T.C.
SÜLEYMAN DEMİREL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME ALGORİTMALARI İLE TÜRKÇE DİLİNDE SAHTE HABER TESPİTİ

Süleyman Gökhan TAŞKIN

Danışman
Prof. Dr. Ecir Uğur KÜÇÜKSİLLE

II. Danışman
Dr. Öğr. Üyesi Kamil TOPAL

DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
ISPARTA - 2020

TEZ ONAYI

Süleyman Gökhan TAŞKIN tarafından hazırlanan "**Derin Öğrenme Algoritmaları ile Türkçe Dilinde Sahte Haber Tespiti**" adlı tez çalışması aşağıdaki jüri üyeleri önünde Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü **Bilgisayar Mühendisliği Anabilim Dalı**'nda **DOKTORA TEZİ** olarak başarı ile savunulmuştur.

Danışman	**Prof. Dr. Ecir Uğur KÜÇÜKSİLLE** Süleyman Demirel Üniversitesi	.....
Jüri Üyesi	**Doç. Dr. Utku KÖSE** Süleyman Demirel Üniversitesi	.....
Jüri Üyesi	**Dr. Öğr. Üyesi Turgay AYDOĞAN** Süleyman Demirel Üniversitesi	.....
Jüri Üyesi	**Dr. Öğr. Üyesi Tuna GÖKSU** Isparta Uygulamalı Bilimler Üniversitesi	.....
Jüri Üyesi	**Dr. Öğr. Üyesi Mehmet Ali YALÇINKAYA** Kırşehir Ahi Evran Üniversitesi	.....

Enstitü Müdürü	**Doç. Dr. Şule Sultan UĞUR**	.....

TAAHHÜTNAME

Bu tezin akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını beyan ederim.

Süleyman Gökhan TAŞKIN

İÇİNDEKİLER

ÖZET

Doktora Tezi

DERİN ÖĞRENME ALGORİTMALARI İLE TÜRKÇE DİLİNDE SAHTE HABER TESPİTİ

Süleyman Gökhan TAŞKIN

Süleyman Demirel Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Ecir Uğur KÜÇÜKSİLLE

II. Danışman: Dr. Öğr. Üyesi Kamil TOPAL

Son yıllarda internet kullanımının artmasıyla insanların bilgi ve haber alma kaynakları da değişmiştir. Radyo, televizyon, gazete ve dergi gibi geleneksel medya araçları yerine sosyal medya araçlarının kullanımı giderek artmaktadır. Geleneksel medyada haberler belirli bir kaynak tarafından gönderilirken, sosyal medyada her kullanıcı bir haber kaynağı olabilmektedir. Bu durum habere erişimi oldukça hızlandırmakta fakat sosyal medyadaki haberlerin bir süzgeçten geçirilmeden paylaşılması, sahte haberlerin büyük bir hızla yayılmasına neden olmaktadır.

Çoğu sosyal medya platformunda, sahte haber tespiti uzmanlar tarafından yapılmaktadır. Yoğun paylaşım trafiği bulunan sosyal medya platformlarında, bu çalışma mantığı ile kısa sürede sahte haber tespiti mümkün olamamaktadır. Bu da sahte haberin kısa sürede çok kişi tarafından paylaşılmasına sebep olmaktadır. Bu nedenle, yarı otomatik ve otomatik sahte haber tespiti sistemleri, uzmanların görev yaptığı sistemlere göre daha kısa sürede sahte haber tespitini sağlayabilmektedir. Sahte haberleri kısa sürede tespit edebilmek için otomatik tespit sistemlerinin geliştirilmesi gerekmektedir.

Bu tez çalışmasında, denetimli öğrenme ve denetimsiz öğrenme kullanılarak, Twitter sosyal ağında sahte haber tespiti yapılmış ve sonuçları incelenmiştir. Ayrıca kullanıcıların takipçi etkileşimleri, sosyal ağ analizi yöntemleriyle incelenmiştir. Denetimli öğrenme algoritmalarında 0,86 ve Derin öğrenme algoritmalarında 0,80 F1-metrik değeriyle başarılı sonuçlar alınmıştır. Denetimsiz öğrenme algoritmalarının F1-metrik değeri ise 0,72'de kalmıştır.

Anahtar Kelimeler: Sahte Haber Tespiti, Derin Öğrenme, Makine Öğrenmesi, Yapay Zeka.

2020, 120 sayfa

ABSTRACT

Ph.D. Thesis

DETECTING FAKE NEWS IN TURKISH WITH DEEP LEARNING ALGORITHMS

Süleyman Gökhan TAŞKIN

Süleyman Demirel University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Prof. Dr. Ecir Uğur KÜÇÜKSİLLE

Co-Supervisor: Asst. Prof. Dr. Kamil TOPAL

In recent years, with the increase of internet usage, people's sources of information and information have also changed. Instead of traditional media such as radio, television, newspapers and magazines, the use of social media tools is increasing. In traditional media, news is sent by a certain source, whereas in social media, each user can be a news source. This situation accelerates access to the news, but sharing the news on social media without a filter causes fake news to spread rapidly.

On most social media platforms, fake news detection is done by experts. In social media platforms with intensive sharing traffic, it is not possible to detect fake news in a short time with such expert systems. This causes fake news to be shared by many people in a short time. Therefore, semi-automatic and automatic fake news detection systems can detect fake news in a shorter time than expert systems. Automatic detection systems should be developed in order to detect fake news in a short time.

In this thesis, fake news was detected in Twitter social network by using supervised learning, unsupervised learning and deep learning algorithms and the results were examined. In addition, the interactions of users with their followers were analyzed using social network analysis methods. Successful results were obtained with 0,86 F1-score in supervised learning algorithms and 0,80 F1-score in deep learning algorithms. The F1-score of unsupervised learning algorithms remained at 0,72.

Keywords: Fake News Detection, Deep Learning, Machine Learning, Artificial Intelligence.

2020, 120 pages

TEŞEKKÜR

Bu araştırma için değerli görüş ve katkılarıyla beni yönlendiren, karşılaştığım zorlukları bilgi ve tecrübesi ile aşmamda yardımcı olan ve bilimsel çalışma disiplinini bana kazandıran değerli Danışman Hocam Prof. Dr. Ecir Uğur KÜÇÜKSİLLE'ye ve her zaman kıymetli vaktini bana ayıran ikinci Danışman Hocam Dr. Öğr. Üyesi Kamil TOPAL'a teşekkürlerimi sunarım.

Her dönemde olduğu gibi, tezimin her aşamasında beni yalnız bırakmayan başta eşim Gamze BAĞÇECİ TAŞKIN olmak üzere tüm aileme sonsuz sevgi ve saygılarımı sunarım.

Süleyman Gökhan TAŞKIN

ISPARTA, 2020

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

SİMGELER VE KISALTMALAR DİZİNİ

ASP.NET	Active Server Pages .NET
DDİ	Doğal Dil İşleme
GRU	Gated Recurrent Unit
HKKA	Hipermetin Kaynaklı Konu Araması
KNN	K-Nearest Neighbor
LDA	Linear Discriminant Analysis
LSTM	Long-Short Term Memory
ML	Machine Learning
MVC	Model-View-Controller
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NMF	Non-negative MAtrix Factorization
RF	Random Forest
RNN	Recurrent Neural Network
SQL	Structured Query Language
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency

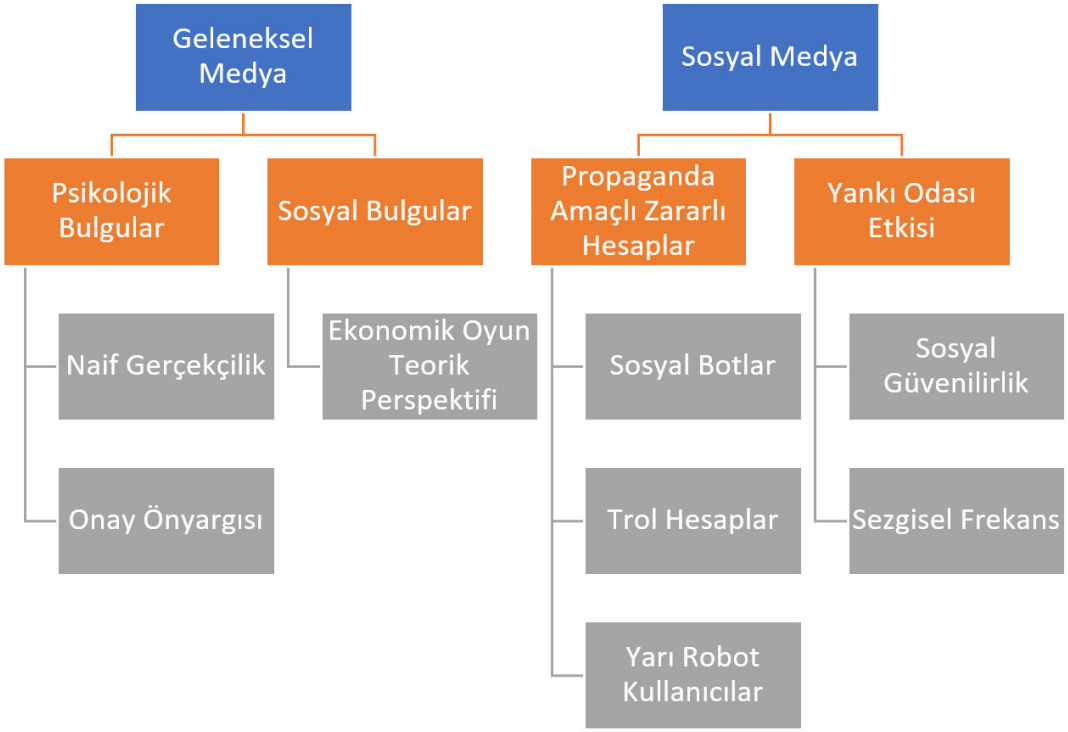
1. GİRİŞ

Sahte haberin en genel tanımı, kasıtlı olarak yapılan ve kesinlikle yanlış olan bir haber makalesi olarak yapılmıştır. Ayrıca, geleneksel medya ve sosyal medyayı sahte haber bakımından karşılaştırmışlardır. Sahte haberlerin, genellikle bot veya trol hesaplardan yapıldığını bildirmişler, trol ve bot hesapları tanımlamışlar ve bu hesapların kısa sürede ve çok sayıda oluşturulduğunu belirtmişlerdir. Aynı anda birden fazla bot veya trol hesaptan paylaşılan bir haberin, normal kullanıcılar tarafından gerçekmiş gibi algılanabildiğini belirtmişlerdir (Shu vd., 2017).

Sahte haber tespiti üzerine çalışmalar 2011 yılında Zhao vd. (2011)'nin çalışmasıyla başlamış olsa da 2016 yılındaki Amerika Başkanlık Seçimlerindeki komplo teorileri ile bu çalışmalar artmıştır. İngilizce dilinde birçok çalışma olmasına rağmen Türkçe dilinde otomatik sahte haber tespiti üzerine çalışmalar oldukça sınırlı sayıda kalmıştır.

