



T.C. DOĞUŞ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**DERİN ÖĞRENME VE KELİME GÖMME MODELLERİ KULLANARAK
PARKİNSON HASTALIĞININ ANALİZİ**

YÜKSEK LİSANS TEZİ

FEYZA ÇEVİK

20172105037

DANIŞMAN: DR. ÖĞR. ÜYESİ ZEYNEP HİLAL KİLİMCİ

İstanbul, 2019



YÜKSEK LİSANS TEZ SINAV TUTANAĞI

Doküman No	FR.1.26
Yürürlük Tarihi	1.11.2017
Revizyon Tarihi	1.11.2017
Revizyon No	1
Sayfa	1 / 1

SOSYAL BİLİMLER / FEN BİLİMLERİ ENSTİTÜSÜ

Tarih : 19./11./2019

Anabilim/Anasanat Dalı : BILGI SAGAR .. MÜHENDİSLİĞİ

Öğrencinin Adı Soyadı : FEYZA .. GEVİK

Öğrenci No : 20172105037

Tez Danışmanının Adı Soyadı : ZEYNEP .. HİLAL .. KILIÇ

İkinci Tez Danışmanının Adı Soyadı :

Tezin Başlığı : DERİN ÖĞRENME VE KULME GÖNME
MODELLERİ KULLANILARAK PARENTERAL HASTALICILARIN
ANALİZİ

Doğuş Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği'nin 32.Maddesi uyarınca yapılan değerlendirmeler sonunda;

tezin kabul edilmesine

tezde düzeltme verilmesine

tezin reddedilmesine

oy birliği / oy çokluğu ile karar verilmiştir.Gereği için arz olunur.

Danışman Üye

Dr. Öğr. Üyesi Zeynep Hilal KILIÇ

Üye

Prof. Dr. Selim Akyağar

Üye

Dr. Öğr. Üyesi Ramazan

Üye

ANAN

Üye

Anabilim/Anasanat Dalı Başkanı Onayı:

Dr. Öğr. Üyesi Yabancı KILIÇ

YEMİN METNİ

Yüksek lisans tezi olarak sunduğum “Derin Öğrenme ve Kelime Gömme Modelleri Kullanarak Parkinson Hastalığının Analizi” adlı çalışmanın, tarafımdan, akademik kurallara ve etik değerlere uygun olarak yazıldığını ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

Feyza ÇEVİK



19.11.2019

ÖNSÖZ

Bu çalışmamın başından sonuna kadar bilgisiyle ve özverisiyle bana yol gösteren, her zaman yardımcı ve destek olan, yeri geldiğinde danışman kimliğini kenara bırakıp arkadaşlık yapan değerli danışman hocam Dr. Öğr. Üyesi Zeynep Hilal KİLİMCİ'ye sonsuz teşekkürlerimi sunarım.

Ayrıca bu günlere kadar beni yetiştiren, eğitimimde maddi manevi desteğini esirgemeyen, canım anneme ve canım babama sonsuz teşekkürlerimi sunarım.

İstanbul, 2019

Feyza ÇEVİK

ÖZET

Amaç

Parkinson hastalığı, hastanın yaşam kalitesini etkileyen, önemli sosyal ve ekonomik etkileri olan ve semptomların aşamalı görünümü nedeniyle erken teşhis edilmesi güç olan yaygın bir nörolojik hastalıktır. Parkinson hastalığının Twitter gibi sosyal medya platformlarında tartışılması, hastaların Parkinson hastalığının hem tanı hem de tedavi aşamasında birbirleriyle iletişim kurduğu bir platform sağlar. Bu çalışmanın amacı, derin öğrenme ve kelime gömme modelleri kullanarak insanların Parkinson hastalığı ile ilgili duygusallık analizlerini değerlendirmek ve karşılaştırmaktır. Bildiğimiz kadarıyla, Parkinson hastalığını sosyal medyadan kelime gömme modelleri ve derin öğrenme algoritmaları kullanarak analiz etmek için yapılan ilk çalışmadır.

Materyaller ve yöntemler

Parkinson hastalığı ile ilgili tweetler, Twitter sayfalarındaki (“ParkinsonsCure”, “Parkinson”, “ParkinsonsTreatment, ParkinsonDiagnosis”) anahtar kelimeleri ile hesaplarını arayarak elde edildi. Parkinson hastalığına ilişkin tüm tweetler, 01.01.2009-09.01.2019 tarihleri arasında Python programlama dilinde yazdığımız Selenium Crawler kullanılarak toplandı. Bu çalışmada Word2Vec, GloVe ve FastText, tweetleri anlamsal, bağlamsal ve sözdizimi açısından zenginleştirmek amacıyla kelime gömme modelleri olarak kullanılmıştır. Evrişimli Sinir Ağları (CNN'ler), Tekrarlayan Sinir Ağları (RNN'ler) ve Uzun Kısa Süreli Bellek Ağları (LSTM'ler) sınıflandırma görevi için uygulanılmıştır.

DeneySEL Sonuçlar

Bu çalışmada, kelime gömme modelleri ve derin öğrenme algoritmaları kullanılarak Parkinson hastalığı hakkında kullanıcı yorumlarının duygularını analiz etmek için kapsamlı deneyler yapılmıştır. Doğruluk, her modelin sınıflandırma performansını ve çalışmamızın katkısını göstermek için deneylerde bir değerlendirme ölçütü olarak kullanılır. Ön işleme yöntemlerinin kelime gömme modelleri üzerindeki etkisi %50 eğitim setinde incelendiğinde, RH ve RU yöntemlerinin kombinasyonunun en iyi doğruluk performansını gösterdiği görülmektedir. Word2Vec kelime gömme modellerinde ön işleme yöntemleriyle %89,34 doğrulukla en iyi sonuçlara sahiptir. Benzer şekilde, ön işleme yöntemlerinin derin öğrenme algoritmaları üzerindeki etkisi incelendiğinde, RH ve RU konsolidasyonunun diğer ön işleme yöntemlerinden daha iyi

sonular verdiđi grlmektedir. LSTM, RU + RH yntemleriyle harmanlanarak %93,63 dođruluk performansına sahip en iyi performans gsteren sınıflandırma algoritmasıdır.

Sonular

Bu alıřma, hastaların gereksinimlerini anlamak iin kelime gmme modelleri ve derin đrenme algoritmalarını kullanma etkinliđini ve duygularını analiz ederek tedavi srecine deđerli bir katkı sađladıđını gstermektedir.

Anahtar Kelimeler: Parkinson hastalıđı; Duygu analizi; Derin đrenme; Uzun kısa sreli hafıza ađları; Kelime gmme modelleri.



ABSTRACT

Objective

Parkinson's disease is a common neurodegenerative neurological disorder, which affects the patient's quality of life, has significant social and economic effects, and is difficult to diagnose early due to the gradual appearance of symptoms. Examining the discussion of Parkinson's disease in social media platforms such as Twitter provides a platform where patients communicate each other in both diagnosis and treatment stage of the Parkinson's disease. The purpose of this work is to evaluate and compare the sentiment analysis of people about Parkinson's disease by using deep learning and word embedding models. To the best of our knowledge, this is the very first study to analyze Parkinson's disease from social media by using word embedding models and deep learning algorithms.

Materials and Methods

Tweets about Parkinson's disease are obtained by searching accounts on Twitter pages with keywords (“ParkinsonsCure”, “Parkinson”, “ParkinsonsTreatment, ParkinsonDiagnosis”). All tweets related to Parkinson's disease are collected from 01.01.2009 to 09.01.2019 using Selenium Crawler, which we write in Python programming language. In this study, Word2Vec, GloVe, and FastText are employed as word embedding models for the purpose of enriching tweets in terms of semantic, context, and syntax. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory Networks (LSTMs) are implemented for the classification task.

Results

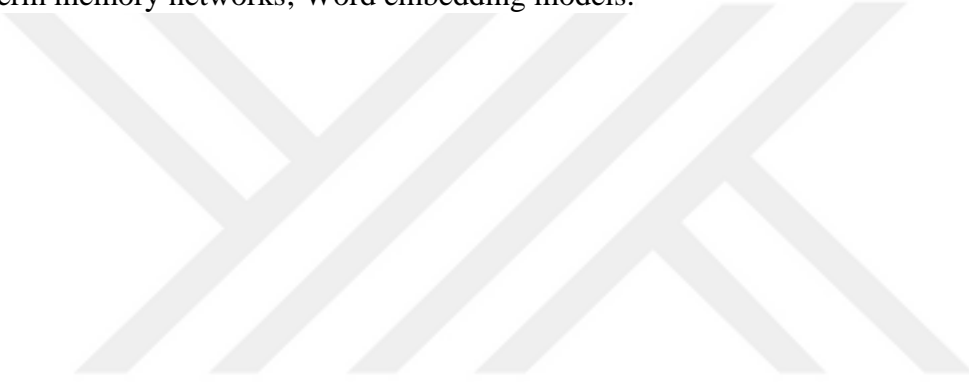
In this study, extensive experiments are conducted to analyze the emotions of user comments about Parkinson's disease using word embedding models and deep learning algorithms. Accuracy is used as an evaluation metric in the experiments to demonstrate the classification performance of each model and the contribution of our study. When the effect of preprocessing methods on word embedding models is examined at 50% training set, it is observed that the combination of RH and RU methods shows the best accuracy performance. Word2Vec has the best results with 89.34% accuracy with preprocessing methods in word embedding models. Likewise, when the effect of preprocessing methods

on deep learning algorithms is analyzed, it is observed that the consolidation of RH and RU gives better results than other preprocessing methods. LSTM is the best performing classification algorithm with 93.63% accuracy performance by blending with the RU + RH methods.

Conclusions

This study demonstrates the efficiency of using word embedding models and deep learning algorithms to understand the needs of patients' and provide a valuable contribution to the treatment process by analyzing sentiments of them.

Keywords: Parkinson's disease; Sentiment analysis; Deep learning; Long short-term memory networks; Word embedding models.



İÇİNDEKİLER

	Sayfa No.
ÖNSÖZ	i
ÖZET	ii
ABSTRACT.....	iv
İÇİNDEKİLER	vi
TABLO LİSTESİ.....	vii
ŞEKİL LİSTESİ.....	viii
KISALTMALAR.....	ix
1. GİRİŞ.....	1
2. LİTERATÜR ÇALIŞMALARI.....	4
3. YÖNTEMLER, MATERYALLER VE ÖNERİLEN ÇERÇEVE	7
3.1. Veri Toplanması ve Önerilen Çerçeve	7
3.2. Kelime Gömme Modelleri	10
3.3. Derin Öğrenme Algoritmaları	22
4. DENEYSEL SONUÇLAR.....	29
5. SONUÇ.....	40
KAYNAKÇA.....	41
ÖZGEÇMİŞ	45

TABLO LİSTESİ

Sayfa No.

Tablo 3.1 6 milyar tonluk bir korpustan seçilen bağlam sözcüklerle hedeflenen ice ve stream kelimeleri için eşzamanlı olma olasılıkları.	14
Tablo 3.2 FastText ve derin öğrenme temelli yöntemler arasındaki karşılaştırma.....	21
Tablo 4.1 Kelime gömme modellerinin eğitim seti yüzdelerine göre sınıflandırma doğrulukları.	30
Tablo 4.2 Derin öğrenme algoritmalarının eğitim seti yüzdelerine göre sınıflandırma doğrulukları.	31
Tablo 4.3 %50 eğitim setinde ön işleme yöntemlerinin kelime gömme modelleri üzerindeki etkisi.....	32
Tablo 4.4 Ön hazırlık yöntemlerinin %50 eğitim setinde derin öğrenme algoritmaları üzerindeki etkisi.....	33
Tablo 4.5 Önerilen modelimizin Parkinson hastalığına yönelik son teknoloji çalışmalarla karşılaştırılması.	38

ŞEKİL LİSTESİ

	Sayfa No.
Şekil 3.1 Önerilen sistemin akış şeması.....	9
Şekil 3.2 3 boyutlu uzayda Word2Vec örneği.....	10
Şekil 3.3 Örnek Word2Vec mimarisi gösterimi.	11
Şekil 3.4 Skip-gram model mimarisi.	12
Şekil 3.5 CBOW modeli mimarisi.	13
Şekil 3.6 Ülke-başkent ilişkileri.....	17
Şekil 3.7 Ülke-başkent ilişkileri.....	18
Şekil 3.8 Ülke → Para Birimi İlişkisi: İngiltere → Euro ilişkisi kurulmuştur.	18
Şekil 3.9 Ülke → Para Birimi İlişkisi: Türkiye → Lira ilişkisi kurulamamıştır.	19
Şekil 3.10 Cinsiyet ilişkisi.	19
Şekil 3.11 Cinsiyet ilişkisi.	20
Şekil 3.12 CNN mimarisi modeli.	22
Şekil 3.13 ReLu fonksiyonunun Feature Map'a uygulanması.	24
Şekil 3.14 Flattening katmanı mimarisi.	25
Şekil 3.15 RNN işlem döngüsü.....	26
Şekil 3.16 Uzun-kısa süreli hafıza ağı mimarisi.	27
Şekil 3.17 Uzun- kısa süreli hafıza ağı katmanı mimarisi.	28
Şekil 4.1 RH + RU bir ön işleme yöntemi olarak ayarlandığında her kelime gömme ve derin öğrenme modelinin eğitim seti yüzdeleri açısından sınıflandırma performansları.....	34
Şekil 4.2 Word2Vec sabitlendiğinde ön işleme yöntemlerinin eğitim yüzdeleri bakımından sınıflandırma doğrulukları.	35
Şekil 4.3 Ön işleme yöntemlerinin LSTM eğitim yüzdeleri cinsinden sınıflandırma doğrulukları.	36

KISALTMALAR

PD	: Parkinson Hastalığı
NLP	: Doğal Dil İşleme
DL	: Derin Öğrenme
WE	: Kelime Gömme
CNN	: Evrişimli Sinir Ağları
RNN	: Tekrarlayan Sinir Ağları
LSTM	: Uzun-Kısa Süreli Bellek Ağları
DBN	: Derin İnanış Ağları
GloVe	: Global Vektörler
CBOW	: Sürekli Kelime Torbası
SWE	: Kelime durdurma eliminasyonu
RH	: Hashtagleri kaldırma
RU	: URL'leri kaldırma
STM	: Stemming
AOT	: Bunların hepsi

1. GİRİŞ

Son yıllarda internet ve mobil teknolojinin gelişmesiyle birlikte, sosyal medya platformları hızla büyüdü. Hastaları ve akrabalarını hastalık süreci hakkında bilgilendirmek, görüş ve deneyimlerini paylaşmak ve tedavi seçenekleri hakkında ayrıntılı bilgi edinmek için önemli bir kaynak haline gelmiştir (Eckler, Worsowicz & Rayburn, 2010; Martino vd., 2017). Twitter, bilgileri olduğu gibi paylaştığı ve gerçek zamanlı olarak başkalarıyla bağlantı kurduğu bilinen en popüler sosyal ağ hizmetidir (Prieto vd., 2014; Sinnenberg vd., 2017). Aylık yaklaşık 350 milyon aktif Twitter kullanıcısı ile Twitter, gerçek dünya hakkında bilgi toplamak için güçlü bir araçtır (Neiger vd., 2013). Twitter, hastaların ve hasta yakınlarının deneyimlerini paylaştığı ve duygularını anlamalarına olanak sağladığı güçlü bir sağlık veri kaynağıdır (Beykikhoshk vd., 2014; Beykikhoshk vd., 2015; Chew & Eysenbach, 2010).

