Konvolüsyonel Sinir Ağları Tabanlı Türkçe Metin Sınıflandırma

Araştırma Makalesi/Research Article

Güler ALPARSLAN[1], Mahir DURSUN[2]

[1]Bilişim Enstitüsü, Gazi Üniversitesi, Ankara, Türkiye
[2]Teknoloji Fakültesi, Gazi Üniversitesi, Ankara, Türkiye
guler.alparslan@gazi.edu.tr, mdursun@gazi.edu.tr
(Geliş/Received:22.08.2022; Kabul/Accepted:09.12.2022)
DOI: 10.17671/gazibtd.1165291

Özet— Bu çalışmada makine öğrenmesi teknikleri ve konvolüsyonel sinir ağları (KSA) tabanlı bir derin öğrenme modeli kullanılarak iki farklı Türkçe metin veri kümesi sınıflandırılmıştır. Metin sınıflandırma çalışmasında Rastgele Orman (RO), Naive Bayes (NB), Destek Vektör Makineleri (DVM), K-En Yakın Komşu (KNN) Algoritmaları ve geliştirilen KSA tabanlı derin öğrenme modeli seçilen veri kümelerine uygulanmıştır. Türkçe dilinde seçilen veri kümeleri, metin ve sınıf adedi olarak birbirinden farklı yapıda tercih edilmiş böylece kelime vektör boyutunun aynı deney ortamında sınıflandırma başarısına etkisi araştırılmıştır. Kelime temsil yöntemi olarak Terim Frekansı-Ters Doküman Frekansı (TF-IDF) belirlenmiş olup, sınıflandırma işlemi öncesi veri kümelerine uygulanan durdurma kelimeleri filtreleme ve kök bulma önişlemlerinin de sınıflandırma sonuçlarına katkısı değerlendirilmiştir. Ayrıca kelime temsil vektörlerine öznitelik seçimi uygulanarak boyutları düşürülmüş, böylece nihai vektör boyutunun da sonuçlara etkisi araştırılmıştır. Bahsedilen tüm ön işlemlerin farklı birleşimleri uygulanarak ortaya çıkan kelime vektörlerinin sınıflandırması sonucunda doğruluk ve F1-skor değerleri karşılaştırılmıştır. Karşılaştırmalar her bir sınıflandırma algoritması özelinde ayrı tablolar halinde sunulmuştur. Ayrıca tüm algoritmaların birbiri ile karşılaştırmasını içeren tablolar oluşturularak sonuçlar analiz edilmiştir.

Anahtar Kelimeler: metin sınıflandırma, makine öğrenmesi, derin öğrenme, konvolüsyonel sinir ağları

Convolutional Neural Networks Based Turkish Text Classification

Abstract— In this study, a text classification has been carried out on two different Turkish datasets using machine learning techniques and a deep learning model based on convolutional neural networks (CNN). In the text classification study, Random Forest, Naive Bayes, Support Vector Machines, K-Nearest Neighbor algorithms and a CNN based deep learning model were used. The datasets selected in Turkish are different from each other in terms of the number of texts and the number of classes. In this way, the effect of word embedding size on classification success was investigated. As a word embedding method, we preferred Term Frequency-Inverse Document Frequency (TF-IDF). The effects of the stopwords eliminating and lemmatizing pre-processes applied before the classification study, on the classification success was also evaluated. In addition, the size of the word embeddings was reduced by applying feature selection, and the effect of the final vector size on the results was investigated. The accuracy and F1-score values were compared as a result of the classification of the feature vectors by applying different combinations of the pre-processes. The comparisons are represented in separate tables for each classification algorithm used. In addition, F1-score comparison tables of the algorithms with each other are presented and the values were analyzed.

Keywords: text classification, machine learning, deep learning, convolutional neural networks

1. GİRİŞ (INTRODUCTION)

Otomatik metin sınıflandırma haber yazılarının etiketlenmesinde, doküman yazarının tespitinde, e-posta sınıflandırma, istenmeyen e-posta tespiti gibi alanlarda yaygın olarak kullanılmaktadır. Metnin içeriğindeki kelimeler ve bunların kullanılma sıklığına göre sınıfının belirlenme işlemi olarak tanımlanan metin sınıflandırma, günümüz teknolojisinde artan veri boyutuyla birlikte manuel olarak yapılması zor bir işlem haline gelmiştir. Bu durum metin sınıflandırma işleminin otomatik yapılmasını gerekli kılmıştır.

Metin sınıflandırma konusunda yapılmış birçok çalışma mevcuttur. Yaygın olarak kullanılan sınıflandırma yöntemleri; Destek Vektör Makineleri, Naive Bayes, Karar Ağaçları, K-En Yakın Komşu Modeli, Yapay Sinir Ağları, Maksimum Entropi Modelleri, Bulanık Mantık Teorisi Yaklaşımları ve Derin Öğrenme algoritmalarıdır [1].

Makine öğrenmesi, sınıflandırma, modelleme ve tahmin gibi günlük hayattaki birçok problemin çözümünde başarılı sonuç veren yöntemler içerir. Destek Vektör Makineleri, Naive Bayes ve K-En Yakın Komşu algoritmaları ile yapılmış sınıflandırma karşılaştırması içeren birçok çalışma bulunmaktadır. Makine öğrenmesi yöntemlerinden denetimli bir öğrenme algoritması olan Rastgele Orman (RO), metin sınıflandırmadaki gibi yüksek boyutlu verilerin sınıflandırması için uygun bir algoritmadır. Liu ve arkadaşları, yaptıkları çalışmada metin sınıflandırma için Semantiğe Duyarlı Rastgele Orman (SARF) sınıflandırıcı önermişlerdir. SARF'ın sınıflandırma performansını 30 adet metin veri kümesi üzerinde değerlendirerek güncel sınıflandırma yöntemleriyle karşılaştırmış, önerilen yaklaşımın üstün performansını gözlemlemişlerdir [2, 3].

Üzerinde çalışılan veri kümesine uygulanan ön işlemler sınıflandırma başarısını büyük ölçüde etkilemektedir. Bu amaçla yapılmış birçok çalıma mevcuttur. Sınıflandırma sırasındaki hesaplama karmaşıklığını azaltmak ve analitik performansı artırmak amacıyla, sınıflandırma başarısına en çok katkı sağlayan özniteliklerin belirlenmesi gerekmektedir. Bu amaçla genetik algoritma optimizasyonu ile öznitelik seçimi yöntemi uygulayan çalışmalar, metindeki tüm özniteliklerin kullanıldığı çalışmalardan daha yüksek bir sınıflandırma başarısı göstermişlerdir [2, 3]. Veri kümesinin alt uzay boyutunu azaltma amacıyla tasarlanan kaotik optimizasyon algoritmasına dayalı metin sınıflandırma modelleri ile, az sayıda öznitelikle daha iyi sınıflandırma performansı elde eden çalışmalar da mevcuttur [4, 5].

Metin sınıflandırmada en önemli çalışma alanlarından ikisi istenmeyen e-posta tespiti ve doküman yazar tespitidir. Makine öğrenmesi algoritmaları ile e-posta sınıflandırma yapan birçok çalışma bulunmaktadır. E-posta içerisinde geçen ve birden fazla anlama gelen kelimeler/cümleler istenmeyen e-posta tespitini oldukça zorlaştırmakta, bu zorlukla mücadele etmek için Naive Bayes algoritması kullanılabilmektedir [6]. Benzer şekilde doküman/metin yazarı tespitinde de Naive Bayes algoritması oldukça etkindir. Özellikle sosyal medya hesaplarından paylaşılan yazıların sahibinin tespitinde önemli bir role sahiptir [7].