Sahte haber tespiti yapılmadan önce sahte haberin dikkatli bir şekilde incelenmesi gerekmektedir. Shu vd. (2017) çalışmalarında, geleneksel medyadaki sahte haberleri, psikolojik ve sosyal bulgular bakımından; sosyal medyadaki sahte haberleri ise propaganda için açılmış sahte hesaplar ve yankı odası etkisi (echo chamber effect) bakımından incelemiştir. Sosyal medyada propaganda için açılmış sahte hesaplar tarafından yayılan sahte haberlerin sosyal botlar, trol hesaplar ve yarı robot hesaplar tarafından yayıldığını belirtmişlerdir. Çalışmada yankı odası etkisini, kullanıcıların aynı fikirde oldukları kullanıcıları takip etme eğiliminin fazla olduğunu ve bu hesaplardan gelen haberlere güvendiklerinden dolayı sahte haber olsa bile kendi ilgisine yakın haberleri paylaşmaya eğilimli olmaları olarak tanımlamışlardır. Ayrıca, benzer grup veya kişileri takip etmeleri nedeniyle, bu kişilerden gelen paylaşımlarla sıklıkla karşılaştıklarından zamanla bu haberleri sahte haber olsalar bile doğru haber olarak algıladıklarını ve paylaştıklarını belirtmişlerdir (Shu vd., 2017). Çalışmada belirtilen sahte haber türleri Şekil 1.1'de verilmiştir.

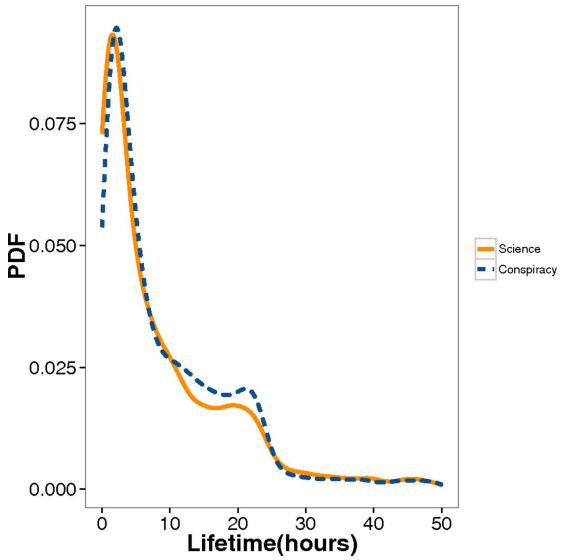
Trol veya bot hesaplardan, birbirine yakın zamanlarda ve tekrar tekrar paylaşılan haberler gerçek kullanıcılar tarafından doğru haber olarak algılanmaktadır. Bu haberleri doğru haber olarak algılayan gerçek hesaplar da bu sahte haberleri paylaşmaktadır.



Şekil 1.1. Sahte haber türleri (Shu vd., 2017).

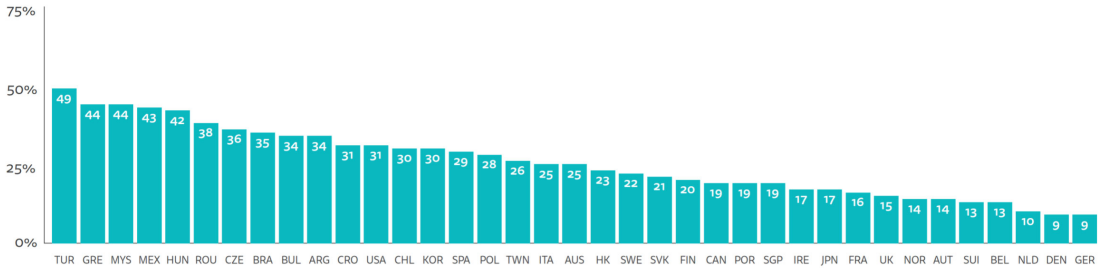
Bu sayede, kısa sürede birçok gerçek veya bot hesap tarafından sahte haber paylaşımı gerçekleşmektedir. Gerçek hesaplardan paylaşılan sahte haberlerin inandırıcılığı bu sayede artmaktadır. Sosyal medya üzerindeki sahte haberler, ilk 2 saatlik zaman diliminde çok yoğun bir şekilde paylaşılmış ve 20. saate kadar paylaşımının devam ettiği görülmüştür (Del Vicario vd., 2016). Bu durum Şekil 1.2'deki grafikte gösterilmiştir. Bu nedenle, sosyal medya üzerindeki sahte haberlerin tespit edilmesi ve okuyucuların bilgilendirilmesi çok hızlı bir şekilde gerçekleşmelidir. Bu da otomatik sahte haber tespiti araçlarıyla mümkün olabilmektedir.

Türkçe dilinde sahte haberle mücadele çalışmaları, İngilizce dilinde sahte haberle mücadele çalışmalarına göre yetersizdir. Literatürde, Türkçe dilinde sahte haberle mücadele için yapılmış çok az çalışma bulunmaktadır. Bu da Türkiye'de yaşayan insanların sahte habere maruz kalma oranını yükseltmektedir. Newman vd. (2018) yayımladıkları raporlarında, ülkelere göre sahte habere maruz kaldığını belirten kişilerin oranının % 49 ile Türkiye'de en fazla olduğunu belirtmişlerdir. Bu durumu gösteren grafik Şekil 1.3'de verilmiştir. Aynı raporda, Türkiye'deki kişilerin % 87'si, haberleri sosyal medya ve çevrimiçi kaynaklardan takip ettiklerini belirtmişlerdir. Bu durumu



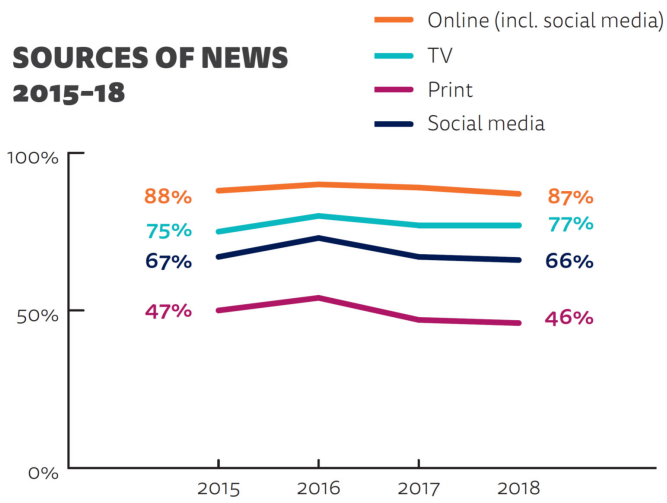
Şekil 1.2. Sahte haberlerin ilk paylaşımı ile son paylaşımı arasında zamana bağlı paylaşılma oranı (Del Vicario vd., 2016)

gösteren grafik Şekil 1.4'de verilmiştir. Yine Newman vd. (2019)'nin raporlarında 38 ülkede yaptıkları araştırmaya göre, bu ülkelerde yaşayan kişilerin ortalama % 55'i internette gerçek ve sahte haberi ayırma yeteneklerinden endişe duymaktadır. Türkiye'de yaşayanlarda ise bu oran % 63'tür. Bu durumu gösteren grafik Şekil 1.5'de verilmiştir. Bu durum, Türkiye'de, sahte haberleri ayırt edemeyen kişiler tarafından sahte haberlerin paylaşılmasıyla, bu haberlerin çok hızlı bir şekilde yayılmasına neden olmaktadır.

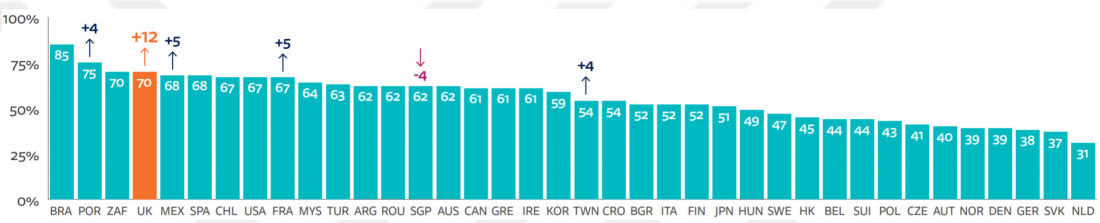


Şekil 1.3. Ülkelere göre sahte habere maruz kaldığını belirten kişilerin oranı (Newman vd., 2019).

Kurumsal hesapları veya siteleri taklit ederek birçok dolandırıcılık yöntemleri kullanılmaktadır. Örneğin; T.C. İletişim Başkanlığı tarafından 04.01.2020 tarihinde Twitter (Twitter, 2006) platformu üzerinden paylaşılan mesajda; kurumun logo ve ismini kullanıp elektrik ve doğalgaz fatura iadesi bildirimi yaparak dolandırıcılık yapılmaya çalışıldığı bildirilmiştir (Twitter, 2020a). Ayrıca 07.01.2020 tarihinde sahte bir Twitter



Şekil 1.4. Türkiye'deki kişilerin haberlere erişim kaynakları (Newman vd., 2018).



Şekil 1.5. Ülkelere göre internet üzerindeki gerçek ve sahte haberi ayırt etme yeteneklerinden endişe duyanların oranı (Newman vd., 2019).

hesabından Eskişehir'de okulların kar dolayısıyla tatil olduğu sahte mesajı çok hızlı bir şekilde yayılmıştır (Twitter, 2020b). İhlas Haber Ajansı tarafından bu haberin yetkililer tarafından yalanlandığı bildirilmiştir (Ihlas Haber Ajansi, 2020).

Türkiye'deki insanların büyük çoğunluğunun sahte haberi ayırt etme yeteneklerinden şüphe etmesi ve sahte haberlerin kısa sürede çok fazla hesap tarafından paylaşılması sahte haber tespitinin önemini göstermektedir. Uzmanlar tarafından yönetilen sistemlerin, sahte haber tespitinde, otomatik sahte haber tespiti sistemlerine göre yavaş kalmaktadır. 2 saatlik dilimde büyük oranda paylaşım yapılan bu sahte haberlerin uzmanlar yerine otomatik tespit sistemleri ile tespit edilmesi önemlidir. Bu çalışmada, Twitter platformunda paylaşılan Türkçe dilindeki mesajlarda sahte haber tespiti yapan otomatik bir sistem üzerine çalışılmıştır (Twitter, 2006). Denetimli öğrenme ve denetimsiz öğrenme algoritmalarıyla Türkçe dilinde sahte haber tespiti yapılmıştır. Denetimli öğrenme algoritmalarında 0,86 ve derin öğrenme algoritmalarında 0,80 F1-metrik değeri ile başarılı sonuçlar alınmıştır. Denetimsiz öğrenme algoritmalarında

ise 0,72 F1-metrik değeri elde edilmiştir. Konu bazlı tweet mesajlarında kullanıcıların takip ilişkisinde de ilginç sonuçlar elde edilmiştir. Literatürde Türkçe dilinde sahte haber tespiti konusunda çalışmalar giriş seviyesinde ve yetersiz kalmışlardır. Bu çalışma, literatürdeki bilinen en geniş kapsamlı Türkçe dilinde sahte haber tespiti çalışmasıdır.

2. KAYNAK ÖZETLERİ

Reuters Gazeteciliği Araştırma Enstitüsü tarafından, 2012 yılından itibaren her yıl Oxford Üniversitesinde, "Digital News Report" ismiyle rapor yayınlanmaktadır. Bu rapor dijital haberlerin durumunu, sosyal medyanın haber alma açısından yeri ve önemi ve birçok ülkenin sosyal medya haberleri de dahil olmak üzere dijital haberler hakkında bilgi sunmaktadır. Ayrıca bu raporda, sahte haberler ile ilgili bölümler de bulunmaktadır (Newman vd., 2018, 2019).

