T.C.

MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ

ANABİLİM DALI

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE İÇERİK TABANLI SMS FİLTRELEME UYGULAMASI GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

ONUR KARASOY

TEMMUZ 2019

MUĞLA

MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

TEZ ONAYI

ONUR KARASOY tarafından hazırlanan **MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE İÇERİK TABANLI SMS FİLTRELEME UYGULAMASI GELİŞTİRİLMESİ** başlıklı tezin, 11/07/2019 tarihinde aşağıdaki jüri tarafından Bilişim Sistemleri Mühendisliği Anabilim Dalı'nda yüksek lisans derecesi için gerekli şartları sağladığı oybirliği ile kabul edilmiştir.

TEZ SINAV JURİSİ

Dr. Öğr.Üyesi Hüseyin ABACI **(Jüri Başkanı)** İmza:

Bilgisayar Mühendisliği Anabilim Dalı,
Adnan Menderes Üniversitesi, Aydın

Doç. Dr. Serkan BALLI **(Danışman)** İmza:

Bilişim Sistemleri Mühendisliği Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Dr. Öğr.Üyesi Osman ÖZKARACA **(Üye)** İmza:

Bilişim Sistemleri Mühendisliği Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

ANA BİLİM DALI BAŞKANLIĞI ONAYI

Dr. Öğr.Üyesi Gürcan ÇETİN İmza:

Bilişim Sistemleri Mühendisliği Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Doç. Dr. Serkan BALLI İmza:

Danışman, Bilişim Sistemleri Mühendisliği Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Savunma Tarihi: 11/07/2019

Tez çalışması sırasında elde ettiğim ve sunduğum tüm sonuç, doküman, bilgi ve belgelerin tarafımdan bizzat ve bu tez çalışması kapsamında elde edildiğini; akademik ve bilimsel etik kurallarına uygun olduğunu beyan ederim. Ayrıca, akademik ve bilimsel etik kuralları gereği bu tez çalışması sırasında elde edilmemiş başkalarına ait tüm orijinal bilgi ve sonuçlara atıf yapıldığını da beyan ederim.

Onur KARASOY

11/07/2019

ÖZET

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE İÇERİK TABANLI SMS FİLTRELEME UYGULAMASI GELİŞTİRİLMESİ

Onur KARASOY

Yüksek Lisans Tezi

Fen Bilimleri Enstitüsü

Bilişim Sistemleri Mühendisliği

Danışman: Doç. Dr. Serkan BALLI

Temmuz 2019, 80 sayfa

Günümüzde SMS (Kısa Mesaj Servisi) yoğun kullanılmasa da halen cep telefonu kullanıcılarına ulaşmanın en hızlı ve düşük maliyetli yollarından birisidir. Bu durum; reklam, bilgilendirme, promosyon vb. ürün tanıtımı yapmak isteyen kurumları, kısa mesaj hizmetini kullanmaya yönlendirmektedir. Fakat SMS kullanıcılarının izni olmadan atılan mesajlar ciddi sorun teşkil etmektedir.

Bu çalışmada, istenmeyen mesajları filtrelemek için geleneksel sınıflama algoritmalarının yanı sıra makine öğrenmesi ve derin öğrenme metotları da kullanılarak içerik tabanlı sınıflandırma yapılmış ve sonuçlar karşılaştırılmıştır. İngilizce ve Türkçe olarak iki ayrı veri seti kullanılmıştır. İngilizce veri setinde Word2Vec derin öğrenme aracı yardımıyla sınıflandırmada kullanılacak model oluşturulmuştur. Oluşturulan bu model sayesinde mesajların Spam ve Ham kelimelerine olan uzaklıkları hesaplanarak iki yeni öznitelik ortaya çıkarılmış ve bu iki yeni öznitelik göz önünde bulundurularak sınıflandırma algoritmalarının performansları karşılaştırılmıştır. Oluşturulan Türkçe veri setinde ise 5 farklı yapısal öznitelik, Word2Vec ile bulunan 2 yeni öznitelik ve her bir mesajın kelime indeks değerleri ile oluşturulan 45 değerden oluşan öznitelik ile beraber toplam 52 öznitelik matrisi ile geleneksel sınıflandırma algoritmaları yanı sıra derin öğrenme algoritmaları karşılaştırılmıştır. İngilizce veri setinde Word2Vec öznitelikleri ve Random Forest (Rasgele Orman) yöntemiyle, %99.64 doğru sınıflandırma oranı ve Türkçe veri setinde ise 52 adet öznitelik kullanılarak oluşturulan CNN (Convolutional Neural Network) yöntemi %99.86 doğru sınıflandırma oranı ile en başarılı algoritmalar olmuşlardır.

Anahtar Kelimeler: Metin Sınıflandırma,Word2Vec, SMS Filtreleme, Derin Öğrenme, Makine Öğrenmesi, CNN

ABSTRACT

DEVELOPMENT OF CONTENT BASED SMS FILTERING APPLICATION WITH MACHINE LEARNING METHODS

Onur KARASOY

Master Thesis

Institute of Science and Technology

Information Systems Engineering

Supervisor: Assoc. Prof. Dr. Serkan BALLI

July 2019, 80 pages

Although SMS (Short Message Service) is not used extensively today, it is still one of the fastest and cost effective ways to reach mobile phone users. This situation, directs institutions that want to promote product with advertising, information, promotion, etc. to using the short message service. However, messages sent without the permission of SMS users constitute a serious problem.

In this study, in order to filter spam messages, content based classification was made by using machine learning and deep learning methods besides traditional classification algorithms and the results were compared. Two separate data sets were used in English and Turkish. In the English data set, a model to be used for classification was created with the help of Word2Vec library. With the help of this model, the distance between the messages "Spam" and "Ham" is calculated and two new features are and the performance of classification algorithms were compared considering these two new features. In the Turkish data set, traditional classification algorithms as well as deep learning algorithms are compared with 5 different structural attributes, 2 new attributes found with Word2Vec, and 45 attributes created with word index values of each message, total 52 attribute matrix. In the English data set, the correct classification rate of 99.64% was obtained by using Word2Vec attributes and Random Forest method, and in the Turkish data set, the Convolutional Neural Network (CNN) formed by using 52 features obtained 99.86% accurate classification rate and they were found the most successful algorithms.

Keywords: Text Classification,Word2Vec, SMS Filtering, Deep Learning, Machine Learning, CNN

ÖNSÖZ

Öncelikle bu çalışmanın ortaya çıkması, şekillenmesi ve faydalı bir kaynağa dönüşmesinde önemli katkılara sahip danışman hocam Doç. Dr. Serkan BALLI'ya tüm destek ve yardımları için teşekkür ederim.

Yaptığım çalışmanın temelini oluşturan veri toplama sürecinde çok yardımcı olan ve başım sıkıştığında desteklerini esirgemeyen Muğla Sıtkı Koçman Üniversitesi Bilgi İşlem Dairesi Başkanlığındaki, başta Osman KELEŞ olmak üzere tüm çalışma arkadaşlarıma çok teşekkür ederim.

Ayrıca desteklerini her zaman hissettiğim, iyi ki hayatımdalar dediğim kardeşlerim Özgür, Öznur ve Gizemnur KARASOY ile birlikte ailemizin yeni üyeleri Sultan ve dünyalar tatlısı Kuzey KARASOY'a teşekkürü bir borç bilirim.

Yaşadıkları onca zorluklara rağmen eğitim hayatımda gerekli olan desteğin katbekat fazlasını veren, adeta bir eğitim bilimci gibi kendilerine has teknik ve taktik ile beni hayata hazırlayan, haklarını ne yapsam ödeyemeyeceğim Annem Birgül KARASOY ve Babam Tuncay KARASOY 'a teşekkür ederim.

İÇİNDEKİLER

ÇİZELGELER DİZİNİ

ŞEKİLLER DİZİNİ

SEMBOLLER VE KISALTMALAR DİZİNİ

SMS	Short Message Service
KNN	K-Nearest Neighbors
YSA	Yapay Sinir Ağı
ANN	Artifical Neural Network
CNN	Convolutional Neural Network
LR	Logistic Regression
NB	Naive Bayes
SVM	Support Vector Machine
DMC	Dynamic Markov Compression
OSBF-Lua	Orthogonal Sparse Bigrams with confidence Factor-Lua
IG	Information Gain
CHI2	Chi-square
Bow	Bag of Word
ML	Message Length
CWR	Capital Letter Weight Rate
SMW	Spam Message Weight
ROC	Receiver Operating Characteristic
ReLU	Rectified Linear Unit

1. GİRİŞ

Her geçen yıl mobil teknolojiler hızla gelişmekte ve cep telefonu üreticileri adeta bir yarış içinde bulunmaktadır. Gelişen bu teknolojiler ile cep telefonu kullanım yaşı düşmekte ve kullanıcı sayısı hızla artmaktadır. Bu artış, mobil teknolojilerin gelişmesine ön ayak olmakla beraber, dolaylı olarak onlara olan bağımlılığı da beraberinde getirmektedir. İnsanların artık bir an bile yanından ayırmadığı cep telefonları, hayatı kolaylaştırmanın yanı sıra istediğimiz kişiye kolayca ve hızla ulaşmamızı sağlamaktadır. Cep telefonu, sadece kullanıcılar için değil kullanıcıya ulaşmak isteyen tüm hizmet sağlayıcılar, kurumlar ve şirketler için de çok önemli bir araç olarak kullanılmaktadır. İnsanlara ulaşmak için genel olarak cep telefonlarına SMS gönderimleri yapılmaktadır.

SMS (Short Message Service-Kısa Mesaj Servisi) günümüzde dahi yaygın kullanılan mobil iletişim kanallarından biridir. Mobil iletişim ortamlarında metin iletimi sağlayan uluslararası bir mesajlaşma servisidir. İlk mesaj 1992 yılında Neil Papworth adında bir mühendis tarafından Vodafone'da çalışan iş arkadaşlarına "MERRY CHRISTMAS" (Mutlu Noeller) içeriği ile gönderilmiştir (Web-9).

SMS 2000'li yılların başında yoğun bir şekilde kullanılmış olsa da son zamanlarda kullanılan WhatsApp, Messenger, Viber gibi internet tabanlı mesajlaşma uygulamaları kısa mesajların yerini almaya başlamıştır. Fakat internet teknolojilerinin yaygınlaştığı bu dönemde, kişi doğrulama ve güvenlik gerektiren uygulamalarda hala çok sık kullanılan bir servis olarak karşımıza çıkmaktadır.

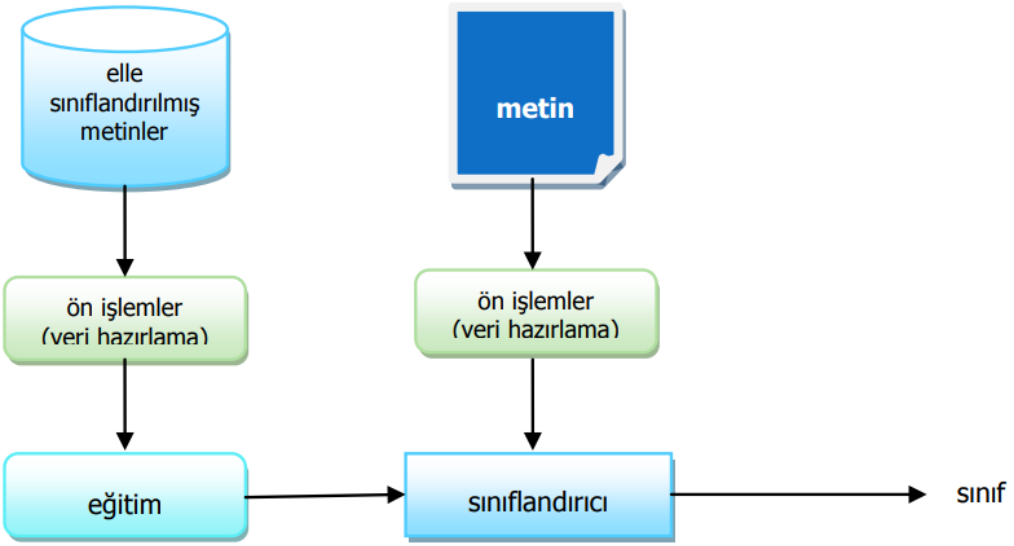
SMS, bankacılık işlemleri gibi önemli kimlik doğrulama gerektiren tüm uygulamaların sık sık kullandığı servislerden biridir. Bu nedenle servisi tamamen kapatmak çok mümkün değildir. SMS, maliyeti düşük ve kolay ulaşılabilir olmasından dolayı özellikle ürün veya hizmet reklamları, özendirmeler ve duyurular için çok sık tercih edilen araçlardandır. (Karasoy ve Ballı, 2016). Kullanıcıya ulaşmanın bu kadar kolay ve ucuz olduğu durumda da indirim, kampanya, tanıtım ve promosyon gibi reklam yöntemlerini kullanmak isteyen kurum ve şirketler, kısa mesajları yüksek oranda tercih

etmektedirler. Bu durum Spam (İstenmeyen reklam mesajı) mesaj problemini de beraberinde getirmektedir.

Spam mesajları tespit etme problemi, metin sınıflandırma problemi olarak ele alınmaktadır. Yıllardır bu alana en yakın araştırma Spam e-posta filtreleme üzerine yapılmıştır. Spam mesaj göndericilerini işaretleyip kara listeye almak etkili bir çözüm gibi dursa da sürekli değişen gönderici profilini takip etmek, belirlemek oldukça zor bir süreçtir. Bu nedenle çalışmalar gönderilen mesajın metninden yola çıkarak mesajın Spam olup olmadığını anlamak üzerine yapılmaktadır.

Metin sınıflama problemlerinde çalışma yapılan metin koleksiyonuna göre özellikler çıkarılıp çözüm üretilmeye çalışılır. Metin madenciliğinin bir alt alanı olan metin sınıflama çalışmalarında tek etiketli veya çok etiketli sınıflandırma yöntemleri kullanılır. Bu problemin türüne göre belirlenir, bazı problemlerde bir metin tek bir sınıfa aitken bazı problemlerde birden çok sınıfa girebilir. SMS filtreleme problemi tek etiketle çözülebilecek bir problemken, bir konuşmanın içeriği sınıflanırken birkaç etiket kullanılabilir. Yani konuşma hem spor hem de siyaset sınıfına girebilir.

Metin sınıflandırma problemlerinde kullanılan makine öğrenmesi, genellikle gözetimli öğrenme ile yapılmaktadır. Şekil 1.1'de de gösterildiği gibi öncelikle işaretlenen eğitim verilerinden bir model oluşturulur ve yeni gelen test verisi bu modele göre sınıfı tahmin etmeye çalışır (Tantuğ, 2016).



Şekil 1.1. Makine öğrenmesi ile metin sınıflama

1.1.Tezin Amacı ve Kapsamı

Neredeyse tüm insanlığa tekabül eden cep telefonu kullanıcıların önemli bir sorunu olan Spam mesaj problemi için henüz başarılı ve kullanışlı bir çözüm bulunmamaktadır. Bu tezde, çıkış noktası olarak önceki çalışmaların doğru sınıflama yüzdesini artırılabilir mi ve kullanışlı bir uygulama haline getirerek önemli bir probleme çözüm bulunabilir mi soruları ele alınmıştır.

Bu çalışmada kullanıcılar arası iletişim ve yeni nesil teknolojilerde kimlik doğrulama için sık kullandığımız SMS (Kısa Mesaj Servisi)'in doğru, güvenilir ve verimli kullanılabilmesi adına Spam (istenmeyen) mesajları, makine öğrenmesi ve derin öğrenme yöntemleri ile sadece mesaj metnini göz önünde bulundurarak, sınıflandırabilecek bir mobil uygulama geliştirilmesi hedeflenmiştir.

Farklı öznitelik (feature) çıkarımlarıyla sistemin doğru sınıflandırma (classification) performansları ölçülmüş ve bu ölçümler sırasında iki ayrı veri seti kullanılmıştır. Kullanılan ilk veri seti daha önce de üzerinde çalışmaların yapıldığı SMS Spam Collection (Almeida vd., 2011) veri setidir. Diğeri ise bu çalışma için yeni oluşturulmuş ve farklı yaşlarda ve bölgelerde bulunan kullanıcılardan toplanarak hazırlanmış olan Türkçe SMS veri setidir.

1.2.Önceki Çalışmalar

Literatürdeki aynı konuda yapılan çalışmaların önemli olanları aşağıda özetlenmiştir:

Healy vd. (2005), SMS filtrelemede KNN sınıflandırma yöntemiyle Naive Bayes (NB) ve Support Vector Machine (SVM) yöntemleri karşılaştırılmaktadır. Farklı çalışma alanları için farklı özellik çıkarımları ve farklı yöntemlerin daha iyi sonuç verebileceği üzerinde durulmaktadır.

Dixit vd (2005) tarafından yapılan çalışmada, mesajların daha önceden kümelenmiş istenmeyen (Spam) mesajlara yapısal benzerliklerine göre kümelenerek filtrelenmesi sağlanmaktadır. Büyük boyutlu istenmeyen mesaj kümelerinden bir formül çıkarılarak yeni mesaj bu formüle göre değerlendirilmektedir.

Hidalgo vd. (2006) tarafından yapılan çalışmada, iki farklı SMS veri tabanında, e-posta filtrelemede kullanılan sınıflandırma algoritmaları karşılaştırılmaktadır. Sonuç olarak, bu sınıflandırma algoritmalarından en etkilisi olarak SVM yöntemi gözlemlenmektedir.

Deng vd. (2006) tarafından yapılan çalışmada, Naive Bayes sınıflandırma algoritması kullanılarak kullanıcı cihazlarında sınıflandırma işlemi gerçekleştirilmiştir. Her bir kullanıcının aldığı mesajlar ile yeniden eğitilen model belirli periyotlarla uygulamayı güncelleyerek değişiklikleri tüm kullanıcılara ulaştırmaktadır. Ayrıca alınan bu dönütler sayesinde Spam mesaj gönderen kullanıcılar işaretlenip filtrelemedeki başarı yüzdesi iyileştirilmektedir.

Cormack vd. (2007) tarafından yapılan çalışmada, istenmeyen blog yorumlarını filtreleme yöntemleri referans alınarak Bogofilter, DMC, Logistic Regression, OSBF-lua, SVM vb. yöntemlerle kısa mesaj filtreleme üzerine çalışılmıştır.

Wu vd. (2008), SMS anahtar kelime, Pinyin Fuzzed anahtar kelime eşleşmesi ve Bayes yöntemiyle SMS filtreleme işlemi yapılmaktadır. Gerçek zamanlı SMS filtreleme işlemini daha hızlı bir şekilde gerçekleştirilmektedir. Bu işlemlerin işlem yükü gereksinimi üzerinde durulmaktadır.

Cai vd. (2008), tarafından yapılan çalışmada, daha basit ve daha az kullanılan Winnow algoritması ile lineer bir sınıflandırma tercih edilmiştir. Bu sınıflandırmanın, çok boyutlu özelliklere sahip çalışma alanlarında, iyi performans gösterdiği vurgulanmaktadır.

Longzhen vd. (2009), tarafından yapılan çalışmada çifte filtreleme yöntemi üzerine çalışmışlardır. Öncelikle KNN algoritmasına göre sınıflandırılmakta ayrıca buna mesajların karakteristik özeliklerine göre sınıflama denetimini de eklenmektedir. KNN ile ulaşılan başarı oranı %82.66 iken çift filtreleme yöntemi ile bu oran %86.75‘e yükseltilmektedir.

Zhang ve Wang (2009) tarafından yapılan çalışmada, Bayes metotla mesajların anahtar kelime özelliklerine dayanan SMS filtreleme uygulaması gerçekleştirilmektedir.

Yoon vd. (2010) tarafından yapılan çalışmada, istenmeyen mesajları filtrelemek ve istenmeyen mesajlarla mücadele etmek için hibrit bir anti-spam sistemi

uygulanmaktadır. Çalışmada içerik tabanlı filtreleme yöntemiyle birlikte gönderici doğrulama metodu (Challenge Response) kullanılmaktadır. Gönderici mesajı gönderdikten sonra kendinin gerçek bir kullanıcı olduğunu ispatlamak için bir doğrulama kodu ile kendini tanıtması gerekmektedir.

Liu vd. (2010) tarafından yapılan çalışmada, örüntü eşleme kullanarak istenmeyen SMS filtreleme üzerine çalışılmaktadır. Gerçek zamanlı uygulamalarda çalışması için örüntü eşleme yönteminin uygun olduğu üzerinde durulmaktadır.

Hu ve Yan (2010), SMS trafiği sıklık analizi yaparak istenmeyen mesajları tespit etmek amaçlanmaktadır. Göndericinin davranışlarını incelemek üzerine durulmaktadır. Bu sistemi geliştirme amaçları gerçek zamanlı uygulamalardaki istenmeyen mesaj filtreleme işlemi için gereken işlem süresi ve yükünden tasarruf etmektir.

Khemapatapan (2010) tarafından yapılan çalışmada, Thai ve İngilizce için iki ayrı filtreleme metodu SVM ve NB algoritması ile incelenmektedir. NB ile anlamsal doğruluk analizi, bölümleme işlemi ve normalizasyon işlemleri, SVM algoritmasına tabi tutularak yapılan incelemede en büyük doğruluk oranına ulaşılmaktadır.

Liu ve Wang (2010) tarafından yapılan çalışmada eğitim verilerindeki farklı kategorilerde istenmeyen ve normal mesajların özelliklerinden yola çıkarak sıklık fonksiyonu ile istenmemezlik skoru hesaplamaya dayalı Index modeli önerilmektedir.

Wang vd. (2010) tarafından yapılan çalışmada, bir kullanıcıdan diğer kullanıcıya giden mesajlar aracılığı ile sosyal ağ analizleri ve spektral gönderim davranış analizlerini birleştirilmektedir. Mesaj loglarından grafikler elde edilmekte ve yöntem için online (çevrimiçi) ve offline (çevrimdışı) olarak İki tür filtre önerilmiştir. Offline filtreleme tek kullanıcının loglardaki davranışından yola çıkarak uygulanmaktadır. Online filtreleme ise kullanıcı aynı periyotta kaç alıcıya mesaj gönderildiği incelenerek uygulanmaktadır.

Rafique vd. (2011) tarafından yapılan çalışmada, evrimsel XCS, SLAVE, UCS, cAnt-Miner ve evrimsel olmayan RIPPER, Naive Bayes, C4.5, C-SVM sınıflandırma yöntemleri kullanılarak giriş katmanında mesaj filtreleme uygulaması yapılmaktadır.

Abdelkader vd. (2011) tarafından yapılan çalışmada iki farklı metot incelenmektedir. Bu metotlar Bloom Filtreleme ve İçerik Özetleme (Content Hashing) tabanlı

listelemedir. Bilgileri kullanıcıların dâhil olduğu ortak bir sosyal ağ oluşturarak toplanan çalışmada, özetleme listesi ile daha başarılı sonuçlar elde edilmektedir.

Mathew vd (2011) tarafından yapılan çalışmada, özellik çıkarımı yapılarak 10 farklı yöntemle buldukları sonuçlar karşılaştırılıp %98 doğruluk oranı ile Bayes yönetiminin daha etkili olduğu sonucu çıkarılmaktadır.

Junaid ve Farooq (2011) tarafından yapılan çalışmada beş gözetimli öğrenme algoritması ile dört evrimsel algoritma sınıflandırması karşılaştırılmaktadır. Yaptıkları çalışma sonuncunda 3000 mesajdan sonra gözetimli sınıflandırma (UCS) sisteminin iyi performans gösterdiği gözlemlenmektedir.

Belem ve Duarte-Figueiredo (2011) tarafından yapılan çalışmada, Bayes sınıflandırma üzerine uygulamalar yapılmaktadır. Fakat önceki çalışmalardan farklı olarak kelime sıklıklarından değil de kelime gruplarının sıklıklarından faydalanılarak daha iyi sonuçlar elde edilmektedir. 1 kelimeden 5 kelimeye kadar gruplama yapıldıktan sonra özellikler belirlenmektedir. Bu özelliklerle yapılan sınıflama işleminde %99 a kadar başarılı bir sonuç elde edilmektedir.

Sohn vd. (2011) tarafından yapılan çalışmada, istenmeyen (Spam) mesajları yasal mesajlardan ayırt edebilen mobil Spam sınıflandırma üzerinde çalışılmaktadır. Makalenin alana iki katkısı gözlemlenmektedir. İlk olarak SMS filtreleme işlemi yapılırken kullanılan hesaplama maliyetini düşürmek amaçlanmaktadır. Bağımsız dil filtreleme yöntemi oluşturulmaya çalışılmıştır. Kısa mesajlar için yapılan özellik çıkarımlarında mesaj uzunluğu, kelime türü, tri-gram, özellik sıklığı, özel karakter, kategori sıklığı, kelime öbeği sıklığı vb. özellikler kullanılmaktadır.