Bu çalışmada ise klasik makine öğrenmesi algoritmaları ile konvolüsyonel sinir ağları (KSA) tabanlı bir derin öğrenme modeli kullanılarak iki farklı veri kümesinde sınıflandırma yapılmıştır. Türkçe dilinde seçilen veri kümeleri, metin ve sınıf adedi olarak birbirinden farklı yapıda tercih edilmiş böylece kelime vektörü boyutunun aynı deney ortamında sınıflandırma başarısına etkisi gözlemlenebilmiştir. Ayrıca bu çalışma, sınıflandırma işlemi öncesi veri kümelerine uygulanan üç farklı ön işlemin de başarıya katkısını değerlendirmek adına önemli bir çalışmadır. Kelime vektörlerine öznitelik seçimi uygulanarak boyut azaltılmış, nihai vektör boyutunun da sonuçlara etkisi böylece gözlemlenebilmiştir. Bahsedilen tüm ön işlemlerin farklı birleşimleri ile ortaya çıkan kelime vektörlerine; Rastgele Orman, Naive Bayes, Destek Vektör Makineleri, K-En Yakın Komşu Algoritmaları ve KSA tabanlı derin öğrenme modeli uygulanarak sınıflandırma doğruluk oranları ve F1-skor değerleri karşılaştırılmıştır.

Çalışmanın devamı şu şekilde düzenlenmiştir. İkinci bölümde ilgili çalışmalar sunulmuş, üçüncü bölümde metin sınıflandırma ana hatları ile anlatılarak başlıca metin sınıflandırma önişlemleri açıklanmıştır. Dördüncü bölümde metin sınıflandırmada kullanılan makine öğrenmesi ve derin öğrenme algoritmaları tanıtılmış, beşinci bölümde sınıflandırma ölçütleri tanımlanmıştır. Altıncı bölümde tasarlanan KSA tabanlı derin öğrenme modeli açıklanmış, yedinci bölümde çalışmada kullanılan veri kümesi tanıtılmıştır. Sekizinci bölümde deneysel sonuçlara, dokuzuncu bölümde ise çalışmanın sonuçlarına yer verilmiştir.

2. İLGİLİ ÇALIŞMALAR (RELATED WORK)

Türkçe metinlerin sınıflandırılması konusunda literatürde kısıtlı sayıda çalışma bulunmaktadır. Türkçe'nin morfolojik yapısı itibariyle diğer dillerden farklı olması, eş sesli kelimelerin fazla miktarda bulunması, anlam darlıkları, özel karakterler, özellikle de sondan eklemeli diller kategorisinde olması sebebiyle metin sınıflandırmada bazı zorluklara sebep olmaktadır. Bir başka problem ise, sonuna eklenen ekler sebebiyle aynı kelimelerin farklı birer öznitelik gibi algılanması, böylece kelime temsil vektörünün boyutunun çok yükselmesidir. Bu da sınıflandırma performansını ve başarısı önemli ölçüde düşürmektedir. Yine de bu zorlukları veri madenciliği ön işlemleri ile aşan, farklı sınıflandırma modelleri ile destekleyen Türkçe metin sınıflandırma konusunda değerli çalışmalar bulunmaktadır.

Acı ve Çırak, Word2Vec metodu ile zenginleştirerek kullandıkları Konvolüsyonel Sinir Ağları (KSA) modelinden elde ettikleri sonuçları, Kılınç ve arkadaşları [8] tarafından yapılmış olan klasik istatistiksel ve makine

öğrenmesine dayalı sınıflandırma çalışması sonuçları ile karşılaştırmışlardır. Karşılaştırma sonucunda daha yüksek doğruluk oranıyla (%93,3) Türkçe haber metinlerinin sınıflandırmasını gerçekleştirmişlerdir [9].

Uçan ve arkadaşları ise yapmış oldukları çalışmada Türkçe sosyal medya metinlerini içerdikleri duyguya göre sınıflandırmışlardır. Deneysel çalışma sonuçlarına göre, önerdikleri ön eğitimli duygu modelinin önceki çalışmalarda kullanılan yöntemlere göre en yüksek başarı oranına sahip olduğu görülmüştür [10].

Aydoğan ve Karcı, eğitim ve sınıflandırma işlemlerinde kullanılmak üzere iki büyük Türkçe veri kümesi oluşturmuş, çeşitli derin öğrenme yöntemleri ile yaptıkları sınıflandırma işlemlerinin sonuçlarını karşılaştırmışlardır. Deneysel sonuçlara göre GRU ve LSTM yöntemlerinin diğer derin öğrenme modellerinden daha başarılı olduğu görülmüştür. Ayrıca ön eğitimli kelime vektörlerinin sınıflandırma doğruluk oranını %5-%7 oranında arttırdığı tespit edilmiştir [11].

Toroslu ve Karagöz ise çalışmalarında bireylerin sosyal medya mesajları ile beş büyük kişilik özelliği arasındaki ilişkiyi denetimli bir öğrenme problemi olarak modellemeyi amaçlamışlardır. Türkçe ve İngilizce iki ayrı veri kümesi üzerinde yaptıkları deneysel sonuçlarda, yapay sinir ağları yaklaşımın kişilik tahmini için başarılı bir şekilde kullanılabileceğini, vektör tabanlı sınıflandırma modellerine benzer bir performansla çalıştığını göstermişlerdir [12].

Yıldırım ve Yıldız, Türkçe haber metinleri üzerinde yaptıkları sınıflandırma çalışmasında, geleneksel kelime torbası yöntemi ile sinir ağı temelli kelime temsil yöntemlerinin başarı oranlarını karşılaştırmışlardır. Deneysel sonuçlarda kelime temsil vektörlerinin oluşturulmasında kullanılan geleneksel yöntemlerin hala yeni nesil yöntemlerle yarışacak düzeyde sınıflandırma başarısı sağladığını göstermişlerdir [13].

Köksal ve Yılmaz, literatürde yaygın şekilde kullanılan iki Türkçe haber veri kümesi üzerinde hem klasik makine öğrenme algoritmaları hem de güncel ön eğitimli dil modellerini kullanarak metin sınıflandırma çalışması yapmışlardır. Yapmış oldukları çalışmada sınıflandırma modellerinin parametre seçimi ve optimizasyonu üzerine yoğunlaşarak başarılı sonuçlar elde etmişlerdir. Ön eğitimli dil modelleri ile yapılan sınıflandırma sonuçları, aynı veri kümesini kullanan benzer çalışmalardan daha yüksek F1 skoru elde etmiştir [14].

3. METİN SINIFLANDIRMA (TEXT CLASSIFICATION)

Metin sınıflandırma araştırmasında kullanılabilecek veriler haber siteleri, online alışveriş platformları, sanal olarak yayınlanan dergiler gibi birçok kaynaktan elde edilebilir. Yapılacak araştırmanın doğruluğunu test edebilmek için metnin kategorisinin etiketler, anahtar kelimeler ile belirtilmiş olması gereklidir.

Metinler ham veri olarak doğrudan sınıflandırma algoritmasında kullanılamamakta, belirli ön işlemlerden geçirilerek öznitelik vektörüne dönüştürülmektedir. Metinlerin öznitelik vektörüne dönüştürülmesinin başarısı doğrudan sınıflandırma başarısını etkilemektedir.