2015 yılında, Carnegie Mellon Üniversitesi'nde öğretim üyesi olan Dean Pomerleau ve "Joostware AI Research Corporation"ın kurucusu Delip Rao tarafından makine öğrenimi ve doğal dil işlemenin sahte haberlerle savaşmakta ne kadar başarılı olabileceğini keşfetmek amacıyla "Fake News Challange-1 (FNC-1)" yarışması düzenlenmiştir. Bu yarışmada amaç, başlıktaki bilginin makale içeriğiyle ilgisinin olup olmadığını tahmin etmektir. Bunun için, eğitim ve test seti bulunan bir veri seti yayınlanmıştır. Bu veri setinde; eğitim seti için 49972 makale-başlık ve test veri seti için 25413 makale-başlık verileri bulunmaktadır. Eğitim setindeki başlık-makale çiftleri, 1689 benzersiz başlıktan ve 1648 benzersiz makaleden oluşturulmuştur. Test setinde ise 894 benzersiz başlık ve 904 benzersiz makale vardır. Eğitim seti; %73,1 ilişkisiz (unrelated), %7,4 onaylama (agree), %1,7 onaylamama (disagree) ve %17,8 tartışma (discuss) sınıf dağılımına sahiptir. Test seti; %72,2 ilişkisiz (unrelated), %7,5 onaylama (agree), %2,7 onaylamama (disagree) ve %17,6 tartışma (discuss) sınıf dağılımına sahiptir. Başlık ve haber çiftlerinin ilişkili olduğunu doğru tahmin eden takım 0,25 ağırlık puanı, ilişkili çiftleri kabul etme (agree), kabul etmeme (disagree) ve tartışma (discussion) olarak doğru etiketleyen takım ise 0,75 ağırlık puanı almaktadır. Toplam 50 yarışmacı bu etkinliğe katılmıştır. Birinci olan "SOLAT in the SWEN" takımı; %82,02 sonucuna ulaşmak için test setinde, Deep Convolutional Model (DCM) ile Gradient Boosted Decision Tree (GBDT) algoritmalarının sonuçlarının ortalamasını kullanmıştır. İkinci olan "Athene" takımı ise her birinin 7 katmanı bulunan, 5 farklı Multi-Layer Perceptron (MLP) modelini kullanarak %81,97 puan almıştır. Üçüncü olan "UCL Machine Reading" takımı da ReLu ve Softmax katmanı ile iki gizli katmanlı bir MLP kullanarak %81,72 puan almıştır (Pomerleau ve Rao, 2015).

2015 yılında "Massachusetts Institute of Technology"de Vosoughi tarafından yapılan doktora tezinde 7000 adet etiketlenmiş İngilizce tweet verileri; Lojistik Regresyon (LR) kullanılarak "iddia", "ifade", "soru", "öneri", "istek" ve "diğer" etiketleri ile sınıflandırılmıştır (Vosoughi, 2015). Tweet mesajlarının konuları; "varlığa yönelik", "olaya yönelik" ve "uzun süreli tartışılan" olmak üzere üç farklı sınıfa ayrılmıştır. Varlığa yönelik konu grubu için; "Red Sox beyzbol takımı" ve "oyuncu Asthon Kutcher" tweet mesajlarını, olaya yönelik konu grubu için; "Boston bombalamaları" ve "Ferguson protestoları" tweet mesajlarını ve uzun süreli tartışılan konu grubu için ise "yemek pişirme" ve "seyahat" tweet mesajlarını toplamıştır. Toplamda altı adet konunun her biri için 1000'in üzerinde tweet toplamıştır. Üç kişiden, toplanan bu tweet mesajlarını daha önce belirlenen altı sınıf ile etiketlemelerini istemiştir. Etiketleme işlemi sonrasında Üç kişinin de aynı etiketi belirttiği mesajları dikkate almıştır. Toplanan tweet mesajlarının %62'sini kullandığını, %38'ini ise dikkate almadığını belirtmiştir. Toplanan ve etiketlenen bu tweet mesajlarını anlamsal ve söz dizimsel olarak incelemiştir. Anlamsal olarak n-gram yöntemi kullanarak inceleme yapmıştır. Söz dizimsel olarak ise soru işareti (?) kullanılan tweet mesajlarının genellikle soru veya istek sınıfında, ünlem işareti (!) kullanılan tweet mesajlarının ifade veya tavsiye sınıfında bulunduğunu belirtmiştir. Ayrıca hashtag (#) işareti kullanılan tweet mesajlarının ifade ve tavsiye sınıfında, @ işareti kullanılan tweet mesajlarında soru, tavsiye veya istek sınıfında ve RT kullanılan tweet mesajlarında ise iddia sınıfında bulunduğunu belirtmiştir. Sınıflandırmak için, denetimli öğrenme algoritmalarından, Naive Bayes (NB), Decision Tree (DT), LR ve Linear Support Vector Machine (L-SVM) algoritmalarını kullanmıştır. LR algoritmasının diğer algoritmalardan daha yüksek F1-metrik değerine sahip olduğu sonucuna varmıştır.

Chen ve Chen (2015) çalışmalarında, telefonlar hakkında Çince bir forum sitesi olan "mobile01.com" üzerindeki forum mesajlarını spam veya spam değil olarak etiketlemişlerdir. Forum mesajlarını zamana bağlı olarak incelemişler ve spam mesajlarının daha çok çalışma günleri ve saatlerinde, spam olmayan mesajların ise çalışma saatleri dışında paylaşıldığını belirtmişlerdir. Scikit-Learn kütüphanesini kullanarak; LR, doğrusal çekirdek fonksiyonu kullanan Support Vector Machines (SVM), radyal tabanlı çekirdek fonksiyonu kullanan SVM ve F1-metrik değerini

optimize eden bir SVM türü olan SVMperf algoritmalarını kullanmış ve F1-metriklerini karşılaştırmışlardır. Üstün bir farkla radyal tabanlı çekirdek fonksiyonu kullanan SVM algoritmasının başarılı olduğunu belirtmişlerdir.

Kayes vd. (2015) çalışmalarında, "Yahoo Answers" platformundaki kullanıcıları sınıflandıran bir model geliştirmişlerdir. Yahoo answers platformunda kullanıcıların kötüye kullanımları bildirdikleri bayraklar bulunduğunu, fazla bayrağa sahip kullanıcıların bir editör tarafından denetlenip hesabının engellendiğini belirtmişlerdir. Fakat, bazı kullanıcıların sosyal ağa çok katkı verdiği halde aldığı bayraklar nedeniyle engellenebildiğini belirtmişlerdir. Yanlış engellenmelerin önüne geçmek için bir sınıflandırma modeli geliştirmişlerdir. Altı kategoride toplam 29 özellik belirlemişlerdir. Sosyal kategorisindeki özellikler, kullanıcıların sosyal ağını belirtmektedir. Etkinlik (Activity) kategorisindeki özellikler, kullanıcıların soru ve cevaplara katkılarını belirmektedir. Başarı (Accomplishment) kategorisindeki özellikler, kullanıcıların ağa katkı verme kalitesini belirmektedir. Bayrak (Flag) kategorisindeki özellikler, bir kullanıcının hem alınan hem de bildirilen bayrak sayılarını belirtmektedir. Sapma puanı (Deviance Score) kategorisi, kullanıcıların etkinliklerine ve bayraklarına göre oluşturulan bir puandır. Sapma Hemofili (Deviance Homophily) kategorisi, sapma ile ilgili olarak kullanıcının benzer kişilerle olarak ilişkisini belirtmektedir. Bu kategorilerin altındaki özellikler kullanılarak kullanıcıları sınıflandırmak için Stochastic Gradient Boosted Trees (SGBT), NB, LR, K-Nearest Neighbor (KNN) ve SVM algoritmalarını karşılaştırmışlardır. Buna göre SGBT algoritmasının diğerlerine göre daha başarılı olduğunu belirtmişlerdir.

Rubin vd. (2016) çalışmalarında, 2015 yılında Amerika ve Kanada gazetelerinde yayınlanan 360 adet çeşitli haber makalesi toplamışlardır. Özellik çıkarımı olarak, TF-IDF, anlamsızlık, mizah, gramer özellikleri (özne, fiil, sıfat, vb.), olumsuz etki ve noktalama işaretlerini dikkate almışlardır. Bu özellikleri farklı kombinasyonlar ile SVM algoritması ile eğitip, test etmişlerdir. Veri setini 260 haber makalesi ile eğitmişler ve 90 haber makalesi ile de test etmişlerdir. Sonuç olarak; TF-IDF, gramer özellikleri, noktalama işaretleri ve anlamsızlık özellikleri ile eğitilen modelin; 0,87 F1-metrik değeri ve 0,90 Kesinlik (precision) değeriyle diğer özellik kombinasyonlarıyla eğitilen modelden daha iyi sonuç verdiğini belirtmişlerdir.

Ahmed (2017) yüksek lisans tez çalışmasında, semantik benzerlik ve n-gram analizi ile sahte haber tespitini araştırmıştır. Bunun için üç farklı veri seti oluşturmuştur. İlk veri seti, 800 sahte hotel incelemesi ve 800 gerçek hotel incelemesinden oluşmuştur. İkinci veri seti, 1000 restoran incelemesi, 800 eşcinsel evlilik ve 800 silahlanma kontrolü konularını içermektedir. Üçüncü veri seti ise, 12600 sahte haber ve 12600 gerçek haber makalesinden oluşmaktadır. Verilen veri setinin %80'ini eğitim seti ve %20'sini test seti için bölmüştür. Eğitim setiyle algoritmayı eğitmek için 5-katlamalı çapraz doğrulama (5-fold cross validation) metodunu kullanmıştır. TF ve TF-IDF yöntemleriyle özellik çıkarma işlemi yapmış ve Stochastic Gradient Descent (SGD), SVM, L-SVM, KNN, LR ve DT olmak üzere 6 farklı makine öğrenme algoritması ile sınıflandırma işlemini yaparak bu algoritmaların sonuçlarını karşılaştırmıştır.

Bajaj (2017), derin öğrenme kullanarak sahte haberlerin tespiti üzerine yaptığı çalışmada, LR, Feedforward Neural Network (FNN), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Long-Short Term Memory (LSTM), Çift Yönlü LSTM (BiLSTM), Maksimum havuzlama ile Convolutional Neural Network (CNN), maksimum havuzlama ve ilgi mekanizması ile CNN metotlarını kullanarak, 63 bin veri içerisinde gerçek ve sahte haberleri sırasıyla 0 ve 1 şeklinde ikili sınıflandırma ile sınıflandırmıştır. Kelime temsili yöntemi için önceden eğitilmiş GloVe yöntemini kullanmıştır. Veri setini karıştırarak, %60 eğitim seti, %20 doğrulama seti ve %20 test seti olarak belirlemiştir. Buna göre LR algoritması 0,65, FNN algoritması 0,80, RNN algoritması 0,70, GRU algoritması 0,84, LSTM ve BiLSTM algoritmaları 0,81, maksimum havuzlama ile CNN algoritması 0,58, maksimum havuzlama ve ilgi mekanizması ile CNN algoritması ise 0,60 F1-metrik değerleri elde etmiştir. GRU algoritmasının hem daha iyi bir F1-metrik değerine sahip olduğunu, hem de diğer algoritmalara göre daha hızlı olduğunu belirtmiştir.