Sentiment Analiz (Duygu/duyarlılık analizi), konuşma veya yazma gibi öznel bir dil ögesinin belirli bir konuda pozitif veya negatif olarak kutuplaşmasıdır. Sınıflandırma her zaman pozitif veya negatif olmayabilir, ancak kullanılacak soruna veya verilere bağlı olarak çok pozitif, pozitif, nötr, negatif, çok negatif gibi sınıflarla detaylandırılabilir. Otomatik duygu analizi, çünkü öznel bilgi istatistiklerine ulaşmak isteyen bir birimin metinler aracılığıyla analiz yapması amaçlar.

TextBlob metinsel verileri işlemek için bir Python kütüphanesidir. Etiketleme, isim ifade çıkarma, duygu analizi, sınıflandırma, çeviri ve daha fazlası gibi ortak doğal dil işleme (NLP) görevleri için basit bir API sağlar.

Parkinson hastalığı (PD), merkezi sinir sisteminin dünya çapında 10 milyondan fazla insanı etkileyen, yaşam kalitesini, motor refleksini, konuşmayı, davranışı, zihinsel süreci ve diğer hayati fonksiyonları etkileyen nörodejeneratif bir hastalıktır (Fahn, 2003; Rijk vd., 1997). Parkinson hastalarında titreme belirtileri, yavaş hareket, kas sertliği ve denge kaybı gibi birçok semptom vardır. Bu semptomlara ek olarak, zihinsel değişiklikler, konuşma bozuklukları, uyku bozuklukları ve depresyon belirtileri de sıklıkla gözlenir ve bu semptomlar zamanla kötüleşebilir. Konuşma bozuklukları hastaların yaklaşık yüzde 90'ında görülür. Ek olarak, bu hastalar düşük ses, donuk konuşma, konuşmaya başlamada zorluk, gürültüyü sürdürmede zorluk, telaffuzda hata ve akıcılığın azalması gibi konuşma bozukluğu belirtileri içerebilir (Fahn, 2003). Parkinson tipik olarak 40-70 yaşları arasında görülür ve yaş ilerledikçe görülme sıklığı artış gösterir.

Hastaların %75 kadarında ilk hareket belirtileri, 60 yaşından sonra ortaya çıkar. Epidemiyolojik arařtırmalarda yař ve cinsiyet haricinde semptomlar diđer risk durumları arasında beyaz ırk, aile öyküsünde Parkinson hastalığı olması, kafa travması, karbon monoksit, ağır metaller ve tarım ilaçları gibi çeřitli toksik maddelere maruz kalınması, kuyu suyu içimi ve kırsal kesimde yařama gibi durumlar söz konusudur.

Derin öğrenme (DL) algoritmaları ve kelime gömme (WE) modellerinin uygulanması son yıllarda görüntü işleme (Image Processing), görüntü sınıflandırma (Image Classification), nesne bulma (Object Detection), nesne takip etme (Object Tracking), doğal dil işleme (Natural Language Processing), stil transferi (Style Transferring) ve makine çevirisi gibi farklı araştırma alanlarında çok popülerdir. DL yöntemleri ve WE modelleri, geleneksel makine öğrenmesi algoritmalarına kıyasla daha iyi tahminler ve sonuçlar sunması nedeniyle arařtırmacılar tarafından tercih edilmektedir. Derin öğrenme modelleri temel olarak, derin sinir ađları aracılıđıyla verilerin anlamlı bir şekilde temsil edilmesini sađlamak için karmařık özellikleri minimal dış destekle eđiterek otomatik özellik çıkarımı sađlamak için kullanılır. Ayrıca, birçok alanda sınıflandırma görevlerinde derin öğrenme yöntemleri de kullanılmaktadır. Evriřimli sinir ađları (CNN'ler), tekrarlayan sinir ađları (RNN'ler), uzun kısa süreli bellek ađları (LSTM'ler) (Kilimci & Akyokuř, 2018), derin öğrenme modelleri olarak derin inanç ađları (DBN'ler) ve Word2Vec (Chen, Corrado, & Dean, 2013; Le & Mikolov, 2014) GloVe (Pennington, Socher & Manning, 2014), FastText (Joulin vd., 2016) kelime gömme modelleri olarak tanınmış mimarilerdir.

Bu çalışmada, Parkinson hastaları için kişisel bir bakım planının geliştirilmesini kolaylařtırmayı ve bireysel ve örgütsel kullanıcıların duygu analizini kullanarak, artan bir Parkinson hasta popülasyonuna kişisel sađlık bilgileri için deđerli bir tıbbi kaynak sađlamayı öneriyoruz. Bu amaçla, Parkinson hastalığı ile ilgili tweet'lerin anlamlarını ve bağlamlarını zenginleřtirmek için Word2Vec, GloVe ve FastText adlı üç farklı kelime gömme modeli kullanılmaktadır. Ayrıca, bireysel ve örgütsel kullanıcıların tweetlerini yorumlamak ve pozitif, negatif ve tarafsız gibi tweetleri sınıflandırarak Parkinson hastalığı algısını anlamak için derin öğrenme modelleri kullanılmaktadır. Bildiğimiz kadarıyla, bu, derin öğrenme ve kelime gömme temelli modelleri kullanarak Parkinson hastalığına iliřkin duyguları analiz etmeye yönelik ilk girişimdir. Deney sonuçları, RU +

RH ve LSTM kombinasyonunun %93,63 doğruluk sonucu diğerlerinden daha iyi performans gösterdiğini göstermektedir.

Makalenin geri kalanı şu şekilde düzenlenmiştir: bölüm 2, Parkinson hastalığı ile ilgili yapılan çalışmaların bir özetini sunar. Bölüm 3, gömme modelleri ve derin öğrenme algoritmaları kelimesini, deney düzenliğini, sonuçları ve sonuçları bölüm 4, 5 ve 6'da verilmektedir.



2. LİTERATÜR ÇALIŞMALARI

Bu bölüm, Parkinson Hastalığı (PD) hakkındaki çalışmaların literatür taramasının kısa bir özetini sunar.

Bir çalışmada yazarlar, Probabilistic Neural Network (PNN) sınıflandırıcısı, Parkinson hastalığının tanısı için kullanılır ve %80,92'lik test başarısı elde eder. Bu çalışmada kullanılan veri seti 23'ü Parkinson hastalığı olan 31 denekten alınan çeşitli biyomedikal ses ölçümlerinden oluşmaktadır (Prashanth & Roy 2018). Başka bir çalışmada, Parkinson hastalarından (42 kişi) oluşan ve hasta başına yaklaşık 200 ses kaydı içeren Parkinson veri setinde sınıflandırılmıştır. Destek vektör makinesi, önerilen modelin sınıflandırıcısı olarak kullanılır. Öte yandan, filtre seçiminde ilgisiz ölçüm özellikleri seçmek için kullanılır. Deney sonuçları, önerilen modelin sınıflandırma performansının %91,40 başarı oranına ulaştığını göstermektedir (Tsanas vd., 2009). Başka bir çalışmada, konvansiyonel önyükleme ya da bir-bir-dışa doğrulama validasyon yöntemi ve karşılıklı olarak bilgiye dayalı bir nitelik seçimi metodu (maksimum alaka düzeyi minimum yedeklilik - MRMR) tarafından toplanan özellikleri kullanır. PD veri seti, 195 denekten oluşan 32 kişinin (24 PD ve 8 sağlıklı) ses kayıtlarından meydana gelmiştir ve %92,9 doğruluk performansı rapor edilmiştir (Sakar & Kurşun, 2010). Başka bir çalışmada, Parkinson hastalarını ayırt etmek için sinir ağları, veri madenciliği sinir ağları modeli (DMneural), regresyon ve karar ağacı olmak üzere dört bağımsız sınıflandırma modeli uygulamıştır. Sinirsel ağ tabanlı sınıflandırıcı %92,9 doğruluk sonucuyla diğerlerinden daha iyi performans gösterir (Das, 2010). Diğer bir çalışmada, hem sınıflandırma görevi hem de özellik seçimi için insan biyomedikal sesini içeren Parkinson hastalığı veri kümesi örneklerinde dilsel olarak güçlü adaptif sinir bulanık sınıflandırıcı (ANFC + LH) kullanılmaktadır. ANFC + LH kullanımının 94.72 sınıflandırma performansı gösterdiğini rapor etmişlerdir (Çağlar, Çetişli & Toprak, 2010).

Başka bir çalışmada, Parkinson hastalığının sınıflandırılması, bulanık C-ortalama kümelemesi temelinde özellik ağırlıklandırma yöntemi kullanılarak önerilmiştir. PD veri seti, Kaliforniya Üniversitesi Irvine makine öğrenim veritabanından elde edildi. Veri setinin içeriği, 31'i erkek, kadın, 23'ü PD tanısı alan 195 uzun vadeli sesli fonan oluşmaktadır. Ağırlıklandırılmış Parkinson hasta veri seti için k-NN sınıflandırma algoritması kullanılarak farklı k değerleri için %97,93'lük bir başarı oranı elde edilmiştir (Polat, 2012). Diğer bir çalışmada, dört tıbbi veri seti, dermatoloji, Pima-Hint diyabeti,

meme kanseri ve Parkinson veri seti kullanılarak, benzerlik sınıflandırıcısı (SC) ve bulanık entropi (FE) özellik seçim yöntemleriyle %85,03 sınıflandırma performansı elde etmektedir (Luukka, 2011).

Diğer bir çalışmada yazarlar, yapay bağışıklık sistemi (AIS) ve klon niteliği seçim yaklaşımlarını kullanarak Parkinson hastalığının tanınmasına odaklanmaktadır. Bu amaçla, veri seti, yukarıda belirtilen çalışmaların çoğunda olduğu gibi Oxford Üniversitesi'nde, Ulusal Ses ve Konuşma Merkezi, Denver, Colorado ile iş birliği içinde oluşturulmuştur. Yazarlar AIS kullanımının %92,70 sınıflandırma başarısı sağladığını bildirmektedir (Kihe & Benyettou, 2011). Başka bir çalışmada, Parkinson hastalığının tanısı için özellik seçim yöntemlerinin vokal ölçümlerden karşılaştırılmasına odaklanmaktadır. Bu amaçla, SVM sınıflandırıcısı altı farklı özellik seçim yöntemiyle harmanlanmıştır. Parkinson hastalığı veri seti ve birleşik Parkinson hastalığı veri seti, iki Parkinson hastalığı veri seti grubu, Parkinson hastalığının tanısında biyomedikal ses parametreleri kullanılarak test edildi. Deneysel sonuçları, özyinelemeli özellik elemeli (SVM-RFE) özellik seçim yöntemine dayanan destek vektör makinesi yöntemlerinin, önerilen sistemin sınıflandırma başarısını diğerlerine göre %95,13 doğruluk değeriyle artırdığını göstermektedir (Eskidere, 2012).

Başka bir çalışmada yazarlar, Parkinson hastalığında klinik ölçekler ve makine öğrenme modelleri kullanarak yeni ve geliştirilmiş aşama tahmini önermektedir. Lojistik regresyon (OLR), destek vektör makinesi (SVM), Adaptif Boost (AdaBoost) ve Random Under sampling Boosting (RUSBoost) tabanlı sınıflandırıcılar gibi makine öğrenme tekniklerini kullanarak PD'nin aşamasını ve ciddiyetini tahmin etmek için tahmin modelleri geliştirirler. Ek olarak, PD veri setindeki özelliklerin önemi rastgele ormanlar kullanılarak belirlenir. AdaBoost tabanlı topluluk modeli %97,46 ile en yüksek sınıflandırma sonucunu sunuyor. Yazarlar hasta anketi ve prediktif modelleme yoluyla Parkinson hastalığının erken saptanmasına odaklanırlar. Bu amaçla, hareket bozukluğu toplum-birleşik Parkinson hastalığı derecelendirme ölçeğinden (MDS-UPDRS) gelen hasta anketi veri seti olarak kullanılmıştır. Kişiyi sağlıklı, normal ve erken olarak sınıflandırabilecek modelleri tahmin etmek için, lojistik regresyon, destek vektör makinesi, ağaçları artırma, rastgele ormanlar gibi yaygın olarak kullanılan makine öğrenme tekniklerini kullanırlar. Bu tekniklerin, erken PD sınıflamasında ROC eğrisi altında (her ikisinde > %95) yüksek doğruluk ve performans gösterdiğini bildirmişlerdir.

Bu çalışma, bu tahmin modellerinin, bir anketin maddelerini makine öğrenmesi yoluyla birleştirerek uzmanların teşhis sürecinde yardımcı olma potansiyeline sahip olduğu sonucuna varılmıştır. Bir başka çalışmada yazarlar derin öğrenme yaklaşımlarını kullanarak Türkçe metinlerden anlamsal çıkarımlara bulunmuştur. Türkçe metinleri toplamak için çeşitli kategorilerde ürün satışı yapan bir internet sitesini kullanmışlardır. Bu siteden toplanan metinleri analiz ederek pozitif negatif anlam içeren kelimelerin vektör temsillerini öğrenerek word2vec modelinde eğitmişlerdir. Eğitilmiş olan kelime setleri kullanılarak Rastgele Orman (Random Forest- RF) modeli geliştirilmiştir. Klasik tekniklere ek olarak, derin öğrenme yaklaşımlarından LSTM ve CNN modelleri duygu sınıflandırma için kullanılmıştır. Sonuç olarak CNN modeline göre LSTM modelinin %94,21 daha iyi sonuç ürettiği görülmüştür (Prashanth & Roy, 2018).

Diğer bir çalışmada, yapay sinir ağı modelinin cümle ve belge modellemesine dikkat çekmek için Zhou ve arkadaşları (2015) çalışma yapmışlardır. Çalışmalarında CNN ve RNN isimli iki temel modeli doğal dilleri anlamada kullanmışlardır. Bu iki mimarinin güçlü yönlerini birleştirmişlerdir. Mimamriler duyarlılık sınıflandırması ve soru sınıflandırması görevleri üzerinde değerlendirilmektedir. Deneysel sonuçlar iki modelin birlikte kullanıldığı yeni model olan C-LSTM'in daha mükemmel performans gösterdiğini ortaya koymuştur. (Zhou, Sun, Liu, & Lau, 2015) Diğer bir çalışmada yazarların yaptıkları çalışmada 16 farklı metin veri seti üzerine sınıflandırma çalışmasını LSTM modelini kullanarak gerçekleştirmişlerdir. Sinir ağları modelleri, ortak ve görev değişmez özelliklerini ortaya koymak için bu çalışma yapılmıştır ve çalışma çok görevli öğrenim için umut vericidir (Liu, Qiu & Huang, 2017).