Metin sınıflandırma alanındaki veri kümeleri diğer birçok veri kümesi ile karşılaştırıldığında öznitelik vektör boyutunun çok yüksek (binler mertebesinde) olduğu görülmektedir. Metinlerde geçen her bir kelime birer öznitelik olarak kullanılırsa ortaya sınıflandırma algoritmasının performansını önemli ölçüde düşürecek, hesaplama karmaşıklığına sebep olacak kadar fazla sayıda öznitelik çıkacaktır. Öznitelik vektör boyutu optimum seviyeye çekilerek hesaplama performansının ve sınıflandırma başarısının arttırılması amacıyla birçok yöntem kullanılabilmektedir. Sıklıkla kullanılan metin sınıflandırma ön işlemleri aşağıda açıklanmaktadır.

3.1. Kök Bulma (Stemming)

Kök bulma (stemming) işlemi, metinde geçen kelimelerin kelime köklerini bularak kullanılması ön işlemidir. Bu sayede birbirinden farklı ekler ile farklı bir kelimeymiş gibi görünen kelimelerin aynı sayılması sağlanmaktadır. Özellikle sondan eklemeli bir dil olan Türkçe metinlerde düşünülecek olursa, öznitelik vektör boyutunun azaltılması ve daha anlamlı vektör ortaya çıkarması açısından kritik bir ön işlemdir. Otomatik kök bulma işleminde dilin özelliklerine göre Zemberek doğal dil işleme kütüphanesi, sözcük eklerini sondan başa doğru sıyırarak çıkarma (affix stripping) ve sözcüklerin ilk n karakterinin kelime kökü olduğunu kabul etme (fixed prefix stemming) yaklaşımlarıyla geliştirilmiş kütüphaneler kullanılabilmektedir.

3.2. Durdurma Kelimeleri Filtreleme (Stopword Filtering)

Durdurma kelimeleri (DK) filtreleme işlemi, metnin sınıflandırmasında etkisiz olan, genellikle tek başın anlamsız ancak çok sık kullanıldığı için frekansı yüksek "ve", "ile", "ya da" vb. kelimelerin işlem dışı bırakılmasıdır. Durdurma kelimeleri her sınıftaki metinde çok sayıda ve benzer sıklıkta kullanıldığı için sınıf belirlenmesinde etkin bir rol oynamadığı gibi öznitelik vektör boyutunu da gereksiz yere büyütmektedir. Bu sebeple durdurma kelimelerinin filtrelenmesi yaygın şekilde kullanılan ön işlemlerden biridir.

3.3. Terim Frekansı-Ters Doküman Frekansı (Term Frequency-Inverse Document Frequency)

Terim frekansı (term frequency-TF), metinde geçen her bir kelimenin metindeki kullanılma sıklığının bulunması önişlemidir. Dokümandaki her bir terimin o dokümanda geçme adedi ile dokümandaki bütün terimlerin toplam adedine oranı şeklinde hesaplanmaktadır.

$$TF = \frac{\text{Terimin dokümanda geçme adedi}}{\text{Dokümandaki toplam terim adedi}} \quad (1)$$

Ters doküman frekansı değeri, bir terimin arandığı tüm dokümanların sayısının, o terimin bulunduğu dokümanların sayısına oranıdır. Terim ne kadar az dokumanda tekrar ediyor ise IDF değeri o kadar büyük çıkar.

$$IDF = \frac{\text{Toplam doküman sayısı}}{\text{Terimin geçtiği doküman sayısı}} \quad (2)$$

Doküman sınıflandırmada sık kullanılan önişlemlerden biri olan Terim Frekansı-Ters Doküman Frekansı (TF-IDF), her bir terim için terim frekansı ile ters doküman frekansının çarpımından elde edilir. Bir terimin bulunduğu dokümanın sınıflandırmasına katkı sağlamadaki önemini gösterir.

$$TF\text{-}IDF = TF * IDF \quad (3)$$

3.4. Word2Vec (Word2Vec)

Word2Vec, Yapay Sinir Ağları (YSA) tabanlı bir kelime vektörü temsil yöntemidir. Bu yöntem sayesinde kelimeler vektörlere dönüştürülerek aralarındaki uzaklıklar hesaplanıp kelimeler arasında analoji kurulabilmektedir. Word2Vec yöntemi son yıllarda metin sınıflaması konusunda oldukça popüler olmuştur. Birçok çalışmada sınıflama yöntemleri uygulanmadan önce Word2Vec kullanılarak veri kümesi zenginleştirilmiş ve metin verileri vektörel hale getirilmiştir [9].

Word2vec yönteminde, Continuous Bag of Words (CBOW) ve Skip-Gram olmak üzere iki alt yöntem kullanılmaktadır. CBOW modelinde her bir kelime, komşu kelimeleri girdi alınarak tahmin edilmeye çalışılır. Skip-Gram modelinde ise her bir kelime girdi alınarak ilgili kelimenin komşu kelimeleri tahmin edilir.

3.5. N-Grams (N-Grams)

Metin sınıflandırma önişlemi olarak kullanılan bir diğer yöntem olan N-Grams yöntemi, incelenen metinde geçen kelimeleri belirlenen pencere boyutunda birlikte gruplandırarak sınıflandırma işlemine girdi sağlar. Unigram 1 kelimelik grupları, bigrams 2 kelimelik grupları, trigrams 3 kelimelik grupları, n-grams ise 3'ten fazla belirlenen n sayısınca kelimelik grupları ifade eder.

Örnek olarak "beğenmedim çünkü çalışırken çok gürültü çıkarıyor" cümlesi bigram olarak bölünmek istenirse aşağıdaki vektör elde edilecektir.

"beğenmedim çünkü çalışırken çok gürültü çıkarıyor":
["beğenmedim çünkü", "çünkü çalışırken", "çalışırken çok", "çok gürültü", "gürültü çıkarıyor"]

3.6. Öznitelik Seçimi (Feature Selection)

Öznitelik seçimi, kullanılan veri kümesinin özniteliklerinden sınıflandırma başarısına en fazla katkı sağlayanların tespit edilip seçilmesi işlemidir. Öznitelik seçimindeki amaç, sınıflandırma başarısını yükseltmek veya eğitim süresini kısaltarak çalışma performansını arttırmaktır. Metin veri kümelerinin öznitelik sayısının yüksek olması sebebiyle, metin sınıflandırmada kritik bir önişlem olarak sıklıkla kullanılmaktadır.

Öznitelik seçimi işleminde; filtreleme, sarmalama ve gömülü yöntemler kategorilerinde çeşitli algoritmalar bulunmaktadır. Bunlardan öznitelik ve sınıf değişken tiplerine uygun olacak şekilde en yaygın kullanılan yöntemler; Pearson korelasyonu, Ki-kare testi, Anova testi ve Bilgi kazanımı yöntemleridir.

4. METİN SINIFLANDIRMADA KULLANILAN MAKİNE ÖĞRENMESİ VE DERİN ÖĞRENME ALGORİTMALARI (MACHINE LEARNING AND DEEP LEARNING ALGORITHMS USED IN TEXT CLASSIFICATION)

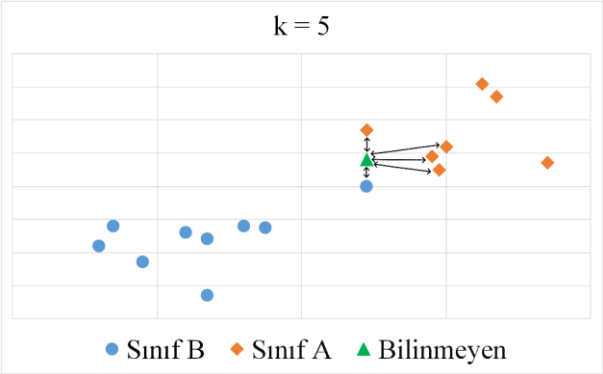
Metin sınıflandırma çalışmalarında sık uygulanan makine öğrenmesi tekniklerinden bazıları: Rastgele Orman, Naive Bayes, Destek Vektör Makineleri, K-En Yakın Komşu Algoritması ve Derin Öğrenme modelleridir.

