

Mobil Oyun Yapım Sürecinde Sınıflandırma Algoritması ile Entegre Edilmesi

Yazarlar: Muhammet Burak Okumuş¹, Ömer Er²

Danışman Hoca: Gülay Cicek³

^{1,2,3}Yazılım Mühendisliği Bölümü, Mühendislik Mimarlık Fakültesi

İstanbul Beykent Üniversitesi, Sarıyer, İstanbul, Türkiye

¹2203013109@student.beykent.edu.tr, ²2203013035@student.beykent.edu.tr, ³gulaycicek@beykent.edu.tr

Özet—Problemin tanımı ve önemi olarak kullanıcıların rastgele veya anlamsız harf kombinasyonlarından gerçek kelimeler oluşturabilecekleri ve kelime tahminlerinde bulunabilecekleri bir mobil uygulama geliştirilmiştir. Günlük bir sorun olan bu uygulamanın amacı, kullanıcıların öğrenme sürecinin farkında olmadan, özellikle dil öğrenimi, kelime dağarcığı geliştirme ve bilişsel uyarılma gibi alanlarda öğrenmelerini sağlamaktır. Bu, eğitim alan öğrenciler ve her yaş grubundan bireyler için günlük yaşamda önemli bir sorundur. Ayrıca, AI destekli kelime tahmini kullanımı sayesinde kullanıcıların kelime bulma becerileri desteklenir ve dil öğrenimi kolaylaştırılır.

Önerilen Yöntem olarak uygulamada Random Forest ve harf sıklığı analizi gibi makine öğrenimi ve derin öğrenme tekniklerinin hibrit modelleri kullanılmıştır. Spesifik olarak, kelime vektörleştirme, n-gram özellik çıkarma ve TF-IDF gibi özellik çıkarma süreçleri kullanılmıştır. (words alpha) gibi büyük İngilizce kelime listeleri veri kaynağı olarak kullanılmıştır. Uygulama, hem mobil (Android/iOS) hem de masaüstü platformlarda çalışacak şekilde hibrit olarak tasarlanmıştır.

Geliştirilen Uygulama olarak uygulanan sistem kullanım kolaylığı açısından dostudur ve 3-7 harfli kelime tahmini, AI tabanlı öneriler, hızlı yanıt ve çevrimdışı destek gibi temel işlevler sunar. Kelime oyunu oyuncuları, öğrenciler ve günlük zihinsel egzersiz arayan kişiler hedef kullanıcılar arasındadır. Kullanım örnekleri arasında kullanıcıların kelimeleri tahmin etmesi, AI'dan öneriler alması ve sonuçları anında görüntülemesi yer almaktadır.

Deneyisel Sonuçlar kısmında modelin doğruluğu, F1 puanı ve ROC-AUC gibi temel performans ölçütleri açısından değerlendirildi; Random Forest tabanlı hibrit model için %99 doğruluk ve F1 puanı elde edildi. Uygulama arayüzü, kullanıcı testleri sırasında kullanıcıların %90'ı tarafından olumlu olarak değerlendirildi ve hız ve verimlilik açısından karakter girişi ve tahmin işlemleri anında gerçekleştirildi. Model, karşılaştırma veri setleriyle karşılaştırıldığında bile iyi sonuçlar verdi.

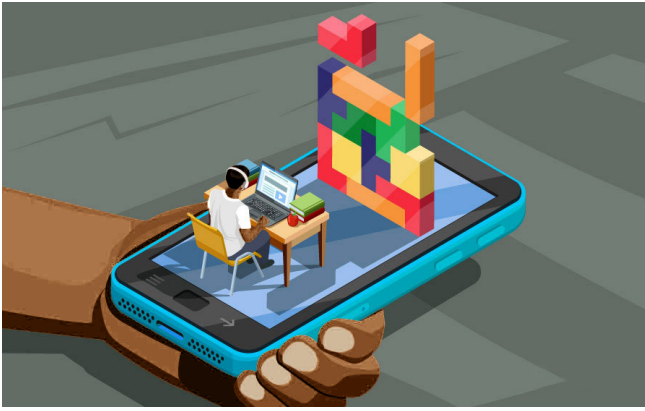
Sonuç olarak AI destekli kelime tahmini ve kullanıcı öğrenimi dinamik modelinin masaüstü/mobil platformlara entegre edilmesidir. Gerçek zamanlı kelime analizi ve oyunlaştırılmış öğrenme sistemini entegre ederek teknoloji, uygulamalar ve literatüre fayda sağlamaktadır. Gelecekte, çok dilli özellikler, sesli asistan işlevselliği ve gerçek zamanlı model güncellemeleri de entegre edilebilir.

Keywords—Kelime Oyunu, Makine Öğrenmesi, Derin Öğrenme, Random Forest, TF-IDF, Mobil Uygulama, Kullanıcı Deneyimi, Doğal Dil İşleme, Hibrit Model, Kelime Sınıflandırma, Yapay Zekâ, Eğitim Teknolojileri

etkileyeceği bile tartışma konusu olmuştur[61]. Ayrıca bu ilerlemelerin hepsinde çoğunlukla anlamlı kelimeler ve düzgün anlaşılır cümleler üzerinde olmuştur. Hatta Alan Turing'in makinesinden günümüze kadar teknolojik gelişmelerin en çok önem sahibi olan YZ oldukça ilerlemiş olmasıyla beraber makale yazıp yazamayacağı bile tartışma konusu olmuştur[11]. Bunlara rağmen anlamı olmayan ve rastgele üretilen metinler üzerinde yapılan çalışmalar sınırlıdır. “Şifre Çılgınlığı” adlı bu çalışma, anlamsız kelimelerden oluşan dizilimlerin belirli ifadeler taşıdığı varsayımına dayanarak, bu dizilimleri analiz edebilecek ve şifrelerini çözebilecek bir sınıflandırma modeli geliştirmeyi amaçlamaktadır. Problemin çözümü için metin madenciliği, dil modelleme ve makine öğrenimi algoritmaları kullanılmakta olup, aynı zamanda derin öğrenme tabanlı Evrimsel Sinir Ağları (ESA) ve Tekrarlayan Sinir Ağları (TSA) gibi gelişmiş modellerden de yararlanacaktır. Mobil tabanlı sistemlerin problemin çözümüne katkısı önem taşımaktadır çünkü mobil platformlar kullanıcı kitlesinin geniş olması nedeniyle veri setinde daima artış gözlenmektedir. Ayrıca kullanıcı dostu bir arayüz tasarlanırsa gerek eğlence gerekse eğitim açısından kullanıcı sayısının da artış göstermesi sağlanabilir. Mobil entegre olan bir sistem hem sürdürülebilir hem de kullanıcıdan gelen veriler sayesinde kendini geliştiren bir YZ modeli oluşturmasına olanak sağlar. Bu çalışma sayesinde dil işleme ve veri madenciliği alanındaki mevcut boşluklar doldurulabilmektedir. Sınıflandırma verileri belirli kategorilere ayırma işlemidir. Sınıflandırma her ne kadar veri işlemesi için önemli bir yöntem olsa da veriler büyüdükçe ve karmaşılaştıkça veri işleme metodlarına da ihtiyaç duyarız[66]. Sınıflandırma sayesinde mobil uygulamada anlamsız olan harflerden bir örüntü tespit edilir. Kullanıcıdan veri aldıktan sonra kelimeler arasındaki şifre bağlantısını bulur. ESA ve TSA gibi modellerle birlikte Unity motorunda kullanılan ONNX ile birlikte başka makine öğrenimi modellerinin eğitimi ve testi sağlanabilir. Ayrıca Tensorflow kütüphanesinden de aynı şekilde yararlanılabilir. Unity mobil uygulama için önemli bir platform olup içinde yeni gelen kullanıcılar için geniş ve çeşitli programlama imkanları sunup uygulama geliştirmeyi kolayca yapmayı sağlamıştır.

1 GİRİŞ

Teknolojinin hızla ilerlemesiyle birlikte yapay zeka (YZ), pek çok alanda etkisini göstermeye başlamıştır. Özellikle dil işleme, metin analizi, metin sınıflandırma, bilgi çıkarımı, duygu analizi ve metin üretimi gibi konularda oldukça önemli derecede ilerlemeler kaydetmiştir. Kaydedilen bu ilerlemelerde en çok göze batan metin sınıflandırma, Türkçe için belli başlı sorunlar ortaya çıkartmaktadır. Türkçe'nin zengin ama karmaşık ve sondan eklemeli dil olması nedeniyle metin sınıflandırması zorlaşmaktadır[48]. Diğer yandan YZ metin üretimi açısından bakılınca ise iş gücünün nasıl



Özellikle genç kitleler tarafından çevrimiçi oyun ve iletişim kültürü etkili olduğu için popülarite zamanla giderek artmaktadır[14]. Mobil uygulamada kullanılan sınıflandırma sistemi sayesinde gerçek zamanlı analiz , verileri depolayarak işleme yeteneği ve mobilde kullanılan optimizasyonlar sayesinde problemi çözmek için alternatif yöntemler sunulmuştur. Makine öğrenimi sistemleri genel olarak görüntü sınıflandırması , konuşma tanınması, oyun oynama ve makine çevirisi gibi alanlarda gelişmiş bir performans sonucu vermektedir[42]. Görüntü işleme metodları sayesinde daha önce manuel olarak yapılan işlemlerden otomatik olarak yapılan işlemlere geçiş yapılabilmektedir[62].Görüntü işleme yöntemleriyle birlikte medikal alanda yapılan işlemlerde görüntü bölme kullanılmıştır. Bu yöntem sayesinde de daha hızlı ve doğru bir şekilde sonuçlar elde edilmiştir[60].Spam filtrelemesi gibi karşılaşılabilecek problemlerde Naive Bayes ve Destek vektör makinesi metodları kullanılabilir. Metin sınıflandırma metodu yoluyla çeşitli ve karmaşık girilen metin bilgileri problemi daha örüntülü ve düzenli bir forma girer. Örnek olarak bilgi alma, konu etiketleme, duygu analizi ve haber sınıflandırma alanları yer alır[24].Mobil uygulamada metin sınıflandırma yöntemi kullanılmıştır. Mobil oyun tasarlanırken kullanılan makine öğrenme yöntemleri geleneksel öğrenme yöntemlerine kıyasla %10 daha etkilidir[55]. Unity oyun motoru sayesinde diğer geliştirilen mobil oyunlarda bağımsız oyun geliştiricilerine büyük bir destek olmuştur[20].Büyüyen piyasayla birlikte Çin gibi ülkeler de zamanla bu alanda avantaja sahip olmuştur[29]. Genel olarak incelendiğinde mobil alanda bu yöntemlerle ileride karşılaşılabilecek sorunlara karşı bir ek önlem alınmıştır. Çeşitli problem çözme metodları ele alındığı için farklı alanlarda farklı çözüm imkanları doğmuştur.Ayrıca kullanıcı incelemeleri ve yorumları da mobil alanında etkili bir yorumlama faktörüdür.Farklı bakış açıları beraberinde farklı yol haritaları getirir[41].

YZ ve dil işleme ile birlikte literatürdeki eksiklikleri gidermek amaçlanmaktadır. Mevcut araştırmalarda veri setlerinin yetersizliği[47], sınıflandırma algoritmalarının çok düşük doğruluğu, anlamsız kelimelerin karmaşıklığından uzak, oyunlaştırma mekanizmalarının eksikliği ve mobil platforma uyarlanabilirlik gibi dikkate alınması gereken sınırlar vardır. Hatta mobil platformda yapılan bu uygulamaya geniş bir kullanıcı kitlesine ulaşmayı amaçlamaktadır. Kullanıcılar aracılığıyla elde edilecek veriler oldukça fazla olacağından gelişmesi de bir o kadar hızlı olacaktır. Çalışmanın özgün olması için yöntem olarak farklı sınıflandırma algoritmalarını ve hibrit modelleri bir araya getirerek [64] Naive Bayes, Destek Vektör Makineleri (DVM), Karar Ağaçları gibi geleneksel makine öğrenimi yöntemleri ile ESA ve TSA gibi derin öğrenme modellerinin analizlerinden yararlanmak amaçlanmaktadır. Bu sayede anlamsız rastgele oluşan kelimelerin anlamlı

hallerinin bir çeşit görsel veya ifade olarak karşılığı eşleştirmede daha yüksek oranda doğru sonuçlar ve doğruluk oranının daha yüksek olması açısından önem taşımaktadır. Ayrıca kullanıcılardan gelen veriler sayesinde kendini sürekli yenileyen ve güncelleyen bir sistem olması amaçlanmaktadır. Gerçek zamanlı veri analizi, sistemin sürekli olarak yeni verilerle eğitilmesini ve gelişmesini mümkün kılarak literatürdeki statik veri seti[9] sorununu ortadan kaldıracaktır. Bununla birlikte kullanıcı dostu bir arayüz ile sistem ve kullanıcının etkileşimde kalıyor olması eğitsel ve eğlenceli bir deneyim kazanılmasını desteklemektedir. Bu sayede çalışma hem akademik olarak ilerleme sağlayacak hem de YZ dil işleme sistemlerinde daha geniş bir uygulama alanına [3] yayılmasına önayak olacaktır.

Yapılacak olan çalışmanın amacı mobil platformda sorunsuz bir şekilde çalışabilecek elde edilen anlamsız kelimeleri düzenleyerek anlamlı kelimeler topluluğuna dönüştürmektir. Ekstra yapay zeka ve kendi kendine geliştirilen bir modelle desteklenmesi amaçlanmaktadır. Kullanıcılardan gelen geri bildirimler sonucunda oluşan veri kümesi büyütülerek modelin doğruluk seviyesinde de bir artış gözlemlenmesi amaçlanmaktadır. Özel hedefleri teker teker sıralayacak olursak:

- Farklı dil grupları ile birlikte sorunsuz çalışması
- Geniş bir veri kümesi oluşturarak kelime daracığını geliştirmek
- Optimizasyon aşamasının eksiksiz tamamlanması

gibi hedefler amaçlanmaktadır.

Makalenin ilerleyen kısımlarında ilk olarak literatür çalışması gelmekte olup çalışmamıza yakın olan makale çalışmalarından bahsedilmektedir. Bu kısımda amaçlanan daha önceki çalışmalarda oluşan açıkları bulup kendi çalışmamızın özgünlüğünü arttırmaktır. Ardından yöntem gelmektedir. Bu bölümde çalışmanın hangi yöntemleri izleyerek kendini geliştireceği ve özgünlüğünü arttırmak amaçlanmaktadır. Ayrıca yöntem zenginliği sayesinde YZ ile olan entegrasyonundan da bahsedilmektedir. Ardından gelen deneysel sonuçlar bölümünde ise çalışmanın özgün algoritması ve doğruluk oranlarından bahsedilmektedir. Sonraki bölümde ise mobil tabanlı sınıflandırma sistemleri bulunmaktadır. Bu bölümde kullanıcı deneyiminden bahsedilerek sınıflandırma sistemleri bulunmaktadır. Daha sonra gelen tartışma bölümünde ise çalışmanın durumu, oluşan açıkları ve gelecekte yapılması amaçlananlardan bahsedilmektedir. Son olarak ise sonuç bölümü gelmektedir. Sonuç bölümünde ise hazırlanmış olan bu makale çalışması ile ilgili özet niteliğinde bilgi verilerek çalışma açıklanmaktadır.

2 LİTERATÜR TARAMASI

Mobil oyun sektörü artan teknoloji ile birlikte büyümektedir.Bunun yanı sıra kullanılan methodlar da gelişmektedir. Kelime tanıma için kullanılan modeller ve bunların entegrasyonu ile oluşan araçların sayısı fazladır.Bu bölümde literatür taraması yapılarak farklı makalelerde bulunan modeller derlendi.Aşağıda **Tablo 1**'de yapılan çalışmalar sonuçlar, eksiklikler ve katkıları biçiminde incelenmiştir.

Yuening Yan (2021) ve Yi Li arkadaşları tarafından yapılan çalışmada aksiyon video oyunu (AVG) sonrasında yüz ifadeleri ve duygusal kelimeler ile davranış tanımlaması yapılmıştır.Yapılan çalışma sonrası uzman ve amatör diye ayrılan iki grup arasında da negatif N400 ve N170 bileşeni gözlenmiştir.ERP ölçümleri sonrasında gruplar arasında farklılık gözlenmemiştir.Duygusal algı becerileri konusunda daha fazla araştırma yapılması gerektiği sonucuna varılmıştır.Çalışmada veri seti doğrudan belirtilmeyip

duyusal ölçümde kullanılan araçlar kısıtlıdır. [68]

İbrahim Yazıcı (2023) ve arkadaşları tarafından yapılan derleme çalışmasında yapay zeka ve makine öğrenimi tekniklerini mobil ağlar (4G ve 5G) üzerinde test edilmiştir. 174 akademik yayın incelenmiştir ve bununla birlikte denetimli ve denetimsiz öğrenme , pekiştirmeli öğrenme , derin öğrenme yöntemleri farklı alanlarla birlikte kullanılmıştır. Bu alanlar siber yönetim , akıllı ulaşım alanları , dijital ikizler ve insansız hava araçlarıdır. Sonuç olarak yapay zeka ve makine öğrenimi tabanlı sonuçların gerçek zamanlı karar alımı ve büyük veri işleme alanlarında optimizasyon konusunda gelişmeler görülmüştür. Mevcut çalışmalar ayrıca sınırları ve gelecek çalışmaları açısından da incelenmiştir.Bu sınırlardan ikisi güncel teknolojiye olan uyumsuzluk ve deneysel veri eksikliğidir.[69]

Kaibin Huang (2023) ve arkadaşları tarafından yapılan çalışmada ise 6G mobil iletişiminin potansiyelleri ve uygulama alanı kapsamı çerçevesinde incelenmiştir. 6G teknolojisi yalnızca insanları ve nesneleri birbirine bağlamayı değil aynı zamanda da zekayı da birbirine bağlamayı hedefleyerek iletişim alanında yeni bir adım atılmıştır.Makalenin yazarları aynı zamanda 5G teknolojisini 6G teknolojisi ile karşılaştırarak yenilikleri ve farklılıklarını nitelemiştir.6G teknolojisinin vaatleri arasında yapay zeka entegrasyonu , yüksek veri hızı ve gelişmiş güvenlik protokolleri yer almaktadır.Yeni olan bu teknolojinin eksikleri olarak gerçek zamanlı performansta düşük sonuçlar vermesi , ağ kaynakları, veri gizliliği ve enerji tüketimi örnek gösterilebilir.[30]

Ayrıca Andreas (2022) ve arkadaşları ile yapılan çalışmada Epic Games ve Unity motorunun (Oyun yapımında kullanılan başlıca motorlardan ikisidir) metaverse ve artırılmış gerçeklik konusunda etkilerini ele almaktadır. Başlıca ele alınan konular şunlardır:

- Artırılmış gerçeklik teknolojilere altyapı oluşturma
- Sosyal medya ve iletişim araçlarında merkez konumunda olma
- Dijital kimlik ve sosyal altyapılarının oluşması

Çalışma ayrıca bu motorların kültüre olan etkilerini de detaylıca bir şekilde anlatmıştır. Bu makale teorik ve gelecek vaatlerine dayanıp deneysel aşamalar bulunmamaktadır. Makalenin eksiklerine gelirse şirket karşılaştırılmasının azlığı , veri alanındaki eksiklik ve bazı alanları kaçırmak gösterilir.[33]