Nuruzzaman vd. (2011) tarafından yapılan çalışmada, bilgisayar destekli bir sisteme ihtiyaç duymayan bağımsız bir filtreleme sistemi önerilmektedir. Naive Bayes yöntemi ve kelime sıklık tablosu kullanılarak filtreleme işlemi sağlanmaktadır.

Almeida vd. (2011) tarafından yapılan çalışmada, birçok öğrenme algoritması karşılaştırması sonuçları sunulmaktadır. 5574 mesajdan oluşan bir veri seti kullanılmaktadır. Bu veri setinin 747 tanesi istenmeyen mesaj 4827 tanesi ise normal mesajdan oluşmaktadır. 13 tane sınıflandırma yönetimi denenmiştir. Naive-Bayes yönetiminin 8 farklı varyasyonu, SVM, En Az Açıklama Uzunluk (Minimum Description Length) sınıflandırması, KNN, Decision Tree C4.5, ve PART kural tabanlı

öğrenme algoritmaları kullanılmaktadır. SVM yönteminin performansının %97.64 doğruluk oranı ve %0.18 FP (False Positive) oranı ile en iyisi olduğu gözlemlenmektedir. Daha sonraki 3 metot ise yaklaşık %97.5 doğruluk oranı ile NB, C4.5 ve PART'dır.

Yadav vd. (2012) tarafından yapılan çalışmada, kullanıcı merkezli bir istenmeyen SMS filtreleme uygulaması üzerine çalışılmaktadır. Bu uygulamada (SMSAssain) kullanıcı tarafından oluşturulan özellikler ile içerik tabanlı makine öğrenme sistemi oluşturulmaktadır. İçerik filtreleme için Bayes sınıflandırma yöntemi kullanılmıştır. Beyaz liste, kara liste, kullanıcıların belirlediği anahtar kelimeler gibi özelliklerle program altyapısı güçlendirilmektedir.

Liu vd. (2012) tarafından yapılan çalışmayla oluşturulan sosyal ağ sayesinde belli zamanda kullanıcılardan alınan bilgilerle kullanıcılardan oluşan bir çizge (graph) oluşturulmaktadır. Mesajlarda bu ağ yardımıyla toplanarak, 3 farklı sınıflandırma metodu (KNN, SVM ve AdaBoost) ile incelenmektedir. Oluşturulan sistemin SVM ve AdaBoost ile iyi sonuçlar verdiği gözlemlenmiştir.

Uysal vd. (2012) tarafından yapılan çalışmada, özellik çıkarım yöntemleriyle, CHI2 ve IG metotlarını kullanarak SMS filtreleme amaçlanmaktadır. Bu yöntemlerden elde edilen sonuçlara göre, istenmeyen mesajlar tespit edilmekte ve CHI2 özellik çıkarım yöntemiyle %90.17 başarı sağlanmaktadır.

Androulidakis vd (2012) tarafından yapılan çalışmada, SMS gönderenin daha önce gönderdiği SMS'ler gönderenin numarasının ilk 5 karakteriyle göndericiyi kimliklendirme, anahtar kelime kara listesi, mesajın URL içerip içermediği vb. özelliklerle istenmeyen SMS belirleme uygulaması gerçekleştirilmiştir.

Uysal vd. (2012) tarafından yapılan çalışmada, geliştirilmiş öznitelik seçimi ve örüntü tanıma yöntemi ile sınıflandırma işlemi gerçekleştirilen "istenmeyen SMS süzgeci" önerilmiştir. Öznitelik seçiminde Gini endeksi temelli bir yaklaşım seçilerek etkili olduğu gözlemlenen özniteliklerden oluşan vektörleri, sınıflama işlemi için KNN ve NB yöntemleri ile sınıflandırmışlardır.

Rafique ve Smıeee (2012) tarafından yapılan çalışmada, akıllı telefonlar ve mobil cihazlarda istenmeyen SMS sınıflandırmaya yarayan yeni bir çizge tabanlı (graph-

based) öğrenme yöntemi üzerine çalışılmaktadır. Sınıflandırma için Kullback Leibler uzaklığı (KL-Divergence) ölçümünden faydalanılmaktadır.

Xu vd. (2012) tarafından yapılan çalışmada, hizmet sağlayıcı tarafından içerik tabanlı olmayan bir SMS filtreleme yönetimi önerilmektedir. Çıkarılan özelliklere göre istenmeyen mesaj göndericisinin keşfine dayalı olan bu incelemede özellikler; son bir haftadaki mesaj sayısı, toplam mesaj uzunluğu, mesaj yanıtları sayısı vb. maddelerden oluşmaktadır. Sınıflandırma için KNN ve SVM algoritmaları üzerinde durulmakta ve SVM'nin KNN ye göre daha başarılı bir sonuç verdiği gözlemlenmektedir.

Mohmoud ve Mahfouz (2012) tarafından yapılan çalışmada, Artificial Immune System (Yapay Bağışıklık Sistemi) olarak adlandırılan bir teknik ile SMS filtreleme üzerine çalışılmaktadır. Biyolojik bağışıklık sisteminden uyarlanarak oluşturulan AIS, telefon numaraları ve kelimelerden kara listeler yaparak istenmeyen SMS'lere karar verme sürecini yönetmektedir.

Delany vd. (2012) önceki yapılmış çalışmalardan yolarak çıkılarak istenmeyen SMS'leri filtrelemek için ne tür yöntemlerin olduğunu, hangi verilerin kullanıldığını anlatıp istenmeyen SMS analizi üzerine bir inceleme hazırlamışlardır.

Ho vd. (2013), yaptıkları çalışmada grafik tabanlı (graph-based) metin temsili (text representation) tekniği ile KNN algoritmasını birleştirerek bir çözüm önermişlerdir. Veri seti, performans değerlendirmeleri için farklı gruplara ayrılmış, bu gruplar grafiklerle temsil edilmiş ve KNN bileşenlerini oluşturmuştur. Önceki yaptıkları çalışmayla karşılaştırmışlar ve bu önerinin %98.9 doğruluk oranıyla daha iyi olduğu sonucuna varılmıştır.

Zhang vd. (2013) tarafından çalışmada benzer algoritmaları İngilizce ve Çince olmak üzere farklı dillerdeki SMS'ler üzerinde denenmektedir. İnceleme sonucunda farklı dillerde farklı metotlar daha etkili olduğu gözlemlenmiştir. Örneğin İngilizce için çok iyi sonuç veren KNN algoritması Çincede başarısız olmaktadır.

Uysal vd. (2013), yaptıkları çalışmada KNN ve SVM sınıflandırma metotlarını önermişlerdir. Öznitelik seçiminde Bow (Bag of Word) ve karakteristik özniteliklerin beraber kullanım kombinasyonları incelenmiştir. (Bow - 2690 Türkçe, 3179 İngilizce öznitelik çıkarılmıştır + 6 karakteristik öznitelik) Karakteristik özniteliklerin farklı

dillerde bile etkili sonuçlar verdiği gözlemlenmiştir. Türkçe veri seti için en iyi sonuç %98, İngilizce veri seti için %96 doğruluk oranı ile SVM metodu ile ulaşılmıştır.

Najadat vd. (2014) tarafından yapılan çalışmada, 12 farklı SMS sınıflandırma yöntemi incelenmektedir. Bu sınıflandırma yöntemleri; AdaBoostM1, Decision Table, J48 (D-tree), Random Forest, KNN, K-Star, Naïve Bayes, NB Multinomial, DMNBtext, SVM, SGD ve VotedPerceptro'dur. Bu yöntemlerin arasında en başarılı sonuç Discriminative Multinomial Naive Bayes ile %96.46 başarı elde edilerek gözlemlenmiştir.

Shahi ve Yadav (2014), Nepali Spam veri tabanında yaptıkları çalışmada Naive Bayes sınıflandırma yöntemiyle SVM sınıflandırma yöntemini karşılaştırılmıştır. SVM'de %87.15 doğruluk oranı, Naive Bayes'in doğruluk oranı %92.74 olarak gözlemlenmiştir.

Kılıç vd. (2014) tarafından yapılan çalışmada 3 katmanlı melez bir SMS filtreleme uygulaması üzerine çalışılmıştır. Uygulama, Beyaz Liste Kontrolü, Kara Liste Kontrolü ve Naive Bayes sınıflandırma yönteminden oluşmaktadır. Belirlenen özellik çıkarımlarından sonra bu değerler Navie Bayes ve KNN algoritmalarıyla incelenerek Naive Bayes ile KNN algoritmalarının başarılarının birbirine yakın olduğu gözlemlenmiştir. Bu nedenle uygulamada işlem karmaşıklığı düşük olan Naive Bayes tercih edilmiştir.

Ahmed vd. (2014) tarafından yapılan çalışmada Naive Bayes yöntemini, Assembly Öğrenme Tekniği ile birlikte test ederek sms filtrelemede daha başarılı sonuç elde edildiği gözlemlenmiştir

Anchal ve Sharma (2014) tarafından yapılan çalışmada, istenmeyen mesajları filtrelemek için Tree Architecture, ICA algoritması ve Yapay sinir ağları algoritması gibi metin sınıflandırma yöntemleri üzerinde çalışılmaktadır. Ayrıca, kullanıcılar tarafından istenmeyen mesaj olarak kabul edilen SMS'lerdeki kelimelerin ve sembollerin özellikleri veri seti olarak kullanılmaktadır. Mesajlardaki kelimeler 14 farklı özellikten oluşmaktadır. Mesajların sınıflandırılmasında yapay sinir ağları yöntemi uygulanmaktadır. Ayrıca, önceden bilinen istenmeyen mesaj göndericileri kontrol edilerek, yapay sinir ağları algoritması kullanmadan SMS filtrelenmesi sağlanmaktadır.

Chen vd. (2014) tarafından yapılan çalışmada, SMS Spam kontrol yöntemi olarak TruSMS adı verilen yöntem kullanılmaktadır. TruSMS, kaynaktan hedefe gönderilen SMS'in spam analizi ve veri trafiği sayesinde SMS'leri güvenlik kontrolünden geçirmektedir. Kullanıcılardan alınan verilerle SMS'leri sınıflandırır. Araştırma sonucunda TruSMS yönteminin SMS filtrelemede doğru, güvenilir ve sağlam olduğu anlaşılmaktadır.

Chan vd. (2014) tarafından yılında yapılan çalışmada SMS filtrelemek için SVM yöntemi uygulanmaktadır. Bu yöntem uygulanırken sadece mesajların ağırlığı değil aynı zamanda kelimelerin uzunluğu da dikkate alınmaktadır. SMS filtrelemede, Good Word Attack (İyi Kelime Saldırısı) ve Feature Reweighting (Özellik Ağırlıklandırma) metotları önerilmektedir.

Ahmed vd.(2014) tarafından yapılan çalışmada, Multinomial Naive Bayes ve Random Forest ve LibSVM yöntemleri uygulanarak filtreleme için birleşik bir yapı (ensemble) oluşturulmaktadır. Pozitif örnek oranının düşük olduğu durumlarda iyi sonuçlar gözlemlenmiştir.

Kim vd. (2015) tarafından yapılan çalışmada, anahtar kelime sıklığına göre Naive Bayes, Decision Tree(J-48) ve Lojistik Regresyon yöntemleri üzerine çalışılmaktadır. Kelime Frekanslarına göre özellik çıkarımları yaparak kolay formüle edilebilen algoritmalar seçilerek istenilen oranda başarı hedeflenmektedir. %96.2 ile Naive Bayes yönteminde en yüksek başarı elde edildiği gözlemlenmiştir.

Bozan vd. (2015) yaptıkları çalışmada uzman sistem yardımı ve sınıflandırma yöntemleriyle birlikte bir çözüm önerisi sunmuşlardır. Veri setinde bulunan ifadeler ve kelimelerden öznitelikler çıkarılmış ve bazı kriterlere göre sonuca etki etmeyeceğini belirledikleri öznitelikler elenmiştir. Sonuç olarak 6622 adet öznitelik elde edilmiş ve CfsSubset öznitelik seçme yönetimi uygulanmıştır. Belirlenen öznitelikler ile SVM, NB ve KNN algoritmalarıyla elde edilen sonuçlar karşılaştırılmış ve %98.61 ile SVM'nin en başarılı sınıflandırıcı olduğu tespit edilmiştir.

Waheeb vd. (2015), yaptıkları çalışmada Scaled Conjugate Gradient (SCG) backpropagation algoritması ile eğittikleri yapay sinir ağının (ANN) performansını incelemişlerdir. Öznitelik seçiminde Gini index değeri kullanılmıştır. Seçilen öznitelik sayısına göre TP FP FN ve TN değerlerini karşılaştırmışlardır. En iyi sonucu 100

öznitelik sayısı ile elde etmişlerdir. Yapılan çalışma sonucunda %99.1 doğruluk oranına ulaşılmıştır.

Fernandes vd. (2015) yaptıkları çalışmada Optimum-Path Forest sınıflandırma modeli ile spam mesaj filtreleme yöntemi önermişlerdir. Sonuçları karşılaştırmak için Spam Yakalama (SC%), Bloklanan Normal Mesajlar (BH%), Doğruluk (ACC%) oranları ve Matthews Korelasyonu Katsayısı değerleri kullanılmıştır. ANN-MLP, KNN, OPF with Complete Graph, OPF with KNN Graph ve SVM sınıflandırma yöntemlerinden en iyi doğruluk oranını (ACC) SVM vermesine rağmen OPF tabanlı sınıflandırıcılar 720 kat daha hızlı, tüm Normal Mesajları (BH%) doğru sınıflandırmıştır.

Akbari ve Sajedi (2015) yaptıkları çalışmada SMS spam sınıflandırma için LPBoost, AdaBoost, TotalBoost, LogitBoost, GentleBoost, RobustBoost, RusBoost, SVM ve NB algoritmalarını karşılaştırılmıştır ve en iyi doğruluk oranı %98.30 ile GentleBoost algoritması önerilmiştir. Ayrıca öznitelik çıkarımı için her bir kelimenin olasılığı hesaplanıp diğerleriyle karşılaştırılmış ve 124 adet kelime özniteliği çıkarılmıştır. Daha sonra kullanılmayan öznitelikleri çıkararak doğruluk oranını etkilenmeden 32 adet kelime özniteliğine düşürülmüştür.

Arifin vd. (2016) yaptıkları çalışmada veri madenciliğinde önemli iki konu olan sınıflama (classification) ve birliktelik kuralı (association) üzerine eğilmişlerdir. FP-growth birliktelikleri belirlemede (frequent pattern) Naive Bayes ise sınıflandırma işleminde kullanılmıştır. FP-growth ile NB birlikte kullanıldıklarında daha iyi sonuç elde edilmiş ve %98.596 doğruluk oranına (accuracy) ulaşılmıştır.

Ma vd (2016) yaptıkları çalışmada Latent Semantic Analysis olasılık teorisine dayanan ve SMS Spam filtrelemede uygun olan bir Message Topic Model (MTM) önermektedirler. Mevcut Spam SMS filtreleme teknolojileri karşılaştırıldığında MTM, mesajların kısalık sorunu ortadan kaldırılabilmektedir. Genellikle spam SMS'lerde görülen sembollere dikkat çekmektedir. MTM modeli ile %97 doğruluk oranına ulaşılmıştır.

Karasoy ve Ballı (2016), yaptıkları çalışmada, yeni bir Türkçe veri seti oluşturup, istenmeyen mesajların engellenmesi için bir mobil uygulama geliştirmişlerdir. Veri setindeki mesajlar karakteristik özelliklerine göre belirlenen 8 ayrı öznitelik kullanmışlardır. Spam ve Normal Mesajın yanı sıra bildiri mesajlarının sonuca olan

etkisi karşılaştırılmıştır. %93.76 doğruluk oranı ile Random Forest ve 2 sınıflı sonuç önerilmiştir.

He vd. (2017), yaptıkları çalışmada dilsel karar ağaçları (LTD) ile gömülü olan, dilsel özellik hiyerarşisi (LAH) önermişlerdir. Veri setinden çıkarılan özellikler semantik açıdan alt kümlere ayrılmıştır. Çalışmada 3 alt seviyeye kadar ayrıştırılmıştır ve LAH'lar bu ayrıştırmalara göre semantik olarak oluşturulmuştur. Farklı ayrıştırmalardaki performanslar karşılaştırılmıştır.

Suleiman ve Al-Naymat (2017), yaptıkları çalışmada makine öğrenmesi platformu H2O kullanılmıştır. Mesaj uzunluğu, kelime sayısı, büyük harf frekansı, URL durumu vb.. 10 adet yapısal mesaj öznitelikleri ile NB, RF ve Deep Learning yöntemlerinin performansları karşılaştırılmıştır. Doğruluk oranı değerleri dikkate alındığında Random Forest yöntemi %97.7 doğruluk oranı ile en iyi algoritma olarak önerilmiştir.

Navaney vd. (2018), yaptıkları çalışmada veri setini kelimelere ayırarak doküman terim matrisi oluşturmuş ve bu değerleri öznitelik olarak kullanmıştır. Oluşturulan öznitelikler ile Naive Bayes, SVM ve Maksimum Entropi yöntemleri ile sınıflama işlemi gerçekleştirilmiş %97,4 doğru sınıflandırma yüzdesi ile SVM algoritmasını önermişlerdir.

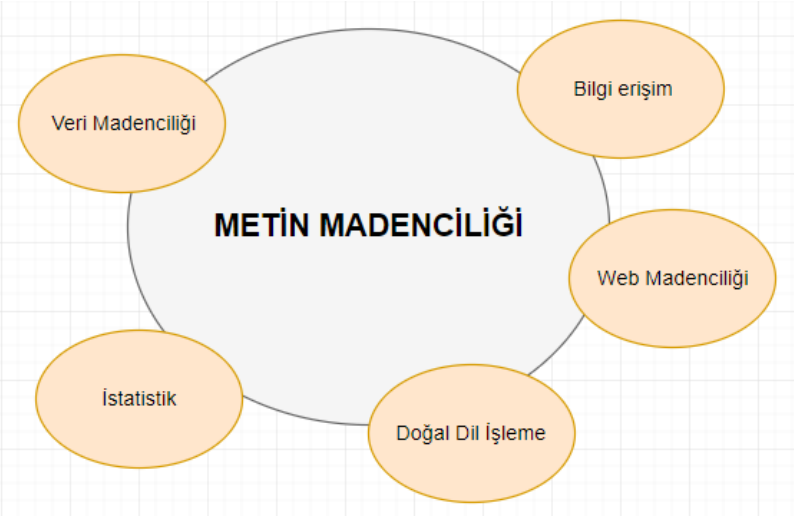
Ali ve Maqsood (2018), yaptıkları çalışmada WEKA araçlarını kullanarak StringToWordVector filtresiyle kelime vektörleri oluşturulmuştur. Oluşturulan bu vektöreler sayesinde 353 öznitelik belirlenmiştir. Bu öznitelikler en yüksek sınıflandırma yöntemini bulmak için farklı sınıflandırma algoritmalarında test edilmiş %98,64 doğru sınıflama yüzdesine ulaşan Random Forest yöntemi önerilmiştir. Önerilen bu model bir C# kütüphanesine dönüştürülmüştür.

Lee ve Kang (2019), yaptıkları çalışmada CBOW yöntemi ile mesajlarda çıkarılan kelime temsilerini öznitelik olarak kullanarak SVM ve oluşturdukları ileri beslemeli yapay sinir ağ (Feedforward Neural Network) yöntemleri ile mesajları sınıflamışlardır. Yaptıkları çalışma sonucunda %95.87 doğru sınıflama oranı ile oluşturulan ileri beslemeli yapay sinir ağı önerilmiştir.

2. METİN MADENCİLİĞİ

Metin Madenciliği, istatistiksel ve matematiksel hesaplamaların yoğun olduğu bir alt yapıya dayanmaktadır. Metin madenciliği, verilen metinlerden yazar tanıma, başlık belirleme, özet çıkarma, anahtar kelime önerme, duygu analizi gibi alanlarda da kullanılmaktadır (Kılınç vd,2016).

Bilimsel çalışmaların hammaddesi veridir. Veri, yapılan araştırmalarda en önemli unsurdur. Veriler nicel olarak büyük olurlar fakat kullanım değerleri oldukça azdır. Veriler düzenlenip kategorize edilir, etiketlenir, belli bir yapıya çevrilir ise işte o zaman bilgiye dönüşmüş olur. Bilgi, nicel olarak küçük olsa da nitel olarak büyük etkiye sahiptir. Veriler yapılandırılmış (Structured Data) ve yapılandırılmamış (Unstructured Data) veri olarak iki gruba ayrılmaktadır. Yapılandırılmış veriler genellikle tablolara satır ve sütunlar şeklinde dökülmüş olarak düzenlenen verilerdir. Yapılandırılmamış veriler ise doküman, kitap, dergi gibi kâğıt üzerinde olan veya web sayfası gibi bilgisayardaki metinlerden oluşmaktadır. İstatistik, yapılandırılmış veriler üzerinde işlem yapabilirken, metin madenciliği yapılandırılmamış veriler üzerinde işlem yapabilmektedir (Oğuzlar, 2011). Şekil 2.1 ‘de metin madenciliğini oluşturan temel alanlar gösterilmiştir.



Şekil 2.1. Metin madenciliğini oluşturan temel alanlar

2.1. Metin Madenciliği Adımları

Metin madenciliği ile ilgili kaynaklar incelendiğinde, yapılan çalışmaların metin koleksiyonu oluşturma, metin önişleme, veri madenciliği, değerlendirme ve yorumlama adımlarından oluştuğu gözlemlenmektedir.

Metin koleksiyonu oluşturma aşaması işlenecek olan konu ile ilgili veri toplama sürecidir. Çevrimiçi olarak ulaşılan veri tabanları dışında kişisel bilgisayarlarda bulunan metin türündeki veriler de bu süreçte kullanılabilmektedir.

Metin önişleme aşamasında ise işaretleme, sözlük oluşturma, sonuca etki etmeyecek kelimeleri (stopwords) ayıklama, kelime köklerini bulma gibi işlemler yapılarak metin koleksiyonu işlemlere hazır hale getirilir. Yapılan çalışmaya göre doğru sonuca ulaştıracak verilerin seçilmesi işlemi özellik seçimi olarak adlandırılır. Tüm veri üzerinde çalışmak yerine seçilen özellikler üzerinde işlem yapılarak iş yükü azaltılmış olur. İşe yarayabilecek kelimelerin belirlenmesi ve sonucu etkilemeyecek kelimelerin çıkarılması işlemleri bu aşamada olur ve yapılandırılmış bir metin koleksiyonu haline gelir (Oğuzlar,2011).

Veri madenciliği, yapılandırılan metinlerin geleneksel istatistik yöntemler ile analiz edilme sürecidir. Değerlendirme ve yorumlama aşaması ise elde edilen sonuçların değerlendirilip anlaşılır bir hale getirilmesidir.

2.2. Doğal Dil İşleme

Doğal dil işleme; doğal dillerin yapısının, bilgisayarlar tarafından tanınması, yorumlanması ve yeniden üretilebilmesi aşamalarını kapsayan bir bilim alanıdır. Dilbilimi, matematik ve yazılım bilim dallarıyla yakından ilgilidir. Genel olarak yapay zekâ ve dilbiliminin alt kategorisi olarak tanımlanmaktadır (Tarcan ve Çakar, 2008). 50 yıldan fazladır doğal dil işleme alanında çalışmalar yapılmaktadır. Fakat değişen problemler ve karmaşık dil yapıları nedeniyle metni anlama ve anlamlandırma çalışmaları devam etmektedir (WEB-11).

Doğal dil işleme ve teknikleri genel çalışma alanları aşağıdaki gibidir (Tarcan ve Çakar, 2008);

- Konuşma dilini tanıma
- Yazılı dili tanıma
- El yazısını tanıma
- Metni konuşmaya çevirme

Günlük hayatımızda karşımıza çıkan metin verisini düşündüğümüzde problemin ne kadar büyük ve zorlayıcı olduğu anlaşılmaktadır. İşaretler, menüler, e-postalar, kısa mesajlar, web sayfaları, gazeteler, dergiler ve daha birçok örnek, metinlerin hayatımızda ne kadar yer tuttuğunu göstermektedir. Konuşmaları da göz önünde bulundurduğumuzda bahsedilen metin verisini tanımanın önemi daha da artmaktadır.

2.3. Zemberek Doğal Dil İşleme Kütüphanesi

Zemberek açık kaynak kodlu Türkçe Doğal Dil işleme kütüphanesidir. Java dili ile geliştirilen bu kütüphane ile kelimenin ekini ve kökünü ayırma, yazım hata denetimi, hatalı kelimeler için öneri, heceleme, metin normalizasyonu, metni kelimelere ayırma (tokenization) ve varlık ismi çıkarma (Named Entity Recognation) gibi işlemleri desteklemektedir. Çizelge 2.1'de Zemberek modülleri kısaca gösterilmiştir.