4.1. Naive Bayes

Hızlı ve kolay uygulanabilir olduğu için metin sınıflandırması işleminde tercih edilen Naive Bayes algoritması, olasılıkçı yaklaşımla sınıflandırma yapan makine öğrenmesi yöntemlerinden biridir [2]. Bu yöntemde tüm özniteliklerin ait olduğu sınıfa göre koşullu olasılığı bulunarak test verisinin sınıfı en yakın olasılıkla tespit edilmektedir.

4.2. K-En Yakın Komşu Modeli (K-Nearest Neighbor Model)

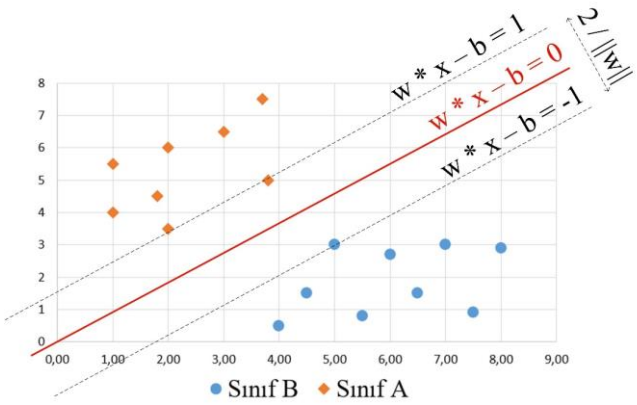
K-En Yakın Komşu (KNN) algoritmasında test örneklerinden biri seçilip, k değerine göre o örneğe en yakın örnek veya örneklerdeki sınıf belirlenerek sınıflandırma işlemi yapılmaktadır. En yakın komşu veya komşuların tespit edilebilmesi için seçilen test örneğinin tüm diğer örneklere uzaklığı hesaplanmalıdır [15]. Örneğin k=5 değeri kullanılarak uygulanan K-En Yakın Komşu Algoritması Şekil 1'de görülmektedir [16]. En yakın 5 örnekten 4'ü A sınıfına ait olduğu için bilinmeyen örneğin sınıfı A olarak tespit edilir.



Şekil 1. k=5 için k-en yakın komşu algoritması
(K-nearest neighbor algorithm with k=5)

4.3. *Destek Vektör Makineleri (Support Vector Machines)*

1992 yılında tanıtılan destek vektör makineleri (DVM), istatistiksel bilgi teorisine ve yapısal risk minimizasyonuna dayalı denetimli bir sınıflandırma algoritmasıdır [8].



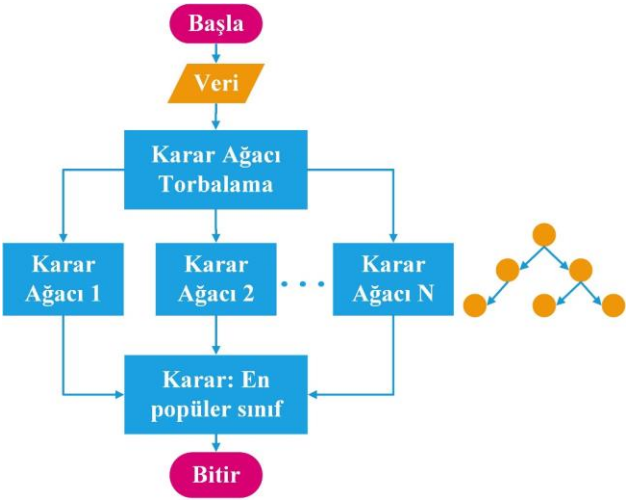
Şekil 2. Destek vektör makineleri
(Support vector machines)

Veri kümesindeki sınıflar arasındaki ayrımı en iyi belirten doğru veya hiper düzlemlerin belirlenmesi yoluyla DVM modeli uygulanır. Örnek bir DVM gösterimi Şekil 2'de görülmektedir.

4.4. *Rastgele Orman Algoritması (Random Forest Algorithm)*

Rastgele Orman (RO) algoritması karar ağacı sınıflandırıcılarından biri olarak kabul edilen denetimli bir makine öğrenmesi algoritmasıdır. Breiman tarafından bulunan sınıflandırma algoritması birden fazla karar ağacını birlikte kullanıp bunları oylayarak en uygun çözümü bulmayı hedefler [17]. Algoritmik basitliği ve yüksek boyutlu veriler için belirgin sınıflandırma performansı nedeniyle, rastgele orman, metin sınıflandırması için popüler bir yöntem haline gelmiştir [5].

Rastgele orman, torbalama ve karar ağacı algoritmalarının birleşimi olarak tanımlanabilir. Rastgele orman algoritması işleyişi Şekil 3'te görülmektedir [18].

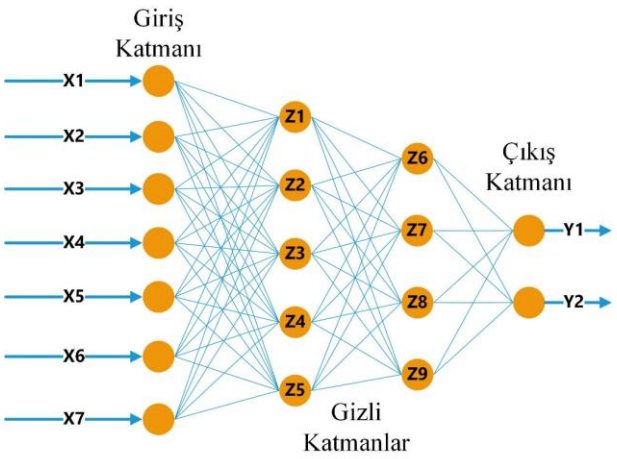


Şekil 3. Rastgele orman algoritması
(Random forest algorithm)

4.5. *Derin Öğrenme Modelleri (Deep Learning Models)*

Yapay sinir ağları, beyindeki nöronların bağlantılarını ve fonksiyonunu benzetme yoluyla geliştirilen, test kategorisindeki en gelişmiş sınıflandırıcılardan biridir. Yapay sinir ağı; giriş katmanı, gizli/ara katman ve çıktı katmanı olarak üç katmandan oluşur [19]. Yapılandırılmamış veriyi girdi olarak alır, nöronlardan oluşan katmanlarda işleyerek çıktı verir. Katman ve nöron sayısı arttıkça daha karmaşık problemleri çözebilir. Yapay sinir ağları işleyişi Şekil 4'te görülmektedir.

Yapay sinir ağları katmanlarında birden fazla gizli katman kullanıldığında derin öğrenme modeli oluşturulmuş olur. Derin öğrenme modelleri üç ana başlıkta toplanabilir: Çok Katmanlı Algılayıcılar (MLP-Multilayer Perceptrons), Konvolüsyonel Sinir Ağı (CNN-Convolutional Neural Networks) ve Tekrarlayan Sinir Ağı (RNN-Recurrent Neural Networks).



Şekil 4. Yapay sinir ağları katmanları
(Artificial neural networks layers)

5. SINIFLANDIRMA ÖLÇÜTLERİ (CLASSIFICATION METRICS)

Sınıflandırma modellerinin yapmış olduğu tahminlerin başarısını ölçmek için çeşitli sınıflandırma ölçütleri kullanılmaktadır. Sınıflandırma başarısını ölçen sınıflandırma ölçütleri karışıklık matrisindeki değerlerden hesaplanır ve sıklıkla kullanılanlar doğruluk, kesinlik, duyarlılık ve F1-skoru değerleridir.