Çalışmamız, yukarıda bahsedilen literatür çalışmalarından farklıdır; bunun sebebi Parkinson hastalığının algı ve duygu analizini, derin öğrenme algoritmaları ve sözcük gömme modelleri kullanan sosyal medyadan saptamaya yönelik ilk girişimi olmasıdır. Literatür araştırmalarından farklı olarak, Parkinson hastaları için kişisel bir bakım planının geliştirilmesini kolaylaştırmayı ve sosyal medyadan gelen metin verilerini kullanarak, artan bir Parkinson hasta popülasyonuna kişisel sağlık bilgileri için değerli bir tıbbi kaynak sağlamayı öneriyoruz.

3. YÖNTEMLER, MATERYALLER VE ÖNERİLEN ÇERÇEVE

Yöntemlerin, malzemelerin ve önerilen çerçevenin bir özeti bu bölümde sunulmaktadır.

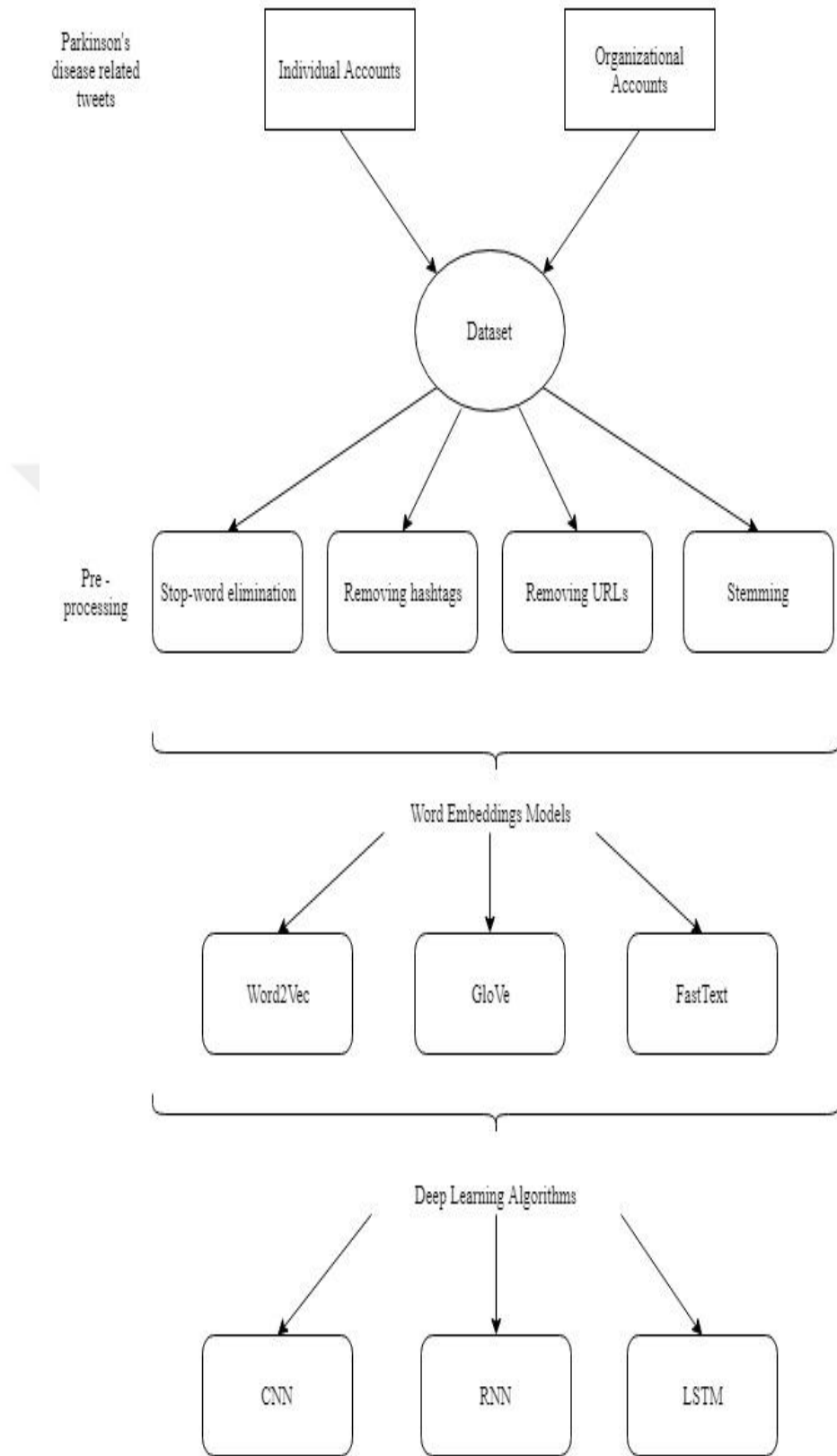
3.1. Veri Toplanması ve Önerilen Çerçeve

Bu çalışmada, Parkinson hastalığına bağlı tweet'lerin duyarlılık analizlerini değerlendirerek Parkinson hastalığının saptanmasına odaklandık. Bu amaçla, Selenium tarayıcısı, Parkinson hastalığı ile ilgili olarak aşağıdaki anahtar kelimelerden oluşan tweetleri toplamak için kullanılmaktadır: “Parkinson”, “ParkinsonDisease”, “ParkinsonsCure”, “ParkinsonsTreatment”, “ParkinsonDiagnosis”, “ParkinsonSymptom”. Twitter'da bireysel ve organizasyonel olmak üzere iki ana kullanıcı hesabı vardır. Parkinson hastalarının ve akrabalarının Parkinson hastalığı hakkındaki görüş ve deneyimlerini anlamak ve yorumlamak için Twitter'da tweet adı verilen bireysel ve kurumsal kullanıcı yorumları toplanmaktadır. Bireysel hesaplar hem Parkinson hastalarına hem de onların yakınlarına ait olabilirken, organizasyon hesapları nörologları, haber kaynaklarını ve hasta olmayanları içerebilir. Bu çalışmada, halka açık tweetli bireysel hesaplar, bazı bireysel hesapların korunan tweet'leri nedeniyle analiz edilmiştir. Toplamda 11.043 tweet hem bireysel hem de kurumsal hesaplardan indirilmektedir. Selenium kullanarak, Twitter API'sinin izin verdiği sınır sorunu hakkında endişelenmeden istediğimiz kadar tweet alıyoruz. Bu çalışmada, denetimli makine öğrenme stratejisine odaklanıyoruz. Bu nedenle, her kullanıcının TextBlob kullanarak Parkinson hastalığına karşı tutumunu belirlemek için kullanıcılardan gönderilen her tweet'i pozitif, negatif veya nötr olarak etiketleme ihtiyacı vardır (Loria, 2018). TextBlob, duyarlılığı belirlemek için Navie Bayes sınıflandırıcısı kullanır ve her tweet için pozitif, negatif veya nötr olarak sınıf olasılığını oluşturur. Kullanıcı yorumlarını anlamada ortalama %79,13 ortalama sınıflandırma başarısı gösterir. Her bir kullanıcıdan toplanan ham veri seti sosyal medya platformlarında oldukça kirli. Bu nedenle, farklı ön işleme teknikleri uygulamasına ihtiyaç vardır. Bu çalışmada, stop-word elimination, removing hashtags, removing URLs, ve stemming teknikleri uygulanmaktadır (Bruns, Kornstadt & Wichmann, 2009).

Twitter'ın karakter kısıtlaması nedeniyle kullanıcıların duygularını yeterince ifade edemedikleri bir gerçektir. Bu sorunu gidermek için Word2Vec, GloVe ve FastText gibi gömme modellerine odaklanıyoruz. Bu şekilde, her yorum kelime gömme modelleri

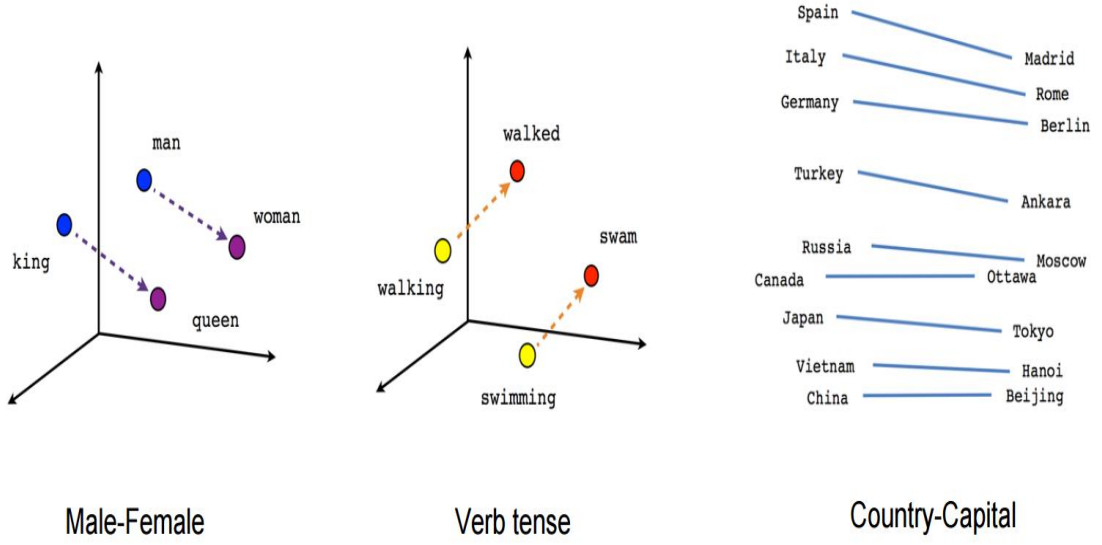
kullanılarak anlam, içerik ve sözdizimi açısından zenginleştirilmiştir. Bu yöntemlerle, Twitter'da fikirlerini ifade etmedeki sınır sorunu kullanıcı duygularını anlamak için bir sorun değildir. Bundan sonra, geleneksel makine öğrenme algoritmaları kullanmak yerine, evrişimli sinir ağları (CNN), tekrarlayan sinir ağları (RNN) ve uzun kısa süreli bellek ağları (LSTM) gibi üç farklı derin öğrenme mimarisi sınıflandırma amacıyla kullanılmaktadır. Önerilen sistemin akış şeması Şekil 3.1'de verilmiştir.





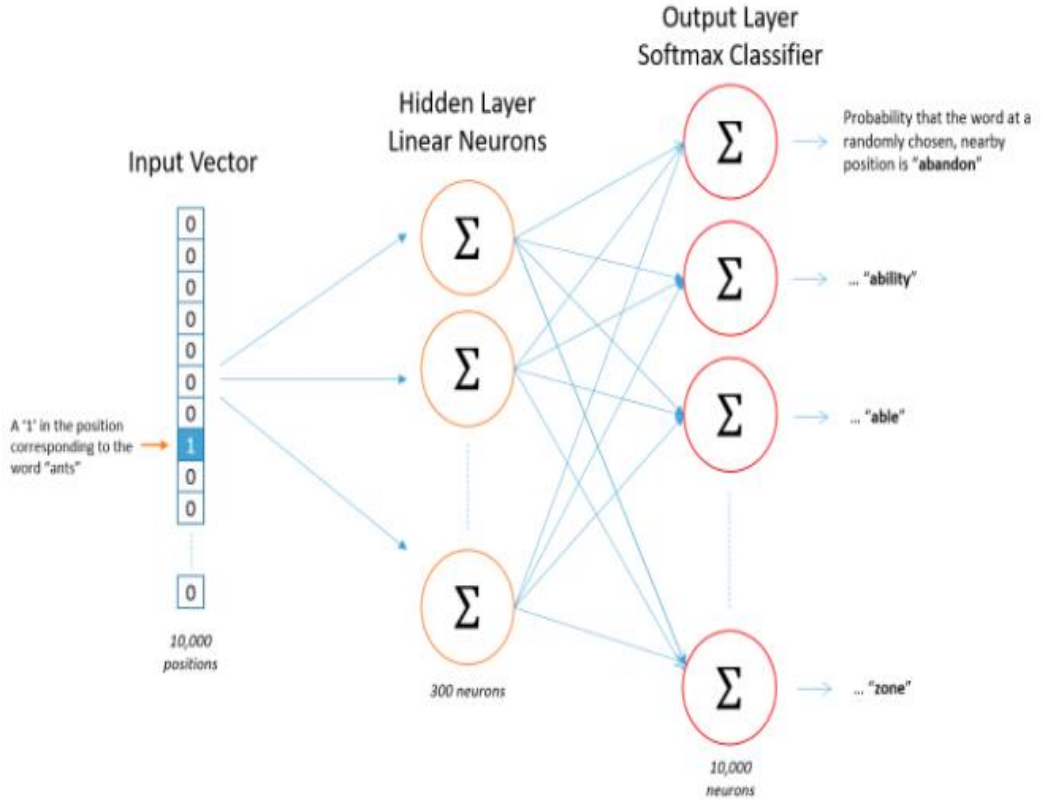
Şekil 3.1 Önerilen sistemin akış şeması

3.2. Kelime Gömme Modelleri



Şekil 3.2 3 boyutlu uzayda Word2Vec örneği.