Çizelge 2.1. Zemberek modülleri

Modül Adı	Maven ID	Açıklama
Core	zemberek-core	Özel veri yapıları ve yardımcı sınıfları içerir.
Morphology	zemberek-morphology	Morfolojik analiz, belirsizlik çözümü, ve kelime üretmeyi sağlar.
Tokenization	zemberek-tokenization	Metinlerden cümleleri çıkarmayı sağlar.
Normalization	zemberek-normalization	Yazım denetleyicisi ve doğru kelime önerileri için kullanılır.
NER	zemberek-ner	Varlık ismi çıkarımı(NER) için kullanılır.
Classification	zemberek-classification	Metin sınıflama modülüdür.
Language Identification	zemberek-lang-id	Hızlı metin dili tanıma için kullanılır.

2.4. Metin Sınıflandırma

Son yıllarda hızla artan dijital verilerden anlamlı ve istenen bilgiye ulaşma önemli bir çalışma alanı oluşturmaktadır. Metin sınıflandırma da bu çalışma alanlarından biridir. İnternetin kullanımıyla birlikte dijital bilgi üretimi hızla artmış durumdadır. Artan bu bilginin büyük çoğunluğu metinlerdir. Bu metinlerin belirlenen özellikler göz önünde bulundurularak çeşitli sınıflar yaratılmasında metin sınıflandırma yöntemleri kullanılmaktadır (Blake,2011). Bu sınıflandırmalar, problemin türüne göre gözetimli ve gözetimsiz öğrenme olmak üzere iki yöntemle yapılmaktadır. Gözetimli öğrenme, bir uzman tarafından önceden tanımlanan sınıflar sayesinde geliştirilen yöntemin yeni metinleri kategorize etmede kullanılmasıdır. Gözetimsiz öğrenme ise etiketlenmemiş ve sınıflanmamış veriler üzerindeki bilinmeyen yapının tespiti için kullanılır (Akba, 2014).

3. MAKİNE ÖĞRENMESİ

Makine öğrenmesi, bilgisayarların karmaşık örüntüleri tanıyarak öğrenip ve daha sonra bu deneyimlerini kullanıp çıkarım yapabilen yapay zekânın alt çalışma alanlarından biridir. Bu yöntem, öğrenme türlerine göre üç şekilde sınıflandırılmaktadır(Atalay ve Çelik,2017):

- Gözetimli Öğrenme (Supervised Learning)
- Gözetimsiz Öğrenme (Unsupervised Learning)
- Pekiştirmeli (Takviyeli) Öğrenme (Reinforcement Learning)

Gözetimli Öğrenme: Bu öğrenme yönteminde etiketlenmiş, yani önceden sınıflandırılmış girdiler ile sınıflar arasında ilişki öğrenilerek model oluşturulur. Daha sonra sınıfı bilinmeyen girdilerde oluşturulan modele göre doğru sınıflara en yakın tahmini üretmesi hedeflenir.

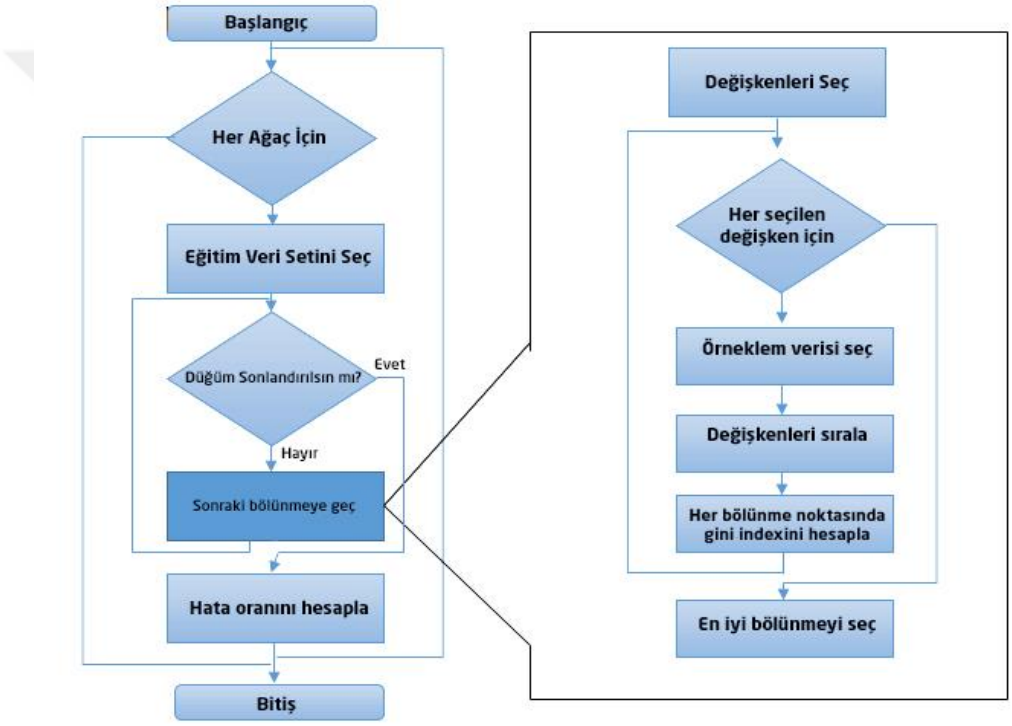
Gözetimsiz Öğrenme: Bu yöntemde ise sınıflandırılmamış girdi değerleri arasında ilişki modellenir. Bu model ile birbirine yakın değerler kümelenmektedir. Yeni girdi modele göre yakın olan kümeye dahil edilir.

Takviyeli öğrenmede ise işleyiş biraz daha farklıdır. Yapılan sınıflamada hedefe ne kadar yakın olduğuna göre ödül veya ceza kriteri ile sistemin sürekli öğrenmeye devam etmesi sağlanır (Sutton ve Barto, 1998).

Metin sınıflandırma problemlerinde çeşitli makine öğrenmesi yöntemleri kullanılmaktadır. Bu tez çalışmasında kullanılan temel yöntemler Random Forest, Naive Bayes Multinomial, SVM, Multilayer Perceptron, Random SubSpace, Logistic Regresyon, k-Nearest, ve derin öğrenme yöntemlerinden LSTM, CNN, Yapay Sinir Ağ kullanılmıştır. Bu yöntemler alt başlıklar olarak devam eden alt bölümlerde anlatılacaktır.

3.1. Rastgele Orman (Random Forest)

Rasgele Orman algoritması, Breiman (2001) tarafından geliştirilen makine öğrenmesi algoritmalarından biridir. Bu algoritma hem veri seti hem de öznitelik setinden rasgele farklı örneklem seçmekte ve sınıflandırmaktadır. Bu sınıflandırmalar hata oranlarına göre oy alırlar. Rasgele orman algoritması da en çok oy alan karar ağacını seçer (Coşkun vd, 2009). Şekil 3.1'te rasgele orman algoritması akış şeması gösterilmiştir (Baraga vd, 2007).



Şekil 3.1. Rasgele orman algoritması akış şeması

3.2. Naif Bayes Sınıflandırıcısı (Naive Bayes)

Bayes teoremi 1812 yılında Thomas Bayes tarafından bulunmuştur. Koşullara bağlı olasılıkçı bir yaklaşım ile elde bulunan verilerin kategorilerini tespit etmeyi, sınıflarını

belirlemeyi hedefler. Bu yöntemde sisteme etiketlenmiş (sınıflandırılmış) veriler verilerek sistemin eğitilmesi amaçlanır. Formül 3.1'de Bayes formülü gösterilmiştir.

$$P(\mathrm{A} \mid B) = \frac{P(B \mid \mathrm{A})P(\mathrm{A})}{P(B)} \tag{3.1}$$

P(A|B) = B olayı gerçekleştiğinde A olayının gerçekleşme olasılığıdır

P(B|A) = A olayı gerçekleştiğinde B olayının gerçekleşme olasılığıdır

P(A) = A olayının gerçekleşme olasılığı

P(B) = B olayının gerçekleşme olasılığı

NB (Naive Bayes), bayes olasılık teoremine dayanmaktadır. Bu algoritma test verilerini formüle göre işleyerek öğrenme işlemini tamamlar ve her bir durum için yüzdelik değer üretir ve bu olasılıklara göre sınıflandırma işlemini gerçekleştirir. C sınıfı temsil ediyor olsun $x =< x1, x2, x3, \ldots, xm >$ ise çıkarılan nitelik değerleri olsun. x test değerlerine göre sınıfı tahmin etmek için Bayes Teoremi olasılık hesaplar:

$$p(C = c_j \mid \mathrm{X} = x) = \frac{p(C = c_j)\, p(X = x \mid \mathrm{C} = \mathrm{c_j})}{p(X = x)} \tag{3.2}$$

Daha sonra yüksek olasılık değerine göre sınıfı tahmininde bulunur. Bu örnekte $X = x$ durumu $X1 = x1 \wedge X2 = x2 \wedge X3 = x3 \wedge \ldots \wedge Xm = xm$ ifade eder. $p(X = x)$ sınıflar arasında değişme göstermediği durumda ihmal edilir ve 3.2 numaralı denklem aşağıdaki şekle gelir.

$$p(C = c_j \mid \mathrm{X} = x) = p(C = c_j)\, p(X = x \mid \mathrm{C} = \mathrm{c_j}) \tag{3.3}$$

$(\mathrm{C} = \mathrm{c_j})$ ve $p = (\mathrm{X} = x \mid \mathrm{C} = \mathrm{c_j})$ öğrenme verilerinden tahmin edilir. $x1, \ldots, xm$ nitelikleri şartlı olarak birbirinden bağımsızdırlar. Böylece 3.3 numaralı denklemin son hali aşağıdadır:

$$p(C = c_j \mid \mathrm{X} = x) = p(C = c_j) \prod_{i=1}^{m} p(X_i = x_i \mid \mathrm{C} = \mathrm{c_j}) \tag{3.4}$$

3.4 numaralı son denklem ile test verileri için tahminde bulunulmaktadır. (Sağbaş ve Ballı, 2016).

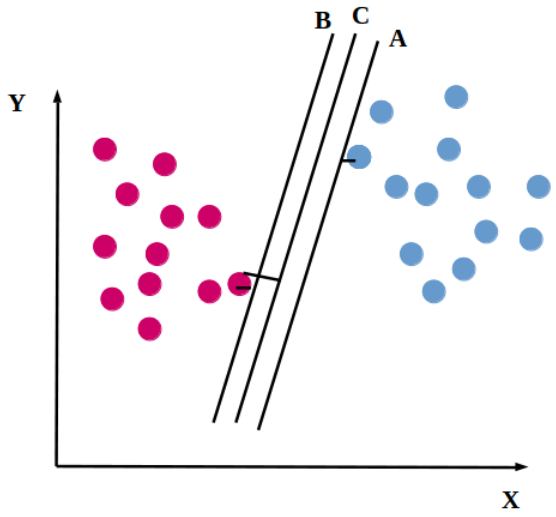
3.3. Destek Vektör Makinesi (Support Vector Machine - SVM)

Koordinat düzleminde bulunan iki grubu aralarına bir sınır çizerek ayırmak mümkündür. Yeni gelecek bireyi sınıflandırmada bu sınırın kullanılacağı düşünüldüğünde; sınırın çizileceği yer iki grubun üyelerine en uzak yer seçilmelidir. Bu sınırın belirlenmesi işlemi SVM tarafından yapılır (Ben-Hur vd, 2001).

SVM, ikili sınıflandırma problemlerinde başarı oranı daha yüksektir. Çünkü iki veri kümesi arasına konulabilecek sınır olasılığı daha çok olacaktır. Eğitim aşamasında düzlemde bulunan her bir üyenin hangi gruba ait olduğu etiketlenmiş olmalıdır (Emhan, 2017).

SVM algoritması ayrıca yeni gelen verinin hangi gruba ait olduğunu ayırt edebilir. SVM, eğitim sürecinde, grup üyelerinin arasındaki mesafeyi en yüksek seviyede ayırabilecek hiper-düzlemi belirlemeye çalışır. Bu süreçte temel olarak, eldeki veri setini daha yüksek bir boyuta dönüştürülerek kümeleri ayrıştırabilecek hiper-düzlemi oluşturmaya çalışılır (Yıldırım 2012; Emhan, 2017).

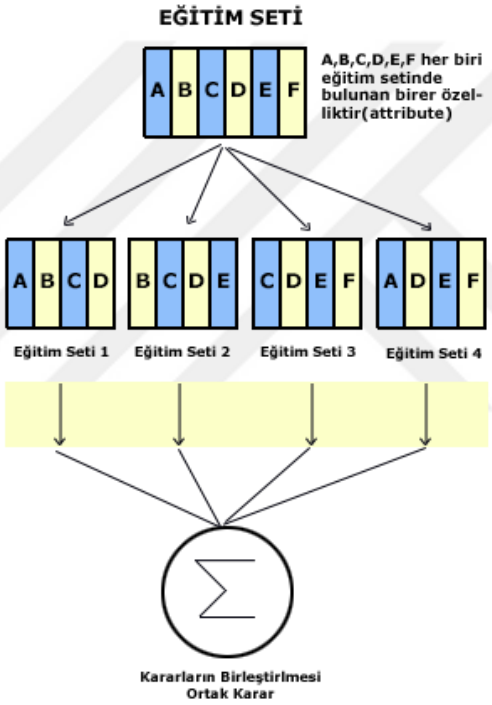
Şekil 3.2'de hiper-düzlemler incelendiğinde C'nin uzaklığının A ve B'ye göre daha yüksek olduğunu gözlemlenmektedir. Yani, C hiper düzlemi seçmek daha mantıklı olacaktır. Daha yüksek uzaklığa sahip hiper düzlemi belirlemek daha güçlü ve kararlı ayrımı yapabilmeyi sağlamaktadır. Düşük uzaklığa sahip bir hiper-düzlem seçilirse yeni bir değeri sınıflarken yapılabilecek hata yüzdesi artmaktadır (WEB-12).



Şekil 3.2. Büyük marjin ayrımı

3.4. Rasgele Alt Uzay (Random SubSpace)

Random Subspace algoritması, veri setindeki her bir öğrenici özniteliklerin tamamını seçmek yerine rasgele farklı öznitelikler seçerek alt eğitim setleri oluşturulur. Ho (1998), öznitelik seçimi yapılırken toplam öznitelik sayısının yarısını kullanmayı önermiştir. Oluşturulan yeni setler eğitilerek farklı kararlar ile birleştirilip nihai karar oluşturulur. Şekil 3.3'te Random Subspace çalışma prensibi örneği gösterilmiştir (Kökçü vd., 2014).



Şekil 3.3. Random Subspace çalışma prensibi

3.5. Lojistik Regresyon (Logistic Regression)

Bu algoritmanın ana hedefi, diğer model yapılandırma tekniklerinde olduğu mümkün olan en az değişken ile maksimum seviyede uyuma sahip değişkenler arası ilişkiyi kurmaktır (Atasoy, 2001). Lojistik regresyon bazı varsayım bozulmalarında diskriminant ve çapraz tablo yöntemlerine alternatif olarak kullanılmaktadır. Ayrıca bağımlı değişken 0 ve 1 yani ikili veya ikiden daha fazla kategori içeren kesikli değişken olduğunda normallik varsayımı bozulmakta ve doğrusal regresyon analizi uygulanmamaktadır (Şahin, 2018).

Lojistik regresyonda, bazı değişken değerlerinden yola çıkılarak ilgilenilen tahmin edilmeye çalışılır. Temel olarak, bir açıklayıcı değişken (explanatory variable) var iken tahmin edilmeye çalışılan durumun olasılığını hesaplamak için kullanılan model formül 3.5 te gösterilmiştir (Binokay, 2018).

$$P(\mathrm{y}) = \frac{1}{1+e^{-(\beta_0+\beta_1 \mathrm{x})}} \tag{3.5}$$

Formül 3.5'deki P(y), Y'nin gerçekleşme olasılığıdır, $\beta_0+\beta_1 x$ katsayısı basit doğrusal regresyondan gelmektedir. Birden fazla açıklayıcı değişken olan durumlarda lojistik model formül 3.6'daki şekle dönüşür (Binokay, 2018).

$$P(\mathrm{y}) = \frac{1}{1+e^{-(\beta_0+\beta_1 \mathrm{X}_1+\beta_2 \mathrm{X}_2+\ldots+\beta_n \mathrm{X}_n)}} \tag{3.6}$$

Lojistik regresyonda doğrusal model elde etmek için logaritmik dönüşüm uygulanır. Formül 3.7 de formül 3.6'nın doğal logaritmasının alındığı hali gösterilmektedir (Binokay, 2018).

$$\mathrm{g(y)} = \ln\left[\frac{P(x)}{1-P(x)}\right] = \beta_0 + \beta_1 \mathrm{X}_1 + \beta_2 \mathrm{X}_2 + \ldots + \beta_n \mathrm{X}_n \tag{3.7}$$

3.6. K-En Yakın Komşu (K-Nearest Neighbor)

K-Nearest Neighbor (K-En Yakın Komşu), KNN olarak bilinen bu sınıflandırma yöntemi temelde 'birbirlerine yakın noktalar muhtemelen benzer küme üyeleridir' yaklaşımına dayanır (Eren, 2008). Bu yöntemin amacı, yeni dahil olan nesnenin özellikleri göz önünde bulundurularak önceden sınıflandırılmış komşu nesneler aracılığıyla sınıflama işlemidir. Henüz kategorileştirilmemiş nesnelere sınama örneği, önceden kategorize edilmiş ve sınıfları belli olan örneklere ise öğrenme örnekleri denilmektedir. Kategorize edilmemiş sınama örneği, önceden sınıflandırılan öğrenme örneklerine olan uzaklıkları hesaplandıktan sonra en yakın k adet komşu nesnenin en fazla hangi sınıfa ait üyesi olduğuna bakılarak tahmin yapılır (Lee vd., 2014).

Eğitim verileri çok olduğunda veya gürültüye sahip öğrenme verilerinde etkili sonuçlar alınabilir. Anlaşılması ve uygulanabilirliği kolay olduğundan sık tercih edilir. k sayısının belirlenmesinin gerekliliği, lazy learner (tembel) bir algoritmadır çünkü eğitim setinden öğrenmez sadece eğitim setleriyle beraber sınıflama işlemi yapar. Bu durum hesaplama zamanının yani maliyetin fazla olması demektir.

3.7. Derin Öğrenme

Derin öğrenme, insan sinir sisteminin belirli bir problemi çözmekte kullandığı yöntem ve kabiliyetlerinin taklit etmeye çalışarak, büyük boyutlardaki verileri kullanıp öznitelik çıkarımı, sınıflandırma, dönüştürme gibi işlemleri gerçekleştirmeyi amaçlayan makine öğrenme yöntemidir. Bu yöntem büyük boyutlardaki denetimsiz verilerin ayırt edici özelliklerini kendisinin öğrenebildiği, gizli katmanlar ve özel işlem basamaklarıyla özelleştirilmiş yapay sinir ağları çeşididir. (Kın, 2019)

Derin öğrenme, yapay sinir ağının ilk evrelerinden itibaren yerine getirmesi gereken görevlere göre çeşitli kategorilere ayrılmıştır (Karakuş, 2018).

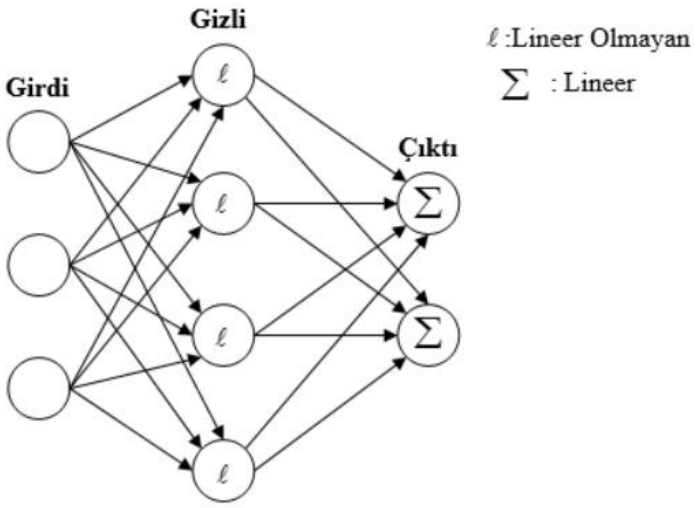
- Tek Katmanlı Algılayıcılar (Perceptron)
- Çok Katmanlı Algılayıcılar(Multilayer Perceptron)
- Konvolüsyonel Sinir Ağları (Concolutional Neural Network)
- Tekrarlanan Sinir Ağları (Recurrent Neural Network)

- Uzun/Kısa Süreli Bellek Ağları (LSTM)
- Sıralı Modeller (Sequence to Sequence Models)
- Sığ Derinlikli Ağlar (Shallow neural networks)
- Gan (Generative Adversarial Nets) Ağları

3.7.1. Çok katmanlı perseptron (Multilayer Perceptron)

Multilayer Perceptron, 1969 yılında M.Minsky ve S.Papert tarafından önerilmiştir. Perseptron sinir ağının genişletilmesi ile oluşturulmuştur. Girdi ve çıktı katmanı arasında bir veya daha çok sinir katmanından oluşur. Önceki katmandan girdiyi alarak sonraki katmana ileten bu katmanlar gizli katman olarak da adlanır (Polat, 2003).

Şekil 3.4'te çok katmanlı perseptron yapısı gösterilmektedir. Bu yapıya göre çıkış değerleri, hata fonksiyonlarının değerini hesaplamak için beklenen çıktı ile karşılaştırılır. Eğer beklenen değer ile üretilen çıktı farklı ise hata olduğu anlaşılır ver geri besleme yöntemiyle hata ağırlıkları dağıtılarak her iterasyonda bu hata oranları azaltılmaya çalışılır. Yeteri kadar eğitim verisi için gerçekleştirilen bu döngüler sayesinde hesaplama hataları düşük olan bazı durumlar elde edilir. Böylelikle oluşturulan ağın bir hedef işlevi öğrenmiş olur (Şanlı, 2018).

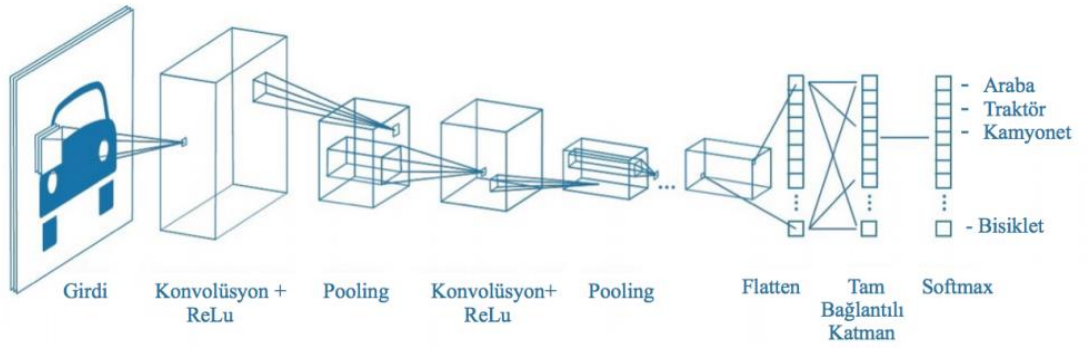


Şekil 3.4. Çok katmanlı perseptron modeli

3.7.2. Konvolüsyonel sinir ağları(Convolutional Neural Networks - CNN)

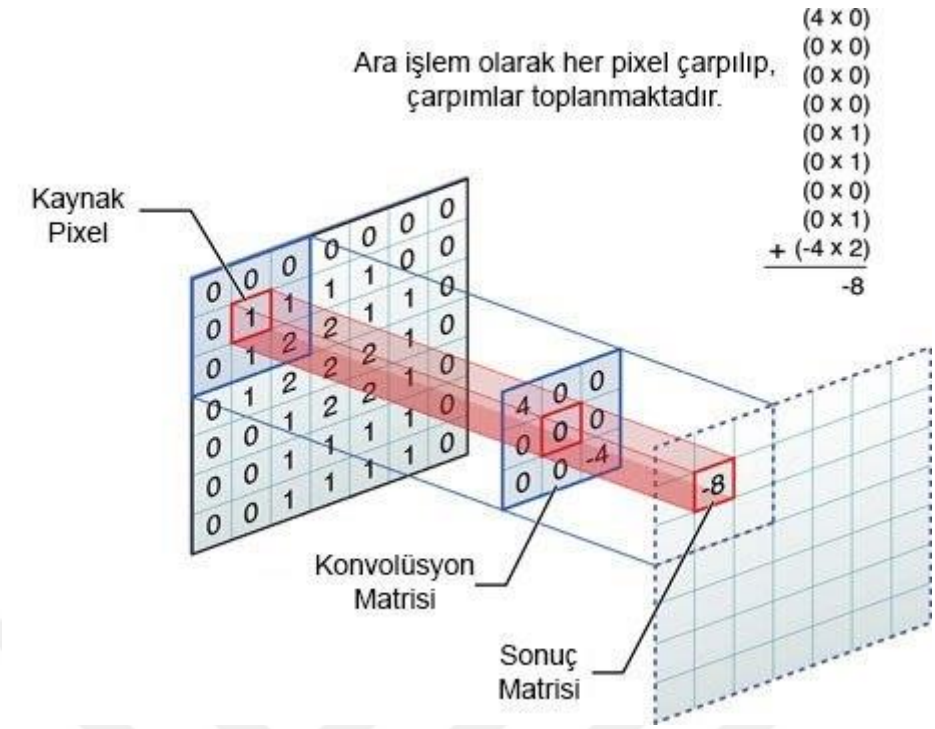
CNN (Convolutional Neural Networks)'ler temelde evrişim işleminin gerçekleştirildiği birden fazla katmandan oluşur. ANN (Artifical Neural Network) ile CNN arasındaki en önemli fark katmanlar arasında bulunan bağlantılardır. CNN'de girişlerin lokal kısımları tek nörona bağlanırken, ANN'de girişler sonraki katmandaki tüm nöronlara bağlı olur. Gizli katman sayıları aynı olduğunda tam bağlı ağlarda sistem eğitimi daha az parametre ile daha kolay hale gelmektedir (Gündüz, 2019).