5.1. Karışıklık Matrisi (Confusion Matrix)

Karışıklık matrisi, bir sınıflandırma işlemindeki tüm doğru ve yanlış tahminlerin sınıflara sayısal dağılımını gösteren matris gösterimidir. "Pozitif" ve "Negatif" sınıflarından oluşan bir sınıflandırma işlemi sonucu örnek bir karışıklık matrisi Tablo 1'de görülmektedir.

Tablo 1. Örnek karışıklık matrisi
(Confusion matrix sample)

Pozitif	Negatif	Tahmin / Sınıf
DP	YN	Pozitif
YP	DN	Negatif

Karışıklık matrisinde, DP doğru tahmin edilen "Pozitif" sayısını, DN doğru tahmin edilen "Negatif" sayısını, YP yanlış tahmin edilen "Pozitif" sayısını ve YN yanlış tahmin edilen "Negatif" sayısını göstermektedir. Diğer sınıflandırma ölçütleri bu değerler kullanılarak hesaplanmaktadır.

5.2. Doğruluk (Accuracy)

Doğruluk, sınıflandırma işlemi sonucundaki bütün tahminlerdeki doğru tahmin oranı olarak tanımlanmaktadır. Doğruluk değerinin matematiksel hesabı aşağıdaki eşitlikte görülmektedir.

$$\text{Doğruluk} = \frac{DP + DN}{DP + DN + YP + YN} \quad (4)$$

5.3. Kesinlik (Precision)

Kesinlik, sınıflandırma işlemi sonucundaki "Pozitif" olarak tahmin edilenlerin hangi oranda gerçekten "Pozitif" sınıfına ait olduğunu gösteren sınıflandırma ölçütüdür. Kesinlik değerinin matematiksel hesabı aşağıdaki eşitlikte görülmektedir.

$$\text{Kesinlik} = \frac{DP}{DP + YP} \quad (5)$$

5.4. Duyarlılık (Recall)

Duyarlılık, "Pozitif" sınıfındaki örneklerin hangi oranda "Pozitif" olarak tahmin edildiğini gösteren sınıflandırma ölçütüdür. Duyarlılık değerinin matematiksel hesabı aşağıdaki eşitlikte görülmektedir.

$$\text{Duyarlılık} = \frac{DP}{DP + YN} \quad (6)$$

5.5. F-1 Skoru (F-1 Score)

F1-Skor ölçütü değeri, kesinlik ve duyarlılık değerlerinin harmonik ortalaması sonucu elde edilmektedir. F1-Skor değerinin matematiksel hesabı aşağıdaki eşitlikte görülmektedir.

$$\text{F1-Skor} = 2 * \frac{\text{Kesinlik} * \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \quad (2.4)$$

6. KONVOLÜSYONEL SİNİR AĞLARI İLE METİN SINIFLANDIRMA (TEXT CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS)

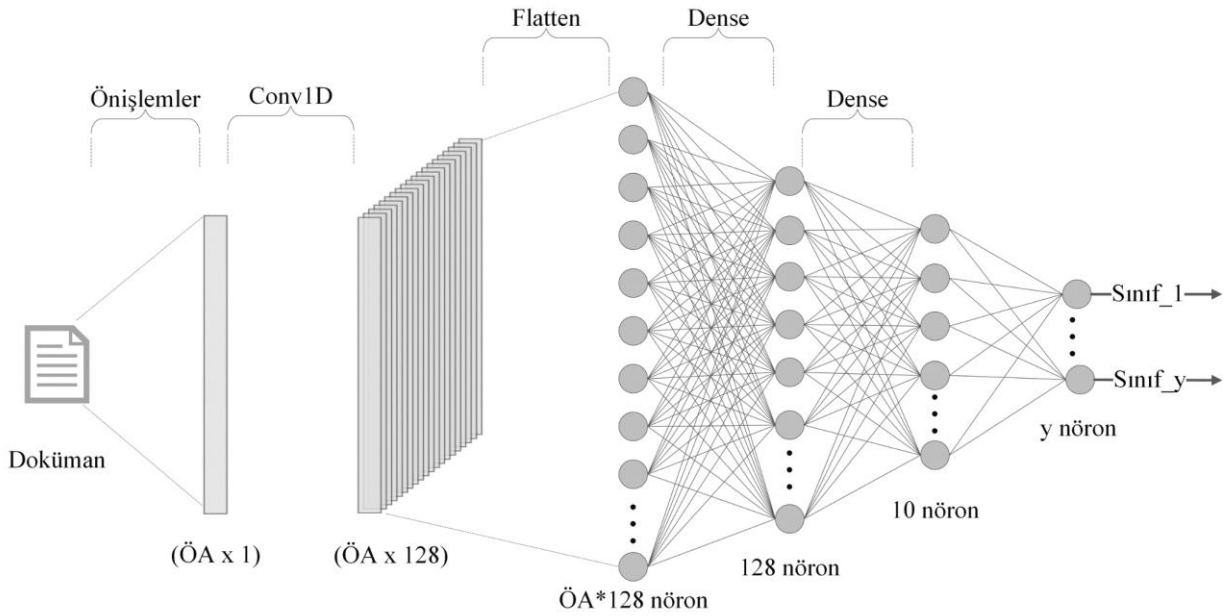
Geliştirilen derin öğrenme modelinde Konvolüsyonel Sinir Ağları (Convolutional Neural Networks-CNN) ve Tam Bağlantılı/Yoğun (Fully Connected/Dense) katmanları kullanılmıştır.

KSA, insan görme sisteminden esinlenerek tasarlanmış, çok katmanlı algılayıcı (Multi Layer Perceptron-MLP) sınıfından olan bir derin öğrenme mimarisidir. Genellikle görüntü işleme çalışmalarında önerilen bir derin öğrenme algoritması olmakla beraber, günümüzde doğal dil işleme alanlarında da etkin bir şekilde kullanılmaktadır [11].

Dense katmanı, derin öğrenme mimarilerinde yaygın ve sık kullanılan bir sinir ağı katmandır. Her nöron bir önceki katmandan girdi aldığı için tam bağlantılı (Fully Connected) katman olarak da bilinir [20].

Deneysel çalışmalarda, konvolüsyonel sinir ağları odağında kalmak kaydıyla çeşitli ek katmanlar ve hiper parametreler kullanarak elde edilen birçok farklı modelle sınıflandırma yapılmıştır. Sonuçlar göz önüne alındığında yüksek sınıflandırma başarısı gösteren, aynı zamanda eğitim süresi açısından çalışma performansı optimum olan bir modelde karar kılınmıştır. Yapılan deneysel çalışmalar neticesinde, Conv1D + Flatten + Dense + Dropout + Dense

katmanlarından oluşan derin öğrenme mimarisi modellenmiştir. Conv1D ve Dense katmanlarında, Doğrultulmuş Doğrusal Ünite (Rectified Linear Unit, ReLU) aktivasyon fonksiyonu uygulanmıştır. Uygulanan KSA tabanlı derin öğrenme modeli Şekil 5'te görülmektedir.



Şekil 5. KSA Tabanlı Derin Öğrenme Modeli, (ÖA): öznitelik adedi
(CNN based deep learning model, (ÖA): number of features)

7. VERİ KÜMESİ (DATASET)

Çalışmada Türkçe dilinde iki farklı veri kümesi kullanılmıştır: Türkçe haber metinlerinden oluşan TTC-4900[13] ve e-ticaret platformlarında yer alan ürünlere yapılmış olan Türkçe müşteri yorumlarından oluşan, çalışmada kullanacağımız kısaltmasıyla, MY-15130.