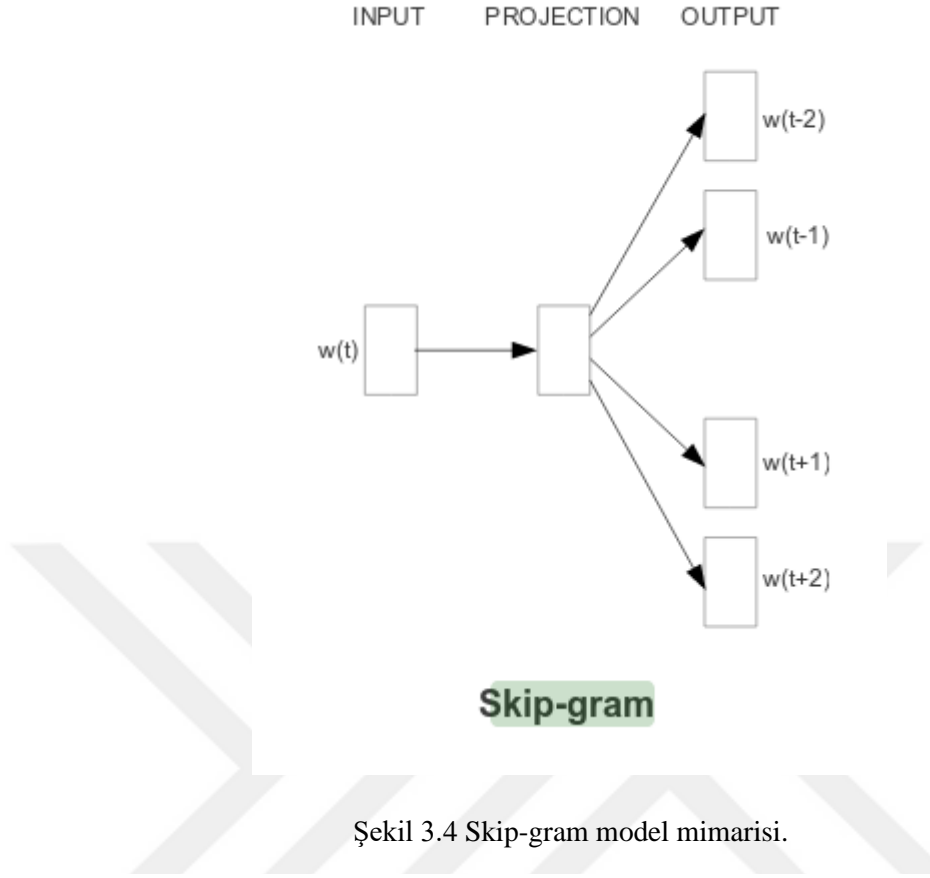
Word2Vec, metni işleyen iki katmanlı bir sinir ağıdır. Word2Vec derin bir sinir ağı olmasa da metni derin ağların anlayabileceği sayısal bir forma dönüştürür. Kelime vektörlerini oluştururken pencere genişliği, embedding boyutu gibi hiper parametreler bulunmaktadır. Pencere genişliği hedef kelimenin sağında ve solunda kaç kelime olması gerektiğini belirtirken, embedding boyutu ise her bir kelimenin kaç boyutlu vektör olarak tanımlanacağını belirtir. Bu durum gizli katmandaki nöron sayısına karşılık gelmektedir. Şekil 3.2’de 3 boyutlu uzayda Word2Vec örneği verilmiştir.



Şekil 3.3 Örnek Word2Vec mimarisi gösterimi.

Yukarıda Şekil 3.3'te örnek olarak Word2Vec mimarisi gösterilmiştir. Gizli katmanda 300 nöron olduğu görülmektedir. Gizli katmanda 300 nöronun bulunmasından her bir kelimenin 300 boyutlu bir vektör olarak gösterdiği anlaşılmaktadır. Sözlükte 10.000 farklı kelime bulunduğu için girdi ve çıktı boyutu 10.000 olarak belirtilmiştir.

Mikolov ve arkadaşları (2013; 2014) Skip-gram ve Sürekli Sözcük Torbası (CBOW) modelleri olmak üzere iki farklı model önermektedir.

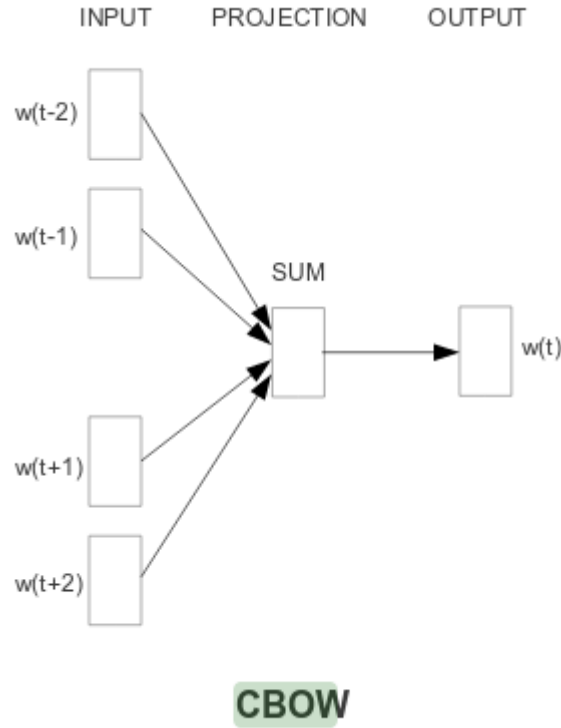


Şekil 3.4 Skip-gram model mimarisi.

Şekil 3.4'te Skip-gram modeli mimarisi verilmiştir. Skip-gram modelleri, bir cümle veya belgedeki bir kelimenin etrafındaki kelimeleri tahmin etmektir (Kilimci & Akyokuş, 2018). Girdi ve çıktıları olasılık olarak benzetip anlamsal olarak en uygun biçimde temsil etmek amaçlanmıştır. Örneğin, 'ben bugün çok fazla çalıştım' cümlesi için, 'çok' kelimesi girdi olarak kullanılır ve pencere boyutu 5 olarak ayarlandığında, 'ben', 'bugün', 'fazla' ve 'çalıştım' kelimeleri çıktı olarak kullanılır. Daha resmi olarak, $w_1, w_2, w_3, \dots, w_T$, Skip-gram modelinin amacı, ortalama kütük olasılığını en üst düzeye çıkarmaktır. Denklem 3.1 Skip-gram modeli ifade eder:

Denklem 3.1

$$J_{\emptyset} = \frac{1}{T} \sum_{t=1}^T \sum_{-j \leq 0 \leq j, j \neq 0} \log p(w_{t+j} | w_t)$$



Şekil 3.5 CBOW modeli mimarisi.

Şekil 3.5'te CBOW modeli mimarisi gösterilmiştir. CBOW (Continuous Bag of Words) skip-gram'a oldukça yakın bir modeldir. Şekil 3.4 ve Şekil 3.5'te de görüldüğü gibi, aralarındaki tek fark girişlerle çıkışların tam tersi olması durumudur. Buradaki fikir bir kelimenin etrafındaki kelimeler verildiğinde hangi kelimenin bu kelimeler içinde görülme olasılığının en yüksek olduğunu bilmek istemesi durumudur.

Daha resmi olarak, CBOW modeli w_T hedef sözcüğünü çevreleyen kelimelerden tahmin eder. Başka bir deyişle, CBOW, w_t hedef sözcüğünü çevreleyen n sözcüklerini eğiterek w_t hedef sözcüğünü öğrenir. CBOW modeli, günlüğe kaydetme olasılığını en üst düzeye çıkararak w_T hedef sözcüğünü tahmin eder. Denklem 3.2 CBOW modelin çalışma mantığını ifade eder:

Denklem 3.2

$$J_{\emptyset} = \frac{1}{T} \sum_{t=1}^T \log p(w_t | w_{t-n}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+1})$$

GloVe, Word Temsili için Global Vektörler, önerilen bir başka kelime gömme modelidir. İstatistikler kontrol edilmeyen algoritmalara dayanır (Pennington, Socher & Manning, 2014). Bir korpustaki kelime oluşumlarının istatistikleri, kelime temsillerini öğrenmek için tüm denetlenmemiş yöntemlerin kullanabileceği temel bilgi kaynağıdır ve bu tür birçok yöntem şu anda mevcut olmasına rağmen, bu istatistiklerden nasıl anlam çıkarıldığına ve sonuçların nasıl ortaya çıktığına dair soru hala devam etmektedir. Kelime vektörleri bu anlamı temsil edebilir.

Tablo 3.1 6 milyar tonluk bir korpustan seçilen bağlam sözcüklerle hedeflenen ice ve steam kelimeleri için eşzamanlı olma olasılıkları.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k ice)/P(k steam)$	8.9	8.5×10^{-2}	1.36	0.96

Tablo 3.1’de gösterilen Water ve fashion gibi ayırıcı olmayan kelimelerin sesleri sadece oranla ortadan kalkar, böylelikle büyük değerler (1’den büyük) ice’ye özgü özelliklerle iyi ilişki kurar ve küçük değerler (1’den çok) steam’e özgü özelliklerle iyi korelasyon gösterir.

Anlamın belirli yönlerinin doğrudan eşzamanlılık olasılıklarından nasıl çıkarılabileceğini gösteren basit bir örnekle başlıyoruz. İlginin belli bir yönünü gösteren iki kelimeyi i ve j düşünün; Somutluk için, $i = ice$ ve $j = steam$ alabileceğimiz termodinamik faz kavramıyla ilgilendiğimizi varsayalım. Bu kelimelerin ilişkisi, birlikte yaşama olasılıklarının çeşitli sonda kelimelerle olan oranı incelenerek incelenebilir, k . İce ile ilgili, ancak steam’le ilgili olmayan k kelimeleri için, $k = solid$ deyin, P_{ik} / P_{ij} oranının büyük olacağını umuyoruz. Benzer şekilde, buharla ilgili olan ancak buza bağlı olmayan k sözcükleri için $k = steam$ deyinince oran küçük olmalıdır. Water veya fashion gibi k kelimeleri için hem ice hem de steam’le ya da ikisiyle de ilgili olmayan oranlar bir orana yakın olmalıdır. Tablo 3.1 bu olasılıkları ve büyük bir korpus için oranlarını göstermektedir ve sayılar bu beklentileri doğrulamaktadır. Ham olasılıklarla karşılaştırıldığında, oran ilgili kelimeleri (solid ve gas) alakasız kelimelerden (water ve

fashion) daha iyi ayırt edebiliyor ve aynı zamanda iki ilgili kelime arasında ayrım yapabiliyor.

Yukarıdaki argüman, sözcük vektörü öğrenmesi için uygun başlangıç noktasının, olasılıkların kendisinden ziyade birlikte ortaya çıkma olasılık oranları ile olması gerektiğini ortaya koymaktadır. Denklem 3.3'te P_{ik} / P_{ij} oranının i, j ve k kelimelerine bağlı olduğunu belirterek, en genel model şeklini alır.

Denklem 3.3

$$F(w_i, w_j, w_{\sim k}) = \frac{P_{ik}}{P_{ij}}$$

$w \in \mathbb{R}^d$ kelime vektörleri ve $w_{\sim} \in \mathbb{R}^d$, bu denklemde rolü tartışılacak olan ayrı bağlam kelime vektörleri olduğunda, sağ taraf korpustan çıkarılır ve F henüz belirtilmemiş bazı parametrelere bağlı olabilir. F için olasılıkların sayısı çok fazla, ama birkaç tane daha zorla uygulayarak benzersiz bir seçim yapabiliriz. İlk olarak, F kelimesinin vektör uzayındaki P_{ik} / P_{ij} oranını gösteren bilgiyi kodlamasını isteriz. Vektör uzayları doğal olarak doğrusal yapılar olduğundan, bunu yapmanın en doğal yolu vektör farklılıklarıdır. Bu amaç doğrultusunda, düşüncemizi, sadece iki hedef sözcüğün farkına bağlı olan F işlevleriyle denklem 3.3'ü sınırlandırabiliriz,

Denklem 3.4

$$F(w_i - w_j, w_{\sim k}) = \frac{P_{ik}}{P_{ij}}$$

Daha sonra, denklem 3.4'ün sağ tarafı ise skaler iken F argümanlarının vektördür F , örneğin bir sinir ağı tarafından parametrelendirilen karmaşık bir fonksiyon olarak görünse de yakalamaya çalıştığımız lineer yapıyı engeller. Bu sorunu önlemek için önce argümanların nokta ürününü alabiliriz,

Denklem 3.5

$$F((w_i - w_j)^T, w_{\sim k}) = \frac{P_{ik}}{P_{ij}}$$

Denklem 3.4 sayesinde F 'nin vektör boyutlarını istenmeyen şekillerde karıştırmasını önler. Sadece $w \leftrightarrow w$ not değil, aynı zamanda $X \leftrightarrow X^T$ 'yi de değiştirmeliyiz.

Son modelimiz bu yeniden etiketleme altında deđişmez olmalıdır, ancak Denklem 3.5 deđildir. Ancak, simetri iki adımda onarılabilir.

İlk olarak, F 'nin $(\mathbb{R}, +)$ ve $(\mathbb{R}_{>0}, \times)$ grupları arasında bir homomorfizm olmasını istiyoruz, yani,

Denklem 3.6

$$F((w_i - w_j)^T, w \sim_k) = \frac{F(w_i^T w \sim_k)}{F(w_j^T w \sim_k)}.$$

Denklem 3.5 ile,

Denklem 3.7

$$F(w_i^T, w \sim_k) = P_{ik} = \frac{X_{ik}}{X_i}$$

çözülür.

Denklem 3.6 ile,

Denklem 3.8

$$w_i^T, w \sim_k = \log(P_{ik}) = \log(X_{ik}) - \log X_i$$

sonucu elde edilir.

Bu modelin ana dezavantajı, nadiren veya hiç olmadıklarında bile tüm eşzamanlıları eşit ağırlığında olmasıdır. Bu tür nadir eşli parazitler gürültüdür ve daha sık olanlardan daha az bilgi taşır- yine de sıfır girdiler bile kelime büyüklüğüne ve korpusa bađlı olarak X 'deki verilerin %75-95'ini oluşturur. Bu problemleri ele alan yeni ağırlıklı en küçük kareler regresyon modeli önerilmiştir. Denklem 3.7'ya ilgili işlemler yapıldığında en küçük kareler problemi olarak ve $f(X_{ij})$ ağırlık fonksiyonunu maliyet fonksiyonuna dahil etmek bize Denklem 3.8'i, GloVe modelini verir.

Skip-gram ve CBOW gibi modeller anlamsal bilgileri yakalar, ancak birlikte çalışabilirlik için istatistik kullanmazlar. Matris ayrıştırma yöntemleri bu istatistikleri kullanmasına rağmen, anlamsal ilişkileri yakalayamazlar. Bu modellerin hiçbir anlamı yoktur. Bu sözcük gömme modeli, bu problemi olasılık istatistiklerini kullanarak yeni bir amaç fonksiyonu yaratarak çözmeyi amaçlar. GloVe modeli kelime gömme işlemlerini

elde etmek için aşağıdaki Denklem 3.9'daki işlevi kullanır (Pennington, Socher & Manning, 2014):

Denklem 3.9

$$J_{\emptyset} = \sum_{i,j=1}^v f(X_{ij})(w_i^T + b_i + \bar{b}_j - \log X_{ij})^2.$$

w_i , ana sözcük için vektördür; w_j , bağlam sözcüğü için vektördür ve b_i , b_j , ana sözcük ve bağlam sözcükleri için skaler hazırlıklardır. GloVe modeli temel olarak kelimelerin olasılık oranlarını da hata fonksiyonunun (J) modellenmesi esnasında kullanır. CBOW ve Skip-Gram daki gibi çerçeve gezdirerek çevreleyen kelimelerin belirlenmesi işlemi kelimelerin birlikte kullanım oranları ile güncellenen hata fonksiyonu ile ortadan kaldırılmıştır.