Bhatt vd. (2017) çalışmalarında, FNC-1 yarışmasının veri setini kullanmışlar ve önerdikleri yöntemi, FNC- yarışmasında en çok puan alan ilk 4 yarışmacının çalışmasıyla karşılaştırmışlardır. Önerdikleri yöntemin özellik çıkarımı adımında; üç özellik kullanmışlardır. İlk özellik ile daha çok soru-cevap sistemlerinde kullanılan 4800 boyutlu Skip-Thought Vektör, ikinci özellik ile haber başlık ve haber içeriklerinin terim sıklığı değerini içeren 5000 boyutlu vektör ve üçüncü özellik ile benzer

kelimelerin sayısı, başlık-gövde çiftlerinin vektör kodlamaları arasındaki kosinüs benzerliği, çiftler arasında eşleşen n-gram sayısı gibi özelliklerden oluşan 50 boyutlu vektör oluşturmuşlardır. Çıkardıkları 3 özellik vektörünü, MLP algoritmasına giriş olarak vermişlerdir. Sonuç olarak önerdikleri yöntemin, FNC-1 puanını 83,08 olarak hesaplamışlar ve FNC-1 yarışmasında birinci olan takımın aldığı 82.05 FNC-1 puanını geçmeyi başardıklarını belirtmişlerdir.

Granik ve Mesyura (2017) çalışmalarında, NB sınıflandırıcı kullanarak sahte haber tespiti için bir yöntem uyguladıklarını belirtmişlerdir. Bunun için veri seti olarak "buzzfeed" internet sitesinde yayımlanan aşırı partizan facebook sayfalarını kullanmışlardır. Bu Facebook sayfalarından toplanan 2282 gönderiyi "çoğunlukla doğru", "çoğunlukla yanlış", "doğru ve yanlış karışık" ve "tam bilgi yok" etiketi ile etiketlemişlerdir. Bazı gönderilerin içeriğinin çekilemediği anlaşılmış ve bu veriler temizlenerek 1771 veri üzerinde çalışmışlardır. Özellik çıkarma aşamasında kelime temsili için kelime torbası (bag of words) yöntemini kullanmışlardır. Sınıflandırma işlemi sonucunda; doğru haberleri sınıflandırma doğruluğu %75,59, yanlış haberleri sınıflandırma doğruluğu %71,73, toplamda ise doğru sınıflandırma doğruluğu %75,40 olarak verilmiştir.

Lazer vd. (2017), düzenledikleri konferansın final raporunda, sahte haberleri azaltmak için dört madde sunmuşlardır. Bunlar; belirli haberlerin sahte olabileceğine dair okuyuculara geri bildirim yapmak, belirli haberlerin sahte olduğunu doğrulayan ve ideolojik olarak uyumlu kaynaklar sunmak, bot ve sahte hesaplar tarafından yayılan haberleri tespit eden ve bu manipülasyonları engelleyen algoritmalar geliştirmek ve sahte haberlerin kaynaklarını tespit edip bu kaynaklardan yayılan haberleri azaltmaktır. Ayrıca araştırma topluluğu olarak yakın zamanda hayata geçirilebilecek üç adet eylem planı oluşturmuşlardır. Bunlar ise; politikadaki yanlış bilgilendirmelerin tartışmasında daha muhafazakâr olmak, gerçeği daha yüksek sesle söyleyebilmek için gazetecilerle daha yakın iş birliği içinde çalışmak ve sosyal medya platformlarında yanlış bilginin varlığına ve yayılmasına ilişkin akademik araştırma yürütmek için çok disiplinli toplum genelinde paylaşılan kaynakları geliştirmektir. İleriye dönük olarak ise; bireylerin ve toplulukların yanlış bilgilendirilmesinin etkilerini en aza indiren sosyal ve bilişsel müdahalelerin yanı sıra, Google, YouTube, Facebook ve Twitter gibi platformların, sahte

haber yayılımını kolaylaştırdığını ve ülkelerin iç politikalarında alınacak hangi kararlarla bu etkileri azaltabileceklerini araştırmaları gerektiğini belirtmişlerdir. Daha geniş anlamda, bir gerçeklik kültürünü teşvik eden bilgi sistemleri için gerekli bileşenlerin neler olduğunu araştırmaları gerektiğini belirtmişlerdir. Bu raporda; Birinci bölüm, mevcut medya ekosistemindeki yanlış bilgi durumunu; İkinci Bölüm, sahte haberlerin psikolojisi ve konferans sırasında kapsanan sosyal sistemlere yayılması ile ilgili araştırmaları; Üçüncü bölüm, konferans sırasında düzenlenen bulguları ve tartışmaları akademik topluluğun yakın gelecekte alabileceği üç eylem sürecine dönüştürdüğünü ve son olarak dördüncü Bölüm ise gelecekte yanlış bilgilendirme yapma becerimizi geliştirecek araştırma alanlarını açıklamaktadır.

Patel (2017) yüksek lisans tez çalışmasında, 2016 yılındaki Amerika Başkanlık seçimlerinde yayımlanan sahte haberleri tespit etmeye çalışmıştır. Bunun için; "snopes.com" ve "politifact.com" sitelerinde 2016 Amerika Başkanlık seçimleri hakkında sahte olarak işaretlenmiş haberlerden oluşan bir veri setini sahte haber veri seti olarak, Google arama motorundan 01 Ocak 2016 ile 01 Aralık 2016 tarihleri arasında güvenilir internet sitelerinde yayınlanan haber makalelerini ise doğru haber veri seti olarak kullanmıştır. Özellik çıkarma aşamasında ise, öncelikle belgelerdeki kelime sayıları ile kelimelerin harf sayılarını incelemiş, daha sonra ise belgelerdeki kelime sayıları ile Google Sentiment Analysis uygulama programlama arayüzünün duygu büyüklüğü ölçüsünü karşılaştırmıştır. Buna göre; kelime uzunlukları daha büyük olan metinlerin doğru haber kategorisinde olabileceğini, kelime sayılarının daha fazla olduğu metinlerde de duygu büyüklüğünün daha fazla olduğu sonucunu çıkarmıştır. Ayrıca Google Duygu Puanı ile Microsoft Duygu Puanını karşılaştırmıştır. Burada da iki farklı uygulama programlama arayüzünün (Google ve Microsoft) birbirinden farklı sonuçlar verdiğini görmüştür. Dolayısıyla hem Google'ın hem de Microsoft'un duygu özelliklerini kullanmıştır. Son olarak farklı bir grafikte ise tarih aralıklarına göre güvenilir makalelerin Google Duygu Puanı ve aynı şekilde güvenilmeyen haberlerin Google Duygu Puanlarını karşılaştırmıştır. Buna göre ise; 1008 güvenilir makaleden sadece 48 tanesi (yaklaşık %5) polar duygu puanına sahipken, 333 güvenilmeyen makalelerin 44 tanesinin (yaklaşık %13) polar duygu puanına sahip olduğundan bahsetmiştir. KNN, SVM ve LSTM olmak üzere 3 farklı makine öğrenme algoritması

kullanmış ve sonuçlarını incelemiştir. LSTM algoritmasının ortalama F1-metrik değerinin %90 olduğunu belirtmiştir.

Pérez-Rosas vd. (2017) çalışmalarında, iki farklı veri seti oluşturmuşlardır. İlk veri setini oluşturmak için spor, iş dünyası, eğlence, politika, teknoloji ve eğitim konularında, Amerika ile ilgili yayın yapan çeşitli haber platformlarından 240 adet haber makalesi toplamışlardır. Topladıkları bu 240 gerçek haber makalesinin sahte haber olarak yayınlanmış olanlarını bulmak için Amazon Mechanical Turk üzerindeki kullanıcılardan 240 adet sahte haber makalesi toplamışlardır. İkinci veri setini oluşturmak için ise ünlüler ile ilgili haberlere yoğunlaşmışlardır. Bunun için haftalık magazin ve eğlence yayını yapan platformlardan makaleler toplamışlardır. Bu makaleleri, gossipcop.com doğrulama sitesi ile doğrulayarak, 100 gerçek 100 sahte haber makalesi toplamışlardır. Özellik çıkarımında, toplanan makalelerin; n-gram, noktalama işaretleri, psiko-dil bilimsel özellikleri, okunabilirlik ve söz dizimi özelliklerini çıkarmışlardır. Çıkarılan tüm özellikleri SVM algoritması kullanarak eğitip test etmişlerdir. Tüm özellikler algoritmaya verildiğinde; birinci veri setinin F1-metrik değerini 0,74, ikinci veri setinin F1-metrik değerini ise 0,73 olarak bulmuşlardır.

Pratiwi vd. (2017) çalışmalarında, Endonezya dilinde php dilinde makine öğrenmesi kütüphanesi olarak geliştirilen PHP-ML kütüphanesini kullanarak, NB sınıflandırıcısı ile sahte haber tespiti üzerine bir yöntem geliştirmişlerdir. Veri seti olarak internet üzerindeki sitelerden 250 adet sahte ve gerçek haber makalesi toplamışlardır. Bu veri setini 3 araştırmacı manuel olarak etiketlemişlerdir. 3 araştırmacıdan en az ikisinin aynı etiketi kullandığı etiketler o makalenin etiketi olarak belirlenmiştir. Bu veri setini, %70 eğitim seti, %30 test seti olarak, %80 eğitim seti, %20 test seti olarak ve %60 eğitim %40 test seti olarak belirlemişlerdir. Özellik çıkarma aşamasında; bir kelimenin kaç belgede geçtiğini gösteren belge sıklığı (document frequency), bir belgede bir terimin kaç kere geçtiğini gösteren terim sıklığı (TF) ve bir terimin sınıfla ilgili ne kadar bilgi içerdiğini ölçen ortak bilgi (mutual information) özelliklerini kullanmıştır. Daha sonra modelin eğitimi ve testi için PHP-ML kütüphanesini kullanarak farklı boyutlarda eğitim ve test setine bölünen 3 veri seti ile modeli eğitmişlerdir. Sonuç olarak %70 eğitim ve %30 test seti için ayırdıkları veri seti ile en yüksek doğruluk metriği olan ortalama %78,6 doğruluk metriğini bulmuşlardır.

Ruchansky vd. (2017) çalışmalarında, sahte haber tespiti için, CSI (Capture, Score, and Integrate) adını verdikleri üç aşamalı bir model önermişlerdir. İlk aşamanın verilen metindeki kullanıcı aktivitelerini incelediğini, ikinci aşamanın kullanıcı davranışlarından kaynak karakteristiğini çıkardığını ve üçüncü aşamanın ise ikinci aşama ile birlikte haberin sahte veya gerçek olduğunu sınıflandırdığını belirmişlerdir. Kendi modellerini, Twitter ve Weibo veri setleri kullanarak, farklı çalışmalarla ve ayrıca LSTM ve GRU algoritmaları ile karşılaştırmışlardır. Kendi önerdikleri modelin diğer modellerden daha başarılı oluğunu deney sonuçlarıyla göstermişlerdir.