TTC-4900 veri kümesi, RSS aracılığıyla 6 farklı Türk haber portalinden toplanan haber metinlerinden hazırlanmıştır. Dünya, ekonomi, kültür, sağlık, siyaset, spor ve teknoloji olmak üzere 7 kategoriden 700'er haber içeren toplam 4900 metinden oluşmaktadır. Türkçe haber veri kümeleri arasında kullanımı kolay ve iyi belgelenmiş bir veri kümesi olan TTC-4900, erişime açıktır [21].

MY-15130 veri kümesi ise e-ticaret platformlarından çeşitli ürün yorumları çekilerek hazırlanmış, 6799 olumlu, 6978 olumsuz ve 1393 nötr, toplamda 15170 yorumdan oluşmaktadır. Kaggle platformunda erişime açık şekilde sunulmaktadır [22]. Bu çalışmada MY-15130 sınıf dağılımının dengeli olmaması sebebiyle nötr yorumlardan ayıklanmış, olumlu ve olumsuz yorumlar aynı sayıda olacak şekilde düzenlenmiştir.

Çalışmada ".csv" formatında sunulmuş olan her iki veri kümesine de sırasıyla TF-IDF dönüştürme, durdurma kelimeleri filtreleme, kök bulma ve öznitelik seçimi önişlemleri uygulanmıştır. Önişlem uygulamaları Python dilinde kodlanmıştır. Kök bulma uygulaması yerel bilgisayarda, diğer önişlem uygulamaları Google Colab platformu üzerinde koşturulmuştur.

TF-IDF dönüşümünde sklearn.feature_extraction.text kütüphanesinde sunulan TfidfVectorizer sınıfı kullanılmıştır. Kullanılan veri kümelerinin boyutlarının farklı olması sebebiyle, TF-IDF dönüşümü sonucunda birbirinden farklı boyutta çıktı oluşmuştur. TTC-4900 için 4900x110917 boyutlu, MY-15130 için 15130x17681 boyutlu vektörler elde edilmiştir.

Durdurma kelimeleri filtreleme önişleminde ise doğal dil işleme kütüphanesi olan Natural Language Toolkit'in sunmuş olduğu Türkçe durdurma kelimeleri kullanılmıştır. Veri kümeleri baştan sona taranarak durdurma kelimelerinden arındırılmıştır. Sonuçta TTC-4900 için 4900x110865 boyutlu, MY-15130 için 15130x17657 boyutlu vektörler elde edilmiştir.

Kök bulma önişleminde Türkçe'nin morfolojik yapısına uygun olarak tasarlanmış Zemberek doğal dil işleme kütüphanesi kullanılmıştır. Java dilinde geliştirilmiş açık kaynak kodlu Zemberek kütüphanesinin Python uygulamasında kullanılabilmesi için JPype kütüphanesinden faydalanılmıştır. Veri kümeleri baştan sona taranarak tüm kelimeler kelime köküne indirgenmiştir. Sonuçta TTC-4900 için 4900x34710 boyutlu, MY-15130 için 15130x8130 boyutlu vektörler elde edilmiştir.

Son olarak iki farklı boyutta çıktı almak üzere, öznitelik seçimi ön işlemi uygulanmış, uygulamada sklearn.feature_selection kütüphanesinden sunulan SelectKBest sınıfı kullanılmıştır. SelectKBest sınıfı Anova, ki-kare, bilgi kazanımı gibi birçok skor fonksiyonuna destek vermektedir. Girdi ve çıktısı

kategorik veri tipinde olan veri kümelerine uyumlu olması sebebiyle bu çalışmada ki-kare skor fonksiyonu tercih edilmiştir. Tüm önişlemlerden sonra her iki veri kümesinin boyutu düşürülerek 2000 ve 5000 adet öznitelik seçilmiş, bu farklı boyutlardaki verilerle sınıflandırma yapılarak sonuçlar kayıt altına alınmıştır.

8. DENEYSEL SONUÇLAR (THE EXPERIMENTAL RESULTS)

Klasik makine öğrenmesi algoritmaları ve uygulanan KSA tabanlı derin öğrenme modeli ile sınıflandırma çalışması Google Colab platformu üzerinde "Sklearn" ve "Tensorflow.Keras" kütüphaneleri kullanılarak geliştirilmiştir. Bu çalışmada veri kümesi eğitim ve test olarak bölünmemiş katlama(folds-k) değeri 10 verilerek çapraz doğrulama(cross-validation) yöntemi kullanılmıştır. Bu teknik sayesinde veri kümesinin tamamı parçalar halinde dönüşümlü olarak öğrenmede kullanılabilmekte, her bir parçadaki sınıflandırma çıktılarının ortalaması alınarak daha doğru sonuçlar elde edilebilmektedir.

Deneysel çalışma sonucunda, karşılaştırması yapılan tüm sınıflandırma algoritmalarının beklendiği şekilde MY-15130 veri kümesinde daha yüksek sınıflandırma başarısı gösterdiği görülmüştür. Yapısal olarak birbirinden farklı olan veri kümelerinden MY-15130'un iki sınıflı, TTC-4900'ün ise çok sınıflı olması bu sonucu ortaya çıkardığı düşünülmektedir. Sınıf sayısı azaldıkça, sınıf tahmininin doğru olma ihtimali yükselmektedir.

8.1. Multinominal Naive Bayes Uygulaması (Classification with Multinominal Naive Bayes Algorithm)

Uygulamada Naive Bayes algoritması çeşitlerinden metin sınıflandırmasına daha uygun olan Multinominal Naive Bayes modeli kullanılmış, sklearn.naive_bayes kütüphanesi MultinomialNB metodu ile geliştirme yapılmıştır. Çalışma sonucunda elde edilen sınıflandırma doğruluk değerleri Tablo 2'de görülmektedir.

Tablo 2. Naive bayes sınıflandırması doğruluk sonuçları
(Accuracy of Naive Bayes classifications)

	Öznitelik seçimi	TTC-4900	MY-15130
TF-IDF	2000	%88,6	%95,7
	5000	%90,0	%96,0
TF-IDF + DK filtre	2000	%88,6	%95,7
	5000	**%90,3**	**%96,2**
TF-IDF + DK filtre + Kök bulma	2000	%89,1	%93,8
	5000	%89,2	%93,9

Sonuçlara göre Naive Bayes sınıflandırmasında her iki veri kümesinde de en yüksek başarı sağlayan ön işlemler TF-IDF + DK filtre + 5000 öznitelik seçimi olarak görülmektedir.

8.2. K-En Yakın Komşu Uygulaması (Classification with K-Nearest Neighbor Algorithm)

K-En Yakın Komşu algoritması sınıflandırma çalışması sklearn.neighbors kütüphanesi KNeighborsClassifier metodu ile gerçekleştirilmiştir. Algoritmadaki k değerinin tek sayı olduğu sınıflandırmalarda yüksek başarı sağlaması sebebiyle k=5 olarak belirlenmiştir. Çalışma sonucunda elde edilen sınıflandırma doğruluk değerleri Tablo 3'te görülmektedir. Sonuçlara göre KNN sınıflandırmasında, her iki veri kümesinde öznitelik sayısı arttıkça sınıflandırma doğruluk oranının belirgin şekilde düştüğü görülmektedir.