Eğitilmiş GloVe modeli için örnek çıktılar;

```
1 word_vectors.most_similar(positive=["londra", "iran"], negative=["ingiltere"])
[('tahran', 0.6414337754249573),
 ('bağdat', 0.5921595096588135),
 ('irak', 0.5417447090148926),
 ('kahire', 0.5351825952529907),
 ('erbil', 0.5221700072288513),
 ('suriye', 0.5216017365455627),
 ('washington', 0.5198656320571899),
 ('başkenti', 0.5075976252555847),
 ('arap', 0.4841397702693939),
 ('york', 0.47895127534866333)]
```

Şekil 3.6 Ülke-başkent ilişkileri.

```
1 word_vectors.most_similar(positive=["londra","fransa"],negative=["ingiltere"])

[('paris', 0.8408543467521667),
 ('berlin', 0.6817123293876648),
 ('viyana', 0.6571167707443237),
 ('york', 0.6064124703407288),
 ('moskova', 0.5933232307434082),
 ('fransanın', 0.5656879544258118),
 ('new', 0.5631628632545471),
 ('amsterdam', 0.5594193339347839),
 ('başkenti', 0.544691264629364),
 ('italya', 0.5445116758346558)]
```

Şekil 3.7 Ülke-başkent ilişkileri.

Şekil 3.6 ve Şekil 3.7’de görüldüğü gibi, Ülke → Para Birimi İlişkisi: İngiltere → Euro ilişkisi kurulmuşken, Türkiye → Lira ilişkisi kurulamamıştır.

```
1 word_vectors.most_similar(positive=["euro","ingiltere"],negative=["almanya"])

[('sterlin', 0.7306982278823853),
 ('avro', 0.7195351719856262),
 ('euroluk', 0.6628521084785461),
 ('bonservis', 0.6358481049537659),
 ('dolar', 0.6208348870277405),
 ('milyon', 0.6104401350021362),
 ('pound', 0.610379159450531),
 ('manchester', 0.5926746129989624),
 ('premier', 0.5664131045341492),
 ('dolarlık', 0.5661766529083252)]
```

Şekil 3.8 Ülke → Para Birimi İlişkisi: İngiltere → Euro ilişkisi kurulmuştur.

```
1 word_vectors.most_similar(positive=["euro","türkiye"],negative=["fransa"])
```

```
[('milyon', 0.5987488031387329),  
( 'dolar', 0.5458033084869385),  
( 'milyar', 0.5335447192192078),  
( 'türkiyenin', 0.5227399468421936),  
( 'avro', 0.5129513740539551),  
( 'bedeli', 0.5129291415214539),  
( 'euroluk', 0.5119384527206421),  
( 'doları', 0.5096079111099243),  
( 'liralık', 0.5044202208518982),  
( 'lira', 0.5037181973457336)]
```

Şekil 3.9 Ülke → Para Birimi İlişkisi: Türkiye → Lira ilişkisi kurulamamıştır.

Şekil 3.8’de ve Şekil 3.9’da ki çıktılarda görüldüğü gibi ülke-başkent ilişkisi düzgün kurulabilmişken aşağıdaki Şekil 3.10 ve Şekil 3.11’de cinsiyet ilişkisi tam olarak kurulamamıştır. Doktor ve hemşire gibi meslekler cinsiyet değildir. Aynı şekilde kral → kraliçe ilişkisi de kurulamamıştır.

```
1 word_vectors.most_similar(positive=["kral","kadın"],negative=["erkek"])
```

```
[('prens', 0.6287834644317627),  
( 'kralı', 0.6282328963279724),  
( 'kraliçe', 0.6226617097854614),  
( 'kralın', 0.5936248302459717),  
( 'prensi', 0.5546481013298035),  
( 'imparator', 0.4998225271701813),  
( 'kraliyet', 0.468278706073761),  
( 'kraliçesi', 0.4632072448730469),  
( 'savaşı', 0.4551733136177063),  
( 'imparatoru', 0.4514875113964081)]
```

Şekil 3.10 Cinsiyet ilişkisi.

```
1 word_vectors.most_similar(positive=["doktor","kadın"],negative=["erkek"])
```

```
[('hemşire', 0.6753973364830017),  
( 'doktorun', 0.5980468392372131),  
( 'doktorlar', 0.574296236038208),  
( 'doktoru', 0.572459876537323),  
( 'hasta', 0.565844714641571),  
( 'hastane', 0.5554147958755493),  
( 'görevlisi', 0.5371353626251221),  
( 'hekim', 0.5328032374382019),  
( 'görevli', 0.531802773475647),  
( 'hastanede', 0.5315940976142883)]
```

Şekil 3.11 Cinsiyet ilişkisi.

FastText, Word2Vec modelinin bir uzantısı olan başka bir kelime gömme modeli yöntemidir. FastText, son birkaç on yılda doğal dil işleme ve makine öğrenen toplulukların getirdiği en başarılı konseptlerden bazılarını birleştirir. Bunlar, cümleleri kelime torbası ve n-gram torbasıyla temsil etmenin yanı sıra alt kelime bilgisini kullanma ve gizli bir temsil yoluyla sınıflar arasında bilgi paylaşımını içerir. Ayrıca hesaplamayı hızlandırmak için sınıfların dengesiz dağılımından faydalanan hierarchical softmax kullanılır. Bu farklı kavramlar iki farklı görev için kullanılır: verimli metin sınıflandırma ve kelime vektör temsillerini öğrenme.

Derin sinir ağları kısa süre önce metin işlemede çok popüler hale geldi. Bu modeller sınırlı laboratuvar uygulamalarında çok iyi performans sağlarken, çok büyük veri setlerinde kullanımlarını sınırlayan eğitmek ve test etmek yavaş olabilir.

FastText bu sorunu çözmeye yardımcı olur. Çok fazla sayıda kategoriye sahip veri kümelerinde etkili olmak için, farklı kategorilerin bir ağaçta organize edildiği düz bir yapı yerine hiyerarşik bir sınıflandırıcı kullanır (liste yerine ikili ağaç). Bu, metin sınıflandırma cihazlarının eğitim ve test zaman karmaşıklığını, sınıf sayısına göre doğrusaldan logaritmaya indirgemektedir. FastText ayrıca, sınıfları temsil etmek için kullanılan ağacı oluşturmak için Huffman algoritmasını kullanarak sınıfların dengesiz (bazı sınıfların diğerlerinden daha sık görünen) gerçeğinden yararlanır. Bu nedenle, çok sık kategorilere giren ağacın derinliği, nadir olanlara göre daha küçüktür ve bu da hesaplama işleminde verimliliği artırır.

FastText ayrıca, metinde görünen kelimelere karşılık gelen vektörleri toplayarak elde edilen düşük boyutlu bir vektörle bir metni temsil eder. FastText'te, kelimelerin her bir kelimesi ile düşük boyutlu bir vektör ilişkilendirilir. Bu gizli gösterim, farklı kategoriler için tüm sınıflandırıcılar arasında paylaşılarak, bir kategori için öğrenilen kelimelerin diğer kategoriler tarafından kullanılmasına izin verilir. Kelime torbası adı verilen bu tür temsiller, kelime sırasını göz ardı eder. FastText'te, birçok metin sınıflandırma problemi için önemli olan yerel kelime sırasını hesaba katan kelime n-gramlarını temsil etmek için vektörleri de kullanılır.

FastText'in diğer derin öğrenme yapılarına göre daha performanslı olduğu Facebook AI Research yaptığı çalışmaları tarafından gözlemlenmiştir. Facebook AI Research tarafından yapılan deneyde birkaç günlük eğitim süreleri ile eğitilen vektörle üzerinde yapılan araştırmalarda kendini kanıtlamış ve diğer yapılara göre daha yetenekli olduğu saptanmıştır.

Tablo 3.2 FastText ve derin öğrenme temelli yöntemler arasındaki karşılaştırma.

	Yahoo		Amazon full		Amazon polarity	
	Accuracy	Time	Accuracy	Time	Accuracy	Time
char-CNN	71.2	1 day	59.5	5 days	94.5	5 days
VDCNN	73.4	2h	63	7h	95.7	7h
fastText	72.3	5s	60.2	9s	94.6	10s

Tablo 3.2'de FastText'in ve bazı derin öğrenme algoritmaları arasında doğruluk oranları açısından karşılaştırma verilmiştir. Yukarıdaki metinde de bahsedildiği gibi, doğrudan kelimeler için vektörleri öğrenmek yerine, FastText her kelimeyi n-gram karakter olarak görüntüler. Örneğin, $n = 3$ ile "artificial" kelimesini seçersek, buradaki FastText aşamaları $\langle ar, art, rti, tif, ifi, fic, ici, ial, al \rangle$, kelimenin başlangıcı ve bitişidir. Bir kelime veya harften ne kadar olduğunu anlamamızı sağlar. Çok sıklıkla geçmeyen kelimelerin n-gramlarının ortaya çıkma olasılığı zayıf olduğu için bu kelimeler daha doğru bir biçimde temsil edilir. FastText, değiştirilmiş bir Skip gram kaybı fonksiyonuna sahip Word2Vec için önerilen negatif örnekleme Skip gram modelini kullanır. $G_w \subset \{1, \dots, G\}$ a kelimesinde görünen n-gram kümesi olsun, kelimenin puanı n-gramın vektör aşamalarının toplamıyla Denklem 3.10'daki FastText genel denklemi elde edilir (Kilimci, & Akyokus 2018):

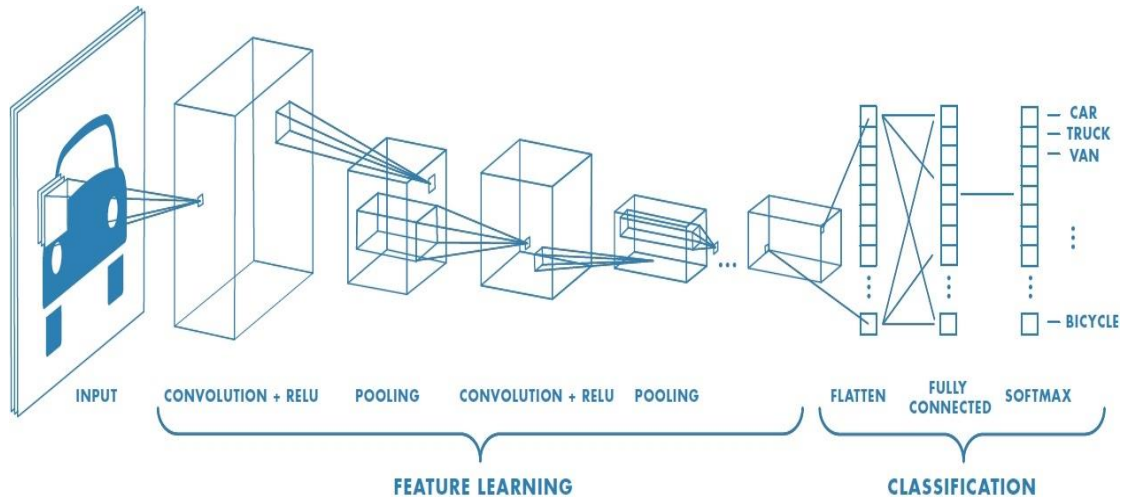
Denklem 3.10

$$s(w, c) = \sum_{g \in G_w} z_g^T v_c$$

Genel olarak FastText sadece metin sınıflandırılması üzerine tasarlanmıştır. FastText diğer metin sınıflandırma yapılarına göre daha hızlı ve performanslıdır. Metin sınıflandırmasının yanı sıra, kelimelerin vektör gösterimlerini öğrenmek için fastText de kullanılabilir. Dillerin morfolojik yapısından faydalanarak İngilizce, Almanca, İspanyolca, Fransızca ve Çekçe gibi çeşitli dillerde çalışmak üzere tasarlanmıştır. Çek gibi morfolojik açıdan zengin diller için çok iyi sonuç veren alt kelime bilgisini dahil etmenin basit ama etkili bir yolunu kullanır, dikkatlice tasarlanmış karakter tasarımı özelliklerinin kelime temsillerini zenginleştirmek için güçlü bir bilgi kaynağı olduğunu gösterir. FastText, popüler kelime gömme modellerinden veya diğer son teknoloji kelime gösterimlerinden daha üstün bir performans sağlayabilir.

3.3. Derin Öğrenme Algoritmaları

Bu çalışmada, Evrişimli Sinir Ağları (CNN'ler), Tekrarlayan Sinir Ağları (RNN) ve Uzun Kısa Süreli Bellek ağları (LSTM) gibi yaygın olarak kullanılan üç derin öğrenme algoritmasına odaklanıyoruz.



Şekil 3.12 CNN mimarisi modeli.

Evrişimli sinir ağları (CNN'ler) bir tür çok katmanlı algılayıcıdır (MLP). Şekil 3.12'de CNN mimarisi modeli gösterilmiştir. Görme merkezindeki hücreler görüntünün

tamamını kapsayacak şekilde alt bölümlere ayrılmıştır. Basit hücreler kenar benzeri özelliklere odaklanırken, karmaşık hücreler tüm alıcıya daha büyük reseptörlerle konsantre olur. İleriye dönük bir sinir ağı olan CNN, hayvanların görsel merkezinden ilham alıyor. Buradaki matematiksel evrişim süreci, bir nöronun uyarıcı alanından uyarıcılara tepkisi olarak düşünülebilir. Tüm CNN katmanları tamamen birbirine bağlanacak şekilde tasarlanmıştır ve her bir evrişim filtresi öğrenilecek özellikleri oluşturur. Bu katlamalı katmanlar, havuzlama yoluyla hem ebatta hem de eğitim süresinde optimizasyon sağlar.

Bu katmanlar;

- Convolutional Layer — Özellikleri belirlemek amacıyla kullanılır
- Non-Linearity Layer — Sisteme lineer olmayanlığın (non-linearity) belirtilmesi için kullanılır.
- Pooling (Downsampling) Layer — Ağırlık sayısını düşürür ve uygunluğu kontrol eder.
- Flattening Layer — Klasik Sinir Ağı için verileri hazır hale getirir.
- Fully-Connected Layer — Sınıflamada işleme alınan Standart Sinir Ağı'dır.

Temel olarak, CNN, sınıflandırma probleminin çözüm yolu için standart Sinir Ağı kullanır ve bilgileri tespit etmek ve bazı özellik durumlarını belirlemek için diğer katmanları kullanır.

Katmanlar ve görevlerini detaylandırarak olursak;

Convolutional katman CNN'nin ana yapı taşıdır. Görevi resmin özelliklerini algılamaktır. Bu katman, resme bazı filtreler uygular. Bunun sebebi, görüntüdeki düşük ve yüksek seviyeli işlevleri ortaya çıkarmaktır. Örneğin, uyguladığı bu filtre kenarları algılanması için bir filtre olabilir. Bu filtreler genellikle boyutları çok fazladır ve piksel değerleri içerirler. (5x5x3) 5 matrisin yükseklik ve genişliğini, 3 matrisin derinliğini temsil eder.

Non-linearity (doğrusal olmayan) katmanı genellikle tüm Convolutional katmanlarından sonra gelir. Görüntüdeki doğrusallık problemi sebebi, tüm katmanlar doğrusal bir fonksiyon olabildiğinden dolayı Sinir Ağı tek bir algı gibi davranır, yani sonuç, çıktılarının lineer kombinasyonu olarak hesaplanabilir. Bu katman aktivasyon katmanı (Activation Layer) olarak adlandırılır çünkü aktivasyon fonksiyonlarından birini

kullanılır. Geçmişte, sigmoid ve tahn gibi doğrusal olmayan fonksiyonlar kullanıldı, ancak Sinir Ağı eğitiminin hızı konusunda en iyi sonucu Rectifier (ReLU) fonksiyonu verdiği için artık bu fonksiyon kullanılmaya başlanmıştır.