TABLE I: MOBİL VE WEB ALANINDA YAPILAN ÇALIŞMALARIN KARŞILAŞTIRMA TABLOSU

| Yazar | Platform | Veri Seti | Yöntemler | Sonuçlar | Eksiklikler | Gelecek Katkıları |
|-------------------------------------|--------------|---|--|----------|---|--|
| Yuening Yan ve ark. (2021)[68] | Web | Doğrudan belirtilmemektedir. | Davranışsal Ölçümler Elektroensefalografi Kaydı Kurumsal Kaynak Planlaması Analizi | %92.17 | Sadece bir oyun türü kullanılması ve duyuşsal ölçüm için sınırlı araç kullanılması. | Duyuşsal algıyı daha kapsamlı yöntem kullanmak ve farklı türde oyunlar kullanılması amaçlanmaktadır. |
| İbrahim Yazıcı ve ark. (2023)[69] | Mobil | Meta-analiz yöntemi kullanılarak yapılmış çalışmaların verileri kullanılmıştır. | Kümeleme yöntemleri Evrişimsel sinir ağları Yapay sinir ağları Karşılaştırma analizi | - | Deneyisel verinin eksikliği, güncel teknolojiye uyumlu olamaması ve veri yetersizliği. | YZ ve makine öğreniminin mobile entegre edilmesi ve gerçek zamanlı veri analizi yapılması amaçlanmaktadır. |
| Kaibin Huang ve ark. (2023)[30] | Mobil | Doğrudan belirtilmemektedir. | In-Situ model indirme algoritması Model havuzu ve uygulama algoritması Uç-bulut işbirliğine dayalı öğrenme algoritması | - | Gerçek zamanlı performans, veri gizliliği, ağ kaynakları, enerji tüketimi gibi konularda sorunlar yaşanmaktadır. | Enerji tüketiminin azaltılması, gelişmiş uç-bulut işbirliğinin artırılması ayrıca güvenlik ve uyumun düzenlenmesi amaçlanmaktadır. |
| Andreas Jungheer ve ark. (2022)[33] | Web | Doğrudan belirtilmemektedir. | Kritik analitik yöntem Veri analizi Araştırma tasarımı | - | Belirli şirketlere odaklanma, sayısal veri eksikliği ve bazı alanları göz ardı etmek. | Bağımsız şirket, ekonomik etkileri ve teknolojik gelişmeler alanlarında daha önemli analizler yapılması amaçlanmaktadır. |
| Tim D. Smithies ve ark. (2021)[58] | Masaüstü | 21,588 Rocket ligi maçına ait 4 farklı oyuncunun rank bilgilerini içermektedir. | Rastgele orman öğrenme algoritması Random forest algoritması | - | Modelin sınırlı olması ile veri setinin sınırlı olması modelin her koşulda geçerli olmasını engelleyebilmektedir. | Modeli gerçek zamanlı olması ve ileri düzey performans analizleri yapabildiği ayrıca başka makine öğrenme dillerini de entegre edebilmesi amaçlanmaktadır. |
| DANIEL ZARATE ve ark. (2023)[59] | Web | 565 bireyin 2 kez oynamak şartı ile 6 ay arayla elde edilen oyun bozukluğu testi ile değerlendirilmektedir. | Makine öğrenimi YZ algoritması k-NN algoritması Naive Bayes | %95 | Zamansal olarak oluşan sıkıntılar ve yöntem olarak kısıtlı algoritmalar kullanılmaktadır. | Uzun vadeli çalışmalar ve yeni özelliklerin eklenmesi amaçlanmaktadır. |
| José Pedro Pinto ve ark. (2021)[50] | Web ve mobil | 118 oyuncunun 490 saat boyunca gerçek zamanlı olarak veri elde edilmektedir. | Konvolüsyonel sinir ağları Çapraz doğrulama yöntemi Derin öğrenme | %99.2 | Mobilde gerçek zamanlı olarak uygulamanın performansı ve hile türlerinin sınırlılığı | Farklı hilelerin incelenmesi, performansın artırılması ve oyun ile hile tespiti arasındaki ilişki |
| Graceline Yasemin ve ark.(2022)[39] | Web ve mobil | Doğrudan belirtilmemektedir. | Greedy Best First algoritması YZ algoritması Genişlik öncelikli arama algoritması | - | Doğrudan belirtilmemektedir. | Doğrudan belirtilmemektedir. |
| İsa Ergin ve ark. (2024)[21] | web | Türkçe dilindeki metinlerden elde edilmiştir ama sayısal bir veri bulunmamaktadır. | Kodlayıcı-kod çözücü model Dikkat mekanizması | %91.9 | Veri seti hakkında belirsizlik ve modelin web veya mobilde nasıl entegre edileceği hakkında bilgi yeterli olmamaktadır. | Daha büyük veri çeşitliliği, mobile entegre edilmek istenilmesi, gerçek zamanlı olarak güncel testler yapılması ve farklı dillerde de yapılması amaçlanmaktadır. |
| Burak DELİCAN ve ark. (2021)[18] | Mobil | 30 mobil uygulama incelenmiştir sayısal veri bulunmamaktadır. | Araştırma modeli Araştırma materyali Veri toplama araçları ve süreci Veri analizi | - | Sınırlı sayıda veri ile araştırma yapılmıştır. | Uygulamaların geliştirilmesi için deneyisel sonuçlar yapılması hedeflenmektedir. |
| Bora Kat ve ark. (2023)[34] | Web | 1255 proje önerisi ile veri seti kullanılmaktadır. | Doğal dil işleme Karar destek sistemi Benzerlik analizi Naive Bayes | %96,4 | Sadece Türkçe metinlere odaklanılmış, güncelleme eksikliği ve açık kaynak erişimi bulunmamaktadır. | Farklı dillerde, YZ destekli ve açık kaynak kodlu uygulama yapılması amaçlanmaktadır. |
| Güler Alparslan ve ark. (2023)[2] | Web | Doğrudan belirtilmemektedir. | Destek vektör makineleri Naive Bayes Karar ağaçları Yapay sinir ağları Derin öğrenme | %96.2 | Zaman ve kaynak kullanımı, performans karşılaştırılması ve veri çeşitliliği | Modeli iyileştirmek, daha fazla algoritma karşılaştırma ve gerçek zamanlı uygulamalar yapılması amaçlanmaktadır. |
| Kısmet Deliveli ve ark.(2022)[19] | Mobil | Doğrudan belirtilmemektedir. | Dijital teknoloji destekli ses tabanlı hece öğretim yöntemi | - | Katılımcı sayısının azlığı, sayısal verilerin olmaması ve veri analizi sınırlı olması | Nitel ve nicel veri toplayarak daha verimli sonuçlar elde edilmesi ve farklı kişiler aracılığı ile veri arttırmak |
| Ali Hakan Işık ve ark. (2022)[22] | Mobil | Doğrudan belirtilmemektedir. | Hibrit şifreleme algoritması | %99.9 | Mobil platform performans sorunları ortaya çıkmaktadır. | Web tabanlı platformlara da entegre edilmesi amaçlanmaktadır. |

Tim D. Smithies (2021) ve arkadaşları ile yapılan bu çalışmada online ve e-spor alanında tanınmış bir oyun olan Rocket League oyununda oyuncuların performansı ve oyun sonu maç istatistiklerini etkileyen belirleyici olan metrikleri tespit etmek için Rastgele Orman (Random Forest-RF) algoritması kullanılmıştır. RF algoritması çok sayıda karar ağacı içinden toplu oynama yöntemini kullanarak tahmin etmeye çalışmışlardır. Model sayesinde oyuncu davranışlarına ilişkin çok boyutlu veriler (Örneğin: atılan şut ve gol sayısı, süperonik hızı kullanma zamanı ve yeri) girdi olarak alınmış ve sonuç olarak yüksek doğrulukla oyuncuların oyun sonundaki skor tahtası sıralamasını tahmin etmiştir. Modelin sadece Rocket League’te kullanılması modeli sınırlamıştır. Veri seti de kendi içinde sınırlıdır. Çalışma e-spor analitiği bakımından başarılı bulunan bir çalışmadır. [58]

Daniel Zarate (2023) ve arkadaşları tarafından yapılan bu çalışmada kullanıcı - avatar bozukluğu üzerine boylamsal veri yardımıyla sonuç alınmıştır. Sonuçların analizinde ise makine öğrenimi yöntemlerinden faydalanılmıştır. Derin öğrenme yöntemlerini ise davranış eğilimlerini tahmin etmekte kullanmıştır. Çapraz Doğrulama yöntemleri de derin öğrenmede kullanılmıştır. Genel olarak oyuncuların avatarla kullandığı bağın psikososyal etkileri incelenmiştir. Veri setindeki örüntülerden faydalanarak yüksek riskli profilleri sınıflandırmıştır. Böylece oyunların dijital bağımlılık ve birçok psikolojik alanda etkileri detaylı bir şekilde ele alınıp incelenmiştir ama zamansal olarak kısıtlamalar ve yöntem yetersizliği eksikliği bulunmaktadır[59].

José Pedro (2021) ve arkadaşları ile yapılan bu makalede çevrimiçi oyunlarda kullanılan hileleri tespit etmek için çok değişkenli zaman serileri kullanan bir derin öğrenme metodu önermişlerdir. Oyuncuların fare ve klavye kullanmaları sonucunda olan verileri elde edilerek "aimbot" ve "trigger" hileleri tespit edilmiştir. Evrişimsel Sinir Ağları (CNN) faydalanarak da davranışları bir örüntü haline sokarak doğruluk oranı arttırılmıştır. Çevrimiçi oyunlarda hile önemli bir faktördür. Bu makale de buna çözüm yöntemi sunmuştur. Eksiklik olarak uygulamanın mobile entegre edilememesi ve incelenen hile türlerinin sınırlı olması verilebilir.[50]

Graceline Jasmineve (2022) arkadaşları tarafından yapılan bu çalışmada video oyunlarında path finding (Yol bulma) algoritmalarını teker teker ele alarak detaylı bir şekilde incelemiştir. Bu algoritmalar kendi içlerinde bilgilendirilmiş (informed), bilgilendirilmemiş (uninformed) ve meta-sezgisel (metaheuristic) olmak üzere üç kategoriye ayrılmıştır. Pratik yaklaşımda, Dijkstra, Genişlik Öncelikli Arama (BFS) ve Açgözlü En İyi Öncelikli Arama (Greedy Best-First Search) metodları kullanılmıştır. Bu algoritmalar yol bulmak için farklı faktörleri ve yöntemleri ele alarak Unity motorunda Grid Search (Izgara Arama) uygulanarak kendi aralarında karşılaştırılmıştır. Karşılaştırma standartları olarak yürütme süresi, toplam iterasyon sayısı, en kısa yol uzunluğu ve grid size (izgara boyutu) bulunmaktadır. Sonuç olarak bu çalışma yapay zeka oyun karakterlerinin daha gerçekçi bir şekilde hareket kabiliyetini geliştirmek için yol bulma algoritmaları kullanılmıştır. Eksiklikler çalışmada doğrudan belirtilmemiştir.[39]

İsa Ergin (2024) ve arkadaşları tarafından yapılan bu çalışmada Türkçe kelimeler elde etmek için kodlayıcı kod çözücü(encoder decoder) mimariyle birlikte dikkat mekanizması da kullanılmıştır. Model uzun kelimelerden anlamlı kelimeler üretmek ve dilbilgisel olarak üretilen kelimelerin doğruluğunu test etmek için kullanılmıştır. Dikkate değer bir unsur olarak özyinelemeli yapay sinir ağları (RNN) sayesinde geçmişte yazılan ve kullanılan kelimelerin hatırlanması sağlanmıştır. Son olarak modelin performans sonuçları sıcaklık örnekleme (temperature sampling) yöntemi

sayesinde farklı eşik aralıklarında tespit edilerek %91.9 doğruluk oranı değerine ulaşılmıştır. Bu makale doğal dil işleme için değerli bir çalışma niteliği taşımaktadır. Her ne kadar doğruluk oranı yüksek olsa da veri seti sınırlıdır ve modelin entegre edilip edilememesi belirsizliği de eksiklik olarak gösterilebilir.[21]

Burak Delican (2021) tarafından hazırlanan bu makalede okuma ve yazma eğitimi için geliştirilen mobil uygulamalar değerlendirilmiştir. Yapılan inceleme daha çok işlevselliği, tasarımları ve işlevleri üzerine yapılan bir inceleme makalesidir. İncelenen 30 mobil uygulama sonucunda birçok uygulamada hata yapıldığı, sessiz harflerin sesli olarak söylendiği ve birkaç uygulama İngilizce tabanlı olduğu için İngilizce seslendirildiği gibi sorunlarla karşılaştırılmıştır. Ek olarak kelimeleri heceye ayırma ve harflerin oluşturma yönlerinde hata olduğu tespit edilmiştir. Çıkan bulgulara göre bu uygulamaların teknik ve pedagojik alanda eksik olduğu sonucuna varılmıştır. Sınırlı sayıda veri ile çalışma yapması çalışmanın eksikliğidir. [18]

Bora Kat (2023) tarafından yapılan bu makalede Türkçe akademik metinler doğal dil işleme yöntemi ile test edilmiştir. Çalışma kapsamında, TÜBİTAK ARDEB’e sunulan 1255 proje için anahtar kelime çıkartma ve öznetelik mühendisliği uygulanarak metinlerin kavramsal açıdan tespitleri bulunmuştur. Sonraki aşamalarda metinler arasındaki benzerlikleri tespit etmek için bir algoritma geliştirilmiştir. Naive Bayes sınıflandırma kullanarak %96.4 oranında doğruluk oranı çıkmıştır. Elde edilen sonuçlar neticesinde bu makale benzerlik analizi ve sınıflandırma için karar destek aşamalarında kullanılabilir. Sadece Türkçe metinlerin kullanılması, güncelleme imkanının olmaması ve açık kaynak erişim kısıtlaması bu makalenin eksiklikleridir.[34]

Güler Arpaslan (2023) ve arkadaşı ile yapılan bu çalışmada Türkçe metinleri sınıflandırmada Konvolüsyonel Sinir Ağları (CNN) destekli bir derin öğrenme modeli kullanılmıştır. Çalışmada bulunan iki farklı Türkçe metin veri kümesi üzerinde Naive Bayes (NB), Destek Vektör Makineleri (SVM), K-En Yakın Komşu (KNN) ve Rastgele Orman (RF) gibi geleneksel olan makine öğrenimi metodları kullanılmıştır. Metin verileri Terim Frekansı-Ters Belge Frekansı (TF-IDF) metodu sayesinde vektörleştirme işlemi yapılmış ve ön işleme adımlarında gereksiz kelime temizliği yapılmıştır. Özellik seçiminde vektör boyutları azaltılmıştır ve çeşitli kombinasyonlar denenerek doğruluk ve F1 skorumun sonuçları karşılaştırılmış. Elde edilen sonuçlar neticesinde CNN metodunun diğer metodlara kıyasla performansı yüksek gelmiştir. Bu çalışmadaki eksiklerden söz edecek olursak zaman ve kaynak kullanımındaki yetersizlik, performans karşılaştırmasına yer vermemesi ve veri çeşitliliğindeki ktlık örnek verilebilir.[2]

Kısmet Deliveli (2022) tarafından hazırlanan bu çalışmada okuma yazma öğretimini inceleyen bir çalışmadır. Çalışma aykırı tipik örnekleme ile ölçüt örnekleme teknikleri kullanılarak bir çocuk, bir öğretmen ve bir annenin içinde bulunduğu bir test grubu üzerinde çalışılmıştır. Elde edilen verilerin incelenmesi, betimsel ve içerik analizleri tekniklerine başvurulmuştur. Bunun sonucunda yöntemi uygulayan anne, öğretmen ve öğrenci tarafından olumlu bir sonuç elde edilmiştir. Bu çalışmada katılımcı sayısı yetersizdir, sayısal verilere hiç yer verilmeyip veri analizinde boşluklar vardır.[19]

Ali Hakan Isık (2022) ve arkadaşı tarafından hazırlanan bu çalışmada mobil cihazlarda destek amaçlı güvenliği yüksek bir hibrit model hazırlanmıştır. Algoritma kendi içinde simetrik şifreleme kullanarak üç aşamadan meydana gelmektedir. Bunlar:

- Statik anahtar kullanımı
- Matematiksel Sabitlerin Entegrasyonu

- Dinamik İndeksleme

Algoritma harf tabanlı olduğu için kendi içinde çeşitli indexler ve fonksiyonlardan da faydalanılır. Ayrıca rastgele anlamlı kelime oluşturma ya da oluşturmama kontrolü Fibonacci dizisindeki elemanları kullanarak sağlanabilir. Testlerde sırasıyla 2kb, 10 kb ve 20 kb boyutlarında veriler üzerinde testi sağlandıktan sonra sonuç olarak %99.98, %99.996 ve %99.9962 doğruluk oranlarına ulaşılmıştır. Oluşturulan algoritma mobil platformlarda sorunsuz bir şekilde çalışacak şekilde yapılmıştır. Ek olarak harf havuzuna eklenen harflerin de sistemde sorunsuz çalışmıştır. Algoritma her ne kadar yüksek sonuç verse de mobil platformda performans sorunlarıyla karşılaşmaktadır.[22]

Bu çalışmada mobil ve web tabanlı uygulamaların sınıflandırmasında kullanılan çeşitli makine ve derin öğrenme yöntemleri incelenmiştir. Bu çalışmaların eksikleri ve potansiyel katkıları da incelenmiştir. Yukarıda yazılanları toplamak ve sonuca erdirmek için aşağıda alt başlıklar halinde açıklama yapılmıştır.

2.1 Başlangıç

2.1.1 Problemin Sınıflandırılması

Mobil ve Web tabanlı platformlarda dili anlamlandırmak ve düzene sokmak önemli bir unsurdur. Öğretici olmasının yanı sıra da zaman geçirmek için Kod Çılgınlığı adında anlamsız kelimeleri anlamlandırarak bir mobil uygulama tasarlanmıştır. Yapılan uygulamada doğal dil işleme ve veri işleme gibi alanlardan da yararlanılmıştır. Geri bildirimler sayesinde de interaktif bir deneyim oluşturulur.

2.1.2 Araştırmanın Amacı ve Önemi

Bu araştırma sadece akademik alanda olmayarak mobil alanda da yapay zeka ile yapılacak fırsatların çokluğuna önem çekmektedir. Self-improving (Kendini geliştirilen) bir model kullanılması sonucunda da zaman içerisinde uygulamanın gelişmesi ve güncel kalması amaçlanır. Makine ve derin öğrenme kapsamlı bir alandır. Bunun mobil entegrasi de aynı şekilde önem taşımaktadır.

2.1.3 Çalışmanın Kapsamı ve Hedefleri

Çalışmada algoritmanın haricinde kullanıcı deneyimleri de hedef alınıp değerlendirilmiştir. Mobil entegrasyonu sağlamak içinse Unity oyun motorundan faydalanılmıştır. Geleceğe olan katkıları için mobilde yapay zekanın önemi ve gelişen mobil oyun piyasasında güvenilir ve optimize uygulamalar geliştirmek ve test etmektir.

2.1.4 Çalışmanın Yapısı

Mobil platforma entegre etmek için kullanılan araçlar problemin yapısına göre değişmektedir. Dil işleme problemi ele alınıp çözüldüğü için bu probleme özgü araç ve gereçler kullanılmıştır. Mobil zamanla gelişen ve değişen bir platformdur. Bu yüzden bu alanda kullanılan yöntemler de çeşitlendirilebilir.

2.2 Mobil ve Web Platformlarında Kullanılan Sınıflandırma Yöntemleri

2.2.1 Mobil ve Web Tabanlı Sınıflandırma Çeşitleri

Mobil ve Web alanında da hem hızlı hem de verimli bir biçimde sınıflandırma yapabilen algoritmalar bulunmaktadır. Tim Smithies (2021) [58] ve arkadaşları tarafından hazırlanan Rastgele Orman algoritmasında hem anlık olarak hem de yüksek doğruluklu oyun

sonucu verilmiştir. Güler Arpaslan (2023) [2] ve arkadaşı tarafından hazırlanan çalışmada kullanılan CNN metoduyla da %96.2 değerinde bir doğruluk değeri ve gerekli ön işleme metodlarıyla hızı teyit edilmiştir. Son olarak José Pedro ve arkadaşları (2021) [50] tarafından kullanılan CNN metoduyla da %99.2 doğruluk oranına ulaşırken hız bakımından da var olan hile sistemlerine üstünlüğünü göstermiştir.

2.2.2 Gerçek Zamanlı Mobil Sınıflandırma

Mobil cihazlar gerçek zamanlı çalışan uygulamalar oldukları için gerçek zamanlı sınıflandırma da bu uygulamalar için hayati önem taşımaktadır. İbrahim Yazıcı (2023) ve arkadaşları derin öğrenme ve pekiştirmeli öğrenmeyi mobil ağlarda kullanarak sonucunda birçok alanda optimizasyona ulaşmışlardır [69]. Diğer yandan Bora Kat (2023) tarafından yapılan doğal dil işleme sonucunda ise metinleri sadece türkçe için denemiştir. Yüksek hesaplama gücü gerektiren diğer diller bu makalenin eksiklerindendir [34]. Ek olarak Vasilcios (2023) ve arkadaşları tarafından geliştirilen davranış eğilimlerine yönelik derin öğrenme veri setini örüntüleme başarılı olsa da zamansal alanda sıkıntı yaşamıştır [59].

2.3 Mobil ve Web Platformlarında Kullanılan Sınıflandırmadaki Kullanımları

2.3.1 Makine Öğrenimi Algoritmaları

2.3.1.1 Naive Bayes

İstatistiksel bir makine öğrenme metodu olan Naive Bayes Vasileios (2023) ve arkadaşları veri seti analizi için sağlamıştır. Spesifik olarak küçük veri setlerinde başarı sağlamaktadır. [59] Ayrıca Bora Kat tarafından yapılan anahtar kelime çıkartmada da önemli bir rol oynamaktadır [34].