Tablo 3. KNN sınıflandırması doğruluk sonuçları
(Accuracy of KNN classifications)

	Öznitelik seçimi	TTC-4900	MY-15130
TF-IDF	2000	%64,1	%84,2
	5000	%55,4	%71,9
TF-IDF + DK filtre	2000	%64,9	**%86,0**
	5000	%54,7	%74,3
TF-IDF + DK filtre + Kök bulma	2000	**%69,0**	%82,7
	5000	%66,3	%70,3

8.3. Destek Vektör Makineleri (DVM) Uygulaması (Classification with Support Vector Machines (SVM) Algorithm)

Destek Vektör Makineleri algoritması sınıflandırma çalışması sklearn.svm kütüphanesi SVC metodu ile gerçekleştirilmiştir. Çalışma sonucunda elde edilen sınıflandırma doğruluk değerleri Tablo 4'te görülmektedir. Sonuçlara göre DVM sınıflandırmasında, TTC-4900 veri kümesinde beklendiği üzere tüm önişlemler başarıya katkı sağlamıştır. Ancak MY-15130 veri kümesinde durdurma kelimeleri filtreleme ve kök bulma önişlemlerinin beklendiğinin aksine sınıflandırma başarısını düşürdüğü görülmektedir. MY-15130 veri kümesinde TTC-4900'a göre kelime çeşitliliğinin az olması sebebiyle bu sonucu ortaya koyduğu düşünülmektedir.

Tablo 4. DVM sınıflandırması doğruluk sonuçları
(Accuracy of SVM classifications)

	Öznitelik seçimi	TTC-4900	MY-15130
TF-IDF	2000	%89,0	%95,8
	5000	%90,8	**%96,2**
TF-IDF + DK filtre	2000	%88,8	%95,5
	5000	%91,1	%96,1
TF-IDF + DK filtre + Kök bulma	2000	%90,3	%94,1
	5000	**%91,2**	%94,1

8.4. Rastgele Orman (RO) Uygulaması (Classification with Random Forests(RF) Algorithm)

Rastgele Orman algoritması sınıflandırma çalışması sklearn.ensemble kütüphanesi RandomForestClassifier metodu ile gerçekleştirilmiştir. Çalışma sonucunda elde edilen sınıflandırma doğruluk değerleri Tablo 5'te görülmektedir. Sonuçlara göre, RO sınıflandırması başarı oranının uygulanan önişlemlerden DVM'e benzer şekilde etkilendiği görülmektedir. Ek olarak öznitelik seçiminde öznitelik sayısının yüksek tutulması başarıyı arttırmıştır.

Tablo 5. RO sınıflandırması doğruluk sonuçları
(Accuracy of RF classifications)

	Öznitelik seçimi	TTC-4900	MY-15130
TF-IDF	2000	%85,9	%93,6
	5000	%86,7	**%93,7**
TF-IDF + DK filtre	2000	%85,7	%93,5
	5000	%86,6	%94,2
TF-IDF + DK filtre + Kök bulma	2000	%87,9	%93,0
	5000	**%88,0**	%93,0

8.5. KSA Tabanlı Derin Öğrenme Modeli Uygulaması (Classification with CNN Based Deep Learning Model)

Derin öğrenme modellerinin eğitim aşaması diğer makine öğrenmesi tekniklerine göre oldukça uzun sürmekte, algoritmanın çalışması sırasında ise yüksek kaynak tüketimine sebep olmaktadır. Seçim işlemi yapılmadan tüm özniteliklerin bulunduğu bir veri kümesi ile yapılan sınıflandırma çalışması, Google Colab platformu gibi yüksek GPU (grafik işlemci ünitesi) kaynağı sağlayan ortamlarda bile saatlerce sürebilmektedir. Bu anlamda öznitelik seçimi işleminin çalışma performansını önemli ölçüde yükselttiği gözlemlenmiştir.

Geliştirilen KSA tabanlı derin öğrenme modeli ile yapılan çalışma sonucu elde edilen sınıflandırma doğruluk değerleri Tablo 6'da görülmektedir. Sonuçlara göre öznitelik seçiminde öznitelik sayısının yüksek tutulması başarıyı arttırmıştır. Ayrıca kullanılan durdurma kelimeleri filtreleme ve kök bulma önişlemlerinin başarıyı bir miktar arttırdığı görülmüştür.

Tablo 6. KSA tabanlı derin öğrenme modeli sınıflandırma doğruluk sonuçları
(Accuracy of CNN based deep learning model classifications)

	Öznitelik seçimi	TTC-4900	MY-15130
TF-IDF	2000	%90,4	%95,6
	5000	%91,2	%95,6
TF-IDF + DK filtre	2000	%90,4	%95,6
	5000	%91,5	**%95,7**
TF-IDF + DK filtre + Kök bulma	2000	%90,3	%94,3
	5000	**%91,7**	%94,0

Çalışmada kullanılan makine öğrenmesi algoritmaları ve geliştirilen KSA tabanlı derin öğrenme modeli ile yapılan sınıflandırma işlemleri sonucunda elde edilen F1-skoru (f-score) değerleri, her bir veri kümesi için Tablo 7'de ve Tablo 8'de görülmektedir.

Sonuçlara göre aynı sınıflandırma algoritması ve önişlemler bütününün uygulandığı iki veri kümesinde birbirinden farklı F1-skorları elde edilmiştir. Uygulanan kök bulma ve durdurma kelimeleri filtreleme önişlemlerinin, KNN sınıflandırıcısı hariç diğer sınıflandırma algoritmalarıyla yapılan sınıflandırma işlemlerinde, elde edilen doğruluk oranlarına en yüksek katkısının yaklaşık %2 olduğu görülmüştür.

KNN algoritmasının, öznitelik sayısı yüksek olan veri kümelerinde sınıflandırma başarısının düşük olduğu bilinmektedir. Sonuçlara göre her iki veri kümesinde de en düşük F1-skoru veren algoritmanın KNN olduğu görülmektedir. TTC-4900 veri kümesinde yapılan sınıflandırma çalışmasında baskın bir şekilde en yüksek F1-skoru veren algoritma **%91,7** ile KSA tabanlı derin öğrenme modeli olmuştur. MY-15130 veri kümesinde ise **%96,2** ile DVM ve Naive Bayes aynı F1-skorunu vermiştir. Sonuçlar genel olarak değerlendirildiğinde ise kelime çeşitliliği az ancak metin adedi fazla olan MY-15130 veri kümesinde yapılan tüm sınıflandırma işlemlerinin, kelime çeşitliliği fazla ancak metin adedi daha az olan TTC-4900'de yapılan sınıflandırmalardan daha başarılı olduğu ortaya çıkmıştır.

Tablo 7**.** TTC-4900 veri kümesi F1-skor(%) karşılaştırması, (Öİ): önişlem, (ÖA): öznitelik adedi
(TTC-4900 data set F1-score(%) comparison, (Öİ): pre-process, (ÖA): number of features)

Öİ	TF-IDF		TF-IDF + DK filtre		TF-IDF + DK filtre + Kök bulma	
ÖA	2K	5K	2K	5K	2K	5K
KNN	64,8	56,6	65,0	55,7	69,9	67,5
RO	85,8	86,7	85,7	86,5	87,8	87,9
DVM	89,0	90,8	88,8	91,0	90,3	91,2
NB	88,5	90,0	88,5	90,2	89,0	89,1
KSA	90,4	91,2	90,4	91,6	90,2	**91,7**

Tablo 8. MY-15130 veri kümesi F1-skor(%) karşılaştırması, (Öİ): önişlem, (ÖA): öznitelik adedi
(MY-15130 data set F1-score(%) comparison, (Öİ): pre-process, (ÖA): number of features)