CNN'ler öznitelik çıkarımı (feature extraction) ve sınıflandırma(classification) olarak iki kısımda incelenebilir. Öznitelik çıkarımı aşamasında ağ, konvolüsyon katmanı (convolution layer) ve havuzlama katmanındaki (pooling layer) bazı işlemler uygulanarak öznitelikler oluşturulur. Sınıflandırma aşamasında ise tam bağlantılı katman (fully connected layer) belirlenen öznitelikleri kullanarak sınıflandırma işlemini girdilere göre olasılık tahminleri üreterek yapmaktadır (Pervan, 2019). Şekil 3.5'te bir görsel üzerinde CNN mimarisinin nasıl uygulandığı gösterilmiştir.



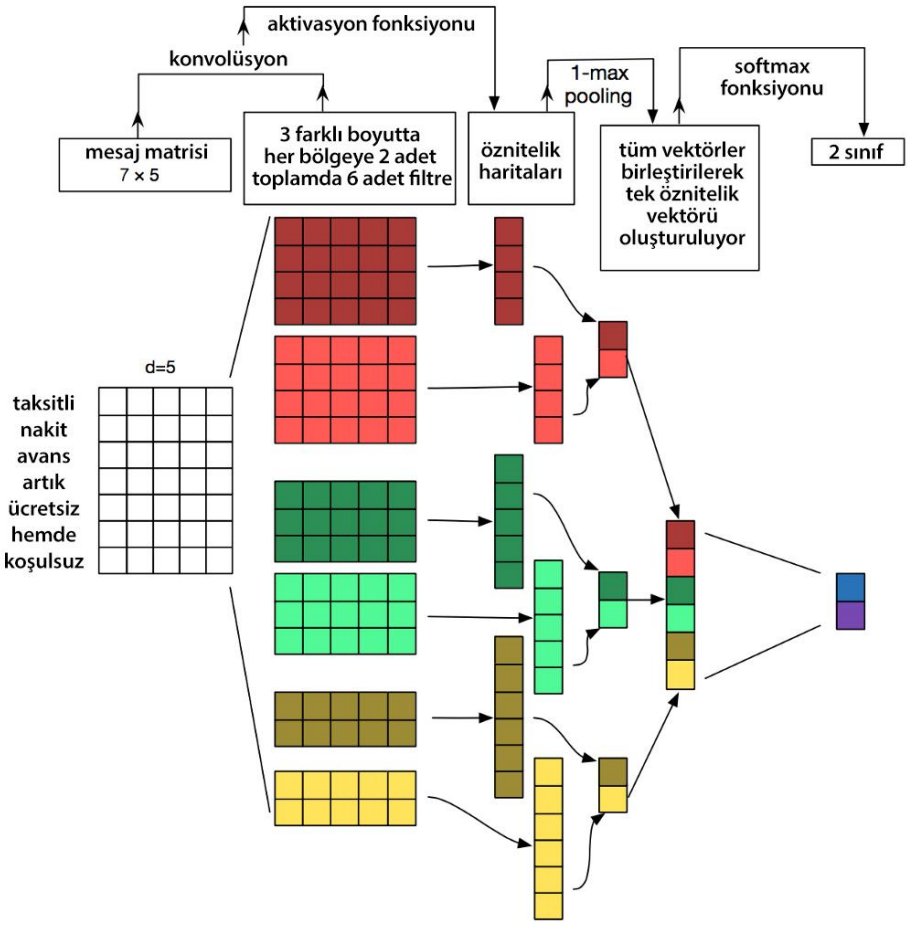
Şekil 3.5. Bir görsel üzerinde CNN uygulanması

Konvolüsyon iki fonksiyonun birleşimden başka bir fonksiyon elde etme işlemidir. Konvolüsyon katmanında öznitelik haritası oluşumu için kare bir filtrenin sol üst köşeden başlayarak, vektör üzerinde dolaşıp matris çarpımı yapılır ve her bir piksel için yeni matrisler üretilir. Şekil 3.6'da bir konvolüsyon örneği gösterilmiştir (Web-13).



Şekil 3.6. Konvülasyon örneği

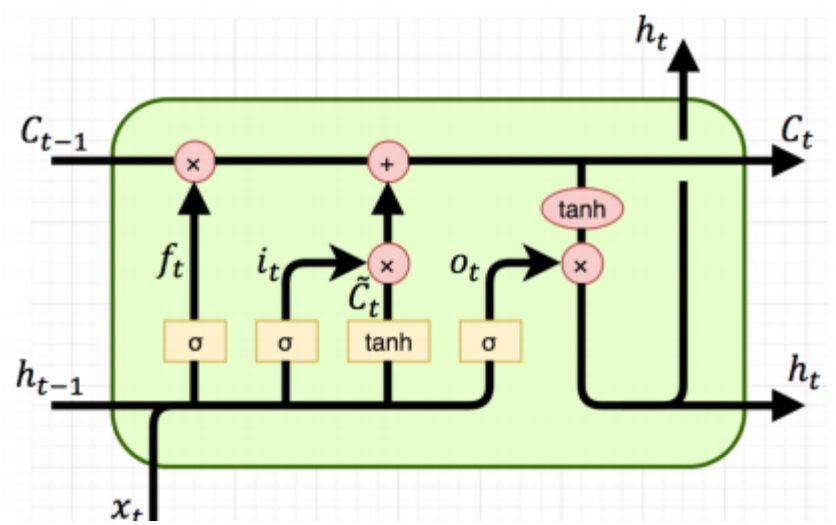
CNN'ler genellikle görsel veriler için kullanılsa da metin veriler sınıflama problemlerinde de tercih edilmektedir. Metin türündeki veriler, 2 boyutlu görsellerden farklı olarak, tek boyutlu giriş verilerinden oluşur. Bu nedenle, konvolüsyon katmanları tek boyutlu tercih edilmektedir. Şekil 3.7'de gösterilen örnek bir metin sınıflandırma mimarisinde, Spam ve normal mesajları içeren bir veri girdisinin temsil katmanı 2 boyutlu matristen oluşmaktadır. Bu matristeki her bir satır sözlükte bulunan benzersiz bir kelimeyi temsil eder ve o satırlardaki tüm elemanlar kelimelere ait vektörlerdir. Örneğin 7x5 boyutlu bir matriste, 7 farklı kelime için 7 satırı bulunur ve her bir kelime 5 elemanlı vektör ile temsil edilir. Değişken boyutlarda filtreler kullanarak hazırlanan matris üzerinde tarama işlemi yapılmaktadır. Değişkenlik gösteren filtre yüksekliği tek seferde kaç kelime kaydırılacağını işaret etmektedir (Karakuş, 2018).



Şekil 3.7. Metin sınıflandırmada CNN mimarisi

3.7.3. Uzun kısa vadeli hafıza (Long-Short Term Memory- LSTM)

LSTM ağları uzun süreli bağımlılıkları öğrenebilen bir RNN (Tekrarlayan Sinir Ağı) mimarisidir. Hochreiter ve Schmidhuber (1997) tarafından ele alınarak tanıtılan bu yöntem farklı problem çözümlerinde kullanılmaktadır. LSTM'yi bir metin sınıflama problemi üzerinden ele alacak olursak herhangi bir kelimenin öncesindeki ve sonrasındaki kelime bilgileri kullanılarak ele alınan kelime hakkında değerlendirmelerde bulunulabilir. Fakat bazen doğru bilgiye ulaştıracak olan kelime üzerinde durulan kelimeden çok uzakta olabilir. İşte tam da bu durumlarda LSTM yaklaşımı ile probleme çözüm üretilir. Uzun süreli bilgi hatırlanmasını sağlayan bu yöntem sinir ağını belirli tekniklerle haberleşmenin sağlandığı 4 ayrı katmanı kullanır (Pervan, 2019).



Şekil 3.8. LSTM İç yapısı ve katmanlar

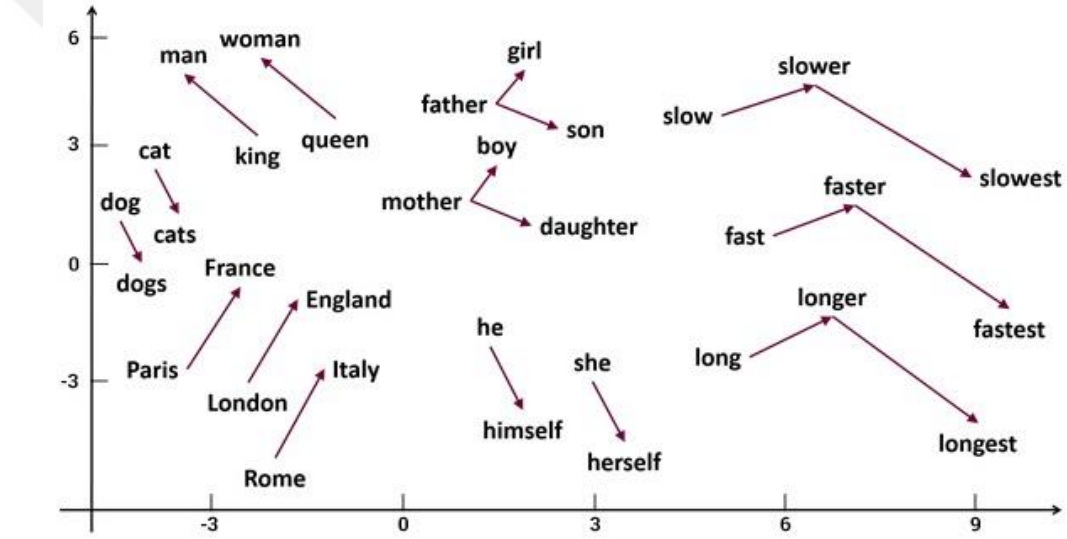
Şekil 3.8'de gösterilen birinci sigmoid katmanı bilginin hücreden atılıp atılmayacağına karar verme aşamasıdır (Gündüz, 2019). Forget gate layer (f_t) (Unut Kapısı Katmanı) olarak adlandırılan bu bölümde t-1 de ki çıkış değeri(h_{t-1}) ile t zamanındaki giriş değerini(x_t) ele alarak hücre durumundaki (C_{t-1}) her bir girdi için 0 ile 1 arasında bir çıktı üretir. 1 değeri "bunu bilgiyi tam olarak sakla" anlamına gelirken 0 ise "bu bilgiyi tamamen sil" anlamına gelmektedir. Sonraki aşama ise depolanacak yeni bilgiye karar verme aşamasıdır. Bu bölüm input gate layer (i_t)(Giriş Kapısı Katmanı) ve aday vektör (C_{tn}) yaratan tanh katmanını birleştirerek C_{t-1} yeni hücre durumuna (cell state) yani C_t ye çekilir.

Daha sonra ise LSTM'nin çıktı değerine karar verme aşamasına geçilir. Çıktı hücrenin son durumunun filtreleme işleminden sonraki değeridir (h_t) (Gündüz, 2019).

3.8. Word2Vec

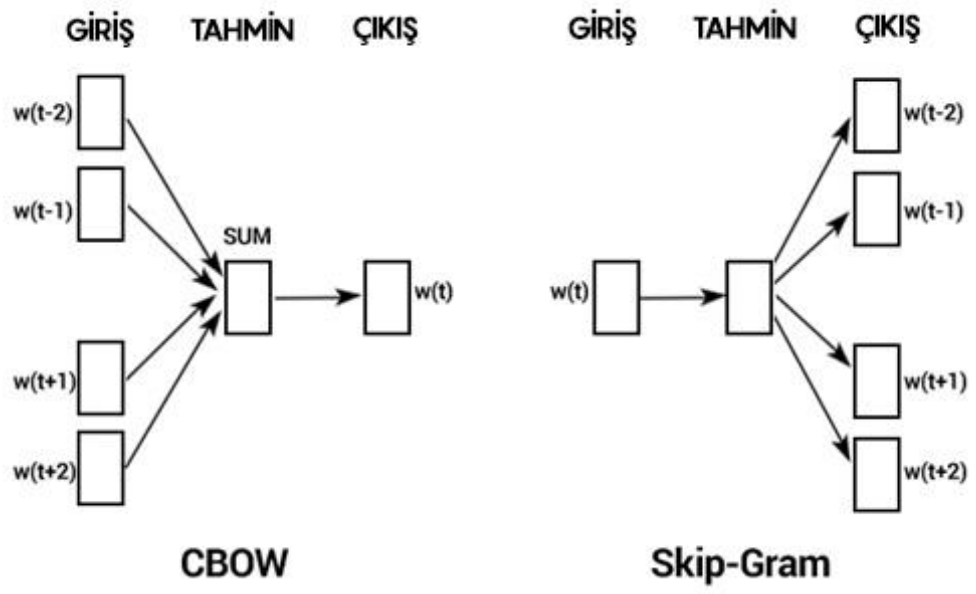
Word2Vec derin öğrenme amaçlı da kullanılan açık kaynak bir araç olarak 2013 yılında Google tarafından yayınlanmıştır (Zhang vd 2015). Bu araç sayesinde kelimeler vektörlere dönüştürülerek aralarındaki uzaklıklar kosünüs benzerliğine (cosine similarity) göre hesaplanıp kelimeler arasında analoji kurulabilmektedir.

Google News metinlerinden oluşan model ile yapılan örneklere bakacak olursak (Web-4), Şekil 3.9'da (Web-5) de görüldüğü gibi birbirleriyle benzerlik gösteren kelimeler yakın vektörler ile temsil edilmiştir. Örneğin dog ve cat birbirine yakın alanlarda temsil edilmektedir. Ayrıca kelimeler arasındaki ilişkiler benzer kelime öbeklerinde de yakın ilişkiler sunmaktadır. Yani "fast", "faster" ve "fastest" arasındaki ilişki "long", "longer" ve "longest" ile benzerlik göstermektedir. Word2Vec ile oluşturulan model kullanılarak kelimelerin birbirine olan benzerliklerine ulaşılabilir. Google News ile hazırlanan modelde "Man" kelimesine en yakın kelime "Woman" (0.69393444061279 benzerlik değeri ile). Belirli bir kelime grubunda alakasız olanı kelimeyi de doesnt'match fonksiyonu ile ayırt edebiliyor. doesnt_match("blue red green yellow book") komutu cevap olarak "book" kelimesini döndürüyor.



Şekil 3.9. Mesaja anahtar kelimeleri eklemek için kullanılan kod bloğu

Word2Vec metinleri işleyen 2 katmanlı sinir ağına sahiptir. CBOW ve Skip-Gram Word2Vec'te kullanılan ana öğrenme modelleridir. Şekil-3.10'da CBOW ve Skip-Gram algoritmalarının çalışma prensipleri gösterilmektedir **(Web-6)**. CBOW modeli hedef kelimeye göre komşu kelimeleri tahmin etmek için, Skip-Gram ise komşu kelimelerden hedef kelimeyi tahmin etmek için kullanılmaktadır (Wensen vd., 2016) (Mikolov vd., 2013).



Şekil 3.10. CBOW ve Skip-Gram modelleri çalışma prensipleri

3.9. Değerlendirme ölçütleri

Sınıflandırma algoritmaları değerlendirilirken Accuracy, Precision (Kesinlik), F-measure (F-Ölçütü), Recall (Duyarlılık) ve ROC Curve (Receiver Operating Characteristic Curve - Alıcı İşletim Karakteristik Eğrisi) parametreleri kullanılmaktadır. Accuracy (ACC) doğru tahminlerin tüm tahminlere oranıdır. Doğruluk oranı için 3.8 numaralı formül kullanılmıştır (Ballı ve Sağbaş 2017). TP (True Positive) değeri Spam olarak sınıflandırılan Spam mesaj sayısıdır. FP (False Positive) değeri, Spam olarak sınıflandırılan ama Spam olmayan mesajların sayısıdır. TN (True Negative) Ham olarak sınıflandırılan Ham mesaj sayısıdır. FN (False Negative) Ham olarak sınıflandırılan ama Spam mesaj sayısıdır.

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} \times 100 \tag{3.8}$$

Precision, pozitif tahminlerin başarı oranını verir ve kesinlik değerini gösterir. Recall doğru sınıflanmış pozitif örnek sayısının toplam setteki pozitif örnek sayısına oranıyla hesaplanır ve hassasiyeti gösterir. Bu iki ölçütü beraber değerlendirmek için F-measure kullanılır. Recall ve Precision'ın harmonik ortalamasıdır.

$$\text{Precision(p)} = \frac{TP}{TP + FP} \tag{3.9}$$

Formül 3.9 ‘da gösterilen precision parametresi kesinlik olarak tanımlanabilir. Tüm verilerdeki doğru tahmin oranını verir. Problem çözümlerinde beklenen, bu değerin yüksek olmasıdır.

$$Recall(\text{r}) = \frac{TP}{TP + FN} \tag{3.10}$$

Formül 3.10’da gösterilen Recall parametresi hassasiyet-duyarlılık olarak adlandırılabilir. Pozitif doğru sınıflandırma oranını verir. Doğru sınıflandırılan pozitif değerlerin tüm pozitif değerlere oranı ile bulunur.

$$F - Measure = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.11}$$

Formül 3.11‘de gösterildiği gibi kesinlik (Precision) ve hassasiyet (Recall) değerlerinin harmonik ortalaması F-Ölçütü değerini verir. İki parametreyi beraber değerlendirme fırsatı verir.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n}} \tag{3.12}$$

Formül 3.12 deki RMSE (ortalama hata kareleri kökü) tahmin edilen değerler ile olması gereken değerler atasındaki uzaklık hesabıyla bulunur ve toplam hata değerini belirlemeye yardımcı olur.

4. UYGULAMA

Bu tez çalışmasında, içerik tabanlı SMS filtreleme için yüksek doğruluk oranına sahip bir yöntem bulunması ve buna yönelik uygulama geliştirilmesi hedeflenmiştir. Uygulama çalışması, literatürde çok kullanılan bir İngilizce veri seti ve yeni oluşturulmuş Türkçe veri setleriyle denemelerin yapıldığı iki bölümden oluşmaktadır.

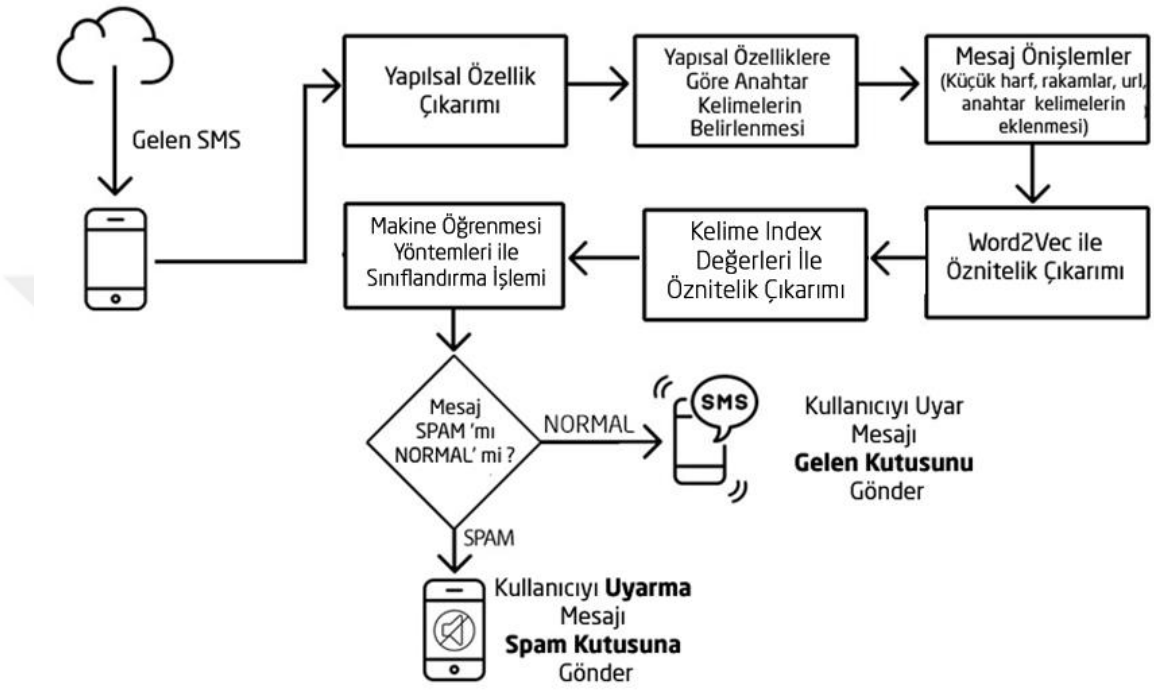
İngilizce veri setinin sınıflandırılmasını anlatan birinci bölümde, gelen mesajın yapısal özellikleri bulunduktan sonra bu yapısal özelliklere göre anahtar kelimeler üretilerek mesaja eklenmiştir. Harf küçültme, rakamların çıkarılması gibi ön işlemlerden sonra Word2Vec aracılığıyla iki yeni öznitelik elde edilmiş ve bu öznitelikler ile Random Forest, Multi Layer Perceptron, SVM, Logistic Regression ve Naive Bayes algoritmalarının doğru sınıflama yüzdeleri karşılaştırılmıştır.

Türkçe veri setiyle SMS sınıflandırma aşamalarından bahsedilen ikinci bölümde ise yapısal öznitelikler, Word2Vec ile oluşturulmuş yeni öznitelikler ve sözlük oluşumu ile hazırlanan mesaj temsillerinden oluşturulan öznitelikler kullanılarak klasik makine öğrenme yöntemlerinin yanı sıra derin öğrenme algoritmaları da kullanılarak mesajlar sınıflandırılmış ve sonuçlar tartışılmıştır.

Oluşturulan metin koleksiyonlarının ilk halleri yalnızca SMS'lerden oluşmaktadır. Bu metinler yapılandırılmamış verilerdir. Daha sonra uygulanan önişlemler ve öznitelik belirlemelerinden sonra oluşturulan veri seti yapılandırılmış veri haline getirilmiştir.

Veri seti oluşturulduktan sonra yapısal özelliklerin belirlenmesi, köklerin bulunması, anahtar kelimelerin çıkarılması ve mesajlara eklenmesi adımlarında C# dilinde hazırlanan fonksiyonlar kullanılmıştır. Word2vec kütüphanesi ve sınıflandırma algoritmaları için Python dili kullanılmış Anaconda (Web-15) geliştirme ortamında çalıştırılmıştır. Yapılan çalışmalar sonunda geliştirilen uygulama Android Stuido (Web-14) ortamında hazırlanmıştır.

Şekil 4.1'de bu çalışmada SMS filtreleme için tasarlanan modelin akış şeması gösterilmektedir. Gelen mesajın ön işlemleri, özniteliklerinin belirlenmesi ve sınıflandırma işlemi alt başlıklarda anlatılmıştır.



Şekil 4.1. Tasarlanan sistemin akış şeması

4.1. İngilizce Veri Seti İçin Spam Tespiti

Tez çalışmasının bu aşamasında aynı konuda çalışan diğer araştırmalarda da kullanılan İngilizce veri seti seçilmiş; sadece mesaj metinleri ele alınarak hazırlanmış olan derin öğrenme tabanlı metin sınıflama algoritmasıyla, Android uygulaması geliştirilme aşamaları anlatılmaktadır.

4.1.1. Veri seti

SMS Spam Collection veri seti Almeida vd.(2011) tarafından hazırlanmıştır. Veri seti 4827 normal, 747 Spam olmak üzere toplam 5574 satır kısa mesajdan oluşmaktadır.

4.1.2. Verilerin hazırlanışı

İçerik tabanlı sınıflama çalışmalarında mesajların yapısal özelliklerinin sonuca olumlu katkıları gözlemlenmiştir (Karasoy ve Ballı 2016). Bu kapsamda, mesajın uzunluğu (ML), büyük harf (CWR) ve duygusal ifade frekansı (Emoji), URL (Url) bulundurma gibi özelliklerin sınıflandırmada yardımcı olabileceği saptanmıştır. Spam mesajların uzunluklarının 160 karakter ve katlarına yakınlığı, genellikle bir URL bulundurmaları, büyük harf kullanım oranlarının yüksekliği gözlemlenmiştir. Normal mesajlarda ise duygusal ifadelerin kullanımı ayırt edici bir özellik olarak karşımıza çıkmaktadır (Ballı ve Karasoy, 2019).