2.3.2 Derin Öğrenme Algoritmaları

2.3.2.1 Konvolüsyonel Sinir Ağları

Genel olarak görüntü işlemede kullanılan bu method son zamanlarda dil işlemede de kullanılmaya başlanmıştır. Güler Arpaslan (2023) ve arkadaşı türkçe metin sınıflandırmasında kullanmıştır. Örüntü öğrenme ve gerçek zamanlı veri işleme alanında başarılı olan bu method ile %96.2 doğruluk oranına ulaşılmıştır [2]. José Pedro ve arkadaşları da CNN sayesinde gerçek zamanlı olarak belirli kalıplara bağlı hileleri tespit ederken kullanmıştır.[50]

2.3.3 Zaman Serisi Verileri ve Sinir Ağı Kullanımı

Yapılan çalışmada zaman serisini bulmak için sinir ağlarından faydalanılmıştır. Güler Alparslan ve arkadaşları ve İbrahim Yazıcı ve arkadaşları tarafından kullanılan bu method gerçek zamanlı veri kullanımı için yararlı bir araçtır.[69][2]

2.4 VERİ TOPLAMA, ÖN İŞLEME VE MOBİL/WEB TABANLI ENTEGRASYON

2.4.1 Veri Toplama ve Kullanıcı Etkileşimi

Mobil uygulamalarda veya oyunlarda veri elde etmek ve verileri verimli kullanabilmek açısından günümüzde olan dijital dönemde kullanıcı davranışını anlamak, kişiselleştirmek ve elde edilen veriler aracılığı ile hizmet kalitesini arttırmak adına önemli bir rol oynamaktadır. Bu süreçte sensörlerden, uygulamalarda bulunan etkileşimlerden veya ağ bağlantıları üzerinden gelen verilerin çeşitli yöntemlerle derlenip bir araya getirilmesi ve analiz edilmesi sayesinde elde edilir.

Mobil cihazlarda, verilerin toplanması uygulama içerisindeki analizlerden, GPS gibi sensörlerden, anketler veya kullanıcıların verdiği izinler doğrultusunda elde edilmektedir. Verilerin analiz edilme süresince ise ön işleme, veri madenciliği veya makine öğrenme algoritmaları kullanılarak veriler elde edilmektedir. Burak Delican'ın ve arkadaşlarının yapmış olduğu makale çalışmasında veri toplama araçları ve süreci hakkında bilgi verilmektedir. Lakin yapılmış olan çalışmada veri sayısı yetersiz olmasından dolayı uygulamaların geliştirilmesi kısıtlı kalmıştır[18].

Web uygulamalarında veri toplama ise kullanıcıların davranışlarını analiz ederek deneyimi verimli hale getirerek kişiselleştirmek ve web platformlarında sistemin performansını arttırmak amaçlanmakta olup yaygın olarak kullanılan bir çeşit yaklaşımdır. Web platformları kullanıcı ile Web uygulamaları arasındaki her türlü etkileşimi çeşitli oldukça fazla olan veri türü toplama imkanı bulunmaktadır.

Web platformlarında verinin elde edilme yöntemlerinden bazıları çerezler(Kullanıcı bilgilerini, geçmiş hareketlerini tarayıcıda kalıcı veya geçici olarak saklamak), tarayıcı API'leri (cihaz bilgisi, ekran çözünürlüğü gibi teknik bilgileri), JavaScript tabanlı izleme yöntemi(Google gibi arama motoru olan araçlar fare hareketleri ile detaylı bilgi toplama) gibi yöntemlerle veri elde edilebilmektedir. Veri analiz sürecinde ise veri temizleme veya kullanıcı segmentasyonu gibi alanlarla yapılmaktadır. Literatür çalışmasından ele alacak olursak Andreas Jungherr ve arkadaşları veri analizi yöntemini kullanarak çalışmalarını sürdürmüştür lakin sayısal veri eksikliği yüzünden verimli çalışma ortaya çıkartılamamıştır[33]. Ayrıca Bora Kat ve arkadaşlarının yapmış olduğu çalışmada ise Benzerlik analizi kullanılarak web üzerinde bulunan Türkçe metinler ile ilgili veri toplanması amaçlanmıştır lakin ne kadar çok Türkçe metin olursa olsun diğer diller de olan veri yoğunluğuna ulaşamadığı için veri sıkıntısı yaşamaktadır. Bunu da web sistemleri YZ ile entegre edilirse sürekli bir veri akışı sağlanması olanağı ortaya çıkması amaçlanmaktadır.[34]

2.4.2 Veri Ön İşleme Teknikleri

Ön işleme tekniklerine gelirsek makine öğrenme yöntemindeki başarısını doğrudan etkileyen önemli adımdır. Kaliteli ve uyumlu şekilde işlenmiş olan verileri modelin en doğru ve en güvenli sonuçları yaratmasını büyük ölçüde sağlamaktadır. Bu süreç içerisindeki en önemli iki aşama veri temizleme ve etiketleme ile veri artırma yöntemleridir.

Veri temizleme ve etiketleme yönteminde bütün makine öğrenme çalışmalarında tanımlanan problemin en doğru şekilde tanımlanması ve yüksek kaliteli veri ile başlatılmaktadır. Gerçe dünya verileri hatalı olabilmekte olup modelin eğitimini olumsuz etkilemesi ortaya çıkmaktadır. Veri temizleme yöntemi bu hataları düzenlemeyi, eksik verileri tamamlamayı ve ihtiyaç olmayan gereksiz veri yığınlarını ayıklamayı amaçlamaktadır. Etiketleme ise denetimlerde önemli role sahiptir. Eğer veriler yanlış etiketlenmiş olursa modelin öğrenimine olumsuz etilerde bulunmakta olur ve ayrıca güvenirliliği düşer. Bu yüzden uzman etiketleme ya da yarı denetimli öğrenme gibi belirli yöntemler kaliteyi arttırmak açısından büyük önem taşımaktadır.

Veri arttırmaya gelince özellikle derin öğrenme gibi veri yoğunluğunun bulunduğu ortamlarda modelin genelleme becerisini arttırmak için kullanılan bir yöntemdir. Eğer ki veri sayısı sınırlı olursa elde edinilmiş veriler üzerinde çeşitli dönüşümler yapılarak yeni örnekler (yani veri çoğaltma gibi) elde edilmesi amaçlanmaktadır. Örneğin görüntü işlemede oldukça sık kullanılmaktadır. En önemli faydası elde olan veriler sayesinde daha çok veri elde

edebilmektedir. Tim D. Smithies ve arkadaşlarının yapmış olduğu çalışmada 21588 veriye sahip olmasında eğer ki veri arttırımı yapılmak istenirse en az iki takına kadar verileri çoğaltabilme olanağı tanımaktadır[58]. Aynı zamanda Bora Kat ve arkadaşlarının yapmış olduğu çalışma içinde geçerli olmaktadır[34].

2.4.3 Mobil/Web Uygulama Entegrasyonu

2.4.3.1 Entegrasyon Yöntemleri

Hem mobilde hem de web programlamada uygulamalar entegre edilirken düşük gecikme , hızlı işlemciler ve optimize olmuş modeller gerekebilir. Bunun için Tim Smithies ve arkadaşları tarafından oluşturulan rastgele orman algoritması e-spor gibi resmi bir alanda kullanılacaktır. Bu sebeple düşük gecikmeli ve oyun sonu skorları hemen yansıtması beklenir.Ayrıca değişen skorlara ve oyuncu listelerine de optimize olması gereklidir.[58] Ek olarak José Pedro Pinto ve arkadaşları tarafından geliştirilen hile tespit programlarında aynı hile için kullanılan farklı programları tespit etmesi ve doğru bir şekilde bulması beklenir[50].

2.5 Algoritmaların Performans Değerlendirilmesi ve Karşılaşılan Zorluklar

2.5.1 Algoritma Performansı ve Başarı Kriterleri

2.5.1.1 Başarı Ölçütleri

Mobil alanında kullanılan yapay zeka algoritmaları geleneksel alana kıyasla başka faktörler de vardır. Bunlara :

- Zamanlama
- Enerji Tüketimi
- Karmaşıklık
- Kullanıcı Etkileşimi

örnek gösterilebilir. José Pedro Pinto ve arkadaşları %99.2 doğruluk oranına tutarlı ve optimize bir örüntü davranışı okuma modeli sayesinde ulaşmışlardır [50].Ali Hakan Işık ve arkadaşları ise kendi hibrit modelinde kendi içindeki fonksiyon ve indexlerin yardımıyla %99.9 oranına ulaşmışlardır. [22]

2.5.2 Zorluklar ve Sınırlamalar

2.5.2.1 Donanım Sınırlamaları

Mobil cihazlarda önemli olan cihaz belleği,enerji gücü ve işlem gücü gibi faktörler gerçek zamanla karar mekanizması ve yüksek doğruluk değerleri için dikkat edilmelidir. Bunu düzeltmek için bazı modeller de geliştirilmiştir. Örneğin, TensorFlow Lite ve ONNX gibi mobil uyumlu model formatları bu amaçla geliştirilmiştir.

2.5.2.2 Veri Gizliliği ve Güvenliği

Mobilde veri güvenliği hem etik hem de yasal açıdan büyük bir önem taşır.Mobilde uygulamaya giriş yaparken ve şifrelemede kullanıcıya özel hassas bilgiler barındırılabilir.Bu sebeple model eğitim aşamalarında yapılabilecek olan veri maskeleyme ve anonimleştirme aşamaları vardır.

2.6 Sonuçlar ve Gelecek Çalışmalar

2.6.1 Literatürün Genel Değerlendirmesi

Başka makale çalışmaları incelendiğinde eksiklik olarak modelin veri setini genişletip farklı verilerle tespit etmek verilebilir. Ayrıca çalışmaların güncel kalması için devamlı güncelleme ve optimizasyon kontrolü sık sık yapılmalıdır. Ayrıca mobil entegre etme aşaması da çalışma bittikten sonra önem taşıyan bir adımdır. Bu yüzden ihmal edilmemelidir.

2.6.2 Gelecek Araştırma Alanları

2.6.2.1 Yapay Zeka Tabanlı Uygulamalar

Yapay Zeka gün geçtikçe gelişen ve stabil olmayan bir alandır. Bu yüzden kullanılan algoritmaların da zamanla bakım yapılması ve kontrol edilmesi lazımdır. Optimizasyon aşaması eğer ki doğruluk sonucu yüksek isteniyorsa yapılması gereken bir adımdır.

2.6.2.2 Daha Güçlü Derin Öğrenme Modelleri

Derin Öğrenme ön işleme adımında gerekli öznitelikleri hesaba katarak modelin geliştirmesi gereken bir aşamadır. Mobil uygulamalarda gerçek zamanlı sınıflandırma gerektirir. Bu yüzden model budama ve sayısal indirgeme aşamalarında ince ayarlar yapılarak bu alanda performans artırılabilir.

3 YÖNTEM

3.1 Veri Seti

Veriler, yapılacak uygulamada normal şartlar altında kullanıcıların yaptıkları aracılığı ile zamanla elde edilmek amaçlanmaktadır, lakin başlangıç olarak veri elde edebilmek ve kullanıcılara başlangıçta bile en iyi performansı sağlayabilmek için gerekli olan veri kaynağı [Github Veri Seti](#)'dir. Bu sayede hem güncel veriler hem de oldukça veri çokluğu sayesinde hazırlanan mobil oyunda kusursuz bir şekilde kullanıcıya sunulmaktadır. Ek olarak bu verilere kullanıcıdan alınan veriler de eklenebilir. Veri tipinde veriler anlamlılık ve anlamsızlık olarak sınıflandırılmıştır. Uygulamada veriler için kaynak kullanıcıdan girilen veridir. Karmaşık, anlamsız olan kelime grupları genellikle kelime listelerinde yer almayan anlamsız oluşturulan harfleri kullanıcı tarafından birleştirerek girdiği kelimeyi anlamlı bir takım kelime haline getirmeye çalışmaktadır. Bu yüzden Doğal Dil İşleme yöntemleri bu uygulamada sık kullanılacak yöntemlerden biridir. Veri örnek sayısı toplamda 15000 adet veri seti içeren 6000 anlamsız kelime, 5000 anlamlıya yakın olan kelimeler ve 4000 anlamlı kelimeler olarak ayrılır. Bu veri seti arttırmakta mümkündür. Rakamlar ve noktalama işaretleri kullanarak ve elde olan anlamsız kelimenin harflerinin yerlerini değiştirerek bu sayıyı anlamsız 20000, anlamlıya benzeyen 15000 ve anlamlı 15000 toplamda 50000 ne kadar kolaylıkla çıkartabilir, hatta daha fazla bile artırılabilir.

3.2 Öznitelik Çıkarma

Mobil uygulamada kullanıcının girdiği veriler anlamlı, anlamsız ve anlamlıya yakın olarak 3 gruba ayrılmakla birlikte oluşan kelime grupları YZ tarafından işlenebilmesi için sayısal verilere dönüştürülmektedir. Kullanıcının girdiği kelime gruplarının hangi kelimenin kaç harften oluştuğu (uzunluğu), sesli harf sayısının kullanımı ki sesli harf kullanımı anlamlı kelime oluşturmak için önemlidir. Ayrıca harf tekrarlarının ne kadar olduğu ki kelime oluşturmada bu sıkça görülür. Örneğin "MAKARNA" kelimesi ele alınırsa 3 kere "A" harfinin tekrar edildiği görülmektedir. Bu durum hem anlamlı hem de anlamsız bütün kelime gruplarında görülmesi mümkündür. Kelime başlangıç harfinin de sınıflandırmaya etkisi vardır çünkü sessiz harflerin fazlalığı sayesinde sessiz harfle başlayan kelimeler çok daha fazladır. YZ'ye göre sesli harfle başlayan kelimelerin çoğunlukla anlamsız kelime olduğunu öne sürmektedir. Bu sayede ilk sesli ise kelimenin anlamlı mı yoksa anlamsız mı olduğunu anlamaktadır. Bir de karakter çeşitliliği de önemlidir. Örneğin "bbbb" diye bir kelime grubu verilirse harf çeşitliliği 1 çeşit olmakta ama "asaftl" kelime grubunda ise 5 çeşit harf çeşitliliği olmaktadır.[37, 44, 51]

Bunlara ilaveten karakter dışı olan rakamlar veya noktalama işaretleri kullanılarak çeşitliliği arttırmak amaçlanmaktadır. Ardından N-gram özelliği gelmektedir. N-gram karakter dizilerinin alt örüntülerinin uygulama modelinin düzenli olarak öğrenmesini sağlar. Örneğin "dolap" kelimesinin 2-gram'larına bakarsak "do", "ol", "la", "ap" gibi yada 3-gram'larına bakarsak "dol", "ola", "lap" gibi belirli bir N değeri kullanıp alt gruplarına ayırmasına olanak tanımaktadır. Anlamsız diziler ile anlamlı diziler arasında ayırım yapmasına olanak tanımaktadır. Bu çalışmaya büyük ölçüde önem katmaktadır.[25, 56]

TF(Terim Frekansı)-TBF(Ters Belge Frekansı) değeri ise bütün dizelerde sık sık geçen kelime gruplarının önemini ölçer. Çoğunlukla anlamlıya yakın dizelerde bazıları tekrar edebilmektedir. Yani bir kullanıcının girdiği kelime kullanıcı tarafından tekrar girilebilir ve ne kadar önemli olduğu ölçülmektedir. Amaç girilen harf dizisindeki gereken parçaların ne kadar ayırt edici olduğunu bulmaktır. Örneğin N-gram.

TF (TERİM FREKANSI)

$$TF(t) = \frac{\text{Belgedeki terim sayısı}}{\text{Belgedeki toplam kelime sayısı}} \quad (1)$$

TBF(TERS BELGE FREKANSI)

$$TBF(t) = \log \left(\frac{N}{1 + df(t)} \right) \quad (2)$$

N: Toplam belge sayısı

df(t): Terimi içeren belge sayısı

TF-TBF

$$TF-TBF(t) = TF(t) \times TBF(t) \quad (3)$$

Bir kelimenin kaç kez tekrarlandığını gösteren denklem olarak 1. denklem, 2. denklemde ise kelimenin kaç belgede geçmekte olduğunu ve hangi kelimelerin daha bilindik olduğunu ölçer. Son olarak 3. denklemde herhangi bir kelimenin belge içerisinde ne kadar önem taşıdığını ölçmek için kullanılan TF-TBF çarpımı yer almaktadır. Buradaki amaç uygulama için kullanıcıların sık kullandıkları kelimelerde düşük puan daha az sıklıkta kullanılan kelimeler için yüksek puan verilmesi amaçlanmaktadır.

Benzerlik skoru da kullanıcının girdiği kelime ile bulunması amaçlanan kelimenin arasında oluşan benzerlik oranını göstermektedir. Benzerlik skoru 3 yolla hesaplanır.

- 1) Levenshtein Distance: Kelimeleri birbirine dönüştürmek için ekleme, silme ve değiştirme işlemleri kullanılır.[35]
- 2) Jaccard Benzerliği: Kelimelerin karakter kümelerindeki kesişimi N-gram yolu ile hesaplaması[4]
- 3) Vektör Tabanlı: Kelimeleri TF-TBF'ye dönüştürüp aralarındaki açısız benzerlikleri hesaplanır.[67]

Örneğin "dolap" kelimesi istenirse kullanıcının girdiği değerlerin kelimelere göre yüzdelik göstermektedir.

- "bokel": benzerlik skoru = 0.20
- "dolek": benzerlik skoru = 0.78
- "dolap": benzerlik skoru = 0.92

Özetle benzerlik skoru kullanıcının girdiği değerlerin gerçek kelimeye olan yakınlığının doğruluğunu sayısal olarak vermektedir.[53]

Son olarak yazım deseni hangi karakterin içerdiğini ve nasıl bir sıralamaya sahip olduğunu analiz eder. Sadece karakter mi yoksa

sembollerinde içerip içermediğini kontrol etmektedir. Aslında burada şu soruların yanıtını da alabiliriz:

- Kelime grubunun şifre olma ihtimali var mı?
- Gelişi güzel mi yazılmış?
- Bu kelime grubu bilinçli bir şekilde yapılandırılmış olabilir mi?

Özetle hangi karakterlerin kullanılıp kullanılmadığını analiz edip gelişi güzel mi, düzenli mi yoksa şifre mi olduğunu anlamaya yardımcı olmaktadır. Yazılım deseninde YZ şifreli kelime gruplarını bulmasında olanak tanır. [45]

3.3 Ön işleme

Veri temizleme, verilerde bulunan gereksiz, hatalı öğeleri tespit ederek düzeltme veya tamamen silme işlemidir. YZ ve makine öğrenme yönteminde eğitilecek veri ne kadar kusursuz ise doğruluğu bir o kadar yüksek olmaktadır. Eğer hatalı veriler temizlenmez ise model yanlış örnekler ile eğitilir bu da doğruluğunda hata olmasına neden olur. Modelde hatalı verileri ortadan kaldırmak için şunlar yapılmaktadır[40, 46]:

- 1) Herhangi bir değer içermeyen boş olarak bırakılan karakter girişleri silinir.
- 2) Çok kısa veya çok uzun kelimeler yani belli bir aralığın dışına çıkan kelimeler silinir.
- 3) Harflerin, rakamların ve belli sembollerin dışında geçersiz olan karakterlerin silinmesi.
- 4) Kullanıcının aynı kelime grubunu tekrar tekrar kullanması ile oluşan gereksiz verinin silinmesi.
- 5) Verinin yapısından sapan diziler YZ ile filtreleme sonucunda silinir.

Ön işlemede ölçeklemede yapılmaktadır. Yapılan ölçeklemede verileri sayısal olarak farklı büyüklüklerden ayırarak aynı ölçek aralığına getirmesidir. Ayrıca makine ve derin öğrenme modellerinde farklı öznitelik aralığı oluyor ise modelin öğrenmesi bozulabilmektedir. Bu sebepten ötürü her şeyi ortak bir aralığa getirmek model için daha verimli olması amaçlanmaktadır.