Öİ	TF-IDF		TF-IDF + DK filtre		TF-IDF + DK filtre + Kök bulma	
ÖA	2K	5K	2K	5K	2K	5K
KNN	84,1	70,8	86,0	73,5	82,6	68,5
RO	93,6	93,7	93,5	94,2	93,0	93,0
DVM	95,8	**96,2**	95,5	96,1	94,1	94,1
NB	95,7	96,1	95,7	**96,2**	93,8	93,9
KSA	95,7	95,6	95,6	95,7	94,3	94,0

Tablo 9. İlişkili çalışmalarla karşılaştırma
(Comparison of related works)

Yıl	Çalışma	Model	F1-skor
2018	Çalışma1[13]	Naive Bayes + Kelime Torbası + Öznitelik seçimi	%90,0
2021	Çalışma2[14]	DVM + DK filtre	%91,8
2022	Bu çalışma	KSA + TF-IDF + DK filtre + Kök bulma	%91,7

TTC-4900 veri kümesini kullanan ilişkili çalışmaların sınıflandırma yöntemleri ve elde edilen F1-skoru değerleri Tablo 9'da karşılaştırılmıştır. Karşılaştırma sonucuna göre, geliştirmiş olduğumuz KSA tabanlı derin öğrenme modeli ve kullanmış olduğumuz önişlemler neticesinde; Çalışma2 ile oldukça yakın, Çalışma1'den ise daha yüksek F1-skor değeri elde edilmiştir.

9. SONUÇLAR (CONCLUSIONS)

Bu çalışmada, iki farklı Türkçe metin veri kümesi makine öğrenmesi teknikleri ve KSA tabanlı bir derin öğrenme modeli kullanılarak sınıflandırılmıştır. Seçilen veri kümelerine Rastgele Orman, Naive Bayes, Destek Vektör Makineleri, K-En Yakın Komşu Algoritmaları ve geliştirilen KSA tabanlı derin öğrenme modeli uygulanmıştır. Metin ve sınıf adedi olarak birbirinden farklı yapıda tercih edilen Türkçe veri kümeleri sınıflandırılarak veri kümesi boyutlarının sınıflandırma başarısına etkisi gözlemlenmiştir. Veri madenciliği önişlemleri olarak uygulanan durdurma kelimeleri filtreleme ve kök bulma önişlemlerinin de başarıya katkısı değerlendirilmiştir. Önişlemler neticesinde ortaya çıkan kelime temsi vektörlerine öznitelik seçimi uygulanarak boyutları düşürülmüş, böylece nihai vektör boyutunun da sınıflandırma sonuçlarına etkisi böylece gözlemlenmiştir. Kullanılan tüm ön işlemlerin farklı birleşimleri ile ortaya çıkan kelime temsil vektörlerinin sınıflandırması sonucunda doğruluk oranları ve F1-skor değerleri karşılaştırılmıştır. İlerleyen çalışmalarda farklı veri madenciliği önişlemleri ve kelime temsil yöntemleri uygulamaya alınarak daha etkili öznitelik çözümü sağlanabilir. Bu şekilde daha kıymetli bilgi içeren düşük boyutlu öznitelik vektörü elde edilerek hem uygulamanın çalışma performansını hem de sınıflandırma başarısını arttırmak hedeflenmektedir.

KAYNAKLAR (REFERENCES)

[1] R. Aşlıyan, K. Günel, "Metin İçerikli Türkçe Dokümanların Sınıflandırılması", **Akademik Bilişim Konferansı**,529-535, 2010.

[2] Y. F. Muliono, F. Tanzil, "A Comparison of Text Classification Methods k-NN, Naïve Bayes, and Support Vector Machine for News Classification", *Jurnal Informatika: Jurnal Pengembangan IT*, 3(2), 157-160, 2018.

[3] J. Liu, J. Li, L. Liu, W. Kang, "A Semantics Aware Random Forest for Text Classification", **28th ACM International Conference,** 1061-1070, 2019.

[4] H. Chen, W. Jiang, C. Li, R. Li, "A Heuristic Feature Selection Approach for Text Categorization by Using Chaos Optimization and Genetic Algorithm", *Hindawi Publishing Corporation Mathematical Problems in Engineering*, 2013(1), 1-6, 2013.

[5] B. Xu, X. Guo, Y. Ye, J. Cheng, "An Improved Random Forest Classifier for Text Categorization", *Journal of Computers*, 7(12), 2913-2920, 2012.

[6] S. Venkatraman, B. Surendiran, P. Arun Raj Kumar, "Spam e-mail classification for the Internet of Things environment using semantic similarity approach", *The Journal of Supercomputing*, 76(2), 756-776, 2020.

[7] R. Abascal-Mena, E. Lopez-Ornelas, "Author detection: Analyzing tweets by using a Na¨ive Bayes classifier", *Journal of Intelligent & Fuzzy Systems*, 39(2), 2331-2339, 2020.

[8] D. Kılınç, A. Özçift, F. Bozyigit, P. Yıldırım, F. Yücalar, E. Borandag, "TTC-3600: A New Benchmark Dataset For Turkish Text Categorization", *Journal of Information Science,* 43(2), 174–185, 2017.

[9] Ç. İnan Acı, A. Çırak, "Türkçe Haber Metinlerinin Konvolüsyonel Sinir Ağları ve Word2Vec Kullanılarak Sınıflandırılması", *Bilişim Teknolojileri Dergisi*, 12(3), 219-228, 2019.

[10] A. Uçan, M. Dörterler, E. A. Sezer, "A study of Turkish emotion classification with pretrained language models", *Journal of Information Science*, 48(6), 857–865, 2022.

[11] M. Aydoğan, A. Karcı, "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, 2019.

[12] İ. H. Toroslu, P. Karagöz, "Personality Analysis Using Classification on Turkish Tweets", *International journal of cognitive informatics & natural intelligence*, 15(4), DOI: 10.4018/IJCINI.287596, 2021.

[13] Ş. Yıldırım, T. Yıldız, "A comparative analysis of text classification for Turkish language", *Pamukkale University Journal of Engineering Science*, 24(5), 879-886, 2018.

[14] Ö. Köksal, E. H. Yılmaz, "Improving automated Turkish text classification with learning-based algorithms", *Concurrency and Computation*, 34(11), e6874, 2022.

[15] Z. Deng, X. Zhu, D. Cheng, M. Zong, S. Zhang, "Efficient kNN Classification Algorithm for Big Data", *Neurocomputing*, 195, 143-148, 2016.

[16] Internet: File:Knn sample plot.png, http://www.scholarpedia.org/w/images/1/13/Knn_sample_plot.png, 27.01.2021.

[17] L. Breiman, "Random forests" *Machine Learning*, 45(1), 5-32, 2001.

[18] L. Xin, "A New Text Classifier Based on Random Forests", **Proceedings of the 2016 2nd International Conference on Materials Engineering and Information Technology Applications (MEITA 2016),** Qingdao, China, 290-293, 24-25 Aralık, 2016.

[19] P. L. Prasanna, D. R. Rao, "Text Classification Using Artificial Neural Networks", *International Journal of Engineering & Technology*, 7(1), 603-606, 2018.

[20] D. Jha, A. Yazidi, M. A. Riegler, D. Jonansen, H. D. Johansen, P. Halvorsen, "LightLayers: Parameter Efficient Dense and Convolutional Layers for Image Classification", **PDCAT**, Shenzhen, China, 285-296, 28-30 Aralık, 2020.

[21] Internet: S. Yıldırım, A Benchmark Data for Turkish Text Categorization, https://www.kaggle.com/ datasets/savasy/ttc4900, 18.11.2022.

[22] Internet: M. Çabuk, E-Ticaret Ürün Yorumları, https://www.kaggle.com/datasets/mujdatcabuk/eticaret-urun-yorumlari/, 18.11.2022.