Singhania vd. (2017) çalışmalarında, haber makalelerini temsil etmek için, sinir ağı tabanlı 3HAN ismini verdikleri bir model önermişlerdir. Veri seti olarak; Polifact internet sitesinde yayınlanan sahte haber sitelerinden 20372 makale ve Forbes internet sitesinde yayınlanan güvenilir haber sitelerinden 20932 makale toplamışlardır. Önerdikleri modeli, kelime çantası, TF-IDF, n-gram gibi kelime sayısı tabanlı modeller ve Glove, GRU gibi sinir ağı tabanlı modeller ile karşılaştırmışlardır. Önerdikleri modelin %96,77 doğruluk oranıyla diğer modellerden üstün olduğunu belirtmişlerdir.

Tacchini vd. (2017) çalışmalarında, Facebook üzerinde paylaşılan sahte haberlerin tespiti için kullanıcıları incelemişler ve bu kullanıcıların beğendiği gönderi ve sayfalara göre paylaştıkları haberlerin sahte olup olmama durumuna karar vermişlerdir. Veri setlerinde, 1 Temmuz 2016 ile 31 Aralık 2016 tarihleri arasında 20 adet sahte haber ve 20 adet bilimsel haber sayfalarında paylaşılan 15.500 gönderi ve 909.236 kullanıcı bulunmaktadır. Bu gönderilerden 8.923 (% 57,6) tanesi sahte haber, 6.577 (% 42,4) tanesi ise doğru haber içeriğidir. Sahte haber gönderilerinin beğenilme oranlarının gerçek haber gönderilerine göre daha fazla olduğunu belirtmişlerdir. Sahte haber ile gerçek haberleri sınıflandırmak için LR ve Harmonic Boolean Label Crowd-sourcing (HBLCS) olmak üzere iki yöntem kullanmışlardır. LR algoritması ile 0,79 doğruluk metriği, HBLCS ile 0,99 doğruluk metriği elde etmişlerdir.

Volkova vd. (2017) çalışmalarında, 130 bin haber makalesini şüpheli veya onaylı olarak sınıflandırma üzerine bir model önermişlerdir. Şüpheli haberleri de hiciv, aldatma, tıklama tuzağı (clickbait) ve propaganda olarak sınıflandırmışlardır. Diğer çalışmalardan farklı olarak, gramer ve söz dizimi özelliklerinin modelin sınıflama performansını

arttırmadığını belirtmişlerdir. Twitter platformunda, haber yayan kullanıcılardan 2016 yılında Brüksel'de yaşanan terörist saldırıları ile ilgili tweet mesajlarını toplamışlardır. Modellerini eğitmek için LR, RNN ve CNN algoritmalarını kullanmışlardır. RNN ve CNN ile eğitilen modellerin LR ile eğitilen modelden daha yüksek F1-metrik değerlerine sahip olduğunu bulmuşlardır.

Wang (2017) çalışmasında, sahte haber tespiti için "LIAR" adını verdikleri yeni bir veri seti oluşturmuştur. Polifact API kullanarak 12836 etiketli kısa cümleler ile bir veri seti oluşturmuştur. Ayrıca bu veri setine, konuşmacı, meslek, yaşadığı yer, mensubu olduğu parti gibi ek özellikler ekleyerek ikinci bir veri seti daha oluşturmuştur. Word2Vec kelime vektörü kullanarak bu kısa cümleleri 300 boyutlu vektör ile temsil etmiştir. LR, SVM, BiLSTM ve CNN kullanarak sadece metinsel ifadelerden oluşan veri setini eğitmiştir. CNN kullanarak ise metinsel ifadelere ek özellikler bulunan veri setlerini eğitmiştir. Ek özellikler bulunan veri seti ile eğitilen CNN modelinin, diğer modellerden daha iyi sonuç verdiğini belirtmiştir.

Ågren ve Ågren (2018) yüksek lisans tez çalışmasında, RNN kullanarak sahte haber tespiti üzerine çalışmıştır. FNC-1 yarışmasında kullanılan etiketli veri setini, Paralel encoder ve koşullu encoder kullanan LSTM ve GRU algoritmaları ile sınıflandırmıştır. Sonuç olarak, ReLu aktivasyon fonksiyonu kullanarak GRU algoritmasının, LSTM algoritmasına göre daha yüksek doğruluk metriği verdiğini ifade etmiştir.

Girgis vd. (2018) çalışmalarında, RNN tabanlı derin öğrenme algoritmalarını kullanarak sahte haber tespiti üzerine çalışmışlardır. Veri seti olarak "LIAR" veri tabanını kullanmışlardır. Veri setindeki metinleri temsil etmek için word embedding kullanmışlar ve Vanilya RNN, GRU ve LSTM algoritmalarına girdi olarak vermişlerdir. Sonuç olarak, en kötü sonucu Vanilya RNN algoritması ile aldıklarını ve GRU algoritmasının, LSTM algoritmasından daha iyi sonuç verdiğini belirtmişlerdir.

Kim vd. (2018) çalışmalarında, kitle destekli çevrimiçi bir algoritma geliştirmişler ve bu algoritmanın adını Curb olarak belirlemişlerdir. Kitle destekli sahte haber sistemlerinin çalışma prensibini açıklamışlardır. Facebook, Twitter ve Weibo gibi sosyal platformların kitle destekli olarak paylaşımları kontrol ettiklerini belirtmişlerdir.

Kullanıcılar tarafından sahte haber olarak paylaşımların işaretlendiği ve daha sonra yeteri kadar sahte haber işareti bulunan paylaşımın güvenilir bir üçüncü kişi tarafından onaylandığını belirtmişlerdir. Kitle kaynaklı bu tarz sistemlerin suiistimal edilerek, üçüncü tarafa gerçek haberlerin de sahte haber olarak gönderilmesinin sahte haber tespitini zorlaştırdığını belirtmişlerdir. Geliştirdikleri algoritma ile hangi verilerin üçüncü tarafa aktarılacağını seçmektedirler. Twiter ve Weibo sosyal ağ platformlarından toplanan verilerle deneylerinin başarılı olduğunu belirtmişlerdir.

Mertoğlu vd. (2018) çalışmalarında, TF-IDF kullanarak sahte ve gerçek haber makaleleri üzerine bir prototip sunmuşlardır. Bu prototipte otomatik, yarı otomatik ve manuel veri toplayan bir kullanıcı arayüzü geliştirmişler ve spor, politika, şehir efsanesi, sağlık ve diğer konulardan oluşan üç farklı veri kümesi oluşturmuşlardır. Kullanıcı arayüzü ile toplam 250 haber toplamışlardır. Özellik çıkarımı adımında terim sıklığı ve n-gram özelliklerini kullanarak bir veri seti oluşturmuşlardır. Ayrıca TF, n-gram, argo kullanımı ve noktalama işareti kullanımı özellikleri ile de ikinci bir veri seti oluşturmuşlardır. İlk veri seti ile eğitilen SVM ve NB algoritmalarının sonuçlarını, ikinci veri seti ile eğitilen SVM ve NB algoritmalarının sonuçlarıyla karşılaştırmışlardır. İkinci veri seti ile kullanılan SVM algoritmasının diğer sonuçlara göre daha yüksek F1-metrik değeri verdiğini belirtmişlerdir.

Potthast vd. (2018) çalışmalarında, partizan haberleri ve sahte haberleri tespit etmek için bir yöntem önermişlerdir. Sağ partizan, sol partizan ve partizan olmayan kullanıcılardan oluşan dokuz siyasi yayıncı grubundan, 1627 haber makalesi toplamışlardır. Üç farklı özellik çıkarımı ile veri setinin özelliklerini çıkarmışlardır. Verileri partizan, sahte ve hiciv haber olarak sınıflandırmak için Ternary Classifier (TC) kullanmışlardır. Partizan olarak sınıflandırmanın F1-metrik değeri 0,78, hiciv sınıflandırmanın F1-metrik değerini 0,81 ve sahte haberin F1-metrik değerini ise 0,46 olarak hesaplamışlardır.

Qian vd. (2018) çalışmalarında, haber makaleleri ve kullanıcıların bu makalelere verdikleri cevapları kullandıkları bir sahte haber tespiti yöntemi önermişlerdir. TCNN-URG adını verdikleri iki seviyeli CNN ve kullanıcı cevap üretici modelini sunmuşlardır. İki aşamalı olan bu modelde, ilk aşama olan TCNN ile algoritmanın otomatik özellik çıkarımı yaptığını ve ikinci aşama olan URG ile sahte haber tespiti için

ihtiyaç duyulan kullanıcı yorumlarını, kullanıcı yorumu olmayan haberler için ürettiğini belirtmişlerdir. Daha sonra oluşan vektörü ileri beslemeli softmax sınıflandırıcı ile eğitip sahte haber tespiti yapmaktadırlar. Modeli eğitmek için en az 500 kelime bulunan ve Twitter platformunda en az bir cevabı bulunan haberleri toplamışlardır. Kendi yöntemlerini, LIWC (Ott vd., 2013), uni-gram, post-gram ve CNN ile farklı boyutlarda eğitim seti kullanarak karşılaştırmışlardır. Sonuç olarak, farklı boyutlardaki eğitim setlerinin tümünde önerdikleri yöntem diğer yöntemlerden daha yüksek doğruluk metriği vermiştir. %90 eğitim seti kullandıkları modelde ise en yüksek doğruluk metriğine sahip olmuşlardır.

Rajendran vd. (2018) çalışmalarında başlık ve içerik çiftlerinin alakalı veya alakasız olarak etiketlenmesi üzerine çalışmışlardır. Alakalı olan başlık-içerik çiftlerini de onaylama, onaylamama ve tartışma olarak 3 farklı etikete bölmüşlerdir. Veri seti olarak, NDTV, CNN gibi haber kuruluşlarından başlık-içerik çiftlerini toplamışlardır. Her başlığı birden fazla haber makalesi ile ilişkilendirmişlerdir. Veri setinin %70'ini eğitim seti, %30'unu ise test seti olarak belirlemişlerdir. Kelimelerin temsili için Skip-Gram yöntemi ile Word2Vec kelime temsili yöntemini uygulamışlardır. Sınıflandırma işlemi için; BiLSTM, LSTM, BiGRU, GRU, basit RNN model, BiGRU ile BiLSTM algoritmalarının birlikte kullanımı ve MLP algoritmalarını kullanmış ve karşılaştırmışlardır. BiLSTM modeli ile %83,5 doğruluk oranı elde ettiklerini belirtmişlerdir.

Tschiatschek vd. (2018) çalışmalarında; Facebook tarafından geliştirilen ve sahte haberlerin tespiti için Facebook kullanıcılarından aldığı dönüşleri kullanan yöntemi daha da geliştirmişlerdir. Facebook tarafından geliştirilen yönteme göre kullanıcılar, Facebook üzerinde paylaşılan haber metinlerini bir bayrak vasıtasıyla gerçek veya gerçek dışı olarak etiketleyebilmektedir. Daha sonra kullanıcılar tarafından işaretlenen bu haberler bir uzmana gönderilmektedir. Tschiatschek ve arkadaşlarının çalışmasında; Facebook sosyal ağının yöntemine ek olarak kullanıcıların haberleri sahte olarak bildirme doğruluğunu ölçmüşlerdir. Kullanıcıların sahte haberleri tespit doğruluğunu öğrenmek için Bayes yaklaşımıyla Detective isimlerini verdikleri bir algoritma geliştirmişlerdir. Bu sayede bir grup sahte hesabın, sahte bir haberi gerçek olarak

işaretleyerek gerçek gibi göstermesinin önüne geçildiğini ve uzmana gönderilen haberlerin sayısının azaltılmasını sağladığını bildirmişlerdir.