Kullanıcıdan girilen kelime gruplarının öznitelik değerleri sayısal veri içerdiğinden modelde gereğinden fazla yük olmasına neden olabilir. Bunu önlemek için gereken min-max normalizasyonu ile [0-1] aralığına getirmek amaçlanmaktadır. [16, 57] Bunun formülü ise:

MIN-MAX NORMALİZASYONU

$$X_{ölçekli} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

- X : Orijinal değer
- X_{\min} : Özelliğin minimum değeri
- X_{\max} : Özelliğin maksimum değeri
- X_{scaled} : Normalize edilmiş değer

Son olarak veri ayırımında ise 50000 veriden oluşan modelde verinin %80'i eğitim ve %20'si test olmaktadır. Bu denli verilerin daha net doğrulanması için K-Fold çapraz doğrulama yöntemini kullanılmaktadır. K-Fold değeri 5 olursa daha hızlı ve iş yükü az olacaktır, 10 olarak alınırsa daha doğru sonuçlar elde edilecektir. Bu model için ise K-Fold değeri 10 olarak almak amaçlanmaktadır; lakin iki şekilde de doğruluk değerlerini kontrol edip deneysel sonuçlar kısmına ilave edilmektedir.

3.4 Öznitelik Seçme

3.4.1 Temel Bileşenler Analizi(TBA)

TBA yüksek boyutlu öznitelik uzayını indirgeme yöntemidir. Verideki varyansların en çok olduğu kısımları korur. Yani anlamlı bilgileri taşıyan doğrulukları bulur, az sayıda boyutun içerisinde bilginin orijinal halini korumaktadır. Bu modelde ise TF-TBF, kelime uzunluğu, n-gram sıklığı gibi sayısal değer taşıyan öznitelikler üzerinde TBA ile analiz yapılmakta olup ilk 10 bileşende toplam varyansın %95 ve üzerini korumaktadır. Bu sayede veri boyutu azaltılıp bilgiler korunmuştur. [1, 23]

3.4.2 Öznitelik Önem Skorlarına Dayalı Seçim(Rastgele orman ile)

Rastgele orman kullanılarak bütün öznitelik için model içerisinde etkisi analiz edilmektedir. Bu sayede her niteliğin sınıflandırmaya olan başarısına verdiği katkıyı hesaplamaktadır. Rastgele orman eğitilerek her nitelik hakkında skorlar elde edilmektedir. Bu yüzden en yüksek skora sahip 7 öznitelik seçilip önemsiz düşük skorlarda bulunanlar çıkartılmıştır. [32]

3.5 Model Seçimi

3.5.1 Makine Öğrenme(MÖ)

Makine öğrenme(MÖ), sistemlerin veri üzerinden öğrenerek tahmin ve sınıflandırma yapmasına olanak tanıyan bir YZ alanıdır. Özellikler sınıflandırmada algoritmalar geçmiş verileri inceleyerek öğrenip gelecekteki verileri bunların üzerine entegre etmeyi öğrenir. MÖ algoritmalarına bakılacak olunursa gözetimli öğrenme, gözetimsiz öğrenme ve pekiştirmeli öğrenme bulunmaktadır. B çalışmada gözetimli öğrenme yani girdi çıktı işlemleri bulunmaktadır. Önemli noktaları anlamsız, anlamlıya yakın ve şifreli dizileri ayırt etmek için karmaşık yapıları öğrenmektedir. Ayrıca uygulamada geçek zamanlı olarak tahmin ve karar alma işlemleri kullanılmaktadır.[70]

3.5.1.1 Destek Vektör Makineleri(DVM)

DVM, veriler arasında bulunan en doğru ayırıcı sınırı bulmaya çalışmaktadır. Ayrıca çok boyutlu öznitelikler içerisinde yüksek doğruluk imkanı verir. Veri seti küçük olsa bile oldukça güçlü performans sergiler. Avantajları ise genelleme yeteneği yüksek ve aşırı öğrenmeye karşı dayanıklıdır. Bu çalışma için ise TF-TBF gibi vektör temelli olan verilerde oldukça iyi çalışır. Anlamsız ve anlamlıya yakın kelime gruplarını net bir şekilde ayırım gücüne sahiptir.[12, 36] Formülü:

$$\min \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{öyle ki} \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$$

3.5.1.1.1 Karar Ağaçları

Karar ağaçları, verileri ağaç yapısına uygun şekilde dallandırarak karar kuralları oluşturur. Her düğümde bir öz nitelik üzerinden karar verip veriyi kademeli olarak sınıflandırmaktadır. Avantajlarından bir kaçısı ise hızlı ve kolay uygulanır olması ile beraber eğitim ve test sürelerinin düşük olmasını sağlamaktadır. Yani düşük işlem gücü sayesinde hızlı sonuç verir buda mobil uygulama için avantajlı olmaktadır. Uygulamaya bu avantajla beraber kelime uzunluğunu belli bir aralığa almasına olanak tanır. Ayrıca anlaşılabilir bir sınıflandırma mantığı sunmaktadır. Karar ağaçları veri setini dallara ayırırken genellikle Bilgi Kazancı veya Gini İndeksi gibi ölçütleri kullanılmaktadır. [15]

BİLGİ KAZANCI

$$IG(D, A) = Entropy(D) - \sum_{v \in \text{Values}(A)} \frac{|D_v|}{|D|} \cdot Entropy(D_v)$$

- p_i : Sınıf i 'nin veri kümesindeki oranı (olasılığı)
- c : Toplam sınıf sayısı

GINI İNDEKSİ

$$Gini(D) = 1 - \sum_{i=1}^c p_i^2$$

- p_i : Sınıf i 'nin örneklerinin, toplam örnekler içindeki oranı
- c : Toplam sınıf sayısı

3.5.1.1.2 Rastgele Orman

Birden fazla karar ağcının özniteliklerle eğitilip birleştirilmesi ile oluşan topluluk yöntemidir. Rastgele ormanda tıpkı DVM gibi aşırı öğrenmeye dayanıklıdır. Ayrıca doğruluk oranı yüksek olması ile birlikte kararlıdır. Bu model için anlamsız kelime gruplarının dahil güçlü bir sınıflandırma yapar. Ayrıca bir diğer avantajı da öznitelik skoru veriyor olmasıdır. [8]

$$\hat{y} = \text{mod}\{h_1(x), h_2(x), \dots, h_T(x)\}$$

- \hat{y} : tahmin edilen çıktı.
- $h_t(x)$: t -inci sınıflandırıcının x girdisi için verdiği tahmin sonucu.
- T : Toplam sınıflandırıcı sayısı.
- mod : En sık çıkan değeri döndürür.

3.5.1.1.3 Naive Bayes (NB)

Bayes teoremi temelli, sınıfların olasılık dağılımını yaparak hesaplar ve sınıflandırmayı çok hafif ve hızlı yapan bir algoritmadır. Avantajlarında ise eğitim ve test süresince hızlıdır. Ayrıca metin sınıflandırmada klasik ve etkili bir yöntem olmaktadır. Çalışma için TF-TBF ile çıkarılmış n -gram temelli özniteliklerle çok iyi çalışmaktadır. [26]

$$P(C | X) = \frac{P(X | C) \cdot P(C)}{P(X)}$$

- $P(C | X)$: X gözlemi verildiğinde C sınıfına ait olma olasılığı.
- $P(X | C)$: C sınıfı verildiğinde X 'in gözlenme olasılığı.
- $P(C)$: C sınıfının önsel (prior) olasılığı.
- $P(X)$: X 'in gözlenme olasılığı.

3.5.1.1.4 K-En Yakın Komşu

Veri noktası, öznitelik uzayındaki en yakın k adet komşusuna göre sınıflandırılmaktadır. Avantajlarında ise parametresiz olup eğitim süreci yoktur. Yani örneklerle karşı esnek ve doğal sonuçlar vermektedir ve çok sınıflı problemler için uygundur. Bu çalışmaya uygun olmasının sebebi eğitimin olmaması ile veriye doğrudan bakar bu yüzden kolay uygulanabilir. Özellikle benzerlik skoru ve karakter örüntülerine göre anlamlıysa benzeyen kelimeleri başarılı şekilde bulabilmektedir. [27, 54] K-En yakın komşu için gerekli kullanılacak formüller ise:

MESAFE HESABI (ÖKLİD MESAFESİ)

Veri noktası aşağıdaki gibi tanımlanır:

$$x = (x_1, x_2, \dots, x_n)$$

Komşu veri noktası ise:

$$x_i = (x_{i1}, x_{i2}, \dots, x_{in})$$

Bu iki nokta arasındaki Öklid mesafe şu şekilde hesaplanır:

$$d(x, x_i) = \sqrt{\sum_{j=1}^n (x_j - x_{ij})^2}$$

- x : Yeni gelen veri noktasıdır.
- x_i : Eğitim veri kümesinde bulunan komşu veri noktasıdır.
- x_j : Yeni veri noktasının j -inci özelliğidir.
- x_{ij} : Komşu x_i noktasının j -inci özelliğidir.
- n : Özellik (boyut) sayısıdır.
- $d(x, x_i)$: İki nokta arasındaki Öklid mesafesidir.

SINIF TAHMİNİ

$$\hat{y} = \text{mod}\{y_i | x_i \in N_k(x)\}$$

- \hat{y} : Tahmin edilen sınıf etiketidir.
- y_i : Komşu veri noktalarının sınıf etiketleridir.
- x_i : Eğitim kümesindeki veri noktalarıdır.
- $N_k(x)$: x veri noktası için en yakın k komşu kümesidir.
- $\text{mod}\{\cdot\}$: En sık tekrar eden (mod) sınıfı temsil eder.

3.5.2 Derin Öğrenme(DÖ)

Derin öğrenme(DÖ), insan beynindeki sinir ağlarına benzer bir şekilde çalışan çok katmanlı yapay sinir ağları ile verilerden örüntü çıkarmayı öğrenmekte olan bir çeşit YZ dalıdır. Makine öğrenmeden farkı olarak elle çıkartılan verilere değilde ham verilerden doğrudan öğrenebilmektedir. Karmaşık ilişkilere daha uyum sağlamaktadır ve büyük veri ile çalışıkça doğruluğu artmaktadır. Önemli yararlarından biri olan doğal dil işleme alanında büyük önem taşımaktadır. Özellikle dilsel ve yapısal olmayan verileri analiz etmede büyük ölçüde önemlidir. [43]

3.5.2.1 Yapay Sinir Ağı(YSA)

Giriş katmanı, gizli katman ve çıkış katmanından oluşan basit bir çok katmanlı sinir ağıdır. Giriş katmanlarında ilk olarak öznitelik sayısı kadar nöron içermektedir. Gizli katmanda ise gizli katman 1 için 128 nöron ve gizli katman 2 için ise 64 nöron bulunmaktadır. son olarak çıkış katmanında ise 3 nöron bulunmaktadır. Ayrıca avantajları eğitim süresinin kısa olması, sayısal özniteliklerle uyumlu olması ve küçük-orta boyutlu veri setlerinde başarılı olmasıdır. Birde aktivasyon fonksiyonlarında ReLU ve Softmax kullanılmaktadır. Bu çalışmada olduğu gibi sayısal olarak elde edilen karakter verilerini analiz eden problemlerde hızlı ve doğru sonuçlar üretmektedir. Özellikle ilk testlerde temel performans karşılaştırması için referans model olarak kullanılabilir. [28]

3.5.2.2 Evrişimsel Sinir Ağı(ESA)

ESA, ilk olarak görüntü işleme için geliştirilmiş olsa da, karakter dizileri gibi sıralı ve lokal örüntüler içeren verilerde de oldukça başarılıdır. 1-boyutlu evrişim katmanlarıyla, harf dizileri üzerinde n -gram benzeri yapıları otomatik olarak öğrenebilmektedir. Katman yapısına gelindiğinde Giriş katmanı vektörleştirilmiş harf dizisi, evrişim katmanı 1 64 filtre ile ReLU aktivasyonu ile,

MaxPooling1D, evrişim katmanı 2 128 filtre ile ReLU aktivasyonu ile, Flatten, dense(tam bağlı) 64 nöronlu ReLU aktivasyonu ile ve son olarak çıkış katmanı 3 nöronlu Softmax aktivasyonu ile oluşmaktadır. Avantajlarında ise N-gram yapılar gibi lokal örüntüler otomatik keşfedilir, eğitim süresi hızlı ve düşük riskli olmaktadır. Bu çalışma için harf kombinasyonlarını algılamada çok başarılıdır.[65]

3.5.2.3 Tekrarlayan Sinir Ağı(TSA)

TSA, sıralı veriler üzerinde çalışmak için geliştirilmiştir. Her bir adımda, sadece o anki girdiyi değil, önceki adımlardan gelen bilgiyi de kullanır. Bu sayede harf dizilerindeki bağlamsal ilişkileri öğrenebilmektedir. Katman yapısı girişte harf dizisi, TSA katmanı 64 nöron ile tanh aktivasyonu, dropout, desen katmanı 64 nöron ReLU aktivasyonu ile ve son olarak çıkışta 3 nöron Softmax aktivasyonu ile sonlanmaktadır. Avantajları sıralı yapıları öğrenme becerisi yüksektir ve düşük parametre sayısı sayesinde hızlı çalışmaktadır. Bu çalışma için karakterlerin sırası, anlamlı/anlamsız ayrımında önemlidir. TSA, bu sırayı dikkate alarak tahmin yapmaktadır ve kısa harf dizilerinde oldukça etkilidirler. [6]

3.5.2.4 Uzun Kısa Süreli Bellek (UKSB)

UKSB, klasik TSA'nın zamanla unutma sorununu çözen bir mimaridir. "Kapı mekanizmaları" sayesinde hangi bilgiyi saklayacağını, hangisini unutacağını kendisi öğrenir. Bu sayede uzun karakter dizilerinde bile bağlam korunabilmektedir. Katman yapısında ilk olarak giriş vektörleştirilmiş karakter dizisi, UKSB katmanı 128 birim geri beslemeli olarak tanh ve sigmoid aktivasyonları, Dropout, dense katman 64 nöron ile ReLU aktivasyonu ve son olarak çıkış 3 nöronlu Softmax aktivasyonu ile oluşmaktadır. Avantajları uzun dizilerde bile önceki karakterleri hatırlayabilir ve "Anlamlıya benzeyen" ya da şifreli dizileri tespit etmede oldukça güçlüdür. Çalışmaya göre UKSB'de bazı kelimelerde anlam sonuna doğru ortaya çıkmaktadır. UKSB bu uzun ilişkileri öğrenerek daha başarılı tahminler yapabilmektedir.[38]

3.5.2.5 Transformatör Kodlayıcı(Mimarisi)(TK)(BERT-benzeri mimari)

TK, sıralı verileri paralel olarak işler. "Dikkat mekanizması" sayesinde her karakterin dizideki diğer karakterlerle olan ilişkisini öğrenmekte olup bu bağlam bilgisini en doğru şekilde yakalayan mimaridir. Katman yapısı da Embedding katmanı, Pozisyonel kodlama, Multi-head attention, Feedforward layer, Dropout ve çıkışta dense 3 nöron Softmax aktivasyonu ile kurulmuştur. Ayrıca ReLU ve GeLU aktivasyonları da bulunmaktadır. Avantajları ise karakterler arası tüm bağlamları aynı anda öğrenebilir, büyük veri ile çalışıldığında çok güçlü performans gösterir ve modern dil işleme uygulamalarında (BERT, GPT) temel mimaridir. Çalışmaya olan katkısı çalışma karakter dizileri üzerinden anlam çıkarmaya çalıştığı için dikkat mekanizması sayesinde karmaşık örüntüleri çok doğru şekilde öğrenebilir.[10, 31]

Bu çalışmada uygulanan makine ve derin öğrenme algoritmaları, karakter dizilerinin anlamlı, anlamsız ya da anlamlıya yakın olup olmadığını sınıflandırma problemine uygun olarak seçilmiştir. Makine öğrenmede DVM, Karar Ağaçları ve Rastgele Orman yorumlanmasına, Naive Bayes düşük kaynak gereksinimine K-En ise benzer diziler üzerinde doğru tahmin yapabilme yeteneği için tercih edilmiştir. Derin öğrenme de ise YSA sayısal öznitelik öğrenme, ESA harf örüntülerini tutma, TSA dizideki sıralı yapıyı

modellemeye, UKSM uzun dizilerde bağlamı koruma ve Transformatör ise dikkat mekanizmasıyla tüm karakterler arası ilişkiyi aynı anda analiz edebilme gücü sahip olduğundan tercih edilmişlerdir. Bunları seçmenin sebebi hem veri seti hem de mobil uygulama uygunluğu göz önüne alınarak seçilmektedirler. [5, 7]

Çalışmada geliştirilen modelde hibrit model olarak TF-IDF yöntemi sayesinde önemli kelimeler belirlenip işaretlenmiştir. Sonra bu işaretlenen kelimeler LSTM modeli kullanılarak birleştirilmiştir.

3.6 Model Eğitimi

3.6.1 Eğitim Süreci

Modelde eğitim verisi oluşturmak için başlıca 3 method kullanılmaktadır:

- **Girdi ve Çıktı Seçimi** Veri setinde bulunan her n-gram dizisi içinden son kelime çıktısı ve geri kalanı girdi olacak şekilde alınır.
- **Pad Aşamaları (Pad Sequences)** Oluşturulan dizinlerin hepsinin aynı uzunlukta olması için başına 0 eklenir (padding).
- **One-Hot Encoding (Tek-Sıcak Kodlama)** Elde edilen çıktı verileriyle kelime sayısı ile aynı boyutta bir vektör oluşturulur.

Model 200 epoch seviyesi kullanarak eğitilmiştir ve sonunda modelin doğruluk değeri gösterilmiştir.

3.6.2 Hiperparametre Ayarı

Öğrenme Oranı: Modelde Keras'ın Adam Optimizer'ı kullanılmıştır. Varsayılan öğrenme oranı olarak da 0.001'e ayarlanılmıştır. **Batch Size:** Modelde batch size değeri de varsayılan olarak 32 alınmıştır. İleride yapılacak geliştirmelerle bu değerde değişiklikler yapılabilir. **Epoch Sayısı:** Modelde epoch 200 değeri üzerinde eğitilmiştir. **Embedding(Gömülü) Boyutu:** Bu değer 10 olarak belirlenip kullanılmıştır. **LSTM Katmanı:** Bu katmanda 100 nöron değeri kullanılmıştır Modelin sonucunu değiştirmek için ekstra olarak Grid Search ve Bayesian Optimizasyon gibi hiperparametre teknikleri ek olarak ilave edilebilir. Bu teknikler denendikten sonra modelin son sonucunda farklı değerler bulunabilir.[49]

3.6.3 Aşırı Öğrenme (Overfitting) ve Yetersiz Öğrenme (Underfitting)

Modelin Aşırı Öğrenme ile başa çıkabilmesi için eğitim ve test verisinin düzgün bir şekilde ayrılıp bu aşamalarda doğruluk değerinin düşmemesine dikkat edilmiştir. Yetersiz Öğrenme sorunu çıkmaması için de veri setinin fazla ve dengeli olmasına dikkat edilmiştir. Ayrıca burada da model aşama aşama detaylıca bir şekilde incelenmiştir.

3.6.4 Değerlendirme Metrikleri

Başarı Ölçütleri ve Formülleri Doğruluk (Accuracy):

- **Doğruluk (Accuracy):**

$$\text{Doğruluk} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

- **Duyarlılık (Recall):**

$$\text{Duyarlılık} = \frac{TP}{TP + FN} \quad (5)$$

- **Özgüllük (Specificity):**

$$\text{Özgüllük} = \frac{TN}{TN + FP} \quad (6)$$

• **F1-Skoru:**

$$F1 = 2 \times \frac{\text{Keskinlik} \times \text{Duyarlılık}}{\text{Keskinlik} + \text{Duyarlılık}} \quad (7)$$

- **AUC-ROC:** Modelin kendi içinde sınıflandırma başarısı, ROC eğrisi altında kalan alan sayesinde ölçülür. Bu değer ne kadar 1'e yaklaşırsa o kadar iyi ne kadar 1'den uzaklaşırsa o kadar kötüdür.

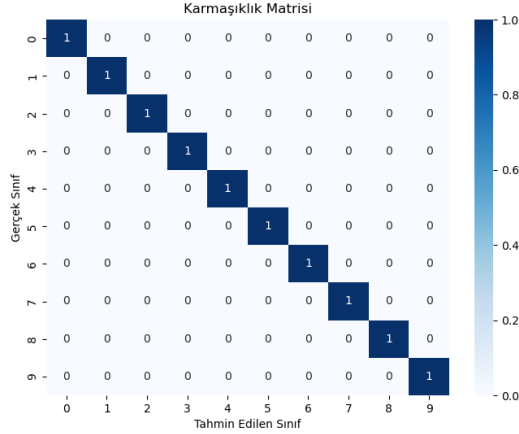
3.6.4.1 Karmaşıklık Tablosu

Karmaşıklık tablosu, modelden elde edilen tahminlerle gerçek değerlerin karşılaştırıldığı bir tablodur.[17, 52] İçerisinde:

- TP (True Positive-Gerçek Pozitif): Sonucu doğru bulunan pozitif değerler
- FP (False Positive-Yanlış Pozitif): Yanlış olarak bulunan pozitif değerler
- TN (True Negative-Doğru Negatif): Doğru olarak tahmin edilen negatif değerler
- FN (False Negative-Yanlış Negatif): Yanlış olarak tahmin edilen negatif değerler

Ayrıca modelin sonucu olarak çıkan karmaşıklık tablosu da aşağıdaki Figür 1'de verilmiştir:

Fig. 1: Karmaşıklık Tablosu



Sonuç olarak elde edilen model bir hibrit modeldir ve özgünlüğü de buradan gelmektedir. Aynı anda hem TF-IDF klasik yöntemi hem de LSTM derin öğrenme modelini kendi içinde kullanarak en iyi sonuç elde edilmiştir.