ReLU Fonksiyonu $f(x) = \max(0, x)$.

ReLU fonksiyonunun Feature Map'a uygulandığında aşağıdaki gibi bir sonuç üretilir.



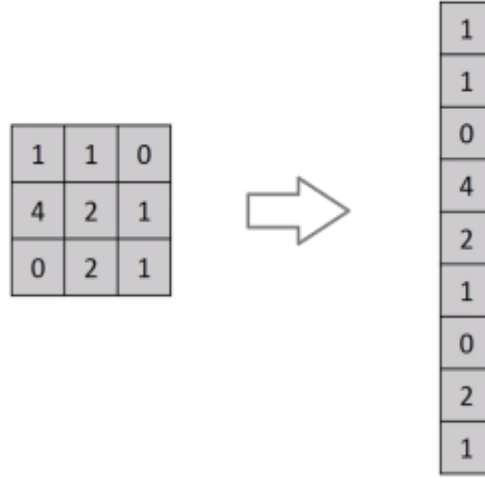
Şekil 3.13 ReLU fonksiyonunun Feature Map'a uygulanması.

Şekil 3.13'te ReLU fonksiyonunun Feature Map'e uygulandığı zaman sonuçların görselini sunmuştur. Feature Map'taki siyah olan değerler negatifi temsil etmektedir. ReLU fonksiyonu işlenmesinin ardından siyah değerler kaldırılır onun yerine 0 gelir.

Pooling katmanı, ConvNet'teki ardışık convolutional katmanları arasına sık sık eklenen bir katmandır. Pooling katmanının işlevi, gösterimin kayma boyutunu ve ağ içindeki parametreleri ve hesaplama sayısını düşürme amacı içindir. Bu katman sayesinde ağdaki uyumsuzluk durumu kontrol altına alınır. Birçok Pooling işlemleri mevcuttur, en çok kullanılan işlem max pooling'dir.

Ayrıca birçok araştırmacı bu katmanı kullanmaktan kaçınır. Pooling katmanından Convolutional katmanında daha büyük Stride (Filtreyi kaydırma işlemi) yönelir.

Flattening katmanının işlevi, son katman olan Fully Connected katmanın girişindeki verileri düzenleyip hazır hale getirmektir. Daha genel olarak, sinir ağı, giriş verilerini tek boyutlu bir diziden getirir. Bu sinir ağındaki veriler ise Convolutional ve Pooling katmanından gelen matrislerin tek boyutlu diziye dönüştürülmüş halidir.

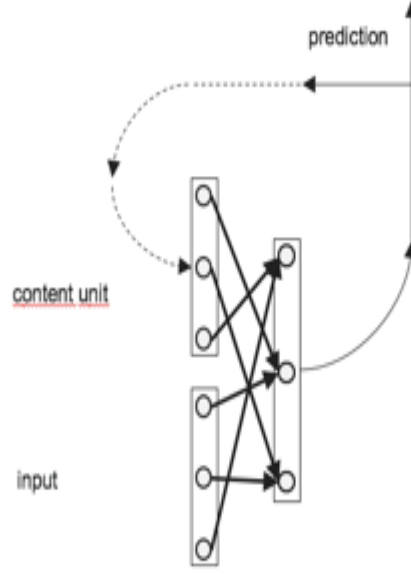


Şekil 3.14 Flattening katmanı mimarisi.

Fully-Connected katmanı, CNN'in son katmanıdır. Yukarıda Şekil 3.14'te Flattening katmanı mimarisi gösterilmiştir. Bu katman en önem durumundaki katmandır. Verileri Flattening işleminden matris olarak alır ve Sinir ağı yoluyla öğrenme işlemini tamamlar (LeCun, Bengio & Hinton, 2015; Schmidhuber, 2015).

Bilgisayarla görme konusunda derin öğrenmenin en önemli kısmı AlexNet'tir. Bu yapıda, bırakma, ağ yapısından rastgele bilgileri elimine eder ve modelin fazla takılma sorununu önler. Bu bir çeşit düzenleme tekniğidir. CNN yapısının aşağıda belirtildiği gibi kullanılmasının avantajları vardır (Krizhevsky, Sutskever, & Hinton, 2012; Voulodimos, vd., 2018):

- Katmanlar daha da derinleşir.
- Hesaplama performansı iyileştirildi (ReLU, bırakma, toplu normalleştirme).
- Ağ katmanları arasındaki bağlantılar arttıkça, backpropagation (geri yayılım) algoritması geliştirildi. Geoffrey Hinton'un 80'li yıllarda backpropagation algoritmasını popüler hale getirdiği bilinmektedir. Bu algoritma şu anda derin öğrenme uygulamalarının hemen hepsinde kullanılmaktadır.



Şekil 3.15 RNN işlem döngüsü.

Şekil 3.15’te RNN işlem döngüsünü gösteren görsel sunulmuştur. Tekrarlayan sinir ağı (RNN), önceki adımdan gelen çıkışın mevcut aşamaya girdi olarak beslendiği bir tür sinir ağıdır (Elman, 1990). RNN, kelimeleri hatırlama gereği nedeniyle önerilmektedir. Bu sorun gizli katman yardımı ile çözülür (Kilimci & Akyokuş 2018; Lipton, Berkowitz & Elkan, 2015). RNN’in en önemli özelliği, bir dizi hakkında bazı bilgileri hatırlayan gizli durumdur. RNN, hesaplananlarla ilgili tüm bilgileri hatırlayan bir “belleğe” sahiptir. RNN, diğer sinir ağlarından farklı olarak, parametrelerin karmaşıklığını azaltır. Çıktıyı üretmek için tüm girdilerde veya gizli katmanlarda aynı görevi gerçekleştirir. Her giriş için aynı parametreleri kullanmak, parametrelerin karmaşıklığını azaltır.

Basit bir RNN, $\mathbf{x} = (x_1, \dots, x_T)$ giriş dizisini kullanan ve bir zaman dilimi kullanan ve gizli vektör dizisini $\mathbf{h} = (h_1, \dots, h_T)$ ve çıkış vektör dizisini hesaplayan dinamik bir sisteme benzeyen bir sinir ağıdır. $\mathbf{y} = (y_1, \dots, y_T)$ aşağıdaki Denklem 3.11 ve denklem 3.12’de $t = 1$ ’den T ’ye kadar tekrarlar:

Denklem 3.11

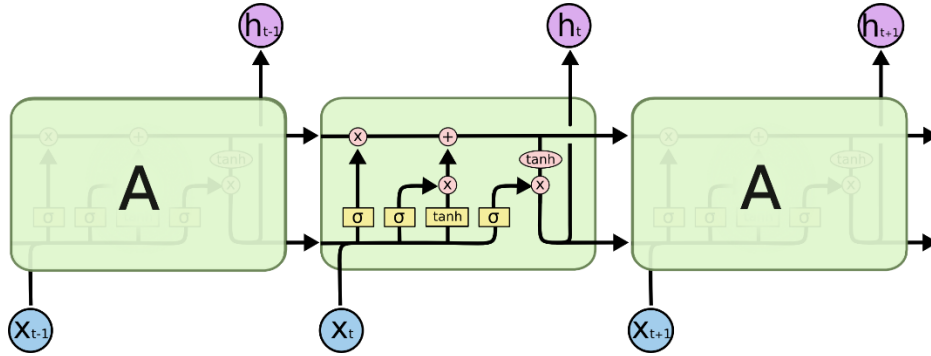
$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

Denklem 3.12

$$y_t = F(W_{hy}h_t + b_y)$$

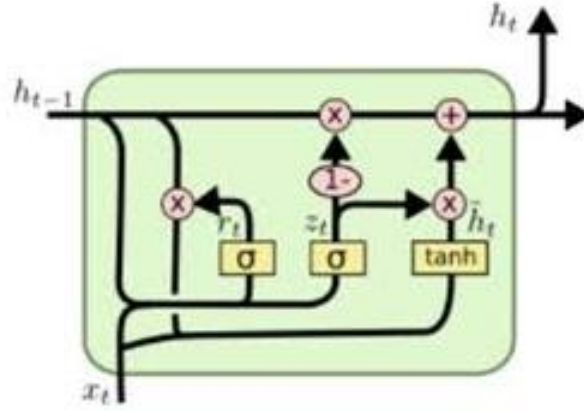
Giriş katmanı için ağırlık matrisleri W_{xh} , W_{hh} ve W_{hy} 'dir. Gizli ve çıktı önyargı vektörleri sırasıyla b_h ve b_y ile temsil edilir. H , durum geçiş işlevidir ve F , çıktı işlevidir.

Uzun Kısa Süreli Bellek ağları genellikle “LSTM'ler” olarak adlandırılır ve uzun vadeli bağımlılıkları öğrenebilen özel bir RNN türüdür. Başlangıç noktası, derin sinir ağlarını eğitirken, geri yayılım algoritmasını kullanarak üssel hata büyümesi sorununa bir çözüm sağlamaktır. Bu sorunun temel nedeni, aktivasyon fonksiyonu tarafından üretilen değerlerin sürekli olarak -1, 1 aralığında olması, böylece bu değerlerin geri yayılma algoritmasına verilmesi ve sifıra çarpılması ile çarpılmasıdır. Bu problemten kaçınmak ve karmaşık yapılar için daha iyi öğrenme algoritmaları tasarlamak için geliştirilen LSTM, uzun vadeli bağımlılıkların ve uzun vadeli bilgilerin hatırlanması gereken problemlerde iyi sonuçlar vermektedir. RNN hücresine ayrıca bellek eşlik eder. Her adımda, öğrenilen hücrelerin hangilerinin atılması gerektiğine ve hangilerinin güncelleneceğine karar verilir. Nöral makine çevirisi için Google tarafından başarıyla kullanılan bir yapıdır. Bu çalışmada, daha önce kullanılan kelimelerin anlamlarını öğrenmek ve bu anlamlara dayalı tahminler üretmek için kullanılmıştır (Kilimci & Akyokuş, 2018; Zhang, Wang & Liu, 2018). Şekil 3.16'da LSTM ağı mimarisi gösterilmiştir.



Şekil 3.16 Uzun-kısa süreli hafıza ağı mimarisi.

Tüm tekrarlayan sinir ağları, bir sinir ağının yinelenen modülleri zincirinin formuna sahiptir. Yalnızca LSTM birimi girdi olarak x_t , $h_{(t-1)}$ ve $c_{(t-1)}$ alır ve aşağıdaki bileşik işlevi kullanarak h_t , c_t üretir. Şekil 3.17'de LSTM'in hafıza ağı katmanının ayrıntılı görseli sunulmuştur.



Şekil 3.17 Uzun- kısa süreli hafıza ağı katmanı mimarisi.

Denklem 3.13

$$i_t = \sigma(W_{xt}x_t + W_{ht}h_{t-1} + W_{ct}c_{t-1} + b_i)$$

Denklem 3.13'te gösterilen i fonksiyonu giriş portal hesabıdır.

Denklem 3.14

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$$

Denklem 3.14'te ifade edilen eşitlik unut portal hesabıdır.

Denklem 3.15

$$c_t = f_t c_{t-1} + i_t \tan h(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

Denklem 3.15'te verilen c fonksiyonu hücre aktivasyon vektörüdür.

Denklem 3.16

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o)$$

Denklem 3.16'da ki o fonksiyonu çıkış portalıdır.

Denklem 3.17

$$h_t = o_t \tan h(c_t)$$

Denklem 3.17'deki h fonksiyonu girdi fonksiyonudur.

σ lojistik sigmoid fonksiyonudur; i , f ve o sırasıyla giriş, unut ve çıkış portallarıdır; c hücre aktivasyon vektörüdür; ve W , ağırlık matrisi olarak sembolize edilir (Kilimci & Akyokuş, 2018).

4. DENEYSEL SONUÇLAR

Bu çalışmada, Parkinson hastalığı ile ilgili kullanıcı yorumlarının duygularını kelime gömme modelleri ve derin öğrenme algoritmaları kullanarak analiz etmek için kapsamlı deneyler yapılmıştır. Her modelin sınıflandırma performansını ve çalışmalarımızın katkısını göstermek için deneylerde bir değerlendirme ölçütü olarak doğruluk kullanılmıştır. Tekrarlanan bekleme yöntemiyle eğitim seti boyutlarını %1, %5, %10, %30, %50, %80 ve %90 olarak değiştirerek deneyler yapıyoruz. Bekletme veri setinde 10 kez uygulanır. Bu yüzdeler, doğruluk yüzdeleriyle karışıklığı önlemek için “ts” ön ekiyle gösterilir (Kilimci, Akyokuş & Omurca, 2016; Schneider, 2004). Bu yaklaşım önceki eğitim çalışmalarına benzerdir, verilerin eğitim için %80’ini ve test için %20’sini kullanır. 300 boyutlu belgeleri göstermek için Google Haberler veri setinden Word2Vec Skip-gram modeli ile önceden eğitilmiş vektörler kullanıyoruz. Ön işleme yöntemleri, kelime gömme modelleri ve derin öğrenme algoritmaları için şu kısaltmalar kullanılmıştır: SWE: Kelime durdurma eliminasyonu, RH: Hashtagleri kaldırma, RU: URL’leri kaldırma, STM: Stemming, AOT: Bunların hepsi, CNN: Evrişimli sinir ağı, RNN: Tekrarlayan sinir ağı, LSTM: Uzun kısa süreli hafıza ağı. Elde edilen en iyi doğruluk sonuçları kalın harflerle belirtilmiştir (Kilimci & Akyokus, 2018; Mikolov vd., 2013).

İlk önce, kelime gömme modellerinin ve derin öğrenme algoritmalarının sınıflandırma performanslarını Tablo 4.1 ve Tablo 4.2’de görüldüğü şekilde birbirleriyle karşılaştırmak için eğitim seti yüzdeleri açısından analiz ediyoruz.

Tablo 4.1 Kelime gömme modellerinin eğitim seti yüzdelere göre sınıflandırma doğrulukları.

