod klonlarının hem olumlu hem de olumsuz etkileri olabilir. Bu nedenle, herhangi bir kod gözden geçirme ve yeniden düzenleme öncesi klonların tanımlanması gerekir. Çok dilli mikroservislerde, platformlar arasında kod klonlarını tespit etmek, onları aynı platformda bulmaktan daha zorlu bir iştir. Bu araştırma, JavaScript, C, C# ve Java ile geliştirilen yazılım bileşenlerinin, birbirinin kod klonları olma derecesini analiz etmek için bir metodoloji sunmaktadır. Spesifik olarak, kodların ne kadar benzer olduğunu belirlemek için Kosinüs Benzerliği, Öklid Uzaklığı ve Manhattan Uzaklığı ölçümlerini kullandık. Deney yoluyla, <https://atcoder.jp> sitesinden kod kopyalarından oluşan bir veri kümesi topladık ve çeşitli yaklaşımlar için deneysel eşikler belirledik. Deneylerimiz, platformlar arası kod klonlarını tespit etmede en etkili yaklaşımın Manhattan Uzaklığı olduğunu (%91,59), bunu Öklid Uzaklığı (%91,08) ve son olarak da Kosinüs Benzerliği'nin (%72,83) izlediğini

gösteriyor. Aynı platformda kod klonlarını tespit etmedeki başarı oranı ise Kosinüs Benzerliği %94,73, Öklid Uzaklığı %91,77 ve Manhattan Uzaklığı %91,15'tir. Bu tekniğin, yazılım geliştirme ekiplerinin çeşitli teknolojiler kullandığı ortamlarda ortaya çıkabilecek platformlar arası benzer kod örneklerini belirlemek için kullanılacağı tespit edilmiştir.

Anahtar Sözcükler mikroservis, mikroservis mimarisi, kod klon tespiti, çok dilli programlama

| | | |
|-------------------------|---|---------------------------------|
| University | : | T.C. İstanbul Kültür University |
| Institute | : | Institute of Graduate Studies |
| Department | : | Computer Engineering |
| Program | : | Computer Engineering |
| Thesis Advisor | : | Assoc. Prof. Akhan AKBULUT |
| Degree Awarded And Date | : | MA –Jan 2023 |

ABSTRACT

To keep up with the demands of today, software architecture is continuously being refined. Although it is the most modern architectural method, adopting a microservice architecture is not without its difficulties. While the polyglot structure of the microservice Architecture provides organizational ease, it also makes detecting code clones more difficult. Code clones refer to duplicate fragments of code in a software system, which increases the time and effort needed to maintain the system and the amount of resources it needs. For software development, code clones can have both positive and negative effects. Therefore, clones need to be identified prior to any sort of refactoring or elimination. In multilingual microservices, recognizing code clones across platforms is a more challenging task than finding them on the same platform. This research presents a methodology for analyzing the degree to which software components developed with JavaScript, C, C#, and Java are code clones of one another. Specifically, we utilized the Cosine Similarity, Euclidean Distance, and Manhattan Distance measures to determine how similar codes were. Through experimentation, we have amassed a dataset consisting of code copies from the site <https://atcoder.jp> and identified empirical thresholds for various approaches. Our research indicates that Manhattan Distance is the most effective approach for detecting cross-platform code clones (91.59%), followed by Euclidean Distance (91.08%), and finally Cosine Similarity (72.83%). The success rate in detecting code clones on the same platform is Cosine Similarity 94.73%, Euclidean Distance 91.77%, and Manhattan Distance 91.15%. It's been established that this technique can be used to

identify instances of code duplication across platforms, which might arise in settings where development teams employ diverse technology stacks.

Keywords: microservice, microservice architecture, code clone detection, polyglot, multilingual programming