3.7 Mobil Uygulama Geliştirme

3.7.1 Uygulamanın Genel Yapısı

Uygulamanın genel işlevi kullanıcıya verilen anlamsız kelime dizinlerinden anlamlı kelime dizinleri oluşturmaktır. Ayrıca bunu yaparken de kullanıcıdan alınan geri dönüşlere dayanarak kendini eğiten bir yapay zeka da geliştirilmesi amaçlanmaktadır. Kullanıcı arayüzü içinse herkese göre bir arayüz kurmayı amaçlandı. Renkler ve şekiller kullanıcı dostu olmalı ve göze güzel görünmesi lazım. Uygulamada kullanıcılardan manuel giriş alarak en anlamlı kelime dizinlerini bulmaları istenecektir.[13]

3.7.2 Mobil Platform Seçimi

Seçilen mobil uygulama platformu Android/iOS'tur. Bunun sebebi de iki platformun da kendi içinde avantajları bulunmaktadır. Android Türkiye'de daha fazla kullanılan bir seçenektir ve birçok cihaza da uyum sağlar.Öte yandan IOS ise uygulamalarına daha güvenli bir erişim imkanı sunar.

3.7.3 Derin Öğrenme/Makine Öğrenme Modellerinin Entegrasyonu

Modelde kullanılacak olan makine ve derin öğrenme uygulamaları için başlıca Tensorflow Lite olmak üzere KNN,DVM,Karar Ağaçları,Rastgele Orman,NB ve derin öğrenme için de YSA,ESA,TSA,UKSB,TK arasından performans değerlendirmesi ve modele uyumluluk açısından hepsi teker teker denenip arasından en iyi performans sonuçları elde edilen modeller içinden uygulama geliştirilecektir. [63]

3.7.4 Performans Optimizasyonu ve Kullanıcı Geri Bildirimlerine Dayalı İyileştirmeler

Uygulamanın son halinde optimizasyonu iyileştirmek istersek birkaç farklı model üzerinden bunlar yapılabilir: Tensorflow Lite ve ONNX bu programlardan birkaçıdır.Ayrıca kullanıcı dostu bir arayüz yaparken bu performans sorunlarını da ele alıp geliştirme yapılabilir.Farklı cihazlarda uyumluluk ve performans için de kullanıcı geri dönüşlerinden yola çıkarak uygulamaya yeni güncellemeler yapılabilir.

4 DENEYSEL SONUÇ

4.1 Veri Seti ve Ön İşleme

4.1.1 Veri Seti Tanımlaması

4.1.1.1 Veri Setinin İçeriği ve Kaynağı

Uygulamada kullanılan veri seti açık kaynak kodunu temel almaktadır. Veri İngilizce dilinde bulunan kelimelerin bir listesini içerir. Toplam 370.106 satırdan oluşmaktadır. Github veri setinin kaynak adresi şu şekildedir: [Github Veri Seti](#). Uygulama 3-7 harfli İngilizce kelimelerden oluşmaktadır ve alfabetik karakterler içermektedir. Kullanılan kelimelerde filtreleme uygulanmaktadır. **Özellikler Tablosu** Aşağıda makalede kullanılan modelin özellik ve türleri tablosu verilmektedir:

| Özellik Adı | Açıklama | Türü |
|----------------|---|-----------------|
| Letter_Pos_1 | 1. harfin one-hot kodlaması (a-z, 26 boyut) | Kategorik (0/1) |
| Letter_Pos_2 | 2. harfin one-hot kodlaması (a-z, 26 boyut) | Kategorik (0/1) |
| Letter_Pos_3 | 3. harfin one-hot kodlaması (a-z, 26 boyut) | Kategorik (0/1) |
| Letter_Pos_4 | 4. harfin one-hot kodlaması (a-z, 26 boyut) | Kategorik (0/1) |
| Letter_Pos_5 | 5. harfin one-hot kodlaması (a-z, 26 boyut) | Kategorik (0/1) |
| Letter_Count_a | 'a' harfi kelime kaç kez geçiyor | Sayısal (0-5) |
| Letter_Count_b | 'b' harfi kelime kaç kez geçiyor | Sayısal (0-5) |
| ⋮ | ⋮ | ⋮ |
| Letter_Count_z | 'z' harfi kelime kaç kez geçiyor | Sayısal (0-5) |

TABLE II: Kullanılan Özellikler ve Türleri

Not: Toplamda kullanılan özellik sayısı 5x26=130 one-hot kod ve 26 harf sayısı olarak 156 özellik bulunmaktadır.

4.1.2 Veri Temizleme ve Ön İşleme

Veri setinde kullanılan 4 ön işleme adımı bunlardır:

- **Filtreleme:** Alfabede bulunan sadece 5 harfli kelimeler arasından seçildi.

| Özellik | Değer |
|---------------------|--|
| Toplam örnek sayısı | 15,921 (5 harfli kelime) |
| Sınıf sayısı | 10 (dummy target, gerçek Wordle'da yok) |
| Özellik sayısı | 156 |
| Özellik türleri | Kategorik (one-hot), Sayısal (harf sayısı) |

TABLE III: Veri Setinin Temel Özellikleri

- **Eksik Veri:** Veri setinde eksik kelime bulunmamaktadır. Bunun sebebi satırların hepsinin kelimelerden oluşmasıdır. Filtreleme ön adımı sayesinde eksikler elenir.
- **Aykırı Değer Yönetimi:** Aykırı değer bulunmamaktadır. Çünkü kelimeler sabit uzunlukta olmakta olup harflerden oluşmuştur.
- **Özellik Mühendisliği:** Her kelimenin her pozisyonu için one-hot kodlama (a'dan z'ye kadar olan tüm harfler ve 5 tane pozisyon) ve her harf için (a'dan z'ye kadar toplam 26 tane harf) olarak 156 tane özellik oluşturuldu.

| Özellik | Orijinal | Dönüştürülmüş |
|----------------|----------------------|--------------------|
| Kelime | String (ör: "apple") | 156 boyutlu vektör |
| Harf Pozisyonu | Karakter | One-hot (26 boyut) |
| Harf Sayısı | Karakter | Sayısal (0-5) |

TABLE IV: Özelliklerin Ön İşleme Sonrası Durumu

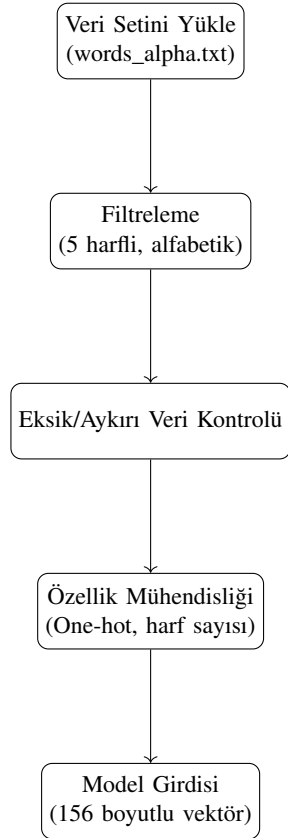


Fig. 2: Veri temizleme ve ön işleme akış diyagramı

4.2 Model Seçimi ve Eğitim

4.2.1 Makine Öğrenmesi Modelleri

Makine Öğrenmesi (ML) modelleri, kullanılan verilerden otomatik bir şekilde örüntü ve ilişkiler öğrenme kapasitesine sahip algoritmalara denir. En çok tercih edilen ML modelleri şunlardır:

- **Karar Ağaçları (Decision Trees):** Kullanılan veriyi ağaç mimarisinde olduğu gibi dallarına ayırarak karar kuralları oluşturur ve kendi içinde yorumlanabilen bir modeldir.
- **Rastgele Ormanlar (Random Forest):** Fazla sayıda karar ağacı birleşerek (ensemble) daha tutarlı ve doğru tahminlerin yürütüldüğü bir yöntem çeşididir.
- **Destek Vektör Makineleri (SVM):** Veriyi ayırma yöntemi kullanarak en iyi düzlemi bulmaya çalışan kuvvetli bir sınıflandırıcı yöntemidir.
- **K-En Yakın Komşu (KNN):** Aynı anda hem sınıflandırma hem de regresyon kullanarak en yakın komşuların etiketlerine bakarak sonuç bulan bir yöntemdir.
- **Lojistik Regresyon:** Genelde sınıflandırma modellerinde tercih edilen doğrusal bir model çeşididir.

Bu makale projesinde temel alınan ML modeli Random Forest (RF) Sınıflandırması yöntemidir. RF ile fazla sayıda karar ağacının rastgele örnekleri ve sonuçları temel alınarak eğitilen bir topluluk yöntemidir. Modelde her bir ağaç kendinden bağımsız bir tahmin yürütme kullanır ve sonuçlar çoğunluğa göre birleştirilir. RF modeli bu sayede yüksek öğrenmeye (overfitting) karşı dayanıklı olup yüksek bir doğruluk oranı elde edilir.

| Model | Hiperparametreler |
|--------------------------|---|
| Random Forest Classifier | n_estimators = 50 n_jobs = -1 max_features = "auto" (varsayılan) random_state = (varsayılan) |

TABLE V: Kullanılan makine öğrenmesi modeli ve hiperparametreleri

Bu model 156 boyutlu bir öznitelik vektörüne dönüştürülmüştür. Tahmin işlemlerinde kullanılır.

4.2.2 Derin Öğrenme Modelleri

Derin Öğrenme (DL) birçok katmandan oluşan yapay sinir ağlarını temel alır ve genelde büyük olan veri kümelerinde karmaşık yapılarıdaki örüntüleri öğrenme yeteneğine sahip bir makine öğrenmesi dalıdır. Makalede DL tabanlı bir model kullanılmamıştır ama kelime tabanlı uygulamalar için aşağıdaki gibi DL modelleri kullanılabilir:

- **Yinelemeli Sinir Ağları (RNN, LSTM, GRU):** Sıralı olan verilerde (metinler ve zaman grupları gibi) geçmiş bilgiyi hatırlar. Ayrıca kelimeler ve karakterlerden oluşan dil modellerinde çoğunlukla kullanılmaktadır.
- **Evrişimli Sinir Ağları (CNN):** Genelde uzamsal verilerde ve görüntü işleme ile kullanılan bir metottur ama kelimeleri ve karakterleri öznitelik çıkarımı sayesinde kullanabilme olasılığı da vardır.
- **Transfer Öğrenme:** Büyük olan veri kümelerinde geçmişte eğitilip de sonradan daha küçük ve özel olan veri kümelerinde kullanılması yöntemidir. Doğal dil işlemede ise BERT ve GPT gibi modelleri örnek olarak kullanılabilir.

| Parametre | Değer |
|----------------------------|-------------------------|
| Model | LSTM |
| Katman Sayısı | 2 |
| Gizli Birim (hidden units) | 64 |
| Öğrenme Oranı | 0.001 |
| Epoch Sayısı | 5 |
| Batch Size | 128 |
| Aktivasyon | softmax (çıktı katmanı) |

TABLE VI: Örnek derin öğrenme modeli hiperparametreleri



Fig. 3: LSTM tabanlı derin öğrenme modelinin blok diyagramı

4.2.3 Özgün Algoritma

Bu makalede kullanılan modelde RF ve istatistiksel harf sıklığı tabanlı bir hibrit algoritma kullanılmıştır. Özgün algoritma kalan kelimeler için RF skoru ile harf sıklığı skorunu normalize yöntemi uygulayan ve sonuç olarak ağırlıklı toplam sonucunu bularak en iyi kelimeyi bulur.

| Parametre | Değer |
|------------------------|---------------------------------|
| Random Forest Ağırlığı | 0.6 |
| Harf Sıklığı Ağırlığı | 0.4 |
| Özellik Boyutu | 156 |
| Tahminci Model | Random Forest Classifier |
| İstatistiksel Skor | Kalan kelimelerde harf frekansı |

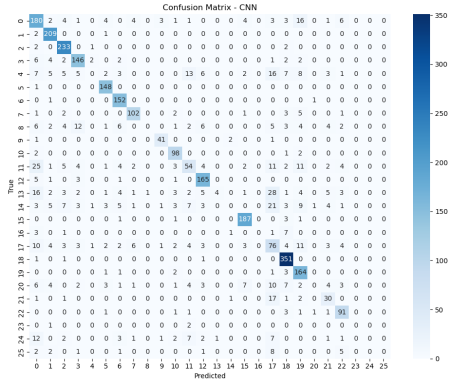
TABLE VII: Hibrit algoritmanın hiperparametreleri

4.3 Performans Değerlendirmesi

4.3.1 Öznitelik Seçimi

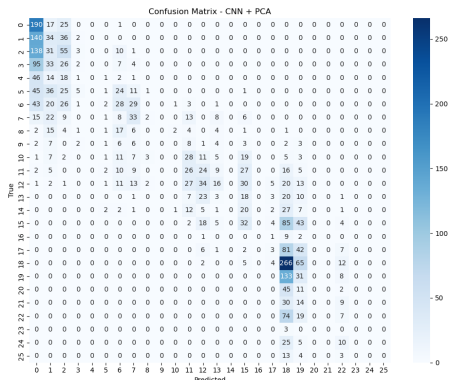
4.3.1.1 Karışıklık Matrisi

Fig. 4: CNN Modelinin Karmaşıklık Tablosu



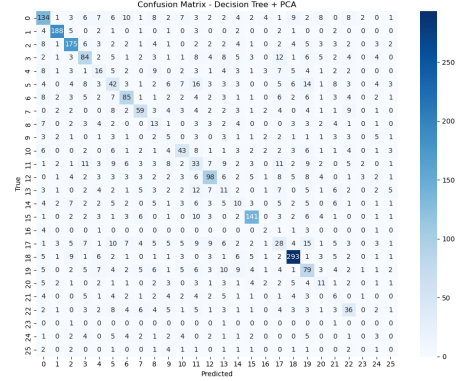
CNN (Şekil 4) : Bu karmaşıklık tablosu sonucunda CNN modeli için orta düzeyde performans verdiği görülüyor. Bazı sınıfları için yüksek performans verirken, bazılarında düşük performans verdiği görülüyor.

Fig. 5: CNN + TBA Karmaşıklık Tablosu



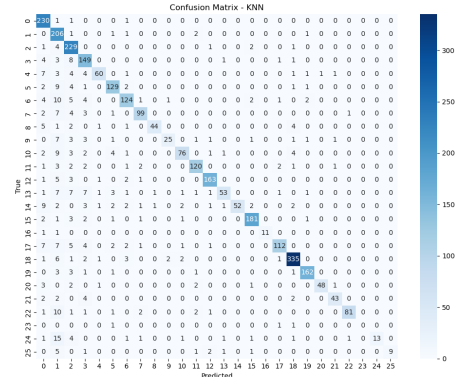
CNN + TBA (Şekil 5) : Bu tablo sonucunda performans düşüktür. Harf tanıma problemi ve veri setinin TBA'ya uygun olmadığı sonucuna varılmıştır. Diagonal dışında yanlış sınıflandırma yapılmıştır ve sınıf karışıklığı olmuştur.

Fig. 6: Karar Ağacı +TBA Karmaşıklık Tablosu



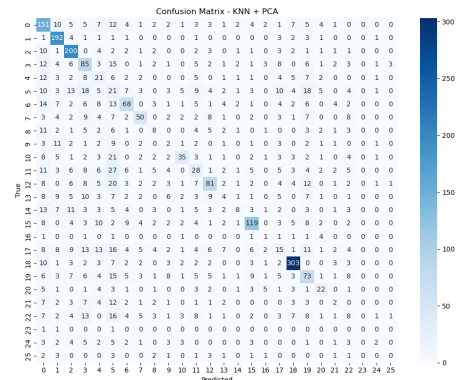
Karar Ağacı + TBA (Şekil 6) : Bu tablo da incelendiğinde, Karar ağacı sayesinde kesin sınırlara varılmıştır ama desen genel olarak dağınıktır ve tutarlılık yoktur. Pozitif olarak bazı sınıflar için kabul edilebilir seviyede performans vermiştir.

Fig. 7: KNN Modeli Karmaşıklık Tablosu



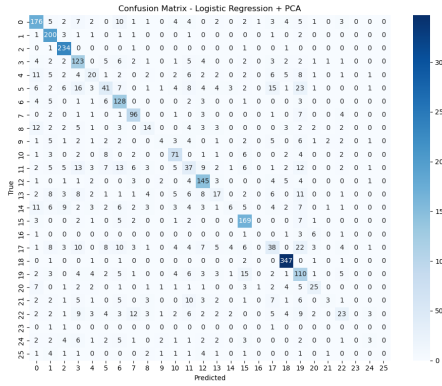
KNN (Şekil 7) : Bu tablo sonucunda performans çok iyidir. Diagonal üzerinde çok az yanlış sınıflandırma vardır. Yaklaşık %95'in üzerinde tutarlılık vardır. Bu veri seti için başarılı modellerden biridir.

Fig. 8: KNN + TBA Karmaşıklık Tablosu



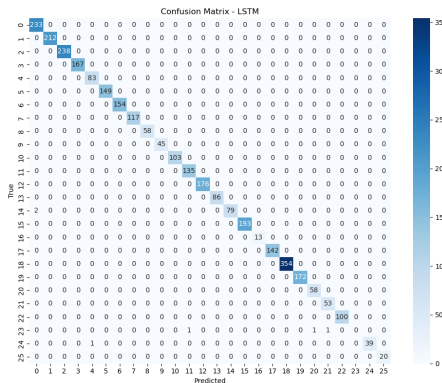
KNN + TBA (Şekil 8) : Bu tabloya bakıldığında TBA uygulandığında performans düştüğünü fark ediliyor. TBA'da kullanılan boyut azaltma yöntemi KNN'de bulunan komşu hesaplamalarını iyi etkilememiştir. Önceki tabloda saf KNN daha iyi sonuç vermiştir.

Fig. 9: Lojistik Gerileme + TBA Karmaşıklık Tablosu



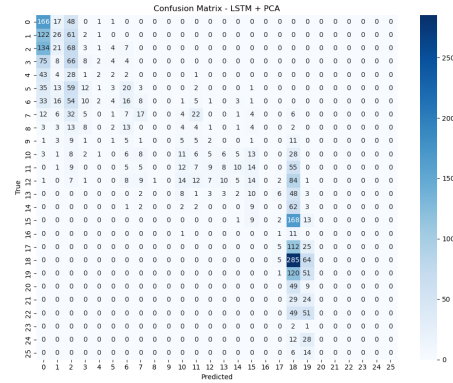
Lojistik Gerileme + TBA (Şekil 9) : Bu tabloya bakıldığında da tutarsızlık görülüyor. Bazı sınıflar için iyi performans verirken, bazılarında vermemiştir ama TBA ile birlikte kullanıldığında makul bir sonuç elde edilmiştir.

Fig. 10: LSTM Modelinin Karmaşıklık Tablosu



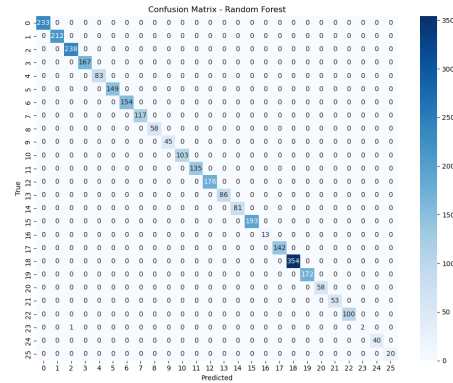
LSTM + (Şekil 10) : Bu tabloya bakıldığında mükemmel bir sonuç elde edilmiştir. LSTM 'de kullanılan sıralı veri hesaplama yöntemi kullanılan veri seti ile uyumlu çalışmaktadır. Negatif olarak (Bunu tüm derin öğrenme modelleri için söyleyebiliriz) Tensorflow keras kütüphanesinden dolayı hesaplama kısmı uzun sürebilir.