Ts	Word2Vec	GloVe	FastText
90	75.20	71.49	71.90
80	76.64	73.46	74.62
50	78.56	75.42	75.33
30	63.52	50.72	51.12
10	58.99	47.89	39.73
5	57.83	42.83	35.87
1	52.61	34.32	19.12

Tablo 4.1 ve 4.2'de, eğitim seti boyutlarının sırasıyla gömme modelleri ve derin öğrenme metodolojileri üzerindeki etkisi incelenmiştir. Tablo 4.1'de en iyi sınıflandırma doğruluğunun Word2Vec, GloVe ve FastText'i kapsayan tüm kelime gömme modelleri için %50 eğitim seti yüzdesinde elde edildiği açıkça görülmektedir. Tüm kelime gömme modelleri için ts50'den ts90'a, sınıflandırma başarısında yaklaşık %2 ila %4'lük bir azalma vardır. Daha düşük eğitim seti boyutlarında, sınıflandırma performansındaki düşüş, ts50 ve ts90 aralığına kıyasla keskin bir şekilde gözlenir. Bu nedenle, %50 antrenman set yüzdesi, deneylerde temel antrenman set büyüklüğü olarak belirlenir. Daha geniş bir bakış açısıyla, Word2Vec ts50'deki diğerlerine kıyasla %78,56 doğrulukla üstün sınıflandırma performansı sergiler. Bunu %75,42 doğrulukla GloVe ve 75,33 doğruluk sonucu FastText takip eder. GloVe ve FastText'in sınıflandırma performansı açısından birbirlerine yakın olduklarını vurgulamak önemlidir, ancak Word2Vec ile karşılaştırıldığında rekabetçidirler. Tablo 4.1'in bir sonucu olarak, Word2Vec'in kullanılması, önerilen sistem için sınıflandırma başarısı açısından avantaj sağlamaktadır.

Tablo 4.2 Derin öğrenme algoritmalarının eğitim seti yüzdelere göre sınıflandırma doğrulukları.

Ts	CNN	RNN	LSTM
90	75.54	76.47	79.46
80	77.98	78.59	80.68
50	78.39	79.36	82.33
30	74.21	74.85	78.34
10	72.46	73.92	75.20
5	56.53	58.68	68.44
1	44.87	42.34	45.89

Tablo 4.2'de, derin öğrenme algoritmalarının sınıflandırma doğrulukları, eğitim seti yüzdesi olarak değerlendirilmiştir. LSTM, tüm eğitim seti seviyelerinde en iyi performans gösteren derin öğrenme modelidir. CNN, RNN'ye kıyasla %1 daha iyi bir başarıya sahiptir, ancak ikisi de birbirlerine çok yakın sınıflandırma performansları sergiler. CNN ve RNN arasındaki sınıflandırma başarısındaki küçük farklılıklar nedeniyle, bir tanesinin diğer derin öğrenme algoritmasından daha iyi olmadığı açıkça görülmektedir. Dahası, LSTM ts50'de diğer derin öğrenme modellerini %3-%4 daha iyi bir doğrulukla geride bırakıyor. Diğer eğitim seti yüzdelere, LSTM ve RNN arasındaki doğruluk farkı, %2 ile %10 arasında değişirken, LSTM, doğruluk yüzdesi bakımından CNN'ye kıyasla minimum %1 ve maksimum %12 artış gösterir. Kelime gömme modellerine benzer şekilde, tüm derin öğrenme teknikleri için en yüksek doğruluk oranı ts50'de elde edilir. Özet olarak, derin öğrenme algoritmalarının sınıflandırma başarısı şu şekilde sıralanır: tüm eğitim seti boyutlarında LSTM > RNN > CNN. Tablo 4.1 ve Tablo 4.2'nin bir sonucu olarak, %50 eğitim seti büyüklüğü, ts50'de gözlenen üstün sınıflandırma performansı nedeniyle tüm deneylerde temel olarak ayarlanmıştır.

Tablo 4.3 %50 eğitim setinde ön işleme yöntemlerinin kelime gömme modelleri üzerindeki etkisi.

Preprocessing methods	Word2Vec	GloVe	FastText
SWE	81.23	80.87	78.98
RH	85.75	81.51	80.72
RU	87.67	82.25	81.86
STM	78.36	76.79	75.69
AOT	82.48	80.59	78.83

İkinci olarak, doğruluk sonuçları açısından en iyi eğitim/test dağılımını belirledikten sonra, ön işleme yöntemlerinin hem kelime gömme modelleri hem de derin öğrenme algoritmaları üzerindeki etkisini Tablo 4.3 ve 4.4'te gösterilmektedir.

Tablo 4.3'te, ön işleme yöntemlerinin %50 eğitim setinde kelime gömme modelleri üzerindeki etkisi yukarıdaki gibi gösterilmiştir. RU, önerilen sistemin başarısını artırarak, %85,75 doğruluk başarısı ile rekabet eden RH ile karşılaştırıldığında %87,67 doğrulukla üstün sınıflandırma performansı sergiler. Bunu %82,48 ile AOT, %81,23 ile SWE ve %78,36 ile STM izlemektedir. Kelime gömme modelleri arasında, Word2Vec ön işleme yöntemi RU olarak ayarlandığında sınıflandırma başarısını %87,67 doğrulukla dikkat çekici bir şekilde göstermektedir. Ayrıca, Word2Vec diğer kelime gömme modellerinden daha iyi performans gösterirken, FastText kelime gömme modellerinde %81,86 ile en düşük sınıflandırma performansına sahiptir. Ayrıca, Word2Vec FastText'in başarısı göz önüne alındığında yaklaşık %6 iyileşme sağlarken, GloVe FastText'in sınıflandırma performansına kıyasla yaklaşık %1 artış sağlar.

Tablo 4.4 Ön hazırlık yöntemlerinin %50 eğitim setinde derin öğrenme algoritmaları üzerindeki etkisi.

Preprocessing methods	CNN	RNN	LSTM
SWE	80.25	81.75	83.81
RH	81.56	84.05	90.57
RU	83.13	86.82	92.50
STM	78.31	78.75	79.64
AOT	80.83	85.56	86.39

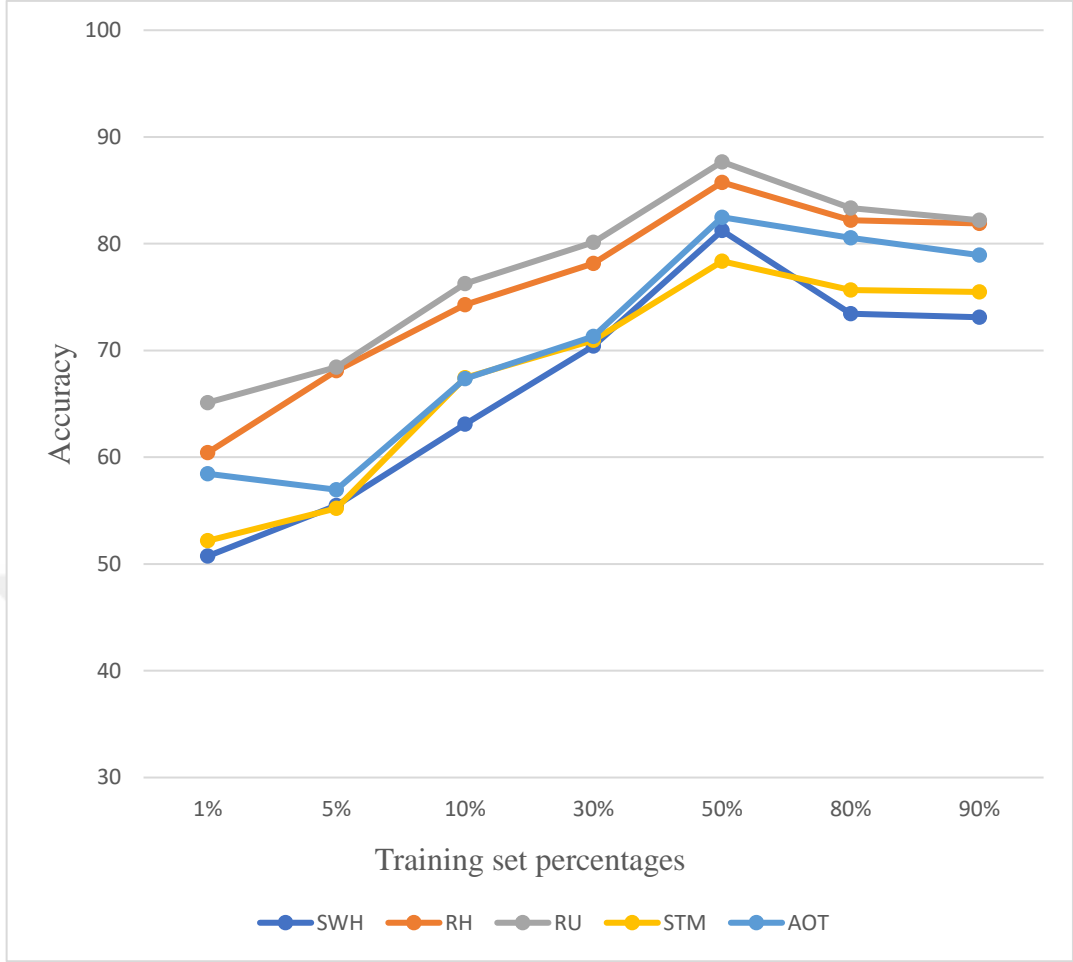
Tablo 4.4'te, RU ve RH, tüm derin öğrenme algoritmaları göz önüne alındığında genellikle en iyi iki ön işleme modelidir. LSTM ve RU'nun kombinasyonu, %92,50 ile dikkate değer bir sınıflandırma başarısı gösterirken, RU'nun CNN üzerindeki etkisi %83,13 doğruluk sonucu olarak değerlendirilir. Ek olarak, LSTM, RNN'ye kıyasla sınıflandırmaya kabaca %6 katkı sağlar. Sonuç olarak, LSTM'nin tüm ön işleme yöntemleri için derin öğrenme sınıflandırıcıları arasında üstün sınıflandırma performansını gösterdiği açıkça görülmektedir. Her derin öğrenme modeli için sınıflandırma başarısı şöyle sıralanır: LSTM> RNN> CNN. %92,50 doğruluk sonucuna sahip RU, LSTM derin bir öğrenme modeli olarak belirlendiğinde önerilen sistemin sınıflandırma başarısını artırır. Sistemin sınıflandırma başarısı, LSTM ve ön işleme modellerinin kombinasyonu ile değiştirilebilir. RU'nun üstün başarısına ek olarak, STM sınıflandırmadaki en kötü performansı göstermektedir. Şaşırtıcı bir şekilde, AOT, en iyi ikinci ön işleme modelidir ve RNN ve RU kombinasyonu göz önüne alındığında %85,56 ile rekabetçidir. Tablo 4.4'ün sonucu olarak, ön işleme modellerinin sınıflandırma doğruluklarının sırası genellikle şöyledir: RU> RH> AOT> SWE> STM. Bu çalışmada, RU ve RH konsolidasyonu, sınıflandırmadaki olağanüstü etkisinden dolayı önerilen sistemin ön işleme modelleri olarak seçilmiştir.

Üçüncü bir adım olarak, en iyi sınıflandırma sonuçlarını, 1 ila 90 arasındaki eğitim seti yüzdelelerini değiştirerek ve Şekil 4.1'de görüldüğü gibi ön işleme yöntemleri olarak RH + RU kullanarak analiz ediyoruz.



Şekil 4.1 RH + RU bir ön işleme yöntemi olarak ayarlandığında her kelime gömme ve derin öğrenme modelinin eğitim seti yüzdeleri açısından sınıflandırma performansları.

Her kelime gömme ve derin öğrenme modelinin sınıflandırma performansları Şekil 4.1’de incelenmiştir. Ts1’den ts50’ye, tüm kelime gömme ve derin öğrenme modelleri için sürekli bir artış gözlenmektedir. FastText, zayıf sınıflandırma performansını sergilerken, LSTM’nin Parkinson hastalığına bağlı tweet’lerin duyarlılık analizini belirlemek için en iyi model olduğu açıkça görülmektedir. Daha yüksek eğitim seti seviyelerinde, her bir modelin hassasiyet sonuçları azalır. Şaşırtıcı bir şekilde, RNN ve CNN’nin sınıflandırma başarıları hem ts90 hem de ts30’da birbirine çok benzer. Bu, eğitim setinin büyümesinin sınıflandırma performansını beklendiği şekilde artıramayacağı anlamına gelir. Şekil 4.1’in bir sonucu olarak, ön işleme aşamasında RU + RH kombinasyonunun kullanılması, Tablo 4.1 ve Tablo 4.2’ye kıyasla ts50’de yaklaşık %2 artış sağlar.

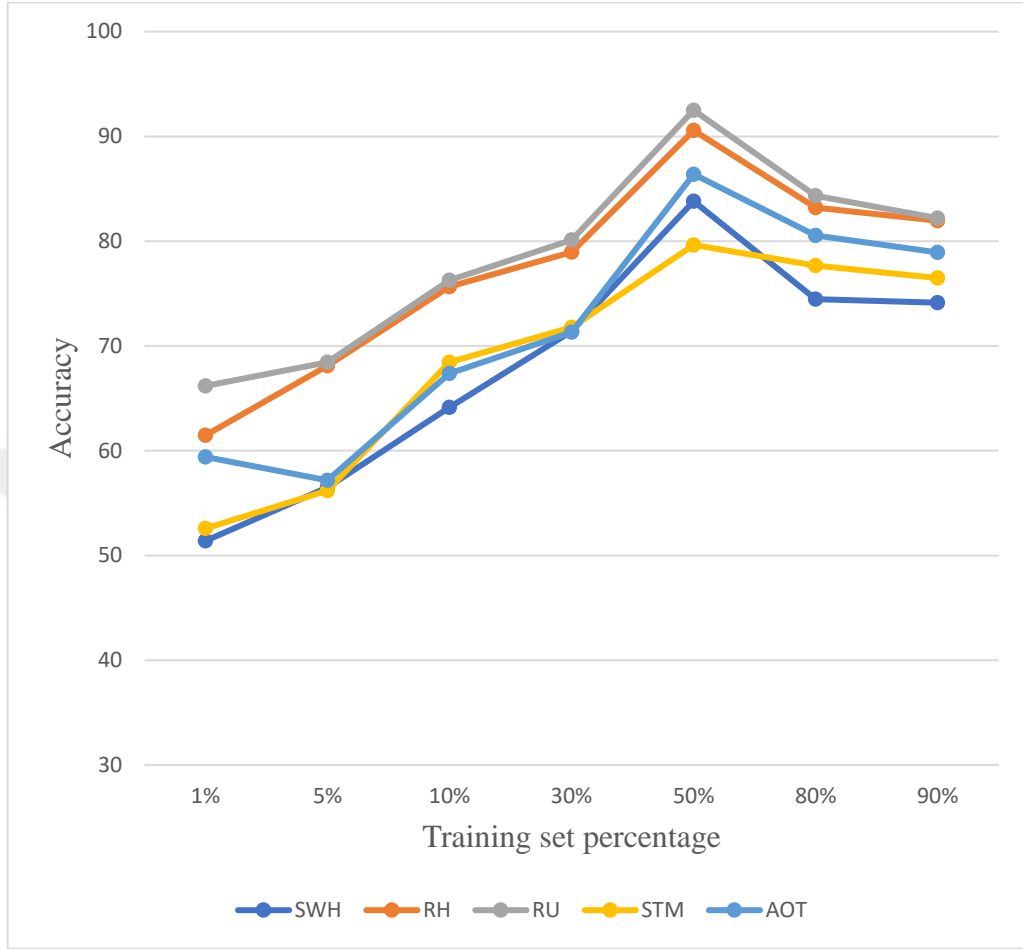


Şekil 4.2 Word2Vec sabitlendiğinde ön işleme yöntemlerinin eğitim yüzdeleri bakımından sınıflandırma doğrulukları.