Çizelge 4.1. Spam mesajlarda sık rastlanan kelimeler

Sınıf	Sık kullanılan kelimeler
Spam	free, the, for, txt, have, from, mobile, com, stop, claim, reply, of, prize, our, only, won, cash, uk, win, send, nokia, new, urgent ..

Ayrıca veri setinde Spam mesajlarda sık geçen 80 kelime seçilerek frekanslarına göre 1 den 80'e kadar puanlar verilmiştir. Çizelge 4.1'de sık geçen kelime örnekleri gösterilmiştir. Bu seçilmiş kelimelerin mesajlardaki bulunma durumlarına göre kelime puanları toplanarak her mesajın Spam Ağırlık Değeri hesaplanmıştır. Çizelge 4.2'de mesajların yapısal özelliklerine ait değerlerin örnekleri gösterilmiştir.

Çizelge 4.2. Mesajların yapısal özelliklerinin belirlenmesi

Sıra	Mesaj	ML	CWR	Url	Emoji	SMW	Class
1	I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today.	109	0.022	0	0	187	Ham
2	SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 info	136	0.153	0	0	694	Spam
3	URGENT! You have won a 1 week FREE membership in our £100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18	155	308	1	0	838	Spam
4	I've been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times.	196	0.019	0	0	807	Ham
5	I HAVE A DATE ON SUNDAY WITH WILL!!	35	0.929	0	0	72	Ham
6	XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here>> http://wap. xxxmobilemovieclub.com?n=QJKGIG HJJGCBL	149	0.176	1	0	448	Spam
7	Oh k...i'm watching here:)	26	0.043	0	0.043	0	Ham
8	Eh u remember how 2 spell his name... Yes i did. He v naughty make until i v wet.	81	0.048	0	0	0	Ham
9	Fine if that?s the way u feel. That?s the way its gota b	56	0.045	0	0	154	Ham
10	England v Macedonia - dont miss the goals/team news. Txt ur national team to 87077 eg ENGLAND to 87077 Try:WALES, SCOTLAND 4txt/ú1.20 POBOXox36504W45WQ 16+	155	0.242	0	0	187	Spam

Özellikler metne çevrilirken Şekil 4.2'deki kod parçacığı kullanılmıştır. Yapılan denemeler sonucunda belirlenen bazı anahtar kelimeler (MaxiSpam, MiniMessage) mesaja ikinci defa eklenip ağırlıkları artırılmıştır.

ML: Mesajın tüm karakterlerinin toplam sayısı

CWR: Mesajdaki büyük harflerin toplam karakter sayısına oranı 0 ile 1 arasında bir değer alır.

Url: Mesajda link olup olmama durumu. 0 veya 1 değerini alır. Birden fazla link olsa da var olarak kabul edilir ve 1 değerini alır.

Emoji: Mesajdaki gülme “:)” üzülme “:(“ gibi emoji diye adlandırılan duygusal ifade sayısının toplam mesaj karakter sayısına oranı. 0 -1 arasında bir değer alır.

SMW: Spam mesaj ağırlık değeri. Daha önce oluşturulmuş Spam mesajlarda sık geçen kelimelerin puanlanmasıyla oluşturulan öznitelik.

Mesajın yapısal özelliklerine göre mesaja eklenen anahtar kelimeler:

- **ML** -> MiniMessage, MaxiMessage
- **CWR**->MiniUpperCase, MidiUpperCase, MaxiUpperCase
- **Url**-> HasUrl, NoUrl
- **Emoji**-> HasEmoji, NoEmoji
- **SMW**-> MiniSpam, MaxiSpam

```
if (SMW < 100) { smsText += " MiniSpm"; }
else { smsText += " MaxiSpm MaxiSpm";}

if (ML < 140) { smsText += " MiniMessage MiniMessage"; }
else { smsText += " MaxiMesssage"; }

if (CWR < Convert.ToDecimal("0,19")){ smsText += " MiniUpperCase"; }
else if (CWR < Convert.ToDecimal("0,6")) { smsText += " MidiUpperCase"; }
else { smsText += " MaxiUpperCase"; }

if (Url > 0) { smsText += " HasUrl"; }
else { smsText += " NoUrl"; }

if (Emoji > 0) { smsText += " HasEmoji"; }
else { smsText += " NoEmoji"; }
```

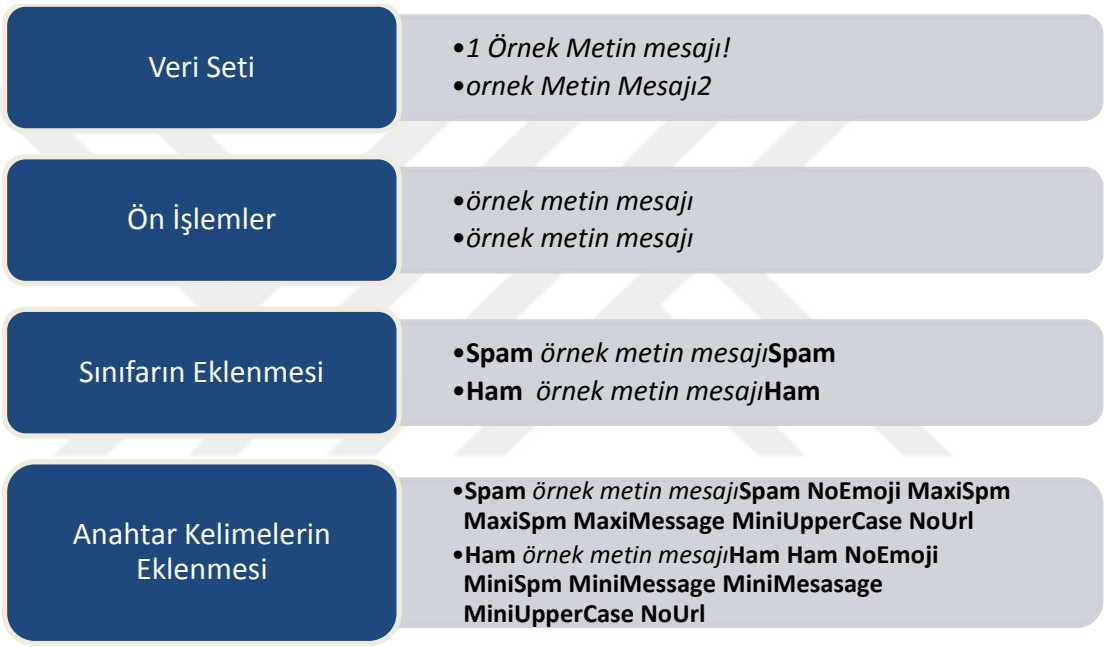
Şekil 4.2. Mesaja anahtar kelimeleri eklemek için kullanılan kod bloğu

4.1.3. Mesajın modele göre düzenlenmesi (Ön Hazırlık)

Mesajın yapısına ait (daha önce belirlenen) özellikleri buradaki veri setine birer anahtar kelime olarak ekleyerek Word2Vec modelinden alınan özniteliklerin

hassasiyeti artırılmaktadır. Bu işlemde kullanılan SMS özellikleri: Spam Mesaj Ağırlığı, Mesaj Uzunluğu, Büyük Harf Frekansı ve URL Durumu 'dur.

Word2Vec model oluşturmak için veri setinde Şekil 4.3'deki adımlarla değişlikler yapılmıştır. Veri setinden öncellikle mesajlardaki rakamlar, noktalama işaretleri ve simgeler çıkarılmıştır. Tüm karakterler küçük harfe çevrilmiş, URL'ler mesajdan çıkarılmıştır. Daha sonra mesajın sınıfı, kelimelerle güçlü bir bağlantı oluşturmak için mesajın başına ve sonuna eklenmiştir. Mesajın yapısal özelliklerini belirten anahtar kelimeler ise mesajın sonuna eklenmiştir (Ballı ve Karasoy, 2019). Ek-A'da, yapılan önişlemlere ait fonksiyonların C# dilinde yazılmış kodları verilmiştir.



Şekil 4.3. Mesaja anahtar kelimeleri eklemek için kullanılan kod bloğu

4.1.4. Word2Vec ile model oluşturma

Hazırlanan veri setinden Word2Vec modeli oluşturmak için açık kaynak bir Phyton kütüphanesi olan Gensim (Web-1) kullanılmıştır. Hazırlanan veri seti, Şekil 4.4'teki kod bloğu ile model oluşturulmuştur. Çizelge 4.3'de kullanılan parametrelerin açıklamaları bulunmaktadır (Ballı ve Karasoy, 2019).

```
import gensim
import logging
sentences = gensim.models.word2vec.LineSentence("datasetwithKeyWords.txt", max_sentence_length=10000)
model = gensim.models.Word2Vec(sentences, size=1600, window=15, min_count=3, workers=5)
model.save("ModelMessages.w2v")
```

Şekil 4.4. Word2Vec modeli oluşturmak için kullanılan Python kodu

Çizelge 4.3. Word2Vec Model oluşturma parametreleri

Parametre	Açıklama
-size	Vektör Uzunluğu
-window	Hesaplanan kelimenin maksimum komşu uzaklık değeri
-min_count	min_count değerinde daha az sayıda frekansa sahip kelimeleri kullanma
-workers	İşlemin daha hızlı çalışması için kullanılan thread sayısı
-sg	Model oluşturulurken kullanılacak algoritmanın seçimi. (Default : 0 – CBow)
-max_vocab_size	Maksimum sözlük boyutu (Default :None - Limitsiz)

4.1.5. Word2Vec ile öznitelik çıkarımı

Yapılan ön işlemlerden sonra Word2Vec modeli oluşturulmuştur. Oluşturulan bu model kullanılarak Word2Vec yardımıyla, her mesajın kelimelerinin Spam ve Ham anahtar kelimesine olan uzaklık değerleri hesaplanmıştır. Bu değerler sınıflara göre ayrı ayrı toplanarak 2 yeni öznitelik oluşturulmuştur.

Örnek mesaj:

> "URGENT! You have won a 1 week FREE membership in our £100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18 "

Ön işlemden Sonra:

> "urgent you have won week free membership in our prize jackpot txt the word claim to no t c lccltd pobox ldnw a rw ***NoEmoji MaxiSpm MaxiSpm MaxiMessage MidiUpperCase HasUrl*** "

Çizelge 4.4. Örnek mesaj öznitelik çıkarımı

Kelime	Spam'a Olan Uzaklık	Ham'a Olan Uzaklık
urgent	0.975194246506	0.438388839784
you	0.655218728478	0.733712784782
have	0.879464504206	0.637038282437
won	0.971993393431	0.447945280534
week	0.927804236488	0.609127003671
free	0.995738243173	0.294745526108
membership	0.986576426483	0.30541150787
In	0.659460629677	0.844060527719
our	0.943573380642	0.561459519238
prize	0.992812816078	0.327789914867
Txt	0.975739475582	0.489539918416
the	0.823232509864	0.646893576175
word	0.964968165891	0.529489068039
claim	0.994731596072	0.368184471241
to	0.948806050559	0.442867815647
No	0.749500382947	0.841420898899
pobox	0.907761091268	0.666196932476
ldnw	0.98065959655	0.47467678971
MaxiSpm	0.754575023181	0.628513943787
MaxiSpm	0.754575023181	0.628513943787
MaxiMessage	0.925637560272	0.546333218334
MidiUpperCase	0.888145625486	0.630623345897
HasUrl	0.959197094166	0.524208530277
TOPLAM	20.6153658002	12.6171416397

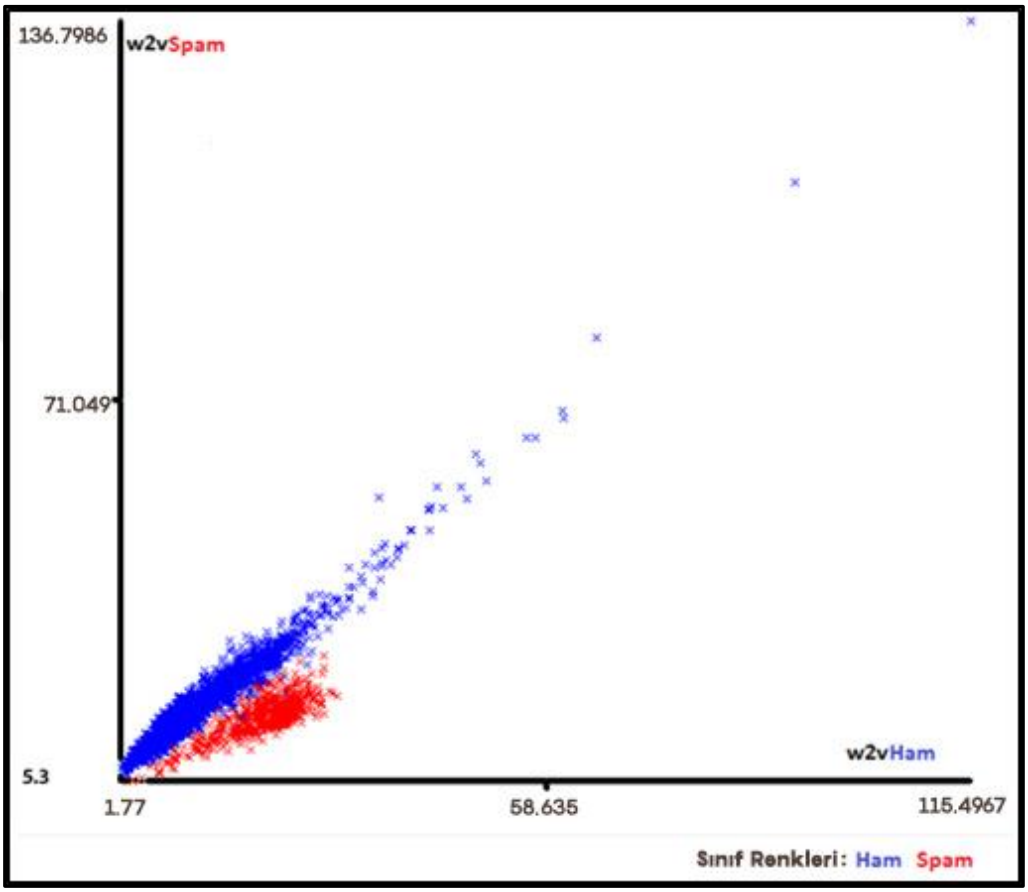
Çizelge 4.4'te seçilen bir örnek mesajın Spam ve Ham anahtar kelimelerine olan uzaklıklarının nasıl hesaplandığı gösterilmiştir. Mesajdaki her bir kelimenin anahtar kelimelere olan uzaklıkları bulunup toplanarak bir özniteliği oluşturulmuştur. Eğer modelde olmayan bir kelime mesajda yer alıyorsa onun anahtar kelimelere olan uzaklığı 0'dır ve hesaba katılmamaktadır. "Jackpot" kelimesi sözlükte olmadığı için "t", "c", "a" gibi harfler ise tek karakter oldukları için işleme alınmamıştır (Ballı ve Karasoy, 2019).

Çizelge 4.5. Sınıflamada kullanılacak veri seti

Mesaj ID	Öznitelik1(Spam Etiketine Uzaklık Değeri)	Öznitelik2 (Ham Etiketine Uzaklık Değeri)	Mesaj Sınıf
1	12.416	19.065	Ham
2	19.295	15.276	Spam
3	22.615	13.213	Spam
4	23.213	25.430	Ham
5	6.549	11.355	Ham
6	18.772	13.564	Spam
7	3.964	10.531	Ham
8	5.955	11.960	Ham
9	5.823	10.573	Ham
10	19.265	14.924	Spam
..	..	..	..

Çizelge 4.5'te veri setinin sınıflamada kullanılmak üzere hazırlanmış son halinden 10 adet örnek kayıt gösterilmektedir.

Şekil 4.5'te w2vSpam ve w2vNormal özniteliklerinin dağılımı gösterilmektedir.

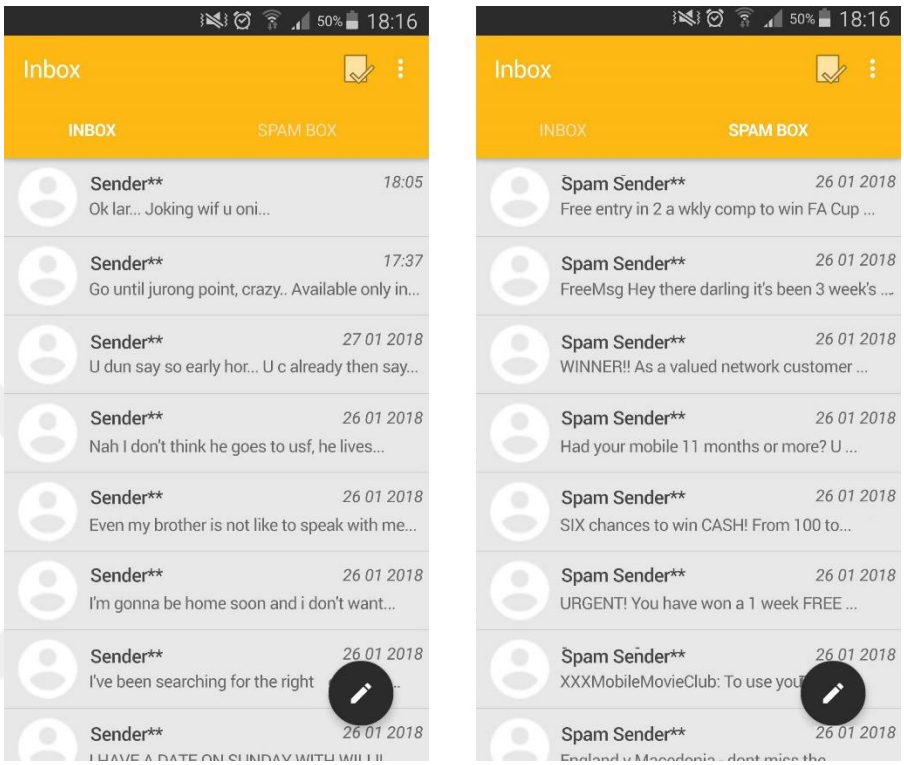


Şekil 4.5. Oluşturulan özniteliklerin dağılımı

4.1.6. Sınıflandırma

Sınıflandırma aşamasında yeni gelen SMS'in yapısal özelliklerine göre anahtar kelimeler belirlenir. Daha sonra bu mesaj bir ön işlemden geçirilir. Mesaj metni küçük harflere çevrilir. Rakamlar, noktalama işaretleri ve URL'ler mesajdan temizlenir ve belirlenen anahtar kelimeler mesaja eklenir. Çalışmada hazırlanan Word2Vec modeli ile mesajın kelimelerinin Spam'a ve Ham'a olan uzaklıkları toplanarak 2 yeni öznitelik bulunur. Bu öznitelikler sınıflandırma algoritması yardımıyla sınıflandırılır. Şekil

4.1’de tasarlanan modele göre sınıflandırılan mesaj Şekil 4.6’da Android işletim sistemi üzerinde geliştirilen uygulamada görüldüğü gibi, eğer Spam ise kullanıcı uyarılmadan mesaj Spam kutusuna gönderilir. Eğer gelen mesaj Ham olarak sınıflandırılırsa kullanıcı uyarılarak gelen kutusuna taşınır (Ballı ve Karasoy, 2019).



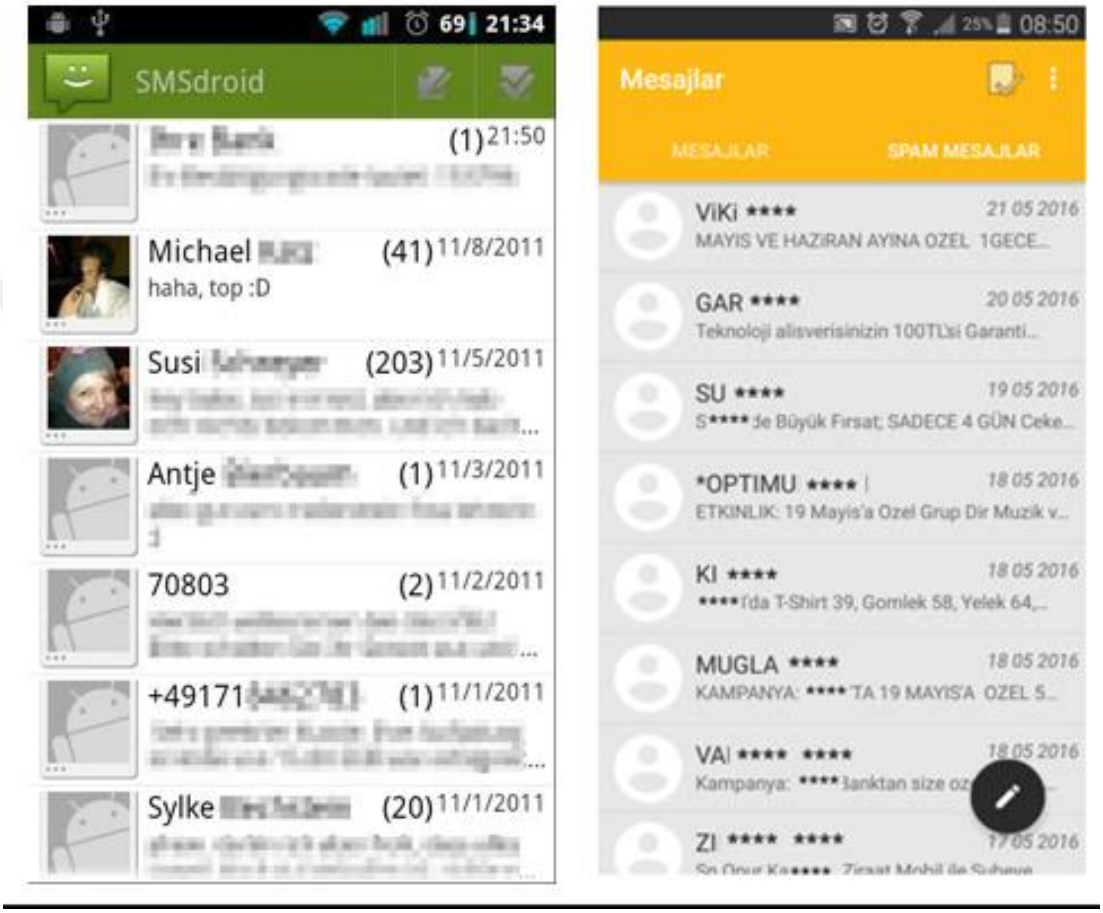
Şekil 4.6. Uygulamanın ekran görüntüsü

4.1.7. Mobil uygulama geliştirilmesi

Yapılan bu çalışmanın amacı, gelen mesaj metinlerinden mesajın normal veya Spam mesaj olup olmadığını belirlemektir. Daha sonra yakalanan Spam mesajları kullanıcıyı rahatsız etmeden Spam mesaj kutusuna gönderebilecek bir mesaj uygulaması tasarlanmıştır. Uygulama iki ayrı gelen mesaj kutusu içerecek şekilde tasarlanmıştır. Ayrıca yapılacak olan uygulama normal bir mesajlaşma uygulamasında bulunan SMS alma, SMS gönderme, MMS alma, MMS gönderme, gelen mesajlarda arama, tüm geçmiş yazışmalara ulaşma, okunmayan mesajları belirtme ve mesaj geldi uyarısı gibi özellikleri de içermesi gerekmektedir. Altyapısı bahsedilen tüm özelliklere sahip olan

açık kaynak kodlu SMSdroid (Web-10) uygulaması kullanılarak hazırlanan BAN adlı yeni bir SMS uygulaması geliştirilmiştir.

Yeni geliştirilen uygulamada gelen kutusu Mesajlar ve Spam Mesajlar başlığında iki sekmeye ayrılmış ve kullanıcı dostu bir ara yüz tasarlanmıştır. Eski uygulama ile yeni geliştirilen uygulamaların ekran görüntüleri Şekil 4.7'de gösterilmiştir.



Şekil 4.7. SMSdroid ve BAN uygulamaları ekran görüntüleri

4.1.8. Bulgular

Yapılan çalışma Word2Vec modelinin oluşumu ve sınıflandırma algoritması olarak iki aşamada gerçekleştirilmiştir. Bu iki aşamada I7 işlemcili ve 8 GB RAM belleğine sahip masaüstü bir bilgisayar kullanılmıştır. Word2Vec modeli, Phyton programlama dilinde (Web-2) ve Gensim kütüphanesi (Web-1) aracılığıyla hazırlanmıştır. Sınıflandırma işlemleri Weka programı (Web-3) kullanılarak gerçekleştirilmiştir.