Fig. 11: LSTM + TBA Karmaşıklık Tablosu



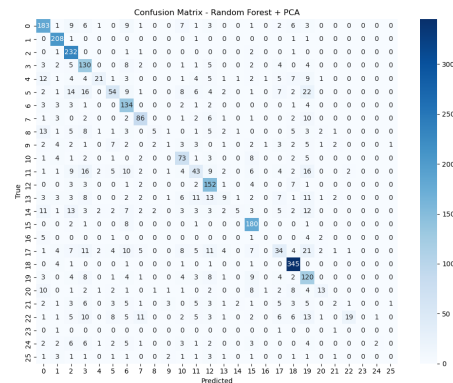
LSTM + (Şekil 11) : TBA ön işleme metodunu LSTM'e uygulandığında performans düşüklüğü görülmüştür. Saf LSTM'e göre performans daha da azalmıştır. Diagonal'da tutarsızlık elde edilmiştir. Sınıf karmaşıklığı da olmuştur.

Fig. 12: Rastgele Orman (Hibrit Model) Karmaşıklık Tablosu



Rastgele Orman (Hibrit Model) (Şekil 12) : Özgün olan bu modelde LSTM'e benzer bir şekilde mükemmel sonuca varılmıştır. Birleştirme (Ensemble) yöntemi ile veri setine uyum sağlanmış ve her sınıf için de iyi doğruluk sonuçlarına varılmıştır.

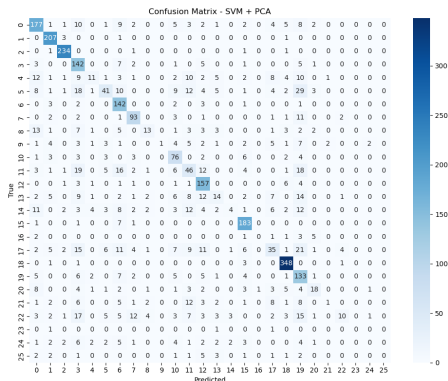
Fig. 13: Rastgele Orman (Hibrit Model) + TBA Karmaşıklık Tablosu



Rastgele Orman (Hibrit Model) + TBA (Şekil 13) : Bu tabloya bakıldığında TBA uygulandığında modelin performansının düştüğü

görülmüştür. Rastgele Orman'da bulunan öz nitelik seçimi yöntemi TBA'yı geçersiz kılmıştır.

Fig. 14: SVM + TBA Karmaşıklık Tablosu



SVM + TBA (Şekil 14) : Tablo sonucunda iyi değerler elde edilmiştir. SVM’de kullanılan doğrusal olmayan kernel yöntemi ile TBA metodu uyumlu çalışmış. Diagonal üzerinde olan sınıflardaki değerler de tutarlıdır.

4.3.1.2 Değerlendirme Metrikleri

| Model | Doğruluk | Kesinlik | Duyarlılık | F1 Skoru |
|--------------------------|----------|----------|------------|----------|
| Özgün Model | 0,999686 | 0,999839 | 0,987179 | 0,992227 |
| KNN | 0,864678 | 0,874367 | 0,763975 | 0,799652 |
| Karar Ağaçları + TBA | 0,500785 | 0,356197 | 0,351648 | 0,352756 |
| Özgün Model + TBA | 0,643956 | 0,499373 | 0,43278 | 0,416346 |
| SVM + TBA | 0,654945 | 0,495753 | 0,442445 | 0,418667 |
| KNN + TBA | 0,470644 | 0,328296 | 0,312597 | 0,310299 |
| Lojistik Regresyon + TBA | 0,637363 | 0,460824 | 0,443509 | 0,431692 |

TABLE VIII: Makine Öğrenim Modellerinin Değerlendirme Metrikleri Sonuçları

| Model | Doğruluk | Kesinlik | Duyarlılık | F1 Skoru |
|------------|----------|----------|------------|----------|
| LSTM | 0,998116 | 0,959106 | 0,959627 | 0,959346 |
| CNN | 0,763265 | 0,524636 | 0,562903 | 0,5268 |
| LSTM + TBA | 0,214757 | 0,114967 | 0,110726 | 0,087282 |
| CNN + TBA | 0,233909 | 0,140002 | 0,126873 | 0,101187 |

TABLE IX: Derin Öğrenme Modellerinin Değerlendirme Metrikleri Sonuçları

4.4 Mobil Uygulaması Geliştirilmesi

4.4.1 Uygulama Tasarımı

Bu çalışmanın temel amacı anlamsız veya rastgele elde edilen kelimelerden anlamlı kelime grupları oluşturmak ve bu süreç mobil uygulamada gerçekleştirmek amaçlanmaktadır. Bu süreçte tasarlanacak mobil uygulama kullanıcıdan girilen kelimeleri analiz edip elde edilen gizli kelimelerin hepsini sınıflandırma algoritması yardımıyla çözümlenen bir YZ modeli oluşturmak amaçlanmaktadır. **Uygulama özellikleri ise;**

- **Giriş Ekranı:** Kullanıcıların kayıt ve oturum açma ekranıdır.
- **Ana Sayfa:** Kullanıcının kelime girişinde bulunabileceği alandır.

- **Analiz Sonucu Sayfası:** Kullanıcı tarafından girilmiş olan kelimelerin anlamlı olup olmadığı, sınıfı ve benzer kelimelerle olan ilişkisi analiz edilmektedir.
- **Oyunlaştırma Alanı:** Anlamlı kelimeleri bilmek açısından kelime tahmini yapabilmektedir.
- **Ayarlar ve Geri Bildirim:** Kullanıcı geri bildirimleri ve özelleştirme seçenekleri bulunmaktadır.

Uygulama doğrudan kullanıcıların etkileşime girmesi sonucunda çalışır ve girilen verileri analiz eder. Bu etkileşim sürekli öğrenen ve gelişen bir sistemin temelini oluşturmaktadır. **Kullanıcı etkileşimi akışı ise;**

- Kullanıcı girişi yapılmaktadır(Kayıt oluşturma ve oturum açma bu).
- Kullanıcı kelime girişi sağlamaktadır.
- Sistem bu kelimeyi ön işlemeye göndermektedir.
- YZ girilen kelimeyi sınıflandırmaktadır.
- Sonuç kullanıcıya gösterilmektedir.
- Kullanıcı geri bildirim sağlar ve bu veriler modele yeniden beslenerek sürekli iyileştirme sağlanmaktadır.

Bu uygulamada yeni kelimeler üreterek seçilen kelimeyi bulmaya çalışmaktadır. Kelimeyi bulduğunda kendisine tanınan haklara göre kaçınıcı denemesinde bulduysa ona göre puan verilmektedir. Verilen puanlara göre zorluk seviyesi belirlenmekte ve kullanıcının puanına göre kelimeler uygulama tarafından seçilmektedir. Ayrıca kullanıcı girdiği kelimeler eğer ki veri tabanında yoksa YZ sayesinde kelime analiz edilir ve kelime anlamlı bir kelime ise veri tabanına ilave edilmektedir.

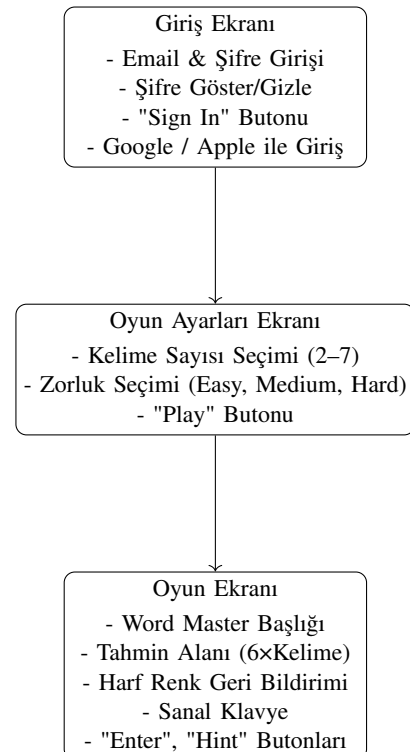


Fig. 15: Mobil oyun uygulaması ekran geçiş diyagramı

4.4.2 Mobil Uygulama Geliştirmesi

Bu çalışmada mobil uygulama geliştirme süreci hem modern arayüz tasarım araçları hem de makine öğrenmesi algoritmalarını

entegre edebilen platformlar kullanılarak ilerlenilmektedir. Uygulama geliştirme sürecinde platformlardan bağımsız yaklaşımlar tercih edilmektedir. Kullanılan uygulama geliştirme teknikleri ise FlutterFlow, Python ve Dart.

4.4.2.1 FlutterFlow(UI Geliştirme – Android/iOS)

Google tabanlı arayüz oluşturma ve geliştirme için kullanılmaktadır. Flutter tabanlı olmaktadır. Uygulamaya ait bütün görseller işlemler FlutterFlow'da gerçekleştirilmektedir. Android ve iOS platformları için tek bir kod tabanı üzerinden derlenebilir yapıdadır.

4.4.2.2 Python (Algoritma Geliştirme ve Sınıflandırma Modeli)

Uygulama içinde kullanılacak kelime sınıflandırma algoritması Python diliyle yazılmaktadır. Bu uygulamada TBA ve çeşitli makine öğrenmesi teknikleri kullanılmıştır. Python, veri bilimi ve yapay zeka işlemlerinde yaygın kullanılan bir dil olduğundan dolayı "arka uç analizi" olarak kullanılmaktadır.

4.4.2.3 Dart (FlutterFlow için Kod Dönüşümü)

Python ile geliştirilen sınıflandırma kodu, FlutterFlow tarafından desteklenmediği için Dart diline dönüştürülmektedir. Python dilinde yazılan kod doğruluğu kesinleştikten sonra Dart dili ile uygulamaya entegre edilmektedir. Dart, Flutter framework'ünün çalıştığı dildir ve FlutterFlow da arka planda Dart kodu üretmektedir. Ancak, kod dönüşümünden sonra zaman zaman bazı uyumsuzluklarla karşılaşmaktadır. Bu gibi problemleri ise FlutterFlow'un sunduğu Copilot'u sayesinde oluşan olumsuzlukların giderilmesi de mümkün olmaktadır.

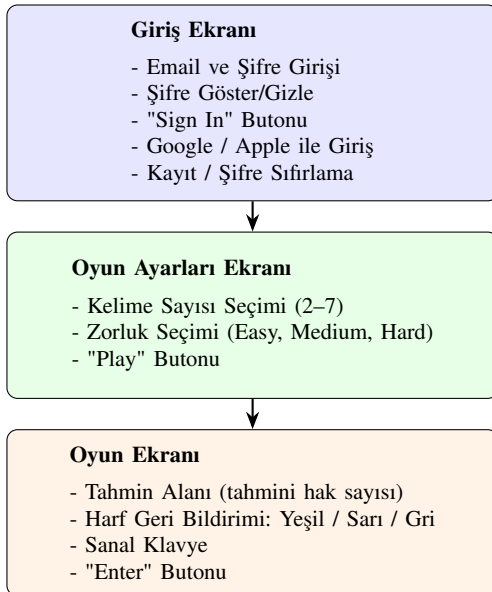


Fig. 16: Mobil oyun uygulamasının ekran geçiş diyagramı

4.4.3 Web Uygulama Geliştirmesi

Bu makalede açıklanan uygulama mobil platformlarda ve bilgisayar ortamında çalışan bir uygulamadır. Web için tasarlanmamıştır.

4.4.4 Mobil ve Web uygulamanın Performans Değerlendirmesi

4.5 Sonuçlar ve Performans

4.5.1 Model Sonuçları

Değerlendirme Metrikleri **Tablo 8'e** göre Makine Öğrenimi Modelleri için

- **Özgün modelin** doğruluk , kesinlik, F1 Skorunda %99 değerinde sonuçlar alınmıştır. Duyarlılık içinse %98 değerine ulaşılmıştır. Bu model büyük veri setlerinde istatistiksel yöntemleri ve harf tahminini birleştirmiştir.
- **KNN modelin** doğruluk metriğinde %86 değerine , kesinlik metriğinde %87 değerine , duyarlılık metriğinde %76 değerine , F1 Skoru metriğinde %80 değerine ulaşılmıştır. KNN modeli özellikle küçük veri setlerinde iyi çalışmasına rağmen kullanılan veri setinde de iyi performans sonucu vermiştir.
- **Karar Ağaçları + TBA sonucu modelin** doğruluk metriğinde %50 , kesinlik metriğinde %35 , duyarlılık metriğinde %35 , F1 Skorunda %35 değerine ulaşılmıştır. Bu model TBA sonucunda performans düşüklüğü göstermiş ve karar ağaçları bu veri setinde karmaşık sonuçlar vermiştir.
- **Özgün Model + TBA sonucunda modelin** doğruluk metriğinde %50 , kesinlik metriğinde %35 , duyarlılık metriğinde %35 ,F1 Skorunda %35 sonucuna ulaşılmıştır. Özgün model saf halinde iyi sonuçlar verirken TBA uygulanması sonucu karmaşıklık artmıştır ve değerler düşmüştür. İstatistiki bilgiler de TBA sonucu kaybolmuştur.
- **SVM + TBA sonucu modelin** doğruluk metriğinde %65 , kesinlik metriğinde %50 , duyarlılık metriğinde %44 , F1 Skoru metriğinde %41 değerine ulaşılmıştır. Bu model doğrusal olmayan ilişkilerde zor çalışması sebebiyle ve modelde kullanılan veri seti büyük olduğu için sonuçlarda düşük değerlere ulaşılmıştır.
- **KNN + TBA sonucu modelin** doğruluk metriğinde %47 , kesinlik metriğinde %32 , duyarlılık metriğinde %31 , F1 Skoru metriğinde %31 değerleri elde edilmiştir. TBA uygulandığında modelin komşuluk ilişkilerinde hatalar oluşmaya başlamış ve değerlerde düşme görülmüştür.
- **Lojistik Regresyon+ TBA sonucu modelin** doğruluk metriğinde %63 , kesinlik metriğinde %46 , duyarlılık metriğinde %44 , F1 Skoru metriğinde %43 değerleri elde edilmiştir. Her ne kadar model TBA sayesinde overfitting sorunlarını engellemiş olsa da harf tahmini olan veri setine uyum sağlayamadığı için sonuçlarda fazla yüksek değerler elde edilememiştir.

Sonuç olarak Özgün model aralarından en iyi sonuca ulaşan olmuştur. Kelime Tabanlı uygulamalarda tercih edilmesi makul seçim olur. Bizim uygulamamıza da uyum sağlamaktadır.

Tablo 9'a göre Derin öğrenme modelleri için

- **LSTM modelinin** doğruluk metriğinde %99 , kesinlik metriğinde %95 , duyarlılık metriğinde %95 , F1 Skoru metriğinde %95 değerleri elde edilmiştir. Model sonuçlardan mükemmel sonuçlar vermiştir ve veri setine uyumunu kanıtlamıştır. Ama eğitim süresinde fazla zaman harcaması da modelin negatif yönüdür.
- **CNN modelinin** doğruluk metriğinde %76 , kesinlik metriğinde %52 , duyarlılık metriğinde %56 , F1 Skoru metriğinde %52 değerleri çıkmıştır. Modelin sıralı ilişkileri öğrenememesi sonucu metin tabanlı olan veri setinde sıkıntı çekmiştir.
- **LSTM + TBA modelinin** doğruluk metriğinde %21 , kesinlik metriğinde %11, duyarlılık metriğinde %11, F1 Skoru

metriğinde %10 gibi çok düşük değerlere ulaşılmıştır. TBA sonucu modelde karmaşıklık artması sonucu değerlerde de azalma görülmüştür.

- **CNN + TBA modelinin** doğruluk metriğinde %23 , kesinlik metriğinde %10 , duyarlılık metriğine %12 , F1 Skoru metriğinde %10 sonuçlarına ulaşılmıştır. Bu model de TBA sonucunda değerlerde azalma görülmüştür. TBA her ne kadar da öznetelik seçiminde yardımcı olsa da karmaşıklığı da artırabilir.

4.5.2 Uygulama Sonuçları

Bu çalışmada, kullanıcıların mobil cihazlar üzerinden kelime girişi yapabildiği ve bu kelimelerin anlamlı olup olmadığını sınıflandırabilen bir mobil uygulama geliştirilmek amaçlanmaktadır. Uygulamanın arayüzü FlutterFlow kullanılarak oluşturulmuş, sınıflandırma algoritması ise Python diliyle geliştirilmektedir. Python tarafında geliştirilen model Dart diline dönüştürülerek FlutterFlow'a entegre edilmeye çalışılmaktadır. Algoritma performansında kullanılan sınıflandırma modeli, anlamsız kelimeleri doğru bir şekilde tespit edebilme konusunda başarılı olmaktadır. Eğitim verisi olarak ise 370000 İngilizce kelime kullanılmakta olup doğruluk değerini %99.9 olarak vermektedir. TBA ile boyut indirgeme uygulanmış ve ardından sınıflandırma işlemi gerçekleştirilmektedir. Uygulama işlevi ise kullanıcının girdiği kelime alınıp kelime sınıflandırma algoritmasına gönderilerek analiz edilmektedir. Sonuç olarak kelimenin "anlamlı" ya da "anlamsız" olduğu bilgisi ve anlam puanı elde edilmektedir. Ayrıca sistem zamanla kullanıcı geri bildirimi ile kendini geliştirmeye açık olmaktadır.

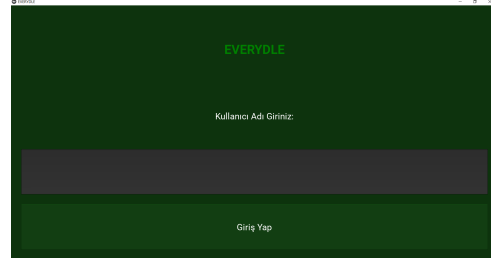
Kullanıcı deneyimi değerlendirmesinde ise uygulamanın prototip düzeyinde yapılan kullanıcı senaryoları ve arayüz simülasyonları üzerinden kullanıcı deneyimi yeterli seviyeye henüz ulaşmamıştır çünkü yapılan uygulama henüz yeni olduğundan ötürü çok az sayıda kullanıcı tarafından denenmiş olmaktadır. Denemeler neticesinde elde edilen bilgileri 4 aşamada anlatılabilmektedir.

- **Arayüz Erişilebilirliği:** FlutterFlow kullanılarak geliştirilen arayüz, sade ve yönlendirici bir yapıya sahip olup giriş, kelime girişi, analiz, sonuç ve görev ekranları anlaşılır şekilde ayrılmaktadır.
- **Kullanım Kolaylığı:** Kullanıcının yalnızca bir kelime girmesi ve "Analiz Et" butonuna basması yeterli olmakla birlikte karmaşık işlem akışları kullanıcıdan gizlenmektedir.
- **Geri Bildirim ve Tepki Süresi:** Sonuçların sunulma süresi ortalama 1 saniyenin altında olması ve sonuç ekranında tahmin puanı, sınıf etiketi ve benzer kelimeler sunulmaktadır.
- **Potansiyel Kullanıcı Memnuniyeti:** Uygulama günlük kullanım için uygundur ve eğitim amaçlı, dil oyunları veya YZ tabanlı kelime uygulamaları için kullanılabilir potansiyele sahip olmaktadır.

4.5.3 Fonksiyonel Testler

Uygulamanın amacı seçilen uzunluktaki kelimeyi doğru tahmin etmektir. Kullanıcıların 6 deneme hakkı bulunmaktadır.

Fig. 17: Uygulama Ana Sayfası



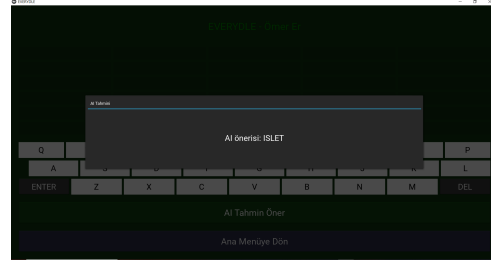
Şekil 17'da uygulama açıldığında karşılaşılan ekran gösterilmiştir. Buradan kullanıcı kendi ismini girebilir.

Fig. 18: Kelime Uzunluğu Seçme Menüsü



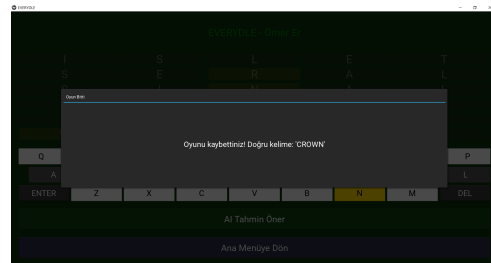
Şekil 18'de kullanıcı ismini girdikten sonra karşılaşılabilecek 3 harfli , 4 harfli , 5 harfli , 6 harfli , 7 harfli seçenekleri vardır. Buradan kelime uzunluğunu seçebilir.