Ön işleme yöntemlerinin doğruluk sonuçları üzerindeki etkisi, Şekil 4.2 ve Şekil 4.3'teki eğitim seti büyüklükleri açısından analiz edilirken, ön işleme modellerinin sınıflandırma performansları Tablo 4.3 ve Tablo 4.4'deki ts50'de verilmiştir. Ön işleme yöntemlerinin sınıflandırma doğrulukları, Eğitim seti yüzdeleri, Word2Vec'in Şekil 4.2'de gösterilen kelime gömme modeli olarak sabitlenmesiyle analiz edilir.

Tüm ön işleme yöntemleri arasında, RH ve RU yöntemleri, ts50'de elde edilen en yüksek hassasiyetli sonuçları verir. Ts80 ve ts90'da RH ve RU diğer ön işleme yöntemlerinden biraz daha yüksek hassasiyete sahiptir, ancak bu istatistiksel olarak anlamlı değildir. Ts50'den başlayarak, RH ve RU doğruluğu ts50'de yaklaşık %2, ts30'da yaklaşık %20, ts10'da yaklaşık %20 ve ts5'te yaklaşık %22 ile diğer ön işleme yöntemlerinden daha iyi performans gösterir. Ön işleme yöntemlerinin eğitim seti yüzdeleri cinsinden sınıflandırma doğrulukları, LSTM'nin Şekil 4.3'te gösterilen derin öğrenme algoritması olarak sabitlenmesiyle analiz edilmiştir. Şekil 4.3'de LSTM derin

bir öğrenme modeli olarak kabul edildiğinde ön işleme yöntemlerinin sırasıyla % doğruluk değerleri verilmiştir.



Şekil 4.3 Ön işleme yöntemlerinin LSTM eğitim yüzdeleri cinsinden sınıflandırma doğrulukları.

Özellikle sosyal medya platformlarından, kelime yerleştirme modellerinden ve derin öğrenme yaklaşımlarından gelen metin verileriyle benzer veri setlerine sahip eserlerin bulunmamasından dolayı sonuçlarımızın performansını diğer çalışmalarla karşılaştırmak zor. Buna rağmen, sınıflandırma modellerini, yüzde olarak doğruluk sonuçlarını ve Tablo 4.5'deki Parkinson hastalığına yoğunlaşan son teknoloji çalışmaların veri kümelerini ele alıyoruz. Ayrıca, önerilen sistemin deney sonuçlarını durumla karşılaştırıyoruz. Farklı veri setlerinin kullanımı göz önünde bulundurularak sınıflandırma doğruluğu açısından son teknoloji çalışmalarıdır. Tablo 4.5'ten açıkça görülüyor ki, Parkinson hastalığına ilişkin son teknolojiye ait çalışmalar, önerilen çalışma dışında vokal kayıt verilerine odaklanıyor. Buna rağmen, önerilen modelimiz, LSTM modelinin kullanımı ve RU + RH yöntemlerinin kombinasyonu ile %93,63 oranında doğruluk sonucu ortaya koymaktadır. Eskidere (2012), destek vektör makinesi temelli

özyinelemeli özelliklerin kaldırılması ile %95,13 doğruluk değeri elde eder. Çalışmamızın sınıflandırma performansı, metinsel verilerin kullanımı, derin öğrenme modelleri ve kelime gömme modellerinin son teknoloji çalışmalara kıyasla göz önüne alındığında oldukça rekabetçi olarak değerlendirilebilir.



Tablo 4.5 Önerilen modelimizin Parkinson hastalığına yönelik son teknoloji çalışmalarla karşılaştırılması.

Çalışmalar	Modeller	Doğruluk Oranı	Veri seti
Ene, M. (2008)	Olasılıksal Sinir Ağı (PNN) (%70 eğitim, %30 test)	80,92	Ses kaydı için sayısal veriler
Das, R. (2010)	Yapay Sinir Ağları (NN) (%65 eğitim, %33 test)	92,9	Ses kaydı için sayısal veriler
Çağlar, M. F., Çetişli, B., & Toprak, İ. B. (2010)	The linguistic strong adaptive nerve-fuzzy sınıflandırıcısı (ANFC + LH) (%50 eğitim, %50 test)	94,72	Ses kaydı için sayısal veriler
Luukka, P. (2011)	benzerlik sınıflandırıcısı + bulanık entropi (SC + FE) (%50 eğitim, %50 test)	85,03	Ses kaydı için sayısal veriler
Sakar, C. O., & Kursun, O. (2010)	Destek vektör makinesi (SVM) + maksimum alaka düzeyi minimum yedeklilik (MRMR) (önyükleme yeniden örnekleme - BR)	92,75	Ses kaydı için sayısal veriler
Tsanas, A., Little, M. A., McSharry, P. E., & Ramig, L. O. (2009)	SVM+ önceden işlenmiş altkümelerin olası her bir kombinasyonu	91,40	Ses kaydı için sayısal veriler
Eskidere, Ö. (2012)	SWM metodları - özyinelemeli özellik ortadan kaldırılması (SVM-RFE) (BR)	95,13	Ses kaydı için sayısal veriler
Bizim çalışmamız	Ön işleme yöntemlerinin LSTM + RU ve RH kombinasyonu (%50 eğitim, %50 test)	93,63	Twitter'dan gelen metinsel veri kümesi

Bildiğimiz kadarıyla, bu Parkinson hastalığını sosyal medya platformlarında analiz etmeye yönelik ilk girişim. Ayrıca, bu çalışmanın diğer bir yeniliği, Tablo 4.5'te belirtilen diğer çalışmaların aksine, Parkinson hastalığının tespiti için kelime gömme modelleri ve derin öğrenme algoritmalarının kullanılmasıdır.



5. SONUÇ

Bu çalışmada, Parkinson hastalığına ilişkin son çalışmaların aksine, kullanıcıların sosyal medya platformlarındaki duyarlılıklarını analiz ederek Parkinson hastalığının tespitine odaklanıyoruz. Kullanıcıların Parkinson hastalığı hakkındaki görüşlerini anlamak ve analiz etmek için derin öğrenme yaklaşımları ve kelime gömme modelleri değerlendirilmektedir. Bu amaçla Konvolüsyonel Sinir Ağları (CNN), Tekrarlayan Sinir Ağları (RNN) ve Uzun Kısa Süreli Bellek Ağları (LSTM) derin öğrenme algoritmaları olarak kullanılmakta ve Word2Vec, Eldiven ve FastText kelime gömme modelleri olarak değerlendirilmektedir. Dahası, önerilen modelin sınıflandırma performansını arttırmak için ön işleme yöntemleri olarak kaynak kelimelerin kaldırılması (SWE), hashtag'lerin kaldırılması (RH), URL'lerin kaldırılması (RU), kaynak yaratma (STM) üzerinde de duruyoruz. Ayrıca, eğitim seti boyutlarının etkisi de sınıflandırma performanslarına göre incelenmiştir. Sonuç olarak, LSTM ile RU + RH kombinasyonunun kullanılması, kullanıcıların Parkinson hastalığına karşı duyarlılıklarını belirlemek için ts50'de en iyi sınıflandırma başarısını gerçekleştirir.

Bildiğimiz kadarıyla, bu sözcük gömme modelleri ve derin öğrenme algoritmaları kullanarak Parkinson hastalığının algısını analiz eden ilk çalışmadır. Bu şekilde, kullanıcıların Parkinson hastalığına ilişkin duyarlılık analizi, Parkinson hastaları ve yakınları için hem erken tanı hem de tedavi sürecinde yeni bir bakış açısı sağlayabilir. Ayrıca, bu çalışma, Parkinson hastalığı ile ilgili tweet'lerden duyulan analizlerin, Parkinson hastalarının bakım sürecinde nelere ihtiyaç duyduğunu ve Parkinson hastalarının ve yakınlarının sağlık sorunları hakkında birbirleriyle nasıl iletişim kurduğunu ortaya çıkaran değerli bir sağlık hizmeti kaynağı olduğunu göstermektedir. Gelecekteki bir çalışma olarak, Parkinson hastalığının erken tespitini güçlendirerek hem metinsel hem de sesli veri kayıt verilerini içeren bir hibrit modeli geliştirmeyi planlıyoruz.

KAYNAKÇA

Beykikhoshk, A., Arandjelović, O., Phung, D., Venkatesh, S., & Caelli, T. (2014, August). Data-mining Twitter and the autism spectrum disorder: a pilot study. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)* (pp. 349-356). IEEE.

Beykikhoshk, A., Arandjelović, O., Phung, D., & Venkatesh, S. (2015, August). Overcoming data scarcity of Twitter: using tweets as bootstrap with application to autism-related topic content analysis. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015* (pp. 1354-1361). ACM.

Bruns, A., Kornstadt, A., & Wichmann, D. (2009). Web application tests with selenium. *IEEE software*, 26(5), 88-91.

Chew, C., & Eysenbach, G. (2010). Pandemics in the age of Twitter: content analysis of Tweets during the 2009 H1N1 outbreak. *PloS one*, 5(11), e14118.

ÇAĞLAR, M. F., ÇETİŞLİ, B., & TOPRAK, İ. B. (2010). Automatic recognition of Parkinson's disease from sustained phonation tests using ANN and adaptive neuro-fuzzy classifier. *Mühendislik Bilimleri ve Tasarım Dergisi*, 1(2), 59-64.

Das, R. (2010). A comparison of multiple classification methods for diagnosis of Parkinson disease. *Expert Systems with Applications*, 37(2), 1568-1572.

De Rijk, M. D., Tzourio, C., Breteler, M. M., Dartigues, J. F., Amaducci, L., Lopez-Pousa, S., ... & Rocca, W. D. (1997). Prevalence of parkinsonism and Parkinson's disease in Europe: the EUROPARKINSON Collaborative Study. European Community Concerted Action on the Epidemiology of Parkinson's disease. *Journal of Neurology, Neurosurgery & Psychiatry*, 62(1), 10-15.

Eckler, P., Worsowicz, G., & Rayburn, J. (2010). Social media and healthcare: an overview. *PM&R*, 2(11), 1046-1050.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2), 179-211.

Ene, M. (2008). Neural network-based approach to discriminate healthy people from those with Parkinson's disease. *Annals of the University of Craiova-Mathematics and Computer Science Series*, 35, 112-116.

Eskidere, Ö. (2012). A Comparison Of Feature Selection Methods For Diagnosis Of Parkinson's Disease From Vocal Measurements. *Sigma*, 30, 402-414.

Fahn, S. (2003). Description of Parkinson's disease as a clinical syndrome. *ANNALS-NEW YORK ACADEMY OF SCIENCES*, 991, 1-14.

Graves, A., Mohamed, A. R., & Hinton, G. (2013, May). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 6645-6649). IEEE.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016). Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Kihel, B. K., & Benyettou, M. (2011). Parkinson's disease recognition using artificial immune system. *Journal of Software Engineering and Applications*, 4(07), 391.

Kilimci, Z. H., Akyokuş, S., & Omurca, S. I. (2016, August). The effectiveness of homogenous ensemble classifiers for Turkish and English texts. In *2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)* (pp. 1-7). IEEE.

Kilimci, Z. H., & Akyokuş, S. (2018). Deep Learning-and Word Embedding-Based Heterogeneous Classifier Ensembles for Text Classification. *Complexity*, 2018.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Le, Q., & Mikolov, T. (2014, January). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188-1196).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

Liu, P., Qiu, X., & Huang, X. (2017). Adversarial multi-task learning for text classification. *arXiv preprint arXiv:1704.05742*.

Loria, S. (2018). *textblob Documentation* (pp. 1-73). Technical report.

Luukka, P. (2011). Feature selection using fuzzy entropy measures with similarity classifier. *Expert Systems with Applications*, 38(4), 4600-4607.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Neiger, B. L., Thackeray, R., Burton, S. H., Thackeray, C. R., & Reese, J. H. (2013). Use of twitter among local health departments: an analysis of information sharing, engagement, and action. *Journal of medical Internet research*, 15(8), e177.

Pennington, J., Socher, R., & Manning, C. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

Pervan, N., & Keleş, Y. *Derin öğrenme yaklaşımları kullanarak Türkçe metinlerden anlamsal çıkarım yapma* (Doctoral dissertation).

Polat, K. (2012). Classification of Parkinson's disease using feature weighting method on the basis of fuzzy C-means clustering. *International Journal of Systems Science*, 43(4), 597-609.

Prashanth, R., & Roy, S. D. (2018). Early detection of Parkinson's disease through patient questionnaire and predictive modelling. *International journal of medical informatics*, 119, 75-87.

Prashanth, R., & Roy, S. D. (2018). Novel and improved stage estimation in Parkinson's disease using clinical scales and machine learning. *Neurocomputing*, 305, 78-103.

Prieto, V. M., Matos, S., Alvarez, M., Cacheda, F., & Oliveira, J. L. (2014). Twitter: a good place to detect health conditions. *PloS one*, 9(1), e86191.

Sakar, C. O., & Kursun, O. (2010). Telediagnosis of Parkinson's disease using measurements of dysphonia. *Journal of medical systems*, 34(4), 591-599.

Schneider, K. M. (2004, October). On word frequency information and negative evidence in Naive Bayes text classification. In *International Conference on Natural Language Processing (in Spain)* (pp. 474-485).

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.

Sinnenberg, L., Bittenheim, A. M., Padrez, K., Mancheno, C., Ungar, L., & Merchant, R. M. (2017). Twitter as a tool for health research: a systematic review. *American journal of public health, 107*(1), e1-e8.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tsanas, A., Little, M. A., McSharry, P. E., & Ramig, L. O. (2009). Accurate telemonitoring of Parkinson's disease progression by noninvasive speech tests. *IEEE transactions on Biomedical Engineering, 57*(4), 884-893.

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience, 2018*.

Zhou, C., Sun, C., Liu, Z., & Lau, F. (2015). A C-LSTM neural network for text classification. *arXiv preprint arXiv:1511.08630*.

Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8*(4), e1253.

ÖZGEÇMİŞ

Feyza Çevik 1995 yılında Antalya’da doğdu. 2018 yılında Doğuş Üniversitesi Matematik bölümünden ve Bilgisayar Mühendisliği yan dal programından mezun oldu. 2018 yılında Doğuş Üniversitesi yüksek lisans eğitimine başladı. Haziran 2018’den itibaren IQBender Teknoloji Şirketi’nde İş Zekâsı Uzmanı olarak görev yapmaktadır. IQBender Teknoloji Şirketi’ndeki çalışma alanları: Talepler ve iş süreçlerinin araştırılması ve geliştirilmesi, teknik analiz ile tüm toplama işlemlerinin gerçekleştirilmesi (Sistem Gereksinimleri), yazılım eksiklikleri ve yeni talepler için ilgili teknik ve teknik olmayan belgelerin hazırlanmasıdır.