Fang vd. (2019) çalışmalarında, CNN algoritmasının uzak kelimeler arasındaki ilişkiyi çözmekte yetersiz kaldığını, önerdikleri CNN algoritması tabanlı, "Self Multi-Head Attention-based CNN (SMHA-CNN)" adını verdikleri modelin ise bu soruna çözüm olarak geliştirildiğini belirtmişlerdir. 12228 sahte haber ve 9762 gerçek haber makalesinden oluşan bir veri seti kullanmışlardır. Kelime temsili için Word2vec ve Glove vektörlerini kullanmışlardır. Önerdikleri modeli LSTM, dikkat mekanizmalı LSTM, BiLSTM ve GRU algoritmaları ile karşılaştırmışlardır. 1200 boyutlu hem Word2Vec hem de Glove kelime vektörü ile önerdikleri modelin, karşılaştırdıkları modellerden daha yüksek F1-metrik değeri verdiğini belirtmişlerdir.

Mertoğlu vd. (2019) çalışmalarında, Türkçe sahte haber tespitinde otomatik anahtar kelime bulma modeli sunmuşlardır. Çalışmalarında evrensel bir veritabanı projesi olan GDELT (Global Data on Events, Languages and Tone) projesinden ve kendi geliştirdikleri kullanıcı arayüzü ile web sitelerinden haber makaleleri toplamışlardır. Toplanan verilerin 1305 tanesi anahtar kelimeleri bulunan verilerden oluşmuştur. Kalan 250 tane anahtar kelimesi bulunmayan veri için ise 7 tane lisansüstü öğrencisi tarafından en az 2, en fazla 4 anahtar kelime ile etiketlenmesini sağlamışlardır. TF-IDF yöntemi ile kelimeler vektör olarak temsil edilmiş ve toplam 1550 verinin 1150 tanesi eğitim setinde ve kullanılmamış olan 400 tanesi ise test setinde kullanılmıştır. Anahtar kelimeleri tahmin etmek için NB algoritması kullanmışlardır.

Monti vd. (2019) çalışmalarında, geometrik derin öğrenme tabanlı sahte haber tespit sistemi önermişlerdir. Modellerini eğitmek için Twitter sosyal medya platformundan tweet mesajları toplamışlardır. Snopes, Polifact ve Buzfeed haber doğrulama platformlarından sahte haber olarak etiketlenmiş konular hakkında tweet mesajlarını tespit etmişlerdir. Ayrıca, Twitter kullanıcılarının güvenilirliğini tahmin etmek için sahte haber yayan kullanıcıları da toplamışlardır. Dört katmanlı graf evrişimsel sinir ağını kullanmışlardır. Önerilen yöntemin, yüksek başarı sağladığını belirtmişlerdir.

Yang vd. (2019) çalışmalarında, sahte haber tespitini denetimsiz öğrenme algoritmaları ile tespit etmeyi denemişlerdir. Bayes ağı ile; haber metni, kullanıcı görüşü ve kullanıcı güvenirliği arasındaki ilişkiyi yakalamaya çalışmışlardır. Veri seti olarak "LIAR" ve "BuzzFeed" veri tabanlarını kullanmışlardır. Çalışmalarını, literatürde bulunan denetimsiz sahte haber tespiti yöntemleri ile karşılaştırmışlardır. Çalışmalarının diğer dört çalışmadan daha iyi sonuç verdiğini deney sonuçları ile göstermişlerdir.

Altunbey Özbay (2020) tez çalışmasında, sürü zekası yöntemlerini kullanarak İngilizce dilinde sahte haber tespiti üzerine çalışmıştır. Veri seti olarak internet üzerinde sahte haber tespiti için kullanımına açık dört farklı veri tabanını kullanmıştır. Bu veri setlerini 33 farklı algoritma ile test etmiş ve sonuçlarını incelemiştir.

Yapılan çalışmalar incelendiğinde, literatürde farklı dillerde sahte haber tespiti üzerine çalışmalar bulunduğu fakat Türkçe dilinde yapılan çalışmaların ise Destek Vektör Makineleri ve Naive Bayes sınıflandırıcısı ile sınırlı kaldığı görülmüştür.

3. MATERYAL ve METOD

Bu bölümde, çalışmada kullanılan veri seti ve yöntemler açıklanmıştır.

3.1. Veri Seti

Bu çalışmada Twitter sosyal ağ platformunda paylaşılan sahte haber içeren tweet mesajlarının tahmin edilmesi amaçlanmıştır (Twitter, 2006). Bu nedenle belirlenen altı konuda tweet mesajları toplanmış ve manuel olarak etiketlenmişlerdir. Daha sonra, bu konulardan sahte haber sayısı fazla olan konular seçilmiş ve sonuç olarak üç konu modellerde kullanılmak üzere veri seti olarak belirlenmiştir.

3.1.1. Verilerin Toplanması

Bu bölümde, Twitter platformundan tweet mesajlarının toplanması, sahte haber tespiti yapılacak tweet konularının seçimi ve bu tweet mesajlarının veri tabanında saklanmasından bahsedilmiştir.

3.1.1.1. Tweet mesajlarının toplanması

Twitter sosyal ağ platformunda paylaşılan meajlara tweet adı verilmektedir. Bu tweet mesajlarının toplanması için Twitter tarafından "Twitter Search API" uygulama programlama arayüzü (API) sunulmaktadır. Bu arayüz "Standart", "Premium" ve "Enterprise" olmak üzere üç farklı sürümde kullanıcılara sunulmaktadır. Ücretsiz olan "Standart" sürümde son yedi gün içinde paylaşılmış olan tweet mesajları çekilebilmekte, ücretli olan "Premium" ve "Enterprise" sürümlerinde ise herhangi bir tarih sınırlaması bulunmamaktadır (Twitter Search API, 2018). Standart API'nin sadece son 7 gün içerisindeki tweet mesajları çekebilmesinden dolayı, etiketsiz tweet mesajlarını toplamak için kullanılmasından vazgeçilmiştir.

TweetScraper uygulaması ile twitter sitesindeki aranan konudaki tüm mesajlar JSON formatında dosyalar halinde çekilip bilgisayarda bir klasöre kaydedilebilmektedir. Bu uygulama ile çekilen veriler, Twitter API ile çekilen veriler kadar kapsamlı değildir.

Fakat Twitter API'den farklı olarak herhangi bir zaman aralığı olmadan tweet mesajları çekilebilmektedir (Github, 2019).

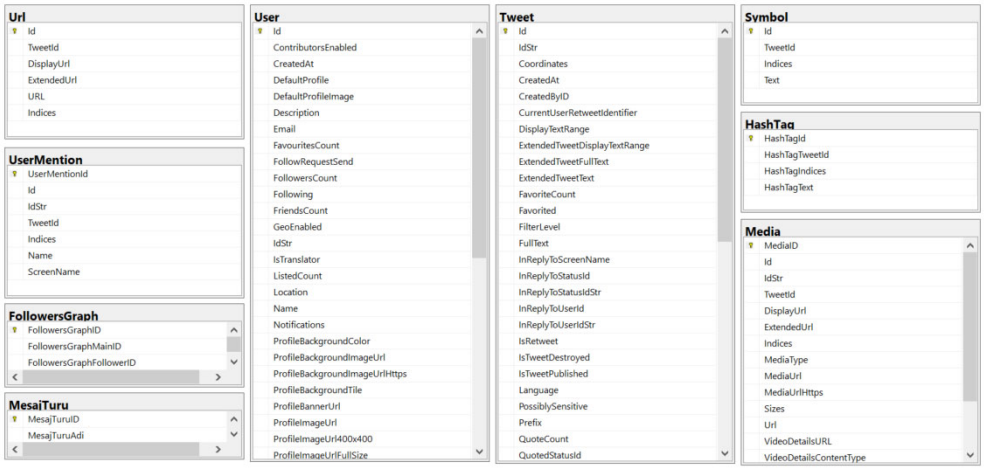
3.1.1.2. Konu seçimi

Haber makalelerinin doğruluğunu kontrol eden birçok site bulunmaktadır. Bu sitelerden biri olan Teyit.org platformu, organizasyonel ve editoryal süreçlerinin, Uluslararası Doğruluk Kontrolü Ağı'nın (IFCN - International Fact-checking Network) prensipleriyle uyuştuğunu gösteren IFCN sertifikasına sahiptir (Teyit.org, 2016). Tweet mesajlarının çekilmesinde konu seçimi için teyit.org haber doğrulama platformunda bulunan sahte haberlerden, sayıca fazla tweet mesajı bulunan konular seçilmiştir.

Tüm hastanelere 4440911 telefon numarasından ulaşılabileceği iddiası, Bayburt Havalimanı'nın yolcu garantili olarak yaptırılması iddiası, hamile kadına saldıran baklavacıların dağıttığı ücretsiz baklavaları almak isteyenlerin kuyruk oluşturduğu iddiası, Radamel Falcao isimli futbolcunun Galatasaray takımına geldiği iddiası, İstanbul Havalimanı'nda hava taksi yolunun çöktüğü iddiası ve Kuleli Askerî Lisesi'nin satıldığı iddiasından oluşan 76738 tane tweet mesajı TweetScraper uygulaması aracılığı ile toplanmıştır.

3.1.1.3. Veritabanının oluşturulması ve tweet mesajlarının kaydedilmesi

Çekilen tweet mesajlarını saklamak için ve Bölüm 3.1.2'de anlatılacak olan manuel etiketleme kullanıcı arayüzünde kullanılması amacı ile ilişkisel veritabanı oluşturulmuştur. Bu veritabanında Tweet mesajları "Tweet" tablosunda ve bu tweet mesajlarını gönderen kullanıcıların bilgileri ise "User" tablosunda tutulmaktadır. "Url", "UserMention", "Symbol", "HashTag" ve "Media" tabloları tweet mesajları ile ilgili detaylı bilgileri barındırır. Manuel etiketleme işleminde kullanılacak etiketler "MesajTuru" tablosunda tutulmaktadır. Her tweet mesajına ait etiket Tweet tablosundaki "MesajTuruID" alanında tutulmaktadır. "FollowerGraph" tablosunda ise kullanıcıların birbirlerini takip etme bilgisi tutulmaktadır. Şekil 3.1'de oluşturulan veri tabanına ait şema gösterilmiştir.



Şekil 3.1. Veritabanı şeması.

TweetScraper ile belirlenen altı konuda toplam 76738 tweet mesajı toplanmıştır. Daha sonra, JSON formatındaki dosyalar içinde bulunan bu tweet mesajlarını, ilişkisel veritabanına aktarmak için C# konsol uygulaması oluşturulmuş ve tüm bu tweet mesajları ilişkisel veritabanına kaydedilmiştir.