Çizelge 4.6 ve 4.7'da farklı iki yöntemle çıkarılan özniteliklerin Random Forest, Multi Layer Perceptron, SVM, Logistic Regression ve Naive Bayes algoritmaları ile ulaşılan sonuçlar gösterilmiştir.

Çizelge 4.6. Sınıflandırma algoritmalarının karşılaştırma sonuçları (sadece Word2Vec)

	SPAM			HAM					Toplam
	Precision	*Recall*	*F-Mea.*	*Precision*	*Recall*	*F-Mea.*	*ROC (AUC)*	*RMSE*	*ACC*
RF	0.970	0.966	0.968	0.994	0.995	0.995	**0.999**	**0.0795**	**%99.1029**
MLP	0.983	0.991	0.987	0.999	0.997	0.998	**1.000**	**0.0584**	**%99.6411**
SVM	1.000	0.931	0.964	0.989	1.000	0.994	**0.966**	**0.0978**	**%99.0431**
LR	0.983	0.987	0.985	0.998	0.997	0.998	**1.000**	**0.0563**	**%99.5813**
NB	0.510	0.534	0.522	0.924	0.917	0.921	**0.880**	**0.3004**	**%86.4234**

Çizelge 4.6 'da mesajın yapısal özellikleri (mesaj uzunluğu, URL durumu vb..) göz önüne bulundurulmadan, sadece kelimelerin türlere (Spam, Ham) olan uzaklıkları hesaplanarak bulunan öznitelikler ile ulaşılan sonuçları gösterilmiştir. Çizelge 4.6'da göre MLP (Multi Layer Perceptron) %99.64 ile en iyi doğru sınıflandırma oranına sahiptir.

Çizelge 4.7. Yapısal özellikler ile beraber sınıflandırma algoritmalarının karşılaştırma sonuçları(Yapısal özellikler + Word2Vec)

	SPAM			HAM					Toplam
	Precision	*Recall*	*F-Mea.*	*Precision*	*Recall*	*F-Mea.*	*ROC (AUC)*	*RMSE*	*ACC*
RF	1.000	0.974	0.987	0.996	1.000	0.998	**0.997**	**0.0651**	**%99.6411**
MLP	0.950	0.983	0.966	0.997	0.992	0.994	**1.000**	**0.0731**	**%99.0431**
SVM	0.991	0.974	0.983	0.996	0.999	0.997	**0.986**	**0.0692**	**%99.5215**
LR	0.991	0.970	0.980	0.995	0.999	0.997	**1.000**	**0.0629**	**%99.4617**
NB	1.000	0.664	0.798	0.949	1.000	0.974	**0.958**	**0.2008**	**%95.3349**

Çizelge 4.7'de ise mesajın yapısal özelliklerinin, birer anahtar kelime olarak mesaja eklendikten sonra mesajdaki kelimelerin türlere olan uzaklık değerlerine göre karşılaştırılan sınıflandırma algoritmalarının sonuçları gösterilmiştir. Çizelge 4.7'ye göre RF(Random Forest) %99.64 ile en iyi doğru sınıflandırma oranına sahiptir.

Çizelge 4.6'daki MLP ve Çizelge 4.7'daki RF yöntemlerinin doğru sınıflandırma oranları eşit gibi görünse de Çizelge 4.8 ve Çizelge 4.9'deki karmaşıklık matrisleri incelendiğine Çizelge 4.9'deki RF algoritmasının hiçbir normal mesajı Spam olarak işaretlemediği görünmektedir. Gerçek yaşam deneyimleri göz önünde

bulundurulduğunda bir mesaj filtreleme uygulamasında, False-Pozitif değeri yani hatalı sınıflandırılan normal mesajlar, önemli bir ölçüttür ve dikkat edilmesi gereken bir durumdur. Bu çalışmada False Pozitif değeri düşük olan Word2Vec ve yapısal özelliklerin beraber kullanımı ile Random Forest yöntemi önerilmektedir.

Çizelge 4.8. Sadece Word2Vec özellikleri için karmaşıklık matrisi

	Random Forest		**Multi Layer Perceptron**	
Sınıf	Spam	Ham	Spam	Ham
Spam	224	8	230	2
Ham	7	1433	4	1436

Çizelge 4.9. Word2Vec + Yapısal özellikler için karmaşıklık matrisi

	Random Forest		**Multi Layer Perceptron**	
Sınıf	Spam	Ham	Spam	Ham
Spam	226	6	228	4
Ham	0	1440	12	1428

Çizelge 4.10. SMS Spam Collection v1 ile sınıflandırma yapılan önceki çalışmalar ile karşılaştırma

Yazar	Veri Seti	Tercih Edilen Yöntem	Doğruluk Oranı
Almeida vd.. (2011)	SMS Spam Colleciton v1	SVM	%97.5
Bozan vd.. (2015)	SMS Spam Colleciton v1	SVM	%98.61
Ho vd.. (2013)	SMS Spam Colleciton v1	Graph-based KNN	%98.9
Fernandes vd.. (2015)	SMS Spam Colleciton v1	OPF with Complete	%92.23
Akbari ve Sajedi (2015)	SMS Spam Colleciton v1	GentleBoost	%98.30
Suleiman ve Al-Naymat (2017)	SMS Spam Colleciton v1	Random Forest	%97.7
Nagwani (2017)	SMS Spam Colleciton v1	SVM	93.45 (first-level) - 96.68 (second-level)
Bu çalışmada önerilen model	SMS Spam Colleciton v1	Word2Vec + RandomForest	**%99.64**

Çizelge 4.10'da SMS Spam Collection v1 veri seti ile yapılmış önceki çalışmalar bulunmaktadır. Gösterilen çalışmaların doğru sınıflandırma yüzdeleri %92 ile %98.9 aralığında değiştiği gözlemlenmektedir. Bu çalışmada kullanılan, Word2Vec yardımıyla çıkarılan öznitelikler ile Random Forest Yöntemi %99.64 ile diğer çalışmalardan daha başarılı doğru sınıflandırma yüzdesi elde edilmiştir.

Bu çalışmada elde edilen gelişimleri göstermek ve diğer çalışmalardan farkı ortaya koymak için N-Way varyans testi kullanılmıştır.

Test:

H_0: Bu çalışmada önerilen yaklaşım, diğer yaklaşımlarla aynı doğruluk oranına sahiptir.

H_1: Diğer çalışmalarla farklı doğruluk oranına sahiptir.

0.05 anlamlılık düzeyinde bulunan p değeri < 0.01 olduğu için H_0 hipotezi reddedilir yani bu çalışmada önerilen yaklaşımın doğruluk oranı ile diğer yaklaşımların ulaştığı doğruluk oranları arasında istatistiksel olarak anlamlı bir fark olduğunu göstermektedir.

4.1.9. Tartışma ve Değerlendirme

Bu bölümde, günümüzde önemli bir problem olan Spam SMS'leri engellemek için içerik tabanlı bir filtreleme sistemi önerilmiştir. Önceki çalışmalardan farklı olarak bu çalışmada önerilen modelde SMS mesajlarındaki kelimelerin arasındaki anlamsal bağ kullanılmıştır. Mesajdaki kelimeler Word2Vec kütüphanesi kullanılarak vektörlere dönüştürülmüş ve kelimeler arasındaki uzaklık hesaplanarak aralarında analoji kurulmuştur.

Yapısal özellikler anahtar kelimelere dönüştürülerek mesajlara eklenerek yeni veri seti oluşturulmuştur. Word2Vec modeli bu yeni veri setinden oluşturulmuş ve bu model kullanılarak iki yeni öznitelik ortaya çıkarılmıştır. Yeni öznitelikler ile önceki çalışmalarda yaygın olarak kullanılan sınıflandırma algoritmaları karşılaştırılmıştır.

İngilizce veri setinde yapılan çalışmada Random Forest %99.6411 doğru sınıflama değeri ile en başarılı yöntem olmuştur. Farklı örnekler ve farklı öznitelik seçimlileriyle alt uzaylar oluşturan Random Forest yöntemi, özniteliklerin farklı birlikteliklerini göz önünde bulundurduğu için doğru sınıflandırma oranı yüksek gözlemlenmektedir. Çizelge 4.10'da gösterilen önceki çalışmalar dikkate alındığında önerilen yöntemin diğer çalışmalara göre doğru sınıflandırma yüzdesinin daha yüksek olduğu gözlemlenmektedir.

4.2. Türkçe Veri Seti İçin Spam Tespiti

4.2.1. Veri seti

Spam SMS'ler, çoğu zaman kullanıcıların üye oldukları, alış veriş yaptıkları veya bir şekilde bağlantılı bulundukları kurum ve kuruluşlardan alınmaktadır. Bu sebeple her kullanıcı benzer Spam SMS'lere maruz kalmaktadır. Doğru tetkik ve analiz için veri seti oluşturulurken farklı yaş grupları ve yaşam alanlarından Türkçe SMS toplanmıştır. Oluşturulan yeni veri setinde 2536 Spam, 2215 Normal toplam 4751 adet SMS bulunmaktadır.

4.2.2. Verilerin hazırlanışı

İngilizce veri setinde olduğu gibi Türkçe veri setinde de yapısal özellikler dikkate alınarak öznitelikler oluşturulmuştur. Mesajların uzunluğu URL içerip içermediği büyük harf kullanım oranı ve duygusal ifadeler ayırt edici özellikler olarak seçilmiştir.

Çizelge 4.11. Spam mesajlarda sık rastlanan kelime örnekleri

Sınıf	Sık kullanılan kelimeler
Spam	Yaz,indir, kampanya, özel, fırsat, http, alışveriş, bonus, gönderi, mersis, iptal, taksit, üzeri, hediye..

Türkçe veri setinde de Spam mesajlarda sık geçen 60 kelime seçilerek frekanslarına göre 1 den 60'e kadar puanlar verilmiştir. Çizelge 4.11'de sık geçen kelime örnekleri gösterilmiştir.

Mesajlarda bulunan Türkçe karakterler normalden daha fazla alan kapladığı için genellikle tercih edilmemektedir. Örneğin "Alışveriş" kelimesi genellikle "alisveris" şeklinde bulunabilmektedir. Bu durum anahtar kelime bulma ve mesaj ağırlık hesaplamada hassasiyeti olumsuz yönde etkilemektedir. Anahtar kelimeler belirlenirken mesajlarla ilgili Türkçe karakter problemi ve kelimelerin farklı çekim ekleri almış hallerini ayırt edebilmek için Zemberek (Web-7) adlı doğal dil işleme kütüphanesini kullanılmıştır.

Sık geçen kelimelerin mesajlarda bulunma durumuna göre her bir kelimenin puanları toplanarak Spam Ağırlık Değeri oluşturulmuştur. Çizelge 4.12'de mesajların yapısal özelliklerine ait tüm değerlerin örnekleri gösterilmiştir.

Çizelge 4.12. Örnek mesajlar ve özellikleri

Mesaj	ML	Url	CWR	ME	SMW	MTİp
TAKSiTLi NAKiT AVANS ARTIK UCRETSiZ! USTELiK 8013 iLE BiTEN KARTINIZLA 9 TAKSiTE OZEL %2,02 YERiNE %1,49 FAiZLE! SON GUN 28 SUBAT bonus.com.tr/3 BiLGi:4440333	158	1	0.610	0	428	Spam
Kusura bakma ben gec yazmisim kardesim :) minibusteyim ;) okulda gorusuruz :)	77	0	0.015	0.045	58	Normal
Internet ucreti odemeden muzik dinlemek icin DINLE yazip 2222 ye gonderin!	74	0	0.094	0	201	Spam
egitim kantindeyim kutuphaneye dogru geliyorum	46	0	0	0	0	Normal

- ML(Mesaj Uzunluğı) Mesajdaki karakter sayısı
- Url Mesajdaki URL Durumu (1 veya 0)
- CWR(Büyük Harf Frekansı) Mesajdaki büyük harf sayısının toplam mesaj uzunluğuna oranı
- ME Mesajdaki Emoji Durumu(1 veya 0)
- SMW Spam Mesaj Ağırlığı. Mesajda bulunan Spam Mesaj özel kelimelerin ağırlıkları toplamı

Bu yapısal özellikler Word2Vec modeli oluşturulurken kullanılmak üzere mesaj metinlere birer anahtar kelime olarak eklenmiştir. Şekil 4.3'te bu işlemin adımları gösterilmiştir. Belirlenen anahtar kelimeler kısa mesajlara eklenirken Şekil 4.8'deki kod bloğu kullanılmıştır.

```
if (SMW < 100) { smsText += " MiniSpm"; }

else { smsText += " MaxiSpm MaxiSpm";}


if (ML < 140) { smsText += " MiniMessage MiniMessage"; }

else { smsText += " MaxiMesssage"; }


if (CWR < Convert.ToDecimal("0,19")){ smsText += " MiniUpperCase"; }

else if (CWR < Convert.ToDecimal("0,6")) { smsText += " MidiUpperCase"; }

else { smsText += " MaxiUpperCase"; }


if (Url > 0) { smsText += " UrlVar"; }

else { smsText += " UrlYok"; }


if (Emoji > 0) { smsText += " DuyguVar"; }

else { smsText += " DuyguYok"; }
```

Şekil 4.8. Mesaja anahtar kelimeleri eklemek için kullanılan kod bloğu

4.2.3. Kelime köklerini bulma (Stemming)

Kelimelerin köklerinin bulmak için Zemberek isimli doğal dil işleme kütüphanesi kullanılmıştır. Bu kütüphane ile ayrıca mesajlardaki Türkçe karakter kullanılması gerektiği halde kullanılmayan kelimler düzeltilmiştir. Örneğin mesajlarda geçen “buyuk” ve “surpriz” kelimeleri “büyük” ve “sürpriz” olarak düzeltilmiştir.

Yapılan ön işleme adımlarıyla mesajların genel yapısını ve anlamını bozmadan kelime sayısını azaltmaktır veri setini sadeleştirerek kullanımı kolay bir hale getirmektir. Zemberek kütüphanesi için mesaj ön işlemeye ilişkin kullanılan kodlar Ek A’da verilmiştir.

Örnek Mesaj : Süvari'den Torium AVM Mağazasına Özel Kampanya; 3 Gömlek 69,90 TL, 3 Pantolon 129,90 TL, Takım Elbise+ Gömlek+ Kravat 199,90 TL. Size Özel Fırsatı Kaçırmayın.

Önişlemlerden sonra*:* süvari torium avm mağaza özel kampanya gömlek tl pantolon tl takım elbise gömlek kravat tl siz özel fırsat kaçır

4.2.4. Kelimeleri ayırma (Tokenization)

Mesajları kelimelere ayırma (parçalama) işlemi bu aşamada yapılmaktadır. Parçalama işlemi için keras.preprocessing.text kütüphanesi kullanılmıştır. Bu kütüphane sayesinde öncelikle metin içinde geçen tüm kelimelerden geçme sıklığına göre sözlük oluşturulmuştur. Toplam 5644 adet benzersiz kelime bulunmuştur. Şekil 4.9'da oluşturulan kütüphaneden örnekler gösterilmektedir.

{'tl': 1, 'için': 2, 've': 3, 'yaz': 4, 'indir': 5, 'sms': 6, 'gönder': 7, 'kampanya': 8, 'fırsat': 9, 'özel': 10, 'ol': 11, 'son': 12, 'al': 13, 'gün': 14, 'ücret': 15, 'alışveriş: 16, 'iste': 17, 'siz': 18…}

Şekil 4.9. Oluşturulan sözlük ve indeks değerleri

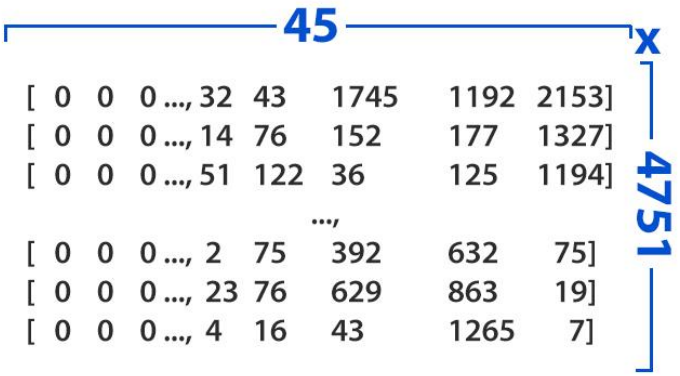
Daha sonra her bir mesaj içinde bulunan kelimeler indeks değerlerine göre işaretlenmiş ve mesaj vektörleri oluşturulmuştur. Şekil 4.10 'da mesaj metinlerindeki kelimelerin indeks değerlerinden oluşan bir vektör örneği gösterilmektedir.

Mesaj Metni: hotic yılbaşı özel net varan indir zaman varan hariç sms al için hotic yaz gönder mersis

Indeks Değerleri : [380, 211, 10, 108, 33, 5, 169, 33, 378, 6, 13, 2, 380, 4, 7, 21]

Şekil 4.10. Mesajlardaki kelimelerin indeks değerlerinden oluşturulan vektör örneği

Bulunan mesaj vektörleri veri setinde bulunan en uzun vektör boyutu ile eşitlenmiştir. Veri setindeki en uzun mesaj 45 kelimelik bir mesajdır. Bu nedenle tüm mesaj vektörleri eksik hanelere 0 atanarak 45 haneli vektörlere dönüştürülmüştür. Ve sonuç olarak 4751x45 boyutunda bir matris elde edilmiştir. Şekil 4.11'de oluşan matrisin genel görüntüsü verilmiştir.



Şekil 4.11. Mesajlardaki kelimelerin indeks değerlerinden oluşan matris.

4.2.5. Kelime temsilleri (Word Embeddings)

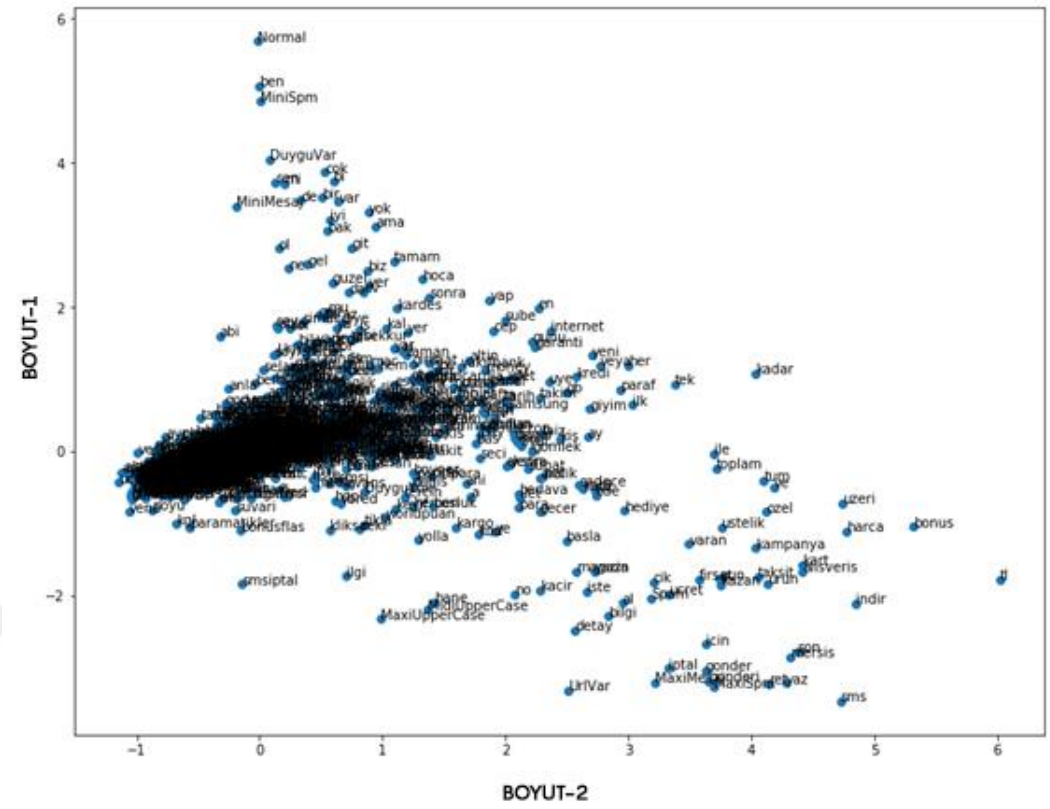
Kelime temsili, her bir kelimenin sayısal olarak temsili için birer vektöre dönüştürülme işlemidir. Bu çalışmada kelime temsillerini hesaplamak için Word2Vec Kütüphanesi kullanılmıştır. Bu kütüphane sayesinde oluşturulan model her bir kelimeyi belirlenen boyutlarda vektörlere çevrilir. Bu çalışmada boyut 300 olarak ayarlanmıştır. Oluşturulan bu vektörler CNN ağı oluşturulurken ön eğitimli (pre-trained) vektörler olarak, ağırlık (weight) özelliğine değer atamak için kullanılmaktadır. Şekil 4.12'de WordVec ile oluşturulan modele ait kelime bulutu gösterilmektedir. 2 boyutlu temel bileşenler analizi (PCA-Principal Component Analysis) görselleştirilmesi kullanılarak kelime konumları gösterilmiştir.

Örneğin fırsat ve tl kelimeleri Word2Vec modelinde;

fırsat, 0.031855 -0.121239 -0.011506 0.053503 0.177177 -0.133212 0.156891 0.078038 -0.076139 -0.215120 0.386383 -…- 0.000126 -0.184152 -0.017607 0.031077 -0.009937 0.109093 0.340587 0.163200 -0.064061

tl, 0.097614 -0.191410 -0.077347 -0.001992 0.318792 -0.133253 0.098148 0.295435 0.001103 -0.274218 0.545888 -0.216871 -…- 0.111804 0.054667 0.390510 0.066539 -0.322426 0.095213 0.112130 -0.287597 0.068989 0.426624 0.247970 -0.069505

şeklinde temsil edilirler.



Şekil 4.12. Word2Vec modelindeki kelime bulutu

4.2.6. Word2Vec ile öznitelik çıkarımı

Oluşturulan Word2Vec model kullanılarak her mesajın kelimelerinin Spam ve Normal anahtar kelimesine olan uzaklık değerleri hesaplanmıştır. Bu değerler sınıflara göre ayrı ayrı toplanarak 2 yeni öznitelik oluşturulmuştur.

WTVSpam: Her bir kelimenin Spam'a olan uzaklıklarının toplamı

WTVNormal: Her bir kelimenin Normal'e olan uzaklıkları

4.2.7. Sınıflandırma

Sınıflandırma aşamasında gelen SMS mesaj uzunluk, büyük harf frekansı, URL barındırma durumu, duygusal ifade barındırma durumu gibi özellikler belirlenir. Daha sonra rakamlar, noktalama işaretleri ve URL'ler mesajdan çıkarılır. Doğal dil işleme yöntemleri ile kelimeler köklerine ayrılır ve Türkçe karakter kullanılmamasından kaynaklı hatalar düzeltilir. Bu çalışmada hazırlanmış olan Word2Vec modeli

kullanılıp mesaj içerisindeki her bir kelimenin Spam kelimesine olan uzaklıkları toplanarak hesaplanan Spam mesaj ağırlığı değeri bulunur. Bulunan bu öznitelikler oluşturulan derin sinir ağ yöntemi ile sınıflandırılır.

Şekil 4.1'de tasarlanan modele göre sınıflandırılan mesaj, Android işletim sistemi üzerinde geliştirilen uygulamada görüldüğü gibi, eğer Spam ise kullanıcı uyarılmadan mesaj Spam kutusuna gönderilir. Eğer gelen mesaj Normal olarak sınıflandırılırsa kullanıcı uyarılarak gelen kutusuna taşınır.