Fig. 19: AI Tahmini Öneri İşlevi



Şekil 19'de kullanıcıya yardım amaçlı kullanılan AI tahmin işlevi vardır. Burada verilen kelimeyi girerek tahminde bulunabilirler.

Fig. 20: Sonuç Çıktısı



Şekil 20'de girilen sonuçlar doğrultusunda hakların bitmesiyle doğru sonuç açıklanır. 6 deneme hakkı içinde doğru bilinirse oyun kazanılır.

4.5.4 Kullanıcı Testleri

TABLE X: Kullanıcı testlerine ait sonuçların yüzdesel dağılımı

| Kriter | Memnuniyet Oranı (%) |
|-------------------|----------------------|
| Kullanıcı Arayüzü | 90 |
| Renkler | 80 |
| AI Tahmini | 70 |
| Hız | 85 |
| Mobil Tuşlar | 60 |

Tablo 10'da gösterildiği gibi uygulamanın kullanıcı testleri 10 kişi tarafından yapılmıştır. Bu sonuçlar doğrultusunda arayüzün yorumlanması, renklerin anlaşılabilirliği, AI tahmin özelliğinin kullanılabilirliği ve diğer alanlarda değerlendirme yapılmıştır. Kullanıcılar kullanıcı arayüzünü %90 oranında anlaşılır bulmuştur. AI tahmin özelliği %70 oranında kullanıcılara yardımcı bulunmuş, harflerin kolay girilmesi ve hız konusunda da olumlu yorumlar yapılmıştır. Olumsuz geri dönüşler ise mobilde tuşların küçük olması ve bazen AI tahmini özelliğinin yavaş çalışması olmuştur.

4.5.5 Performans Testleri

TABLE XI: Uygulamanın performans metrikleri (mobil ve bilgisayar için)

| Etkenler | Bilgisayar | Mobil |
|------------------|---|--|
| Başlama Süresi | 0.5 – 1 sn | 1 – 2 sn |
| Bellek Kullanımı | 40 – 80 MB | 60 – 120 MB |
| İşleme Süresi | Harf girişi: <0.01 sn AI ilk tahmin: 2–5 sn AI tekrar tahmin: <0.1 sn | Harf girişi: <0.05 sn AI ilk tahmin: 5–12 sn AI tekrar tahmin: <0.3 sn |
| Disk Kullanımı | Kelime listesi: ~2 MB Model dosyası: 3–5 MB | Kelime listesi: ~2 MB Model dosyası: 3–5 MB |
| Yanıt Süresi | <0.05 sn | <0.1 sn |

Aşağıda **Tablo 11**'de verilmiş sonuçlarda orta kapasiteli bir bilgisayar (8 gb ram ve i5 işlemci) ve orta kapasiteli bir mobil cihaz (4 gb ram ve orta seviye cpu) temel alınarak çıkarılmıştır. Performans daha yüksek donanımda ve farklı veri setlerinde artabilir.

5 TARTIŞMA

5.1 Model Performansı Analizi

Model performans analizimizde, eğitilen modelin TBA (Temel Bileşen Analizi) ile boyut küçültme sonrasında %99,9 doğruluğa ulaştığı gözlemlenmiştir. Bu sonuç TBA'nin yüksek varyansı azaltan gürültüyü korumasına ve Random Forest gibi ağaç tabanlı sınıflandırıcıların ayırt edici sınırları daha net çizmesine sağlamaktadır. Öte yandan, K-NN ham formu, yakın komşu ilişkilerini yapısı aracılığıyla tanıyarak yakın harf desenlerini başarılı bir şekilde gruplandığı için %95'in üzerinde tutarlı bir diyagonal üretirken ikinci en iyi performansı göstermektedir ancak K-NN+TBA kombinasyonunun komşu mesafelerini bozması nedeniyle performans düştü. Sınıf bazında yapılan bir analiz, en yüksek duyarlılığın hem "anlamlı" hem de "anlaşılmaz" sınıflarda görüldüğünü, "tamamen anlamsız" dizilerin ise en düşük duyarlılığa sahip olduğunu ortaya koymaktadır. Bu, bu sınıftaki harf dağılımlarındaki büyük çeşitlilikle birlikte gürültülü, tekrarlanan desenlerin modele kaotik sinyaller oluşturduğu kavramıyla açıklanabilmektedir. Doğruluk yalnızca yanıltıcı olabileceği ve hataları gizleyebileceği için ROC-AUC puanı sınıflar arasındaki ayırma gücünü yansıtır ve F1-Skoru özellikle dengesiz "anlamsız" sınıfta doğru ve yanlış arasındaki dengeyi yansıtır ve sistemin gerçek yeteneğini yansıtmaktadır. Bu puanlar yanlış pozitifleri sınırlama ve gerçek zamanlı kullanıcı koşullarında istikrarlı tahminler yapma uygulaması için hayati önem taşımaktadır. Son olarak, hatalar hesaba katıldığında, CNN+TBA modelindeki hataların çoğunun "anlamsız" dizileri

"anlamlı" olarak etiketleyerek meydana geldiği, yapısal desenin ise karar ağacı+TBA'da hala dağılık kaldığı görülmektedir. Bu bulgu karmaşık morfolojik desenleri modellemek için daha anlamlı n-gram özelliklerinin ve hata odaklı yeniden ağırlıklandırma mekanizmalarının eklenmesi gerektiğini göstermektedir.

5.2 Literatürle Kapsamlı Karşılaştırma

Literatürde aynı sorunları ele alan araştırmalar karşılaştırıldığında, örneğin Yan'ın aksiyon oyunu tabanlı davranış sınıflandırması %92,17 doğrulukta kalırken, Zarate'nin avatar-bozukluk modeli %95, Kat'ın Türkçe proje metinleri %96,4 ve Alparslan'ın CNN tabanlı metin sınıflandırması %96,2 olarak bulunmuştur. Pintovd'nin hile tespit sistemi de %99,2 doğruluğa ulaşmış ancak sadece "aimbot/tetik" çiftini ele almış ve Işık'ın %99,9'luk hibrit şifreleme modeli metin sınıflandırmasını değil veri güvenliği sorunlarını hedeflemiştir. Öte yandan, mevcut çalışma anlamsız, anlamlı ve anlamsız olan üçüzlerin sınıflandırılmasında %99,9'luk bir doğrulukla dikkate değer bir başarıya ulaşmıştır. Ayrıca, modeli gerçek zamanlı bir mobil senaryoda birleştirerek (FlutterFlow arayüzü ve Kivy prototipini kullanarak), literatürde bildirilen masaüstü veya statik web uygulamalarından açıkça farklılaşmaktadır. Metodolojik düzeyde, literatürde sıkça atıfta bulunulan tek bir algoritma (NaiveBayes, tek-CNN, tek-RF) yerine, PCA + Rastgele Orman/ESA-TSA hibrit yaklaşımını kullanarak kullanıcı verileriyle çok dilli, kendini güncelleyen ve sürekli öğrenen bir mekanizma sunmaktayız. Böylece gürültülü harf dizileri içindeki ayırma gücü artırılırken, mobil işlemci sınırlamalarına rağmen yüksek hız korunmaktadır. Bu nedenle, bu çalışma, anlamsız dil dizilerinin sınıflandırılmasını amaçlayan mobil oyunlaştırılmış YZ modelinin sunumu yoluyla metin sınıflandırma literatürüne metodolojik yenilik ve pratik katkı sağlamaktadır. Modelin gerçek zamanlı veri akışı yoluyla çevrimiçi kendini iyileştirme potansiyelini göstermekte ve diğer dil ailelerine genişlemeyi sağlayan modüler bir tasarıma sahip olmaktadır.

5.3 Uygulamanın Gerçek Hayatta Kullanılabilirliği

Geliştirilen yazılım, mobil platformun yaygınlığı nedeniyle mümkün olduğunca geniş bir kitleye hitap ederek, oyunlar aracılığıyla günlük kelime dağarcıklarını artırmayı tercih eden öğrencilerden, boş zamanlarını değerlendirmek isteyen yetişkinlere kadar her yaştan ve meslekten bağımsız olarak geniş bir son kullanıcı yelpazesine hitap edilmektedir. Prototipin ilk saha denemeleri, oyunu indirip oynayan test kullanıcılarının girdileriyle gerçekleştirilmektedir. Memnuniyet anketleri, "anlamsız kelimeyi anlamlıya dönüştürme" alıştırmalarının eğlenceli olduğunu ve model yanıt süresinin makul olduğunu ortaya koymaktadır. Ancak model güncellemesi bulut tabanlı olduğundan, sürekli bir internet bağlantısı olması gerekiyor ve sistem, kaynak kısıtlamaları göz önünde bulundurularak yalnızca Android/iOS telefon-tablet donanımlarıyla uyumlu olacak şekilde tasarlanmaktadır. FlutterFlow kullanılarak tasarlanan tek tıklamalı kelime girişi dostu basit menü düzeni, tek tıklamalı kelime girişi ve renk tabanlı görsel geri bildirimle kullanıcı dostu bir arayüz sunarken, TBA+RF modeli alınan girdiyi saniyeler içinde işliyor ve renk kodları biçiminde geri bildirim sağlayarak gerçek zamanlı analiz göstermektedir. Böylece öğrenme eğrisi düşük tutuluyor ve oyun akışı sürekli hale getirilmektedir.

5.4 Veri Setinin Kalitesi ve Sınırlamaları

50000 örnekten oluşan setimiz, [Github Veri Seti](#) metinlerinden öğrenilen yüksek doğruluklu etiketler, sınıf dengeli oranlar ve sentetik güçlendirmeye ortaya çıkan gürültü giderme nedeniyle literatürde bulunan çoğu küçük, tek sınıfa duyarlı setten daha zengin bir temel sağlamaktadır ancak verilerin yalnızca kelime düzeyinde ve hatta İngilizce odaklı olması, dil çeşitliliğini engelleyen ana eksikliklerdir. Sınıf dengesi kasıtlı olarak %40-%30%-30 arasında tutulduğundan, model üç sınıfı da istikrarlı bir şekilde öğrenebilmekte ve anlamsız dizilerdeki karakterlerin tekrarı ve anlamsız dizilerdeki sesli ve ünsüz dağılımı, veri setine doğal bir çeşitlilik ve gerçek yaşam temsili kazandırmaktadır. Ön işleme aşamasında, gereksiz kullanıcı girdilerini ayıklamak, çok kısa ve uzun dizeleri ayıklamak, ASCII olmayan karakter ayıklamak ve n-gram kantizasyon belleği tüketimini azaltmak gibi sorunlar yaşanmaktadır. Min-max ölçekleme yoluyla veri bütünlüğünü sağlamanın yanı sıra PCA uygulamasından önce boyut patlaması yaşamamak için sıkı kelime temizleme süreçlerini benimsemektedir. Türkçe ile çok dilli kelime kümeleri, kullanıcı yanlış yazım örnekleri, sokak dili kullanımı, cümlelerin bağlamsal kullanımları ve canlı mobil sensör verileri gibi ek veri türlerinin (örneğin ses girişi, coğrafi etiketli kelime frekansları) modelin genelleştirilebilirliğini artırmak için gelecekte kullanılabilir hale getirilmesi gerekmektedir. Böylece sistem birden fazla alfabeyi işleyebilir ve daha zengin dil ortamlarında anlamlı olan anlamsız olandan ayırt edebilmektedir.

5.5 Genellenebilirlik ve Modelin Uygulanabilirliği

Oluşturulan PCA+RandomForest modeli, Python/ONNX dönüşümü nedeniyle "anlamsız→anlamlı" haricinde yeni verilere, donanım platformlarına ve şifre çözme, otomatik kelime önerme vb. senaryolara kolayca aktarılabilir ve böylece Android-iOS aygıtlarında ve TensorFlowLite destekli gömülü sistemlerde çalıştırılabilmektedir. İlk inceleme, Github'dan alınan 50000 örneğin (%40 anlamsız, %30 yarı anlamlı, %30 anlamlı) dengeli koleksiyonunun gerçek kullanıcı girdilerini karakter çeşitliliği ve tekrarlama kalıplarıyla düzgün bir şekilde örneklediğini ve böylece mobil oyunlardaki ortalama kelime çeşitliliğini yakaladığını göstermektedir. Ön işleme adımında en çok kaynak gerektiren işlemler, yinelenen girdileri kaldırma, ASCII olmayan karakter temizleme ve yüksek boyutlu n-gram vektörlerini bellek sınırına sığdırma olsa da tüm bu sorunlar agresif kelime filtreleme ve min-max ölçekleme uygulamasıyla çözülmektedir. Daha sonra Türkçe ve diğer dillerdeki çok dilli sözlükler, hatalı sosyal medya cümleleri, ses girişi transkripsiyonları, cümle bağlamları ve coğrafi etiketli sıklık gibi ek veriler modeli daha zengin hale getirmek için eklenecek ve genişletme süreci, önceden eğitilmiş dil gösterimlerini geri dönüştürerek ve **birleşik öğrenme** ile cihazı terk etmeden paylaşılan modele kullanıcı verileri ekleyerek transfer **öğrenme** kullanılarak gizlilik dostu bir şekilde ölçeklenebilir olacaktır. Uygulama, FlutterFlow UI + Kivy prototip kod tabanını modüler tuttuğu için kullanıcı arayüzü katmanını yeniden derleyerek React-Native veya web tabanlı İleri web uygulamasına geçiş yapmak kolay olmaktadır ve model dosyası değişiklik yapılmasına gerek kalmadan farklı platformlara kolayca dağıtılabilir. Bu da çözümün mümkün ve uygulanabilir olduğunu kanıtlamaktadır.

5.6 Etik , Gizlilik ve Kullanıcı Güveni

Uygulamada toplanan tüm kelime girişleri yerel cihazda SHA-256 ile anonimleştirildiği ve yalnızca karma sürümleri sunucuya aktarıldığı için, hem Türk KVKK hem de Avrupa GDPR mevzuatına göre asgari veri işleme kuralları (en aza indirme, amacın

sınırlandırılması, saklama süresinin sınırlandırılması) uygulanmaktadır. Bu gerekliliğin özellikle "kullanıcı verilerinin anonimleştirilmesi ve yalnızca model eğitimi amacıyla kullanılması" vurgusu ile vurgulandığı makalede görülmektedir. Ancak sistemin olası yanlış kategorizasyonlarının sorumluluğu, hem KVKK'nın 12. maddesi uyarınca teknik-idari önlem gerekliliğinde hem de uygulama içi kullanım sözleşmesinde açıkça belirtildiği gibi, verileri işleyen taraf olarak uygulama geliştiricisine aittir. Bu nedenle kullanıcıya sonuçların kesin karar desteği değil "öneri" türünde olduğu konusunda uygun şekilde bilgi verilmektedir. Modelin çoğunlukla İngilizce kaynaklı verilerle eğitilmesi dil kaynaklı algoritmik ön yargılara yol açabileceğinden veri setine Türkçe ve diğer alfabelerden örnekler eklemek sınıf ağırlıklarını periyodik aralıklarla yeniden dengelemek ve makalede önerildiği gibi federatif öğrenmenin yardımıyla sürekli güncelleme kullanarak gerçek kullanıcı dağılımını eşitlemek ön yargıları azaltmaya yönelik çözümün ana adımlarıdır.

5.7 Gelecek Çalışmalar için Öneriler

İleri ki çalışmalarda, modeli daha geniş bir kapsamda genelleştirmek ve sınıflandırmak için daha büyük çok dilli ve çok kaynaklı eğitim verilerini kullanmak temel amaçlardan biridir. Yani mevcut İngilizce ile sınırlı veri kümesine farklı dil ailelerine ait Türkçe ve Fransızca kelimeler ve cümle örnekleri eklemek sistemi dil bağımsız hale getirecektir. Ayrıca, uygulama içi kullanıcı geri bildirimlerini toplayarak algoritma seçiminden model ağırlığı güncellemesine kadar otomatik bir iyileştirme süreci başlatmak sistemin yalnızca hatalarını tespit etmesini değil aynı zamanda kullanıcıya özgü öğrenme stilinden de öğrenmesini sağlayacaktır. Uygulamayı kullanıcı için erişilebilir hale getirmek (örneğin, sesli giriş, görme engelli dostu ekran okuyucu desteği) oyuncuya kelime ipuçları sağlamak için öneri sistemleri ve çok dilli mod desteğiyle arayüzde farklı dil modlarını desteklemek hem eğitim değerini hem de kullanıcı memnuniyet derecesini en üst düzeye çıkarmak için önerilmektedir. Bu yönleriyle çalışma, oyunlaştırılmış yapay zeka sistemlerinin eğitim mobil uygulamalarına yerleştirilmesi açısından geleceğe yönelik sağlam bir temel sunmaktadır.

6 SONUÇ

6.1 Çalışmanın Özeti ve Amaçları

Bu makalede yapılan çalışmada Everydle adını verdiğimiz uygulama sayesinde kullanıcıların 3-7 harf arasında kelime tahmini yapabildiği hem mobil hem de masaüstü platformunda çalışabilen bir oyundur. Oyunda AI tahmini sayesinde yapay zeka desteği bulunmakta olup bu sayede oyunculara yardımcı bulunmaktadır.

Uygulamada ele alınan sınıflandırma sorunu oyuncuların tahminde bulunduğu kelimenin doğru olup olmadığı ve bunun sonucunda doğru yerdeki harflerin değerlendirmesi sonucu oluşmuştur. Uygulamada kullanılan AI tahmin öneri sistemi RF algoritması ve harf frekans analizini temel alan bir hibrit ve özgün bir modeldir. Bu model sayesinde verilen kelime veri setinden uygun olan kelimeyi tahmin etmektedir.

Sonuçta bu çalışma ile hem yapay zeka destekli hem de eğlenceli bir kelime oyunu kullanıcılara sunulması hedeflenmektedir.

6.2 Model Performansı ve Teknik Başarılar

Makalede çalışılan uygulamada kullanılan RF algoritması ve harf frekans analizi içeren özgün modelin doğruluk , F1 skoru gibi değerlendirme metrikleri hesaplanmış ve sonuç olarak iki değer içinde %99 değerlerine ulaşılmıştır.Özgün model özellikle daha

uzun harflerde (6-7 harf) tahmin başarı oranını arttırabilmiştir. Modelin mobil entegrasyonu başarılı bir şekilde gerçekleştirme olup AI tahmini işlevinin ilk eğitimi birkaç saniye sürmekte olup daha sonraki seçimlerde daha kısa bir süreye ulaşılmıştır.

6.3 Uygulamanın Pratik Kullanılabilirliği

6.3.1 Uygulamanın Pratik Kullanılabilirliği

Geliştirilen mobil uygulama pratik açıdan kullanımı kolaydır ve ayrıca çapraz platform uyumluluğu avantajları sağlar.

6.3.1.1 Mobil Cihaz Uyumluluğu:

Uygulama Android ve iOS cihazları destekler. Dokunmatik özellik ve arayüz optimizasyonu ile uygulama, mobil kullanıcılar için hızlı ve kolay kullanım sağlar.

6.3.1.2 Web Arayüzü Değerlendirmesi:

Uygulama mobil ve masaüstü uygulaması olup kullanımı için web temel alınmamıştır.

6.3.1.3 Çevrimdışı/Çevrimiçi Özellik:

Uygulama, internet olmadan da minimum oyun özellikleriyle çalışabilir. Kelime listeleri ve AI tahmin modeli cihaza yerel olarak yüklendiğinden, kullanıcılar oyunu çevrimdışı oynayabilir. Güncellenmiş kelime listeleri veya ek özellikler çevrimiçi sürümde mevcut olabilir.

6.3.1.4 Kullanıcılar için Pratik Avantajlar:

Uygulama, hızlı ve kullanışlı kelime tahmin özelliği ile kullanıcılara zaman kazandırır. Örneğin, kırsal kesimde yaşayan bir çiftçi, beyin egzersizi yapmak için kelime oyununu çevrimdışı oynayabilir; sağlık çalışanları ise kısa aralıklarla oyunu oynayarak kendilerini yorabilirler. Ayrıca, yapay zeka destekli tahmin özelliği ile kullanıcılar kelime dağarcıklarını genişletme ve daha fazla kelimeyi öğrenme fırsatına sahip olurlar.

6.4 Kullanıcı Deneyimi ve Geri Bildirim Potansiyeli

Uygulamanın kullanıcı arayüzü (UI) ve kullanıcı deneyimi (UX) basit ve anlaşılırdır. Her yaştan ve teknik beceri düzeyinden kullanıcının kolaylıkla kullanabilmesi için büyük düğmeler, kontrast renkler ve mantıklı menüler ön plana çıkarılmıştır. Oyun ekranı, harflerin sorunsuz bir şekilde girilebilmesi için basitleştirilmiş ve renkli geri bildirimler kullanıcıya anında bilgi sağlamaktadır.