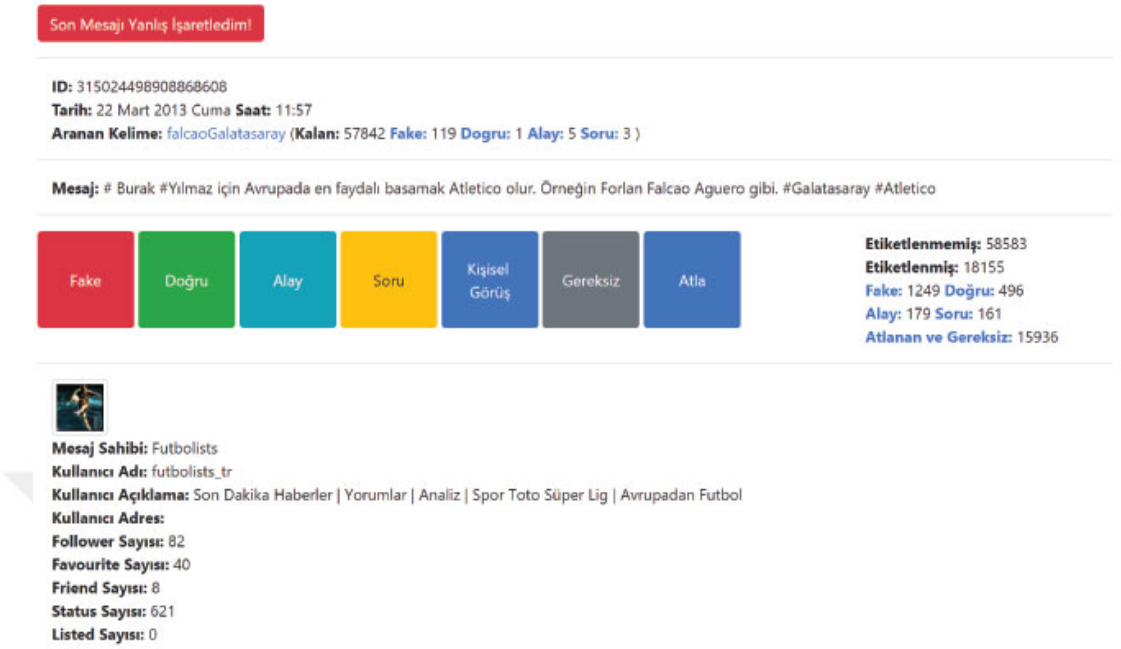
3.1.2. Verilerin Etiketlenmesi

Bu çalışma kapsamında kullanılacak olan denetimli öğrenme algoritmaları, sınıflandırma işlemini yapabilmeleri için etiketli verilere ihtiyaç duymaktadırlar. Bu nedenle, Twitter platformundan çekilen tweet mesajlarının kullanıcı tarafından manuel olarak etiketlenmesi amacıyla, bir kullanıcı arayüzü geliştirilmiştir.

3.1.2.1. Manuel etiketleme yapılabilmesi için kullanıcı arayüzünün oluşturulması

Manuel etiketlemenin yapılabilmesi için, ASP.NET MVC çatısı kullanılarak C# programlama dili ile bir kullanıcı arayüzü oluşturulmuştur. Toplanan tweet mesajları etiketleyiciye bir arayüz ile gösterilmiş ve etiketleyiciden "Sahte", "Doğru", "Alay", "Soru", "Kişisel Görüş", "Gereksiz" ve "Atla" etiketlerinden birini tercih etmesi istenmiştir: Oluşturulan arayüze ait ekran görüntüsü Şekil 3.2'de gösterilmiştir. Aynı zamanda kullanıcı arayüzünde bilgilendirme amacıyla, konu bazlı etiketli mesaj sayıları ve tüm etiket sayıları da gösterilmektedir. "Aranan Kelime" kısmında, sayısal bilgilerin önündeki "Sahte", "Doğru", vb. başlıklar tıklandığında, o konu ile ilgili etiketlenmiş

mesajların listesi görülebilmektedir. Sağ tarafta bulunan etiketler tıklandığında ise tüm konular için "Sahte", "Doğru", vb. ile etiketlenen mesajların listesi görülebilmektedir.



Şekil 3.2. Manuel etiketleme yapılabilmesi için oluşturulan kullanıcı arayüzü.

Belirlenen konular, teyit.org haber doğrulama platformu üzerinden ve internet haber sitelerinden teyit edildikten sonra, tweet mesajları, oluşturulan kullanıcı arayüzü aracılığıyla etiketlenmiştir (Teyit.org, 2016). Eğer bir tweet mesajı yanlış etiketlenirse veya yanlışlıkla farklı bir etiket seçilirse kullanıcı arayüzünün en üstünde bulunan "Son Mesajı Yanlış İşaretledim" butonu ile son etiketlenen mesajın etiketi iptal edilebilmektedir.

Veritabanında bulunan 76738 adet tweet mesajının 18021 tanesi etiketlenmiştir. Bu etiketlenen tweet mesajlarından 1249 tanesi sahte, 496 tanesi doğru, 179 tanesi alay, 161 tanesi soru, 40 tanesi kişisel görüş ve 15896 tanesi ise gereksiz olarak etiketlenmiştir.

3.1.2.2. Etiketlenen mesajların seçilmesi

Etiketlenen mesajlardan etiketi, sahte ve doğru olan tweet mesajları dikkate alınmıştır. Diğer etiketler çalışmada kullanılmamıştır. Tweet mesajları etiketlendikten sonra, etiket bazında (sahte, doğru) en az tweet mesajı sayısı 140 olarak görülmüştür. Her konuda eşit sayıda ve eşit etiketlerde tweet mesajları alınmıştır. Bu nedenle, 140 doğru ve 140

sahte tweet mesajına sahip olan konular seçilmiştir. Daha sonra iki metin arasındaki benzerliği hesaplayarak benzerlik puanı oluşturan, Levenshtein mesafe algoritması kullanılarak tweet mesajlarının her birinin diğerleri ile benzerlikleri ölçülmüş ve 0,5 değerinin altında benzerlik oranı çıkan tweet mesajları denetimli öğrenme ve denetimsiz öğrenme modellerinde kullanılmak üzere seçilmiştir.

$$lev_{\mathrm{a,b}}(\mathrm{i},\mathrm{j}) = \begin{cases} max(i,j), & \text{Eğer, } min(i,j) = 0, \\ min \begin{cases} lev_{\mathrm{a,b}}(\mathrm{i\text{-}1},\mathrm{j}) + 1 \\ lev_{\mathrm{a,b}}(\mathrm{i},\mathrm{j\text{-}1}) + 1 \\ lev_{\mathrm{a,b}}(\mathrm{i\text{-}1},\mathrm{j\text{-}1}) + 1_{(a_i \neq b_i)} \end{cases} & , \quad \text{diğer durumlar.} \end{cases} \tag{3.1}$$

İki metin arasındaki Levenshtein Uzaklığı Denklem 3.1 ile hesaplanır. $a_i = b_i$ olduğu zaman, $1_{(a_i \neq b_i)}$ gösterge fonksiyonu 0, diğer durumlarda 1'dir. $lev_{\mathrm{a,b}}(\mathrm{i},\mathrm{j})$, a metninin ilk i karakteri ile b metnin ilk j karakteri arasındaki uzaklığı hesaplar (Levenshtein, 1965).

Levenshtein Uzaklığı uygulanan, benzer tweet mesajları çıkarıldıktan sonra her konudan 90 doğru, 90 sahte tweet mesajı alınmıştır. Bu 3 konudaki tweet mesajları aşağıda açıklanmıştır.

Konu-1: Bayburt Havalimanı'nın T.C. Ulaştırma ve Altyapı Bakanlığı tarafından yıllık 2 milyon yolcu garantili olarak yaptırıldığı iddiasına ait gerçek tweet mesajına örnek olarak "Bayburt 'un yılda 2 milyon yolcu kapasiteli havalimanı olacakmış ahxnbabxbabd" tweet mesajı verilebilir. Sahte tweet mesajına örnek olarak ise "Bayburt 'a 2 milyon yolcu garantili havalimanı. Kendine güvenen bir ekonomi profesörü bunu bize anlatsın." tweet mesajı verilebilir. Sahte ve gerçek tweet mesajı örneklerine bakıldığında, "garantili" ve "kapasiteli" kelimelerinin tweet mesajının gerçek veya sahte olduğunu gösterdiği görülmektedir. Bu konudaki tweet mesajları, "Bayburt havalimanı" arama cümlesi kullanıldığında, Temmuz 2009 ile Eylül 2019 tarihleri arasındaki tweet mesajlarını kapsamaktadır.

Konu-2: Galatasaray futbol kulübünün Radamel Falcao isimli futbolcuyu transfer ettiği iddiasına ait gerçek tweet mesajına örnek olarak "Kafayı Falcao ile bozduk doğrudur gelecekse gelsin artık Galatasaray" veya "Bu gece de bekledik gelmedin @FALCAO ultrAslan Galatasaray" tweet mesajları verilebilir. Sahte tweet mesajına örnek olarak ise "@FALCAO Galatasaray camiamıza hayırlı olsun." veya "Ve Galatasaray Falcao ile anlaştığını Borsaya bildirdi!" gibi tweet mesajları verilebilir. İlk konudan farklı olarak bu konudaki tweet mesajlarında, sahte tweet mesajları ile gerçek tweet mesajları tamamen farklı kelimelerden oluşabilmektedir. Bu konudaki tweet mesajları, "Falcao Galatasaray" arama cümlesi kullanıldığında Nisan 2011 ile Ağustos 2019 tarihleri arasındaki tweet mesajlarını kapsamaktadır. Bu tarihlerde Galatasaray futbol takımı tarafından Radamel Falcao isimli futbolcunun transferi gerçekleşmemiş fakat 02 Ekim 2019 tarihinde bu transfer gerçekleşmiştir.

Konu-3: Kuleli Askerî Lisesi'nin T.C. Kültür ve Turizm Bakanlığı tarafından satıldığı iddiasına ait gerçek tweet mesajlarına örnek olarak, "Kültür ve Turizm Bakanı Mehmet Nuri Ersoy, Kuleli Askerî Lisesi 'nin satıldığına ilişkin sosyal medya ve basına yansıyan haberlerin tamamen asılsız olduğunu belirtti." tweet mesajı verilebilir. Sahte tweet mesajlarına örnek olarak ise "Kuleli Askerî Lisesi araplara satılmış" tweet mesajı verilebilir. Bu konudaki tweet mesajları incelendiğinde; sahte tweet mesajlarında "satılmış" kelimesi geçerken, gerçek tweet mesajları ise farklı kelimelerden oluşmaktadır. Bu konudaki tweet mesajları, "Kuleli Askerî Lisesi" arama cümlesi kullanıldığında Mayıs 2009 ile Ağustos 2019 tarihleri arasındaki tweet mesajlarından oluşmaktadır.

3.1.3. Kullanıcı ve Takipçilerinin Toplanması

Vosoughi vd. (2018) çalışmalarında, sahte haberin yayılımının Twitter sosyal ağ platformunda çok hızlı olduğunu ve sahte haberleri takip edenler ile sahte haberleri yayan kullanıcıları takip edenlerin sayısının oldukça fazla olduğunu belirtmişlerdir. Bu nedenle sosyal ağ analizi için, etiketli verilerden sahte haber yayan kullanıcılar ile gerçek haber yayan kullanıcılar ve bu kullanıcıların takipçileri toplanmıştır.

Kullanıcıların bilgilerinin ve takipçilerinin çekilmesi için "Twitter Search API" kullanılmıştır. Toplam 7.996.407 adet kullanıcı bilgisi "User" tablosuna aktarılmıştır. Bu kullanıcılardan sahte ve gerçek haber gönderen kullanıcıların 20.483.954 adet takip etkileşimi bilgisi "FollowerGraph" tablosuna kaydedilmiştir. Bu kullanıcılar sosyal ağ analizinde kullanılmak üzere veritabanında tutulmaktadır.