Çizelge 4.13. Çıkarılan tüm öznitelikler

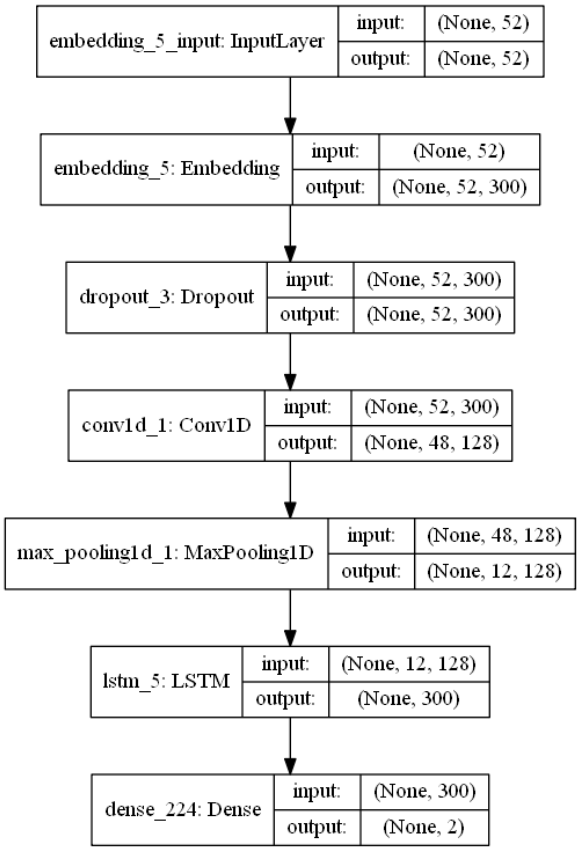
Kelime Indeks Matrisi					**Yapısal Özellikler**					**Word2Vec ile üretilen öznitelikler**		**Sınıf**
t1	**t2**	**t-**	**t44**	**t45**	**ML**	**URL**	**CWR**	**SMW**	**ME**	**WTV Spam**	**WTV Normal**	**MSinif**
0	0	..	4	5	3	0	0.000	0	0.333	1.025	5.247	2
0	0	..	4	5	1	0	0.000	0	0.000	1.536	4.643	2
0	0	..	4	5	3	0	0.000	0	0.000	1.536	4.643	2
146	2813	..	4	5	43	0	0.026	0	0.000	2.869	7.224	2
0	0	..	4	5	20	0	0.167	0	0.000	2.372	6.455	2
0	0	..	4	5	27	0	0.043	0	0.000	2.612	9.203	2
1086	757	..	4	5	93	0	0.013	0	0.000	5.038	14.554	2

Şekil 4.13'te oluşturulan modelin python kodu verilmiştir. Oluşturulan model bir Sequantial (ardışık) modeldir. İlk katman, oluşturulan sözlükte bulunan kelime sayısı kadar büyüklüğe sahip bir Embeding katmanından oluşur. Veri setinde toplam 5646 adet farklı kelime bulunmaktadır.Bu nedenle embedding input_dim değeri 5646'dır. Giriş katmanından gelen değere göre input_length 52 olarak ayarlanmıştır. Daha sonra aşırı öğrenmenin önüne geçmek için nöron değerlerinde 0.2 oranında değişikliğe gidilmiştir. Dropout (seyreltme) işlemi tamamlanınca tek boyutlu konvolüsyon katmanı oluşturulmuştur.Konvolüsyon katmannında girişlere filtre 5li filtre uygulanır ve veriler Relu aktivasyon fonksiyonuna tabi tutulur. Konvolüsyon katmanıyla çoğalan öznitelikler MaxPooling1D katmanı ile basitleştirilmiştir. Veri artırmak adına LSTM katmanıyla model desteklenmektedir. Son olarak Ham ve Spam olmak üzere iki farklı durum olduğu için son çıkış katmanı 2 nörondan oluşmaktadır ve sigmoid aktivasyon fonkisyonu ile çıkışlar 0 ve 1 e yakınlaştırılır.

```
def create_CNN_model():
    model_conv = Sequential()
    model_conv.add(Embedding(5646, 300, input_length=52))
    model_conv.add(Dropout(0.2))
    model_conv.add(Conv1D(128, 5, activation='relu'))
    model_conv.add(MaxPooling1D(pool_size=4))
    model_conv.add(LSTM(300))
    model_conv.add(Dense(2, activation='sigmoid'))
    model_conv.compile(loss='binary_crossentropy', optimizer='adam',  metrics=['accuracy'])
    plot_model(model_conv, to_file='CNNmodel_plot.png', show_shapes=True,
show_layer_names=True)
    return model_conv
```

Şekil 4.13. CNN modeli Python kodu

Şekil 4.14'te Şekil 4.13'teki kodun çıktısı olarak oluşturulan modelinin giriş, çıkış ve gizli katmanları gösterilmiştir.



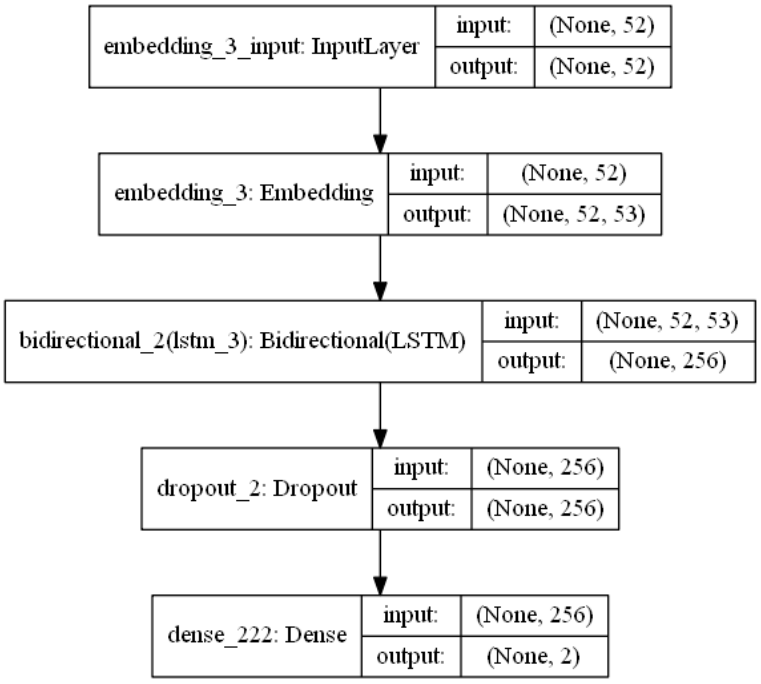
Şekil 4.14. Oluşturulan CNN modeli

Şekil 4.15'de oluşturulan LSTM ağının python kodları gösterilmektedir. CNN modeli ile benzerlikler taşıyan LSTM modeli yine embedding katmanıyla başlar ve bidirectional katmanı ile dizinin tersine sıralamasını da hesaba katarak LSTM katmanı oluşturulur. Ve aşırı öğrenmenin önüne geçmek için bir Dropout işlemiyle sınıflama işlemi gerçekleştirilir.

```
def create_LSTM_model():
    modelstm = Sequential()
    modelstm.add(Embedding(5646, 53, input_length=52))
    modelstm.add(Bidirectional(LSTM(128)))
    modelstm.add(Dropout(0.2))
    modelstm.add(Dense(2, activation='sigmoid'))
    modelstm.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    plot_model(modelstm, to_file='LSTMmodel_plot.png', show_shapes=True,
show_layer_names=True)
    return modelstm
```

Şekil 4.15. LSTM modeli Python kodu

Şekil 4.16'da Şekil 4.15'te çalıştırılan kodun çıktısı olan LSTM ile hazırlanılan model grafiği gösterilmiştir.



Şekil 4.16. Oluşturulan LSTM modeli

4.2.8. Bulgular

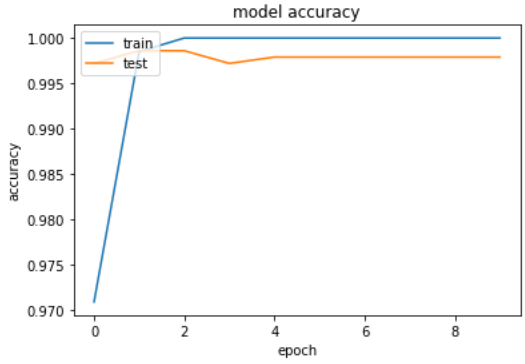
Yapılan çalışma Word2Vec modelinin oluşumu, kelimelerin köklerine ayrılması kelime temsil vektörlerinin oluşturulması, mesaj yapısal özelliklerinin belirlenmesi ve bu öznitelikler kullanılarak yapay sinir ağın oluşturulup mesajın sınıflandırılması gibi aşamalardan oluşmuştur. Bu aşamalarda I7 işlemcili ve 8 GB RAM belleğine sahip masaüstü bir bilgisayar kullanılmıştır. Word2Vec modeli, Phyton programlama dilinde (Web-2) ve Gensim kütüphanesi (Web-1) aracılığıyla hazırlanmıştır. Sınıflandırma işlemleri Weka programı (Web-3) kullanılarak gerçekleştirilmiştir. Derin öğrenme ile sınıflandırma işlemleri Keras (Web-8) kütüphanesi kullanılarak yapılmıştır.

Yapılan önişlemlerden sonra Çizelge 4.13'de gösterilen 52 öznitelik değerinden oluşan bir veri seti elde edilmiştir. Bu öznitelikler farklı kombinasyonlarda seçilerek farklı sınıflama yöntemleri ile test edilmiştir. Hazırlanan modellerde oluşturulan veri setinin %65'i eğitim %35 'i test verisi olarak kullanılmıştır. Bu oranlar aşırı öğrenme(overfitting) ve eksik öğrenme(unerfitting) durumları göz önünde bulundurularak tercih edilmiştir. Çizelge 4.14'te bu testlerin sonuçları gösterilmiştir.

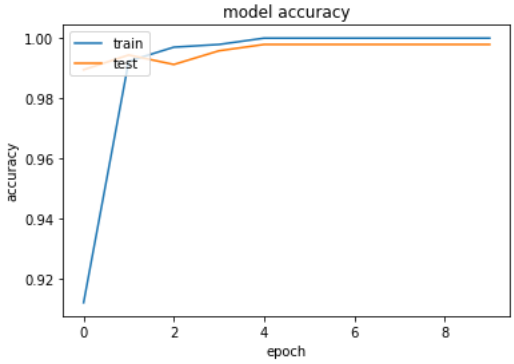
Çizelge 4.14. Sınıflandırma algoritmalarının belirlenen özniteliklere göre doğruluk oranları

Yöntem	Yapısal Öznitelikler	Tokens	Yapısal + Tokens	Yapısal + Token +W2VFeatures
	Doğru Sınıflama Oranı (Accuracy)			
Random Forest	99.57	95.36	99.7	99.32
NaiveBayes Multinomial	91.64	87.13	90.8	90.98
SVM	99.27	89.18	99.28	98.95
Multilayer Perceptron	99.52	89.96	99.4	99.321
Random SubSpace	99.76	94.41	99.51	99.1358
Logistic Regression	99.64	85.74	98.8	99.07
k-NN	99.34	87.79	98.31	96.48
LSTM	99.16	99.51	99.79	99.72
CNN	99.09	99.51	99.79	**99.86**
CNN+Word2Vec	-	97.77	-	-

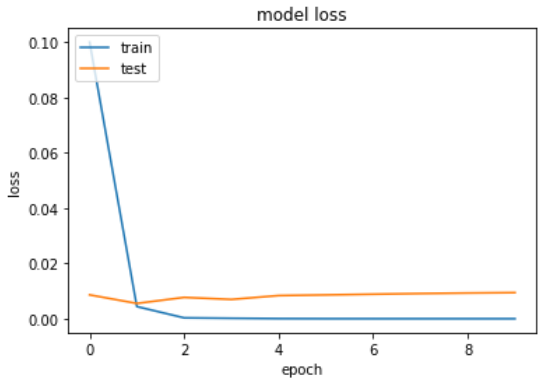
Çizelge 4.14'te bulunan CNN+Word2Vec ile yapılan denemede kelimelere karşılık gelen Word2Vec vektörleri, ağın yapısına ağırlık değerleri olarak atandığından sadece token öznitelikleri ile kullanılmıştır.



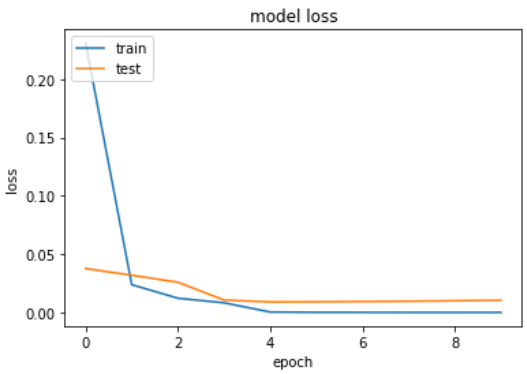
Şekil 4.17. CNN Accuracy değeri grafiği



Şekil 4.19. LSTM Accuracy değeri grafiği



Şekil 4.18. CNN Loss değeri grafiği



Şekil 4.20. LSTM Loss değeri grafiği

Şekil 4.17, 4.18, 4.19 ve şekil 4.20'de en yüksek doğru sınıflama yüzdesi değerine sahip olan iki yöntemin(CNN ve LSTM) Accuracy (doğru sınıflama oranı) ve Loss (tahmin edilen değerin gerçek değerden ne kadar faklı olduğunu gösterir) değerlerinin grafikleri gösterilmiştir. Şekil 4.17'de oluşturulan CNN ağı ile her bir epoch (eğitim turu) değerinde doğru sınıflama yüzdeleri ve Şekil 4.18'de hatalı tahmin değerleri gösterilmiştir. Epoch değer arttıkça doğruluk değerinin 1'e yaklaştığı ve loss değerinin ise 0'a yaklaştığı gözlemlenmektedir.

4.2.9. Tartışma ve Değerlendirme

Yeni oluşturulan Türkçe veri seti ile uygulama yapılan bu bölümde, öncelikle öznitelik çıkarım işlemi yapılmıştır. Mesajların uzunluk, büyük harf frekansı, duygusal ifade ve URL durumu gibi 5 adet yapısal öznitelik bulunmuştur. Daha sonra kelimelerin analojik yakınlıklarından yola çıkarak Word2Vec kütüphanesi ile 2 yeni öznitelik daha üretilmiştir. Ayrıca mesaj da geçen kelimelerden oluşturulan sözlük aracılığı ile her bir mesaj da bulunan kelimelerin indeks değerleri ile 45 (bir mesaj en fazla 45 kelimeden oluşmaktadır) yeni öznitelik daha üretilmiştir. Toplamda 52 öznitelik farklı varyasyonlarla farklı sınıflandırma algoritmalarında denenmiştir. Çizelge 4.14'te deneme sonuçlarında ulaşılan doğru sınıflama yüzdeleri gösterilmektedir. Gözlemlenen sonuçlar doğrultusunda oluşturulan tüm öznitelikler kullanılarak kurulan CNN ağ ile %99.86 gibi yüksek bir doğru sınıflama oranına ulaşılmıştır. Birinci uygulamada daha iyi sonuçların alınmasına vesile olan Word2Vec ile çıkarılan özniteliklerin, bu uygulamada daha iyi sonuç alınmasını sağlamadığı görülmüştür.

5. SONUÇ VE ÖNERİLER

Yapılan bu çalışmada, SMS sınıflama problemi ve Spam tespiti için çözüm önerisi sunulmuştur. 2 ayrı veri setinde yapılan denemeler ve bu denemelerin sonuçlarında bulunan değerler tartışılmıştır. Farklı diller için farklı modeller oluşturulduğundan bu problemler ayrı ayrı ele alınmış ve farklı çözümler üretilmiştir.

Öncelikle üzerinde birçok çalışmanın da yapıldığı Spam SMS Collection veri seti kullanılmış ve önerilen sistemde Word2Vec ile çıkarılan iki yeni öznitelik yapısal özniteliklere eklenerek öznitelik sayısı artırılmıştır. Bu öznitelikler kullanılarak Random Forest yöntemi ile sınıflandırılan veri setinde %99.64 oranında doğru sınıflama değerine ulaşılmış ve yapılan önceki çalışmalarda daha iyi bir sonuç olduğu gözlemlenmiştir. Word2Vec ile bulunan yeni özniteliklerin başarımı arttırdığı görülmüştür.

İçerik tabanlı metin sınıflama problemlerinde metin yapısının yanı sıra metin dilleri de önemli bir değişkendir. Bir dilde oluşturulan model diğer dillerde aynı başarıyı gösterememektedir. Henüz ulaşabilir olan Türkçe bir veri seti olmadığından bu çalışmada kullanılmak üzere yeni bir Türkçe SMS veri seti oluşturulmuştur.

İkinci çalışma olarak Türkçe SMS'lerden oluşturulan veri seti ele alınmıştır. Yeni oluşturulan bu veri setinde önceki çalışmalarda kullanılan yöntemlere ek olarak derin öğrenme teknikleri ile sınıflandırma işlemi gerçekleştirilmiştir. Farklı öznitelik çıkarım teknikleri denenmiş ve sonuçlar bu özniteliklerin kullanımlarına göre tartışılmıştır. Yapılan bu denemeler sonunda mesajların tüm özelliklerinden oluşan 52 adet öznitelik ile oluşturulan CNN ağı kullanılarak %99.86 gibi yüksek bir doğru sınıflandırma başarı oranına ulaşılmıştır. Word2Vec ile bulunan yeni öznitelikler bu veri seti için başarımı arttırmamıştır.

Bu çalışmalar sonucunda, sadece kısa mesaj metni ele alınarak yapılan sınıflandırma işleminin oldukça başarılı olduğu gözlemlenmiş bu deneyler doğrultusunda elde edilen

modeller kullanılarak mobil uygulama geliştirilmiştir. Önerilen sistem hâlihazırda bulunan SMS uygulamasına bağlı olarak arka planda çalıştırılabilmektedir.

İlerleyen çalışmalarda, geliştirilen bu yöntem Whatsapp, Messenger, Instagram, Viber vb. online mesajlaşma uygulamalarında da kullanılabilirliği incelenebilir ve her veri seti için öğrenme işleminden sonra model yeniden yaratılıp doğru sınıflama başarısı artırılabilir. Ayrıca SMS'den daha büyük metinlere sahip e-mail mesajları için yine bu çalışmada geliştirilen model rahatlıkla uygulanabilir.

6. KAYNAKÇA

Ahmed,I., Guan, D. ve Chung, T. (2014) Sms classification based on naive bayes classifier and apriori algorithm frequent itemset, *International Journal of Machine Learning and Computing*, 4:183–187.

Akba, F. (2014) *Duygu analizinde öznitelik seçme metriklerinin değerlendirilmesi: Türkçe film eleştirileri*, Yüksek Lisans Tezi, Hacettepe Üniversitesi, Ankara 76s.

Akbari F. ve Sajedi H. (2015) SMS spam detection using selected text features and Boosting Classifiers, *2015 7th Conference on Information and Knowledge Technology (IKT)*, Urmia, 2015, 1-5.

Ali S. S. ve Maqsood, J. (2018) .Net library for SMS spam detection using machine learning: A cross platform solution, *15th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, Islamabad, 470-476.

Almeida, T. A., Gómez Hidalgo, J. M., ve Yamakami, A. (2011) Contributions to the Study of SMS Spam Filtering: New Collection and Results, *11th ACM Symposium on document engineering*, 19-22 September 2011, 259-262.

Androulidakis, I., Vlachos, V. ve Papanikolaou, A. (2012) Spam goes mobile: Filtering unsolicited SMS traffic, *20th Telecommunications Forum*, Belgrade, Serbi 1452 – 1455.

Atalay M. ve Celik E. (2017) Büyük veri analizinde yapay zekâ ve makine öğrenmesi uygulamaları, *Mehmet Akif Ersoy University Journal of Social Science Institute*, 9(22): 155-172

Atasoy, D. (2001) *Lojistik regresyon analizinin incelenmesi ve bir uygulaması*, Yüksek Lisans Tezi, Cumhuriyet Üniversitesi, Sivas 73s.

Ay Karakuş (2018) *Derin Öğrenme Ve Büyük Veri Yaklaşımları İle Metin Analizi*, Doktora Tezi, Fırat Üniversitesi, Elazığ, 248s

Ballı S. ve Karasoy O. (2019) Development of content-based SMS classification application by using Word2Vec-based feature extraction, *IET Software,* DOI: 10.1049/iet-sen.2018.5046, Baskıda

Ballı, S. ve Sağbas, E.A. (2017) The usage of statistical learning methods on wearable devices and a case study: activity recognition on smartwatches, advances in statistical methodologies and their application to real problems, Hokimoto T. (editör), *Advances in Statistical Methodologies and Their Application to Real Problems*, Intech, Rijeka.

Belem, D. ve Duarte-Figueiredo, F.(2011) Content filtering for SMS systems based on Bayesian classifier and word grouping, *Network Operations and Management Symposium* 10-11 Oct. 2011, Quito, Ecuador 1-7.

Ben-Hur, A., Horn, D., Siegelmann, H., ve Vapnik, V. N. (2001) Support vector clustering, *Journal of Machine Learning Research*, 2: 125–137.

Binokay H. (2018) *Lojistik Regresyon ve Farklı Sınıflama Modellerinin Performanslarının Karşılaştırılması* Yüksek Lisans Tezi, Çukurova Üniversitesi Biyoistatistik Anabilim Dalı, Adana, 149s.

Blake, C. (2011) Text mining. *Information Science and Technology*, 45: 121-155.

Bozan Y. S., Çoban Ö., Özyer G. T. ve Özyer B. (2015) SMS spam filtering based on text classification and expert system, *23nd Signal Processing and Communications Applications Conference (SIU)*,16-19 Mayıs 2015, Malatya, Türkiye, 2345-2348.

Braga, P. L., Oliveira, A. L., Ribeiro, G. H., & Meira, S. R. (2007) Bagging predictors for estimation of software project effort. In *2007 International Joint Conference on Neural Networks*,12-17 August 2007, Orlando, FL, USA pp. 1595-1600.

Breiman, L. (2001) Random forests. *Machine learning*, 45(1): 5-32.

Cai, J., Tang, Y., ve Hu, R. (2008) Spam filter for short messages using winnow, *international conference on advanced language processing and web information technology*, Dalian Liaoning,454–459.

Chan, P.P.K., Yang, C., Yeung, D. ve Wing, W.Y.N. (2014) Spam filtering for short messages in adversarial environment, *Neurocomputing* 155: 167–176.

Chen, L., Yan, Z., Zhang W. ve Kantola, R. (2014) TruSMS: A trustworthy SMS spam control system based on trust management, *Future Generation Computer Systems* 49: 77–93.

Cormack, G. V., Hidalgo, J. M. G., ve Sánz, E. P. (2007) Spam filtering for short messages, *16th ACM conference on information and knowledge management (CIKM'07)*, 6-10 November 2007, Lisbon, Portugal, 313–320.

Coşgun E., Karabulut E., ve Karaağaoğlu E. (2009) Random forest ve destek vektör makinası yöntemleri ile gen seçimi ve sınıflaması, *VI. Ulusal İstatistik Kongresi*, Mayıs 2009, Antalya.

Delany S.J., Buckley M. ve Greene D. (2012) Sms spam filtering: Methods and data, *Expert Systems with Applications*, 39:9899–9908.

Delvia Arifin , Shaufiah ve Bijaksana M. A. (2016) Enhancing spam detection on mobile phone Short Message Service (SMS) performance using FP-growth and Naive Bayes Classifier, *2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, 13-15 Sept. 2016, Bandung, Indonesia, 80-84.

Deng, W.-W., ve Peng, H. (2006) Research on a Naive Bayesian Based Short Message Filtering System, *International conference on machine learning and cybernetics*,13-16 August 2006, Dalian, China, 1233–1237.

Dixit, S., Gupta, S., ve Ravishankar, C. V. (2005) LOHIT: An online detection & control system for cellular SMS spam, *International conference on communication, network and information security (IASTED)*, Phonex, AZ, USA, 48–54.

Emhan Ö. (2017) *Yukarı – Aşağı İmleç Hareketine İlişkin Eeg Kayıtlarının Ayrık Dalgacık, Knn Ve Dvm İle Sınıflandırılması,* Yüksek Lisans Tezi, Dicle Üniversitesi Fen Bilimleri Enstitüsü, Diyarbakır, 96s.

Eren Ö. (2008) *Alerjen proteinlerin otomatik olarak sınıflandırılması*, Yüksek Lisans Tezi, Başkent Üniversitesi, Ankara 92s.

Ergün, K. (2012) *Metin Madenciliği yöntemleri ile ürün yorumlarının otomatik değerlendirilmesi*, Doktora, Sakaraya Üniversitesi, Sakarya, 91s.

Fernandes D., D. Costa K. A. P., Almeida T. A. ve Papa J. P., (2015) SMS Spam Filtering Through Optimum-Path Forest-Based Classifiers, *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, Miami, FL, 133-137.

Gündüz H. (2019) *Derin Öğrenme Yöntemleri İle Zaman Serisi Tahmini* Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 142s.

He H., Watson T., Maple C., Mehnen J. ve Tiwari A. (2017) A new semantic attribute deep learning with a linguistic attribute hierarchy for spam detection, *2017 International Joint Conference on Neural Networks (IJCNN)*, 14-19 May 2017, Anchorage, AK, USA, 3862-3869.

Healy, M., Delany, S., ve Zamolotskikh, A. (2004) An assessment of case-based reasoning for short text message classification. *Proceedings of the 15th. Irish Conference on Artificial Intelligence and Cognitive Sciences (AICS'04)*, Castlebar 257–266.

Hidalgo, J. M. G., Bringas, G. C., Sanz, E. P., ve Garcia, F. C. (2006) Content based SMS spam filtering, *ACM symposium on document engineering*, 10-13 October 2006 Amsterdam, The Netherlands, 107–114.

Ho, T.K. (1998) The Random subspace method for constructing decision forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20: 832-844.

Ho T.P., Kang H.S. ve Kim S.R. (2013) Graph-based KNN Algorithm for Spam SMS Detection, *Journal of Universal Computer Science*, 19: 2404-2419.

Hochreiter, S. ve Schmidhuber, J. (1997) Long short-term memory, *Neural computation*, 9(8), 1735–1780.