Kullanıcı testleri sonucunda, kullanıcıların büyük çoğunluğu uygulamanın kullanım kolaylığını ve arayüzün basitliğini olumlu olarak değerlendirmiştir. Test sırasında, kullanıcıların %90'ı arayüzü anlaşılır bulmuş ve %85'i oyundaki akışın hızlı ve sorunsuz olduğunu fark etmiştir. Dokunmatik ekran desteği sayesinde mobil cihazlarda kullanıcı deneyimi de geliştirilmiştir.

Uygulama, hedef pazarının ilgisini çekecek şekilde tasarlanmıştır (kelime oyunu hayranları, öğrenciler ve günlük zihinsel uyarılmaya ihtiyaç duyan kişiler). Uygulama ayrıca, kullanıcı görüşlerini toplamak ve değerlendirmek için kolayca entegre edilebilen bir altyapıya sahiptir. Bu, gelecekteki güncellemelerin kullanıcı önerilerine göre geliştirilmesini sağlar.

6.5 Sınırlamalar

Geliştirilen uygulama bazı platform ve veri kaynağı sınırlamalarına sahiptir.

6.5.1 Düşük Kaliteli Cep Telefonlarında Karşılaşılan Sorunlar:

Uygulama yeni akıllı telefonlarda sorunsuz çalışsa da, RAM ve işlem kapasitesi daha düşük olan eski telefonlarda ilk model eğitimi sırasında gecikme yaşanabilir. Özellikle AI tahmin özelliği ilk kez kullanıldığında, model eğitimi birkaç saniye sürebilir ve bu da düşük kaliteli telefonlarda kullanıcı deneyimini olumsuz etkileyebilir.

6.5.2 Veri Setinin Güncelliği ve Kısıtlamaları:

Kullanılan veri kümesi terimi, yaygın İngilizce kelimeleri içerir. Ancak, veri kümesi zamanla güncelliğini yitirebilir ve yeni kelimeler veya modern terimler eksik olabilir. Ayrıca, program İngilizce kelimelerle sınırlıdır; diğer diller veya çok dilli destek için başka veri kümeleri gereklidir. Çevresel sınırlamalar nedeniyle, uygulama yalnızca önceden yüklenmiş kelime listelerini çevrimdışı olarak destekleyebilir ve internet bağlantısı olmadan verilerin güncellenmesi mümkün değildir.

6.6 Gelecek Çalışma için Öneriler

Mevcut uygulama, temel işlevlerini iyi bir şekilde yerine getirmektedir ancak gelecekte aşağıdaki genişletme ve platform iyileştirmeleri planlanmaktadır:

- **Çoklu Cihaz Uyumluluğu:** Uygulama, kullanıcılar arasında erişilebilirliği artırmak için giyilebilir cihazlar (akıllı saatler) ve IoT(Nesnelerin İnterneti) tabanlı platformlarda kullanım için optimize edilebilir.
- **Yeni Arayüzler ve Çoklu Dil Desteği:** Daha verimli ve dinamik arayüzler kullanılarak kullanıcı deneyimi iyileştirilebilir. Ayrıca, çok dilli destek sayesinde uygulamanın farklı dilleri desteklemesi beklenmektedir.
- **Gerçek Zamanlı Analiz ve Sesli Asistan Entegrasyonu:** Kullanıcılar, oyun içinde gerçek zamanlı analiz ve uyarılar alabilir ve sesli asistan entegrasyonu sayesinde uygulamaya daha kolay erişebilir.
- **Dinamik Model Güncellemeleri (Çevrimiçi Öğrenme):** Amaç, kullanıcılardan toplanan verilere dayalı olarak modeli çevrimiçi olarak güncellemek ve böylece zaman içinde tahmin doğruluğunu artırmaktır.

Bu tür iyileştirmeler, uygulamanın teknik özellikleri ve kullanıcılara sağladığı avantajlar üzerinde önemli bir etki yaratabilir.

REFERENCES

- [1] Majid Hameed Ahmed, Sabrina Tiun, Nazlia Omar, and Nor Samsiah Sani. A multi-view representation technique based on principal component analysis for enhanced short text clustering. *Plos one*, 19(8):e0309206, 2024.
- [2] Güler Alparslan and Mahir Dursun. Konvolüsyonel sinir ağıları tabanlı türkçe metin sınıflandırma. *Bilişim Teknolojileri Dergisi*, 16(1):21–31, 2023.
- [3] Tariq Alqahtani, Hisham A Badreldin, Mohammed Alrashed, Abdulrahman I Alshaya, Sahar S Alghamdi, Khalid Bin Saleh, Shuroug A Alowais, Omar A Alshaya, Ishrat Rahman, Maged S Al Yami, et al. The emergent role of artificial intelligence, natural learning processing, and large language models in higher education and research. *Research in social and administrative pharmacy*, 19(8):1236–1242, 2023.
- [4] Altynbek Amirzhanov, Cemil Turan, and Alfira Makhmutova. Plagiarism types and detection methods: a systematic survey of algorithms in text analysis. *Frontiers in Computer Science*, 7:1504725, 2025.

- [5] Özlem Aydın and Hüsein Kantarcı. Türkçe anahtar sözcük çıkarımında lstm ve bert tabanlı modellerin karşılaştırılması. *Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 17(1):9–18, 2021.
- [6] Murat Aydoğan and Ali Karci. Improving the accuracy using pre-trained word embeddings on deep neural networks for turkish text classification. *Physica A: Statistical Mechanics and its Applications*, 541:123288, 2020.
- [7] Charaf Eddine Benarab and Shenglin Gui. Cnn-trans-enc: A cnn-enhanced transformer-encoder on top of static bert representations for document classification. *arXiv preprint arXiv:2209.06344*, 2022.
- [8] Abdulkadir Bilen and Ahmet Bedri Özer. Siber saldırılar için rastgele orman algoritması kullanılarak öznelik seçimi. *Fırat Üniversitesi Fen Bilimleri Dergisi*, 34(1):31–37, 2021.
- [9] Federica Biotti, Sarah Ahmad, Racquel Quinn, and Rebecca Brewer. Development and validation of the interoceptive states static images (issi) database. *Behavior Research Methods*, pages 1–22, 2022.
- [10] Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Junichi Tsujii. Characterbert: Reconciling elmo and bert for word-level open-vocabulary representations from characters. *arXiv preprint arXiv:2010.10392*, 2020.
- [11] Samet Büyükkada. Akademik yazımda yapay zekâ kullanımının etik açıdan incelenmesi: Chatgpt örneği. *Rize İlahiyat Dergisi*, 26(26):1–12, 2024.
- [12] Özer Çelik and Burak Can Koç. Tf-idf, word2vec ve fast-text vektör model yöntemleri ile türkçe haber metinlerinin sınıflandırılması. *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, 23(67):121–127, 2021.
- [13] Chenghong Cen, Guang Luo, Lujia Li, Yilin Liang, Kang Li, Tan Jiang, and Qiang Xiong. User-centered software design: user interface redesign for blockly–electron, artificial intelligence educational software for primary and secondary schools. *Sustainability*, 15(6):5232, 2023.
- [14] Yun-Gyung Cheong, Alaina K Jensen, Elín Rut Guðnadóttir, Byung-Chull Bae, and Julian Togelius. Detecting predatory behavior in game chats. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(3):220–232, 2015.
- [15] Mamata Das, PJA Alphonse, et al. A comparative study on tf-idf feature weighting method and its analysis using unstructured dataset. *arXiv preprint arXiv:2308.04037*, 2023.
- [16] Lucas BV de Amorim, George DC Cavalcanti, and Rafael MO Cruz. The choice of scaling technique matters for classification performance. *Applied Soft Computing*, 133:109924, 2023.
- [17] Isaac Martín De Diego, Ana R Redondo, Rubén R Fernández, Jorge Navarro, and Javier M Moguerza. General performance score for classification problems. *Applied Intelligence*, 52(10):12049–12063, 2022.
- [18] Burak Delican. Okuma yazma öğretimine yönelik geliştirilmiş mobil uygulamalar; olanaklar ve sınırlılıklar. *Abant İzzet Baysal Üniversitesi Eğitim Fakültesi Dergisi*, 21(2):682–703, 2021.
- [19] Kismet Deliveli. Dijital teknoloji destekli ses tabanlı hece öğretim yöntemi ile okuma yazmanın öğretilmesi (vaka çalışması). *Journal of Computer and Education Research*, 10(19):103–136, 2022.
- [20] John P Doran. *Unity 2022 Mobile Game Development: Build and publish engaging games for Android and iOS*. Packt Publishing Ltd, 2023.
- [21] İsa Ergin and Timur İnan. Kodlayıcı kod çözücü ve dikkat algoritmaları kullanılarak karakter tabanlı kelime üretimi. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 39(3):1999–2010, 2024.
- [22] Ömer Can Eskicioğlu and Ali Hakan Isık. Mobil uyumlu çoklu dil destekli hibrit şifreleme algoritması. *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, 24(72):1007–1019, 2022.
- [23] Maximiliano García, Sebastián Maldonado, and Carla Vairetti. Efficient n-gram construction for text categorization using feature selection techniques. *Intelligent Data Analysis*, 25(3):509–525, 2021.
- [24] Andrea Gasparetto, Matteo Marcuzzo, Alessandro Zangari, and Andrea Albarelli. A survey on text classification algorithms: From text to predictions. *Information*, 13(2):83, 2022.
- [25] Tsvetanka Georgieva-Trifonova and Mahmut Duraku. Research on n-grams feature selection methods for text classification. In *IOP Conference Series: Materials Science and Engineering*, volume 1031, page 012048. IOP Publishing, 2021.
- [26] Alfin Gerliandeva, Yulison Chrisnanto, Herdi Ashaury, et al. Optimasi klasifikasi sentimen pada komentar online menggunakan multinomial naïve bayes dan ekstraksi fitur tf-idf serta n-grams. *Jurnal Pekommas*, 9(2):260–272, 2024.
- [27] Kuncoro Hadi and Ema Utami. Analysis of k-nn with the integration of bag of words, tf-idf, and n-grams for hate speech classification on twitter. *JUITA: Jurnal Informatika*, 12(2):289–298, 2024.
- [28] Hulya Hark, Meral Karakurt, Cengiz Hark, and Ali Karci. Glove kelime gömmeleri ve sinir ağları ile haber metinlerinin sınıflandırılması. *International Journal of Pure and Applied Sciences*, 9(1):175–187, 2023.
- [29] Yang Hu. The research on the plight of chinese mobile game development and the development direction in future. In *2022 International Conference on Science Education and Art Appreciation (SEAA 2022)*, pages 1387–1395. Atlantis Press, 2022.
- [30] Kaibin Huang, Hai Wu, Zhiyan Liu, and Xiaojuan Qi. In-situ model downloading to realize versatile edge ai in 6g mobile networks. *IEEE Wireless Communications*, 30(3):96–102, 2023.
- [31] Yunpeng Huang, Jingwei Xu, Junyu Lai, Zixu Jiang, Taolue Chen, Zenan Li, Yuan Yao, Xiaoxing Ma, Lijuan Yang, Hao Chen, et al. Advancing transformer architecture in long-context large language models: A comprehensive survey. *arXiv preprint arXiv:2311.12351*, 2023.
- [32] Nasir Jalal, Arif Mehmood, Gyu Sang Choi, and Imran Ashraf. A novel improved random forest for text classification using feature ranking and optimal number of trees. *Journal of King Saud University-Computer and Information Sciences*, 34(6):2733–2742, 2022.
- [33] Andreas Jungherr and Damien B Schlarb. The extended reach of game engine companies: How companies like epic games and unity technologies provide platforms for extended reality applications and the metaverse. *Social Media+ Society*, 8(2):20563051221107641, 2022.
- [34] Bora Kat. Mühendislik alanındaki türkçe akademik metinler için makine öğrenmesi destekli doğal dil işleme çalışmaları ve bir karar destek sisteminin geliştirilmesi: Tübitak projeleri örneği. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 38(3):1879–1892, 2023.

- [35] Madiha Khalid, Muhammad Murtaza Yousaf, and Muhammad Umair Sadiq. Toward efficient similarity search under edit distance on hybrid architectures. *Information*, 13(10):452, 2022.
- [36] Doğançan Kınık and Aysun Güran. Tf-idf ve doc2vec tabanlı türkçe metin sınıflandırma sisteminin başarımlarının ardışık kelime grubu tespiti ile artırılması. *Avrupa Bilim ve Teknoloji Dergisi*, 21(21):323–332, 2021.
- [37] Victor Kuperman, Sascha Schroeder, and Daniil Gnetov. Word length and frequency effects on text reading are highly similar in 12 alphabetic languages. *Journal of Memory and Language*, 135:104497, 2024.
- [38] Mehmet Salih Kurt and Eylem Yücel Demirel. Web page classification with deep learning methods. *Uludağ Üniversitesi Mühendislik Fakültesi Dergisi*, 27(1):191–204, 2022.
- [39] Sharmad Rajnish Lawande, Graceline Jasmine, Jani Anbarasi, and Lila Iznita Izhar. A systematic review and analysis of intelligence-based pathfinding algorithms in the field of video games. *Applied Sciences*, 12(11):5499, 2022.
- [40] Ga Young Lee, Lubna Alzamil, Bakhtiyar Doskenov, and Arash Termehchy. A survey on data cleaning methods for improved machine learning model performance. *arXiv preprint arXiv:2109.07127*, 2021.
- [41] Xiaozhou Li, Zheyang Zhang, and Kostas Stefanidis. Mobile app evolution analysis based on user reviews. In *New Trends in Intelligent Software Methodologies, Tools and Techniques*, pages 773–786. IOS Press, 2018.
- [42] Yao Li, Minhao Cheng, Cho-Jui Hsieh, and Thomas CM Lee. A review of adversarial attack and defense for classification methods. *The American Statistician*, 76(4):329–345, 2022.
- [43] Minhui Liang and Tiansen Niu. Research on text classification techniques based on improved tf-idf algorithm and lstm inputs. *Procedia Computer Science*, 208:460–470, 2022.
- [44] Zainab Mansur, Nazlia Omar, Sabrina Tiun, and Eissa M Alshari. A normalization model for repeated letters in social media hate speech text based on rules and spelling correction. *Plos one*, 19(3):e0299652, 2024.
- [45] Jiazhi Mo, Hailu Kuang, and Xiaoqi Li. Password strength detection via machine learning: Analysis, modeling, and evaluation. *arXiv preprint arXiv:2505.16439*, 2025.
- [46] Alhassan Mumuni and Fuseini Mumuni. Automated data processing and feature engineering for deep learning and big data applications: a survey. *Journal of Information and Intelligence*, 2024.
- [47] Kuzma Mustač, Krešimir Bačić, Lea Skorin-Kapov, and Mirko Sužnjević. Predicting player churn of a free-to-play mobile video game using supervised machine learning. *Applied Sciences*, 12(6):2795, 2022.
- [48] Mustafa Özkan and Görkem Kar. Türkçe dilinde yazılan bilimsel metinlerin derin öğrenme tekniği uygulanarak çoklu sınıflandırılması. *Mühendislik Bilimleri ve Tasarım Dergisi*, 10(2):504–519, 2022.
- [49] Cevahir Parlak. Konuşma duygu tanıma uygulamalarında hiper parametre optimizasyonu ile derin öğrenme metodlarının geliştirilmesi. *Karadeniz Fen Bilimleri Dergisi*, 14(4):1955–1975, 2024.
- [50] José Pedro Pinto, André Pimenta, and Paulo Novais. Deep learning and multivariate time series for cheat detection in video games. *Machine Learning*, 110(11):3037–3057, 2021.
- [51] Md Rafi-Ur-Rashid, Sami Azam, and Mirjam Jonkman. Feature extraction using deep generative models for bangla text classification on a new comprehensive dataset. *arXiv preprint arXiv:2308.13545*, 2023.
- [52] Oona Rainio, Jarmo Teuvo, and Riku Klén. Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, 14(1):6086, 2024.
- [53] Tri Puspa Rinjeni, Ade Indriawan, and Nur Aini Rakhmawati. Matching scientific article titles using cosine similarity and jaccard similarity algorithm. *Procedia Computer Science*, 234:553–560, 2024.
- [54] Nova Adi Saputra, Khurotul Aeni, and Nurul Mega Saraswati. Indonesian hate speech text classification using improved k-nearest neighbor with tf-idf-icspf. *Scientific Journal of Informatics*, 11(1):21–30, 2024.
- [55] I Gede Wiarta Sena and Andi WR Emanuel. Mobile legend game prediction using machine learning regression method. *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, 9(2):221–230, 2023.
- [56] Yudi Setiawan, Nur Ulfa Maulidevi, and Kridanto Surendro. The optimization of n-gram feature extraction based on term occurrence for cyberbullying classification. *Data Science Journal*, 23(1), 2024.
- [57] Saichon Sinsomboonthong. Performance comparison of new adjusted min-max with decimal scaling and statistical column normalization methods for artificial neural network classification. *International Journal of Mathematics and Mathematical Sciences*, 2022(1):3584406, 2022.
- [58] Tim D Smithies, Mark J Campbell, Niall Ramsbottom, and Adam J Toth. A random forest approach to identify metrics that best predict match outcome and player ranking in the esports rocket league. *Scientific reports*, 11(1):19285, 2021.
- [59] Vasileios Stavropoulos, Daniel Zarate, Maria Prokofieva, Noirin Van de Berg, Leila Karimi, Angela Gorman Alesi, Michaela Richards, Soula Bennet, and Mark D Griffiths. Deep learning (s) in gaming disorder through the user-avatar bond: A longitudinal study using machine learning. *Journal of Behavioral Addictions*, 12(4):878–894, 2023.
- [60] S Suganyadevi, V Seethalakshmi, and Krishnasamy Balasamy. A review on deep learning in medical image analysis. *International Journal of Multimedia Information Retrieval*, 11(1):19–38, 2022.
- [61] Burcu Türkmen and Sema Koçlu. Yapay zeka destekli haber metni üretimi ve çevirilerinin karşılaştırmalı bir analizi: Chatgpt-4o örneği. *IU Journal of Translation Studies*, 21(21):212–229, 2024.
- [62] Jorge Valente, João António, Carlos Mora, and Sandra Jardim. Developments in image processing using deep learning and reinforcement learning. *Journal of Imaging*, 9(10):207, 2023.
- [63] Ülku Veranyurt, Ahmet Deveci, M Fevzi Esen, and Ozan Veranyurt. Makine öğrenmesi teknikleriyle hastalık sınıflandırması: Random forest, k-nearest neighbour ve adaboost algoritmaları uygulaması. *Uluslararası Sağlık Yönetimi ve Stratejileri Araştırma Dergisi*, 6(2):275–286, 2020.
- [64] Laura von Rueden, Sebastian Mayer, Rafet Sifa, Christian Bauckhage, and Jochen Garcke. Combining machine learning and simulation to a hybrid modelling approach: Current and future directions. In *Advances in Intelligent Data Analysis XVIII: 18th International Symposium on Intelligent Data Analysis, IDA 2020, Konstanz, Germany, April 27–29, 2020, Proceedings 18*, pages 548–560. Springer, 2020.
- [65] Haitao Wang, Jie He, Xiaohong Zhang, and Shufen Liu. A short text classification method based on n-gram and cnn.

- Chinese Journal of Electronics*, 29(2):248–254, 2020.
- [66] Le Wang, Meng Han, Xiaojuan Li, Ni Zhang, and Haodong Cheng. Review of classification methods on unbalanced data sets. *Ieee Access*, 9:64606–64628, 2021.
 - [67] Adi Widiyanto, Eka Pebriyanto, Fitriyanti Fitriyanti, and Marna Marna. Document similarity using term frequency-inverse document frequency representation and cosine similarity. *Journal of Dinda: Data Science, Information Technology, and Data Analytics*, 4(2):149–153, 2024.
 - [68] Yuening Yan, Yi Li, Xinyu Lou, Senqi Li, Yutong Yao, Diankun Gong, Weiyi Ma, and Guojian Yan. The influence of action video gaming experience on the perception of emotional faces and emotional word meaning. *Neural Plasticity*, 2021(1):8841156, 2021.
 - [69] İbrahim Yazıcı, İbraheem Shayea, and Jafri Din. A survey of applications of artificial intelligence and machine learning in future mobile networks-enabled systems. *Engineering Science and Technology, an International Journal*, 44:101455, 2023.
 - [70] Meliz Yuvalı, Belma Yaman, and Özgür Tosun. Classification comparison of machine learning algorithms using two independent cad datasets. *Mathematics*, 10(3):311, 2022.