Hu, X., ve Yan, F. (2010) Sampling of mass SMS filtering algorithm based on frequent time-domain area, *Third International Conference On Knowledge Discovery And Data Mining*, 9-10 Jan. 2010, Phuket, Thailand, 548 –551.

Junaid, M. B. ve Farooq, M. (2011) Using evolutionary learning classifiers to do mobile spam (SMS) filtering, *13th annual conference on Genetic and evolutionary computation, GECCO '11*, 12 – 16 July 2011, Dublin, Ireland, 1795-1802.

Karaca, M.F. (2012) *Metin Madenciliği yöntemi ile haber sitelerindeki köşe yazılarının sınıflandırılması*, Yüksek Lisans, Karabük Üniversitesi, Karabük, 99s.

Karadağ, K. (2013) *ECoG Tabanlı Parmak Hareketlerinin KNN ve DVM Yöntemleri ile Sınıflandırılması* Yüksek Lisans Tezi, Dicle Üniversitesi Fen Bilimleri Enstitüsü, Diyarbakır, 69s.

Karakuş, S. (2018) *Derin Öğrenme Yöntemlerinin Kullanılarak Dijital Deliller Üzerinde Adli Bilişim İncelemesi*, Yüksek Lisans Tezi, Fırat Üniversitesi, Elazığ, 96s.

Karasoy, O ve Ballı S. (2016) İçerik Tabanlı İstenmeyen SMS Filtreleme için Mobil Uygulama Geliştirilmesi ve Sınıflandırma Algoritmalarının Karşılaştırılması, *International Artificial Intelligence and Data Processing Symposium (IDAP'16)*, 13-18 Eylül 2016 47-53.

Khemapatapan C. (2010) Thai-English spam SMS filtering, *Communications 16th Asia-Pacific Conference*, 31 Oct.-3 Nov. 2010, Auckland, New Zealand, 226 – 230.

Kılıç E., Arslan,S.N. ve Guvensan, M. A. (2014) 3-Tier hybrid approach for SMS filtering, *22nd Signal Processing and Communications Applications Conference*, 23-25 April 2014, Trabzon, Turkey, 1950 – 1953.

Kılınç, D., Borandağ, E., Yücalar, F., Tunalı, V., Şimşek, M. ve Özçift, A. (2016) KNN Algoritması ve R Dili ile Metin Madenciliği Kullanılarak Bilimsel Makale Tasnifi, *Marmara Fen Bilimleri Dergisi*, 28 (3): 89-94.

Kın Z. B. (2019) *Türk İşaret Dili Alfabesinin Derin Öğrenme Yöntemi ile Sınıflandırılması*, Yüksek Lisans Tezi, Başkent Üniversitesi, Ankara, 92s.

Kim, S. E. , Jo, J.T. ve Choi, S.H. (2015) SMS Spam Filterinig Using Keyword Frequency Ratio, *International Journal of Security and Its Applications* 9: 329-336.

Kökçü B. N., Köse R. D., Bulut F., Amasyalı M. F. (2014) Kolektif öğrenme algoritmalarıyla çocuklarda obezite hastalığına yakalanma olasılıklarının

hesaplanması, *Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu*, Ekim 2014, İzmir, Türkiye, 200-205.

Lahmadi A, Delosiere L ve Festor O. (2011) Hinky: defending against text-based message spam on smartphones, *2011 IEEE International Conference on Communications*, 5-9 June 2011, Kyoto, Japan, 1–5.

Lee, H. ve Kang, S. (2019) Word Embedding Method of SMS Messages for Spam Message Filtering. *IEEE International Conference on Big Data and Smart Computing (BigComp),* Kyoto, Japan 1-4.

Lee S., Kang P. ve Cho S. (2014) Probabilistic Local Reconstruction for k-NN Regression and Its Application to Virtual Metrology in Semi Conductor Manufacturing, *Neurocomputing*, 131,427–439.

Liu J., Ke H., ve Zhang G. (2010) Real-time sms filtering system based on bm algorithm, *International Conference on Management and Service Science*, 24-26 August 2010, Wuhan, China, 1 – 3.

Liu J.Y., Zhao Y.H., Zhang Z.X. ve Lei, H. (2012) Spam short messages detection via mining social networks. *Jorunal of computer science and technology*, 27,3: 506–514.

Liu, W. ve ve Wang, T. (2010) Index-based online text classification for SMS spam filtering. *Journal of Computers*, 5: 844–851.

Longzhen, D., An, L., ve Longjun, H. (2009) A new spam short message classification, *international workshop on education technology and computer science*, Wuhan, Hubei, 2:168 –171.

Ma, J., Zhang Y., Liu J., Yu K. ve Wang X. (2016) Intelligent SMS Spam Filtering Using Topic Model, *2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, 7-9 September 2016, Ostrawva, Czech Republic, 380-383.

Mathew, K. ve Issac, B. (2011) Intelligent spam classification for mobile text message, *Computer Science and Network Technology* 1:101 -105.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., ve Dean, J. (2013) Distributed representations of words and phrases and their compositionality, *In Advances in neural information processing systems* 3111-3119.

Mohmoud, T. M. ve Mahfouz, A. M. (2012) SMS Spam Filtering Technique Based on Artificial Immune System, *International Journal of Computer Science*, 9: 589-597.

Nagwani N. K. (2017) A Bi-Level Text Classification Approach for SMS Spam Filtering and Identifying Priority Messages, in *The International Arab Journal of Information Technology*, July 2017, 14(4):473-480.

Navaney P., Dubey G. ve Rana A. (2018) SMS Spam Filtering Using Supervised Machine Learning Algorithms, *2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence),* Noida, 43-48.

Nuruzzaman M. T., Changmoo L. ve Deokjai C. (2011) Independent and personal SMS spam filtering, *11th, IEEE International Conference on Computer and Information Technology*, 31 Aug.-2 Sept. Pafos, Cyprus, 429 – 435.

Oğuzlar, A. (2011) *Temel metin madenciliği*, 1.Baskı, Bursa, 156s.

Pervan N. (2019) *Derin Öğrenme Yaklaşımları Kullanarak Türkçe Metinlerden Anlamsal Çıkarım Yapma* Yüksek Lisans Tezi, Ankara Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 79s.

Pilavcılar, İ.F. (2007) *Metin Madenciliği ile metin sınıflandırma*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, İstanbul, 64s.

Polat Ö.E. (2003) *Yapay sinir ağları ve bir işletmede maliyet/üretim miktarı ilişkisinin yapay sinir ağı ile belirlenmesi* Yüksek Lisans Tezi, Sakarya Üniversitesi 77s.

Rafique M.Z., Alrayes N. ve Khan M.K. (2011) Application of evolutionary algorithms in detecting SMS spam at access layer, *13th annual conference on Genetic and evolutionary computation*, 12-16 July 2011, Dublin, Ireland, 1787-1794.

Rafique, M.Z. ve Smıeee, M.A., (2012) Graph-based learning model for detection of SMS spam on smart phones, *8th International Wireless Communications and Mobile Computing Conference*, 27-31 Aug. 2012, Limassol, Cyprus, 1046 – 1051.

Sağbaş E.A ve Ballı S. (2016) Akıllı Telefon Algılayıcıları ve Makine Öğrenmesi Kullanılarak Ulaşım Türü Tespiti. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi* 22(5) : 376-383

Shahi, T. B. ve Yadav, A. (2014) Mobile SMS Spam Filtering for Nepali Text Using Naïve Bayesian and Support Vector Machine, *International Journal of Intelligence Science,* 4: 24-28.

Sohn, D. N., Lee, J. T., ve Rim, H. C. (2009) The contribution of stylistic information to content-based mobile spam filtering, *ACL/AFNLP 2009*, 2-7 August 2009, Singapore, 321–324.

Suleiman D. ve Al-Naymat G. (2017) SMS Spam Detection using H2O Framework, *In Procedia Computer Science,* 113: 154-161.

Sutton, R. S., & Barto, A. G. (1998) *Introduction to reinforcement learning* (Vol. 2, No. 4). Cambridge: MIT press.

Şahin, Ç. (2018) *Bursa İli İnegöl İlçesinde Ortaokul Öğrencilerinin Sayısal Derslerindeki Başarısını Etkileyen Etmenlerin Lojistik Regresyon Yöntemi ile incelenmesi* Yüksek Lisan Tezi Kahraman Maraş Sütçü İmam Üniversitesi Fen Bilimleri Enstitüsü, Kahramanmaraş, 59s.

Şanlı E. (2018) *Yapay Sinir Ağı Kontrollü Otonom Rc Araç Uygulaması* Yüksek Lisans Tezi, İstanbul Gelişim Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 98s.

Tantuğ A. (2016) Metin Sınıflandırma. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 5 (2).

Tarcan, A. ve Çakar, F. (2008) Bilgisayarlı Dil Tanımlamada Dil Bilimsel Yaklaşımlar ve Bir Yazılım Denemesi, *Elektronik Sosyal Bilimler Dergisi,* 7(26):64-70.

Uysal A. K., Gunal S., Ergin S. ve Gunal E. S. (2013) The impact of feature extraction and selection on SMS spam filtering, *Elektronika ir Elektrotechnika*., 19(5):67–72.

Uysal, A.K., Gunal, S., Ergin, S. ve Gunal, E.S. (2012) Detection of SMS spam messages on mobile phones, *20th IEEE Signal Processing and Communications Application*, 18-20 April 2012, Mugla, Turkey, 1 – 4.

Waheeb W., Ghazali R. ve Deris M. M. (2015) Content-based SMS spam filtering based on the Scaled Conjugate Gradient backpropagation algorithm, *12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 15-17 Aug. 2015 Zhangjiajie, China, 675-680.

Wang, C., Zhang, Y., Chen, X., Liu, Z., Shi, L., Chen, G., vd.. (2010) A behavior-based SMS antispam system, *IBM Journal of Research and Development*, 54, 3:1–16.

WEB-1: GENSIM, https://radimrehurek.com/gensim/models/word2vec.html Erişim: 04.03.2019

WEB-2 Phyton, https://www.python.org/ Erişim: 04.03.2019

WEB-3 Weka, https://www.cs.waikato.ac.nz/~ml/weka/ Erişim: 04.03.2019

WEB-4 Word2Vec Tutorial, https://rare-technologies.com/word2vec-tutorial/ Erişim: 04.03.2019

WEB-5 NLP with gensim (word2vec), http://www.samyzaf.com/ML/nlp/nlp.html Erişim:04.03.2019

WEB-6 Introduction to Word2Vec, https://deeplearning4j.org/word2vec Erişim: 04.03.2019

WEB-7 Zemberek-NLP https://github.com/ahmetaa/zemberek-nlp Erişim: 04.03.2019

WEB-8 KERAS https://keras.io/ Erişim: 04.03.2019

WEB-9 Hppy bthdy txt! http://news.bbc.co.uk/2/hi/uk_news/2538083.stm Erişim 04.03.2019

WEB-10 SMSdroid https://github.com/felixb/smsdroid Erişim 04.03.2019

WEB-10 SMSdroid https://github.com/felixb/smsdroid Erişim: 04.03.2019

WEB-11 What Is Natural Language Processing? https://machinelearningmastery.com/natural-language-processing/ Erişim: 04.03.2019

WEB-12 Destek Vektör Makineleri https://veribilimcisi.com/2017/07/19/destek-vektor-makineleri-support-vector-machine/ Erişim: 04.03.2019

WEB-13 Matlab Görüntü İşleme – Konvolüsyon, Prewitt ve Sobel Filtreleri https://erencelik.com/matlab-goruntu-isleme-konvolusyon-prewitt-ve-sobel-filtreleri/ Erişim: 04.03.2019

WEB-14 Android Studio https://developer.android.com/studio Erişim 04.03.2019

WEB-15 Anaconda https://www.anaconda.com/ Erişim: 04.03.2019

Wensen L., Zewen C., Jun W. ve Xiaoyi W. (2016) Short text classification based on Wikipedia and Word2Vec, *2nd IEEE International Conference on Computer and Communications (ICCC)*, 14-17 Oct. 2016, Chengdu, China, 1195-1200.

Wu, N., Wu, M., ve Chen, S. (2008) Real-time monitoring and filtering system for mobile SMS, *3rd IEEE conference on industrial electronics and applications*, 3-5 June 2008, Singapore, 1319–1324.

Xu, Q., Xiang, E.W., Yang, Q., Du, J. ve Zhong, J. (2012) SMS Spam Detection Using Noncontent Features, *IEEE Intelligent Systems*, 27:44-51.

Yadav, K., Kumaraguru, P., Goyal, A., Gupta, A., ve Naik, V. (2011) SMSAssassin: Crowdsourcing driven mobile-based system for SMS spam filtering. *12th Workshop On Mobile Computing Systems And Applications*,1-6.

Yıldırım, M. (2012) *Kandaki Glikoz Ve HbA1c Değerlerinin Avuç İçi Nem Parametreleri Kullanılarak Destek Vektör Makinesi İle Sınıflandırılması*, Yüksek Lisans Tezi, Dumlupınar Üniversitesi Fen Bilimleri Enstitüsü, Kütahya, 61s.

Yoon, J. W., Kim, H., ve Huh, J. H. (2010) Hybrid spam filtering for mobile communication. *Computers & Security*, 29: 446–459.

Zhang D., Xu H., Su Z. ve Xu Y. (2015) Chinese comments sentiment classification based on word2vec and SVM, *Expert Systems with Applications*, 42(4):1857-1863.

Zhang L., Ma J. ve Wang Y.(2013) Content Based Spam Text Classification: An Empirical Comparison between English and Chinese, *5th International Conference on Intelligent Networking and Collaborative Systems*, 9-11 Sept. 2013, Xi'an, China, 69 – 76.

Zhang, H.Y. ve Wang W. (2009) Application of Bayesian Method to Spam SMS Filtering, *international conference on information engineering and computer science*, 19-20 December 2009, Wuhan, China 1 – 3.

Ek. A: Mesaj Önişlemler Fonksiyonları ve Zemberek Kullanımı

```
using net.zemberek.erisim;
using net.zemberek.yapi;
using net.zemberek.tr.yapi;

public string mesaj_temizle(string msg)
{
    string snc = "";
    string fullBook = msg;
    fullBook = Regex.Replace(fullBook, "(http://([\\w+?\\.\\w+])+([a-zA-Z0-
9\\~\\!\\@\\#\\$\\%\\^\\&amp;\\*\\(\\)_\\-
\\=\\+\\\\\\\/\\?\\.\\:\\;\\'\\,]*)?)", " ");//Http:/liler
    fullBook = Regex.Replace(fullBook, "[a-zA-Z0-9\\._-
]+\\.(com|edu|net|org)(\\.[a-zA-Z0-9\\._-]+)?(/[a-zA-Z0-9\\&amp;%_\\./-~-
]*)?", " ");//com lu olanlar
    fullBook = Regex.Replace(fullBook, "www([a-zA-Z0-
9\\~\\!\\@\\#\\$\\%\\^\\&amp;\\*\\(\\)_\\-
\\=\\+\\\\\\\/\\?\\.\\:\\;\\'\\,]*)?", " ");//www lu olanlar
    fullBook = fullBook.Replace("*", " ").Replace("+", " ").Replace("-", "
").Replace("?", " ").Replace("!", " ").Replace("/", "").Replace("_", "
").Replace("\"", "").Replace(":", " ").Replace(";", " ").Replace(")", "
").Replace("(", " ").Replace("%", " ").Replace("£", " ");
    fullBook = Regex.Replace(fullBook, "\\.|;|:|,|[0-9]|'", " ");
    fullBook = Regex.Replace(fullBook, @"\s[a-z]\s", "");
    fullBook = Regex.Replace(fullBook, @"\s+", " ");

    fullBook = fullBook.ToLower(); //Küçük harflere çevir
    snc = fullBook;
    return snc;
}



public string kelime_islem(string word) { //kelime denetleme fonksiyonu
    string sonc = "";
    Zemberek zemberek = new Zemberek(new TurkiyeTurkcesi());
    if (!zemberek.kelimeDenetle(word)) // kelimenin doğru yazılıp
yazılmadığını denetle
    { //yanlış ise
        String[] sonuclar = zemberek.asciidenTurkceye(word);//kelimenin türkçe
karakterden kaynaklanan sorununu düzeltir. alisveris -> alışveriş
        if (sonuclar.Count() > 0) {
            sonc = kok_bul(sonuclar[0]);
        }
        else
        {
            sonc = word;
        }
        // foreach (String s in sonuclar)
    }
    else { //Doğru ise
        sonc = kok_bul(word);
    }

    return sonc;
}
```

```
public string kok_bul(string word) { //kelime kök bulma
    string snc = "";
    Zemberek zemberek = new Zemberek(new TurkiyeTurkcesi());
    Kelime[] cozumler = zemberek.kelimeCozumle(word);
    Kelime kelime1 = new Kelime();
    try
    {
        kelime1 = cozumler[0];
        snc = kelime1.kok().icerik();
    }
    catch (Exception)
    {
        snc = word;
    }
    return snc;
}

public bool UrlDurum( string msg) {//Mesaj içinde URL var mı kontrolü

      bool snc = false;
      string sc = "(http://([\\w+?\\.\\w+])+([a-zA-Z0-
9\\~\\!\\@\\#\\$\\%\\^\\&amp;\\*\\(\\)_\\-
\\=\\+\\\\\\\/\\?\\.\\:\\;\\'\\,]*)?)|([a-zA-Z0-9\\._-
]+\\.(com|edu|net|org)(\\.[a-zA-Z0-9\\._-]+)?(/[a-zA-Z0-9\\&amp;%_\\./-~-
]*)?)|(www([a-zA-Z0-9\\~\\!\\@\\#\\$\\%\\^\\&amp;\\*\\(\\)_\\-
\\=\\+\\\\\\\/\\?\\.\\:\\;\\'\\,]*)?)";
       Regex regx = new Regex(sc, RegexOptions.IgnoreCase);
       MatchCollection mactches = regx.Matches(msg);
       if (mactches.Count > 0) snc = true;
       return snc;
 }

public double BuyukHarfFrekans(string msg) {//Mesaj büyük harf frekansı
      double snc = 0;
      double count = msg.Count(c => char.IsUpper(c));
      snc =  count / Convert.ToDouble(msg.Replace(" ","").Trim().Length);
      return snc;
}

public double DuygufFrekans(string s) //Mesaj duygusal ifade frekansı
{
    double sonc = 0;
    double toplam = 0;
    string[] words = { ":\\)", ":-\\)", ":\\(", @":\\", " ok ", " hmm ", " Ok
", "Hmm", ":'\\(", "x\\)", "x-\\)", ":p", ":-p", ":\\*", ":-\\*", ":D", ":-D",
";\\)", ";-\\)", ":-0", ":-<", ":-Q", ":Q" };
    for (int i = 0; i < words.Count(); i++)
    {
        int j = Regex.Matches(s, words[i]).Cast<Match>().Count();
        toplam += j;
    }

    sonc = toplam / Convert.ToDouble(s.Replace(" ", "").Trim().Length);
    return sonc;
}
```

```
public void frekanslar(string msj, int tur) // Tüm Mesajlardaki Kelime
Sözlüğünü oluşturma ve Tekrar Sayılarını Belirleme
{
    string fullBook =  msj;
    MatchCollection wordCollection = Regex.Matches(fullBook, @"[\w]+",
RegexOptions.Multiline);
    LinkedList<string> wordList = new LinkedList<string>();
    Hashtable frequencyHash = new Hashtable();
    LinkedList<string> uniqueWord = new LinkedList<string>();
    for (int i = 0; i < wordCollection.Count; i++) // 1) Kelimelere Ayırma
    {
        wordList.AddLast(wordCollection[i].ToString().ToLower().Trim());
    }
    int j = 0;
    foreach (var word in wordList) // 2) Kelimelere Tekrarlarını Bulma
    {
        j++;
        if (uniqueWord.Contains(word))
        {
            int wordCount = int.Parse(frequencyHash[word].ToString());
            wordCount++;
            frequencyHash[word] = wordCount;
        }
        else
        {
            uniqueWord.AddLast(word);
            frequencyHash.Add(word, 1);
        }
    }

    List<kelime_f> kf = new List<kelime_f>(); //Kelime ve Frekans İkilisi
    int z = 0;
    foreach (DictionaryEntry gg in frequencyHash)//3) Kelime Listesi Oluşturma
    {
        z++;
        kelime_f oge = new kelime_f();
        oge.kelime = gg.Key.ToString();
        oge.say = Convert.ToInt32(gg.Value);
        kf.Add(oge);
    }
        frekans_islem(kf,tur); //veri tabanına oluşan sözlük ve frekans
sayılarını yazma işlemi

}
```

7. ÖZGEÇMİŞ

1. KİŞİSEL BİLGİLER

Ad Soyad	:	Onur KARASOY
Uyruk	:	T.C.
Doğum Yeri ve Tarihi	:	Bakırköy-1987
Telefon	:	0555 554 700 1456
E-posta	:	onrkrsy@gmail.com

2. EĞİTİM

Alınan Derece	Aldığı Kurum/Üniversite	Mezuniyet Yılı
Lise	Borusan Asım Kocabıyık A. M. L.	2005
Lisans	Muğla Üniversitesi / Teknik Eğitim Fakültesi	2011
Lisans	Pamukkale Üniversitesi / Mühendislik Fakültesi	2019
Yüksek Lisans	Muğla Sıtkı Koçman Üniversitesi	2019

3. İŞ TECRÜBESİ

Yıl	Yer	Pozisyon/ Görev
2006-2011	Aqua Club Dolphin, Bilgi İşlem, İstanbul	Bilgi İşlem Personeli
2011-…	Muğla Sıtkı Koçman Üniversitesi Bilgi İşlem Dairesi Başkanlığı	Öğr. Gör. Yazılım Geliştirme Uzmanı

4. YAYIN BİLGİLERİ

SCI veya SCI Expanded, SSCI, AHCI tarafından taranan dergilerde yayımlanan tam makale

1. Ballı S. ve Karasoy O., 2019, "Development of content-based SMS classification application by using Word2Vec-based feature extraction", *IET Software,* DOI: 10.1049/iet-sen.2018.5046, Baskıda

Uluslararası kongre, sempozyum, panel, çalıştay gibi bilimsel, sanatsal toplantılarda sözlü olarak sunulan ve tam metin olarak yayımlanan bildiri

1. Karasoy O., Ballı S., 2017, "Derin Öğrenme Aracı Word2Vec ile Türkçe SMS Sınıflandırma", *Uluslararası Bilgisayar Bilimleri ve Mühendisliği Konferansı (UBMK 2017)*
2. Karasoy O., Ballı S., 2016, "İçerik Tabanlı İstenmeyen SMS Filtreleme için Mobil Uygulama Geliştirilmesi ve Sınıflandırma Algoritmalarının Karşılaştırılması", *IDAP 2016 : International Artificial Intelligence and Data Processing Symposium*

Uluslararası kongre, sempozyum, panel, çalıştay gibi bilimsel, sanatsal toplantılarda sözlü olarak sunulan ve özet metin olarak yayımlanan bildiri

1. Gürüler H., Yılmaz Y., Karasoy O., Ayvaz U., 2018, "PinPaper: Web-Based Smart Conference Management System", *International Conference on Data Science and Applications* (ICONDATA 2018)
2. Üstünel E., Abi,M., Sarıman G., Sariman G., Karasoy O., 2017, "Foreign Language in Professional Training of Specialists: Issues and Strategies", *International Scientific and Practical Conference*

Ulusal kongre, sempozyum, panel, çalıştay gibi bilimsel, sanatsal toplantılarda sözlü olarak sunulan ve tam metin olarak yayımlanan bildiri

1. Karasoy O., Ballı S., 2016. "Google Maps ve Genetik Algoritmalarla GSP Çözümü İçin", *AKADEMİK BİLİŞİM KONFERANSI - AB 2016*

2. Karasoy O., Güvenç E., Ballı S., 2015, "OpenPGP ile E-posta şifreleme: Muğla Sıtkı Koçman Üniversitesi Uygulaması", *AKADEMİK BİLİŞİM KONFERANSI -- AB 2015*
3. Sarıman G., Karasoy O.,Tarlacı M.F., Durmuş B., 2014. "Yazılımlar için web servis destekli bütünleşik hesap yönetimi", *AKADEMİK BİLİŞİM KONFERANSI - AB 2014*
4. Sarıman G., Tarlacı M.F., Karasoy O., Durmuş B., 2014. " Eduroam ve SMS destekli E-Posta Kullanıcı Yönetim Modeli: MSKU Örneği", *AKADEMİK BİLİŞİM KONFERANSI - AB 2014*

Bilimsel Araştırma Projelerinde (BAP) görev alma (araştırmacı, eğitmen, danışman, vb. olarak)

1-) Proje Durum: Tamamlandı. Projedeki Görev: Araştırmacı. Konu: . Proje Türü: Yükseköğretim Kurumları tarafından destekli bilimsel araştırma projesi. Yabancı dil olarak Türkçe öğretiminde yakın alan iletişim teknolojisinin kullanımı.. 2014-2016