3.2. Doğal Dil İşleme (DDİ)

Doğal Dil İşleme, insanların iletişimde kullandıkları doğal dilleri, bilgisayarların anlaması için işleyen bir bilim dalıdır (Vosoughi, 2015). İlk örnekleri, doğal dillerin birbiri arasında çevrilmesini konu alan makine çevirisi ile başlamıştır. DDİ ilk zamanlarda geleneksel yöntemlerle yapılmış fakat ilerleyen zamanlarda makine öğrenmesi kullanarak etkili sonuçlar vermeye başlamıştır. Makine öğrenmesinin kullanılması ile DDİ tanınır hale gelmiş ve yapılan çalışmalar artmıştır. Bu çalışmada da makine öğrenmesi kullanarak DDİ yöntemleri ile sahte ve gerçek haber tespiti yapılmıştır.

3.3. Özellik Çıkarımı

Tweet mesajlarının makine öğrenmesi algoritmalarını verilmesi için öncelikle kelimelerin vektörler ile temsil edilmesi gerekmektedir. Bu bölümde, bu çalışmada kullanılan özellik çıkarımı yöntemlerinden bahsedilmiştir.

3.3.1. TF-IDF (Term Frequency - Inverse Document Frequency)

Makine öğrenmesi ile yapılan DDİ çalışmalarında, matematiksel işlemlerin yapılabilmesi için metinsel ifadelerin vektörel ifadelere dönüştürülmesi gerekmektedir. Bu yöntemlerden biri olan TF-IDF doğal dil işleme alanında sıkça kullanılmaktadır. TF-IDF, makine öğrenmesine girdi olarak verilecek olan verilerde tahmin yapmak için bir kelimenin ne kadar önemli olduğunun değerini bulan istatistiksel yöntemdir. Bu yöntemde önce terim sıklığı değeri hesaplanır. Terim sıklığında, her kelimenin her bir dokümanda kaç defa geçtiği hesaplanır. TF değerini hesaplamak için Denklem 3.2 kullanılır.

$f(t,d)$ = d dokümanındaki t teriminin sıklığı

$$TF(\mathrm{t,d}) = \frac{f(\mathrm{t,d})}{n_d} \tag{3.2}$$

Burada, n_d, d doküman dizisinin boyutunu (bu çalışmada tweet mesajlarının sayısı) temsil etmektedir. TF skorunun hesaplanmasından sonra ise ters doküman sıklığı değeri hesaplanır. Ters doküman sıklığında ise, bir kelimenin kaç farklı dokümanda geçtiği kontrol edilmektedir. Tersine doküman sıklığını hesaplamak için Denklem 3.3 kullanılır.

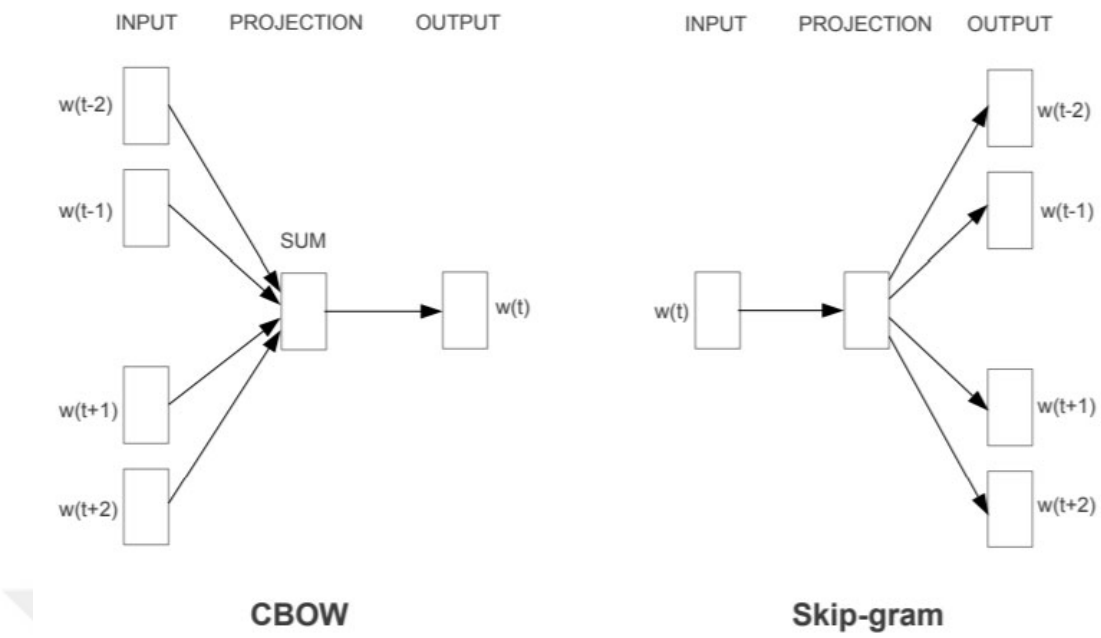
$$IDF(\mathrm{t}) = log(\frac{1+D}{D_\mathrm{t}}) \tag{3.3}$$

Denklemde D doküman dizisini (bu çalışmada, tweet mesajlarının dizisini), D_t ise t teriminin geçtiği doküman dizisini temsil etmektedir. Denklemde bulunan log fonksiyonu ise sönümleme fonksiyonudur (dampening function). d dokümanındaki her bir t teriminin TF-IDF skoru Denklem 3.4 ile hesaplanır;

$$TF - IDF(\mathrm{t,d}) = TF(\mathrm{t,d}) \times IDF(t) \tag{3.4}$$

3.3.2. Word2Vec

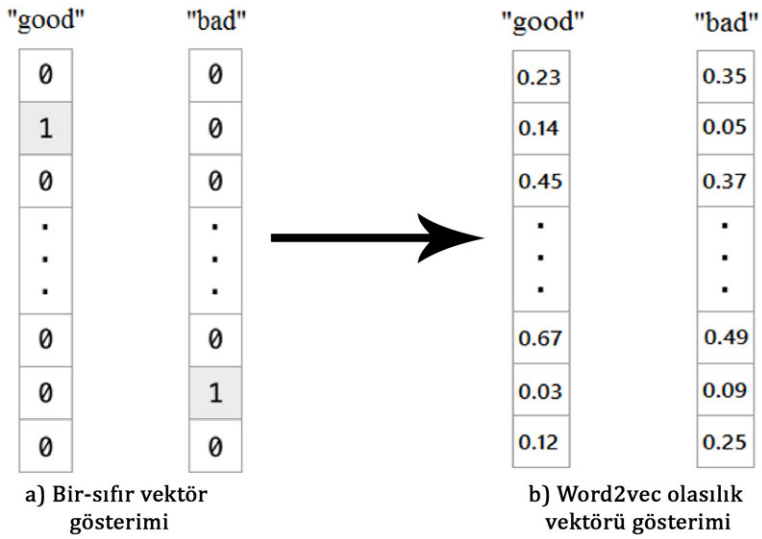
Word2vec kelime temsili modeli, bir kütüphanedeki kelimeleri girdi olarak alarak sinir ağı ile eğitip, kelimeler arasındaki benzerlikleri istenen boyutta bir vektör ile temsil eden yöntemdir. Skip-Gram ve CBOW (Continous Bag of Words) olmak üzere iki yöntem ile sinir ağı eğitilmektedir. Şekil 3.3'de bu yöntemlere ait şekil gösterilmiştir. Skip-Gram yönteminde bir kelime modele girdi olarak alınır ve girdi olarak modele giren kelimenin öncesi ve sonrasındaki n adet kelime tahmin edilir. Bu n sayısına pencere boyutu adı verilir. CBOW yönteminde ise girdi olarak bir kelimenin öncesi ve sonrasındaki n adet kelime alınır ve model tarafından o kelime tahmin edilir (Mikolov vd., 2013).



Şekil 3.3. CBOW ve Skip-Gram yöntemleri (Mikolov vd., 2013).

Word2vec, yapay sinir ağlarını kullanarak her kelimeye karşılık gelen bir vektör oluşturur. Vektör boyutunun, pencere boyutunun ve eğitilecek metinlerin algoritmaya önceden verilmesi gerekmektedir. Algoritma, kelimeleri bir-sıfır vektörüne dönüştürür. Algoritmaya verilen metinlerde geçen tüm kelimeler tespit edilir. Bu kelimeler alfabetik olarak sıralanır. N kelime sayısı olmak üzere, "1xN" boyutunda vektörler oluşturulur. İlk kelime için vektörün 1. elemanı bir, diğer elemanları sıfır alınır. 2. kelime için vektörün 2. elemanı bir, diğerleri sıfır alınır. Bu şekilde son kelimeye kadar devam edilir (Şekil 3.4 a). Cümlelerde geçen kelimelerin bir-sıfır vektörleri bulunur. Cümleler, cümle içinde geçen kelimelerin bir-sıfır vektörlerinden oluşan matrislere çevrilir. CBOW yönteminde; bir kelimenin tüm cümlelerde geçen pencere boyutu kadar önceki ve pencere boyutu kadar sonraki kelimeleri sinir ağına verilir. Sinir ağı, olasılıksal olarak kelimeye ait istenen boyutta vektörü oluşturur. Skip-Gram yönteminde ise bir kelime algoritmaya verilir ve algoritmanın bu kelimeden önce ve sonra gelen pencere boyutu kadar kelimenin istenen boyutta olasılıksal vektörünü oluşturur (Şekil 3.4 b).

Word2Vec yöntemi tarafından, her kelime için belirlenen boyutta bir vektör oluşturulmaktadır. Algoritmalara verilen matris boyutlarının eşit olması gerekmektedir. Tweet mesajları incelendiğinde, 50 kelimenin üstünde tweet mesajlarının sayısının çok



Şekil 3.4. Kelimelerin vektörel gösterimleri (Vo vd., 2019).

az olduğu görüldüğünden dolayı, bir tweet mesajındaki kelime sayısı en fazla 50 olarak belirlenmiştir. 50 kelimeden az olan tweet mesajlarını 50 kelimeye tamamlamak için sonuna Word2vec vektör boyutuyla aynı boyutta olan sıfır vektörleri eklenmiştir. 50 kelimeden fazla olan tweet mesajlarının ise ilk 50 kelimesi dikkate alınmıştır.

3.3.3. Doc2Vec

Doc2Vec modeli ise, Word2Vec modeline benzer şekilde kelime temsili için bir vektör hesaplamaktadır. Word2Vec modelinde her bir kelime için belirlenen boyutta vektör oluşturulurken, Doc2vec yönteminde her bir doküman (bu çalışmada her bir tweet mesajı) için belirlenen boyutta bir vektör oluşturulur (Le ve Mikolov, 2014). Word2vec modelinin CBOW yöntemine benzer şekilde dokümanda bulunan kelimeleri girdi olarak alır ve her doküman için belirlenen boyutta dokümanı temsil edecek bir vektörü çıktı olarak vermektedir.

3.4. Denetimli Makine Öğrenmesi

Makine öğrenmesi, sensörler ve algılayıcılardan toplanan veya veritabanında biriken çok sayıda verinin sınıflandırılması, kümelenmesi veya tahminini olanaklı kılan algoritmaların tasarım ve geliştirme süreçlerini konu alan bir bilim dalıdır. Makine öğrenimi ile bilgisayarlara karmaşık örüntüleri tanıma ve karar verme becerisi

kazandırılması amaçlanmaktadır. İstatistik, olasılık, veri madenciliği, örüntü tanıma ve yapay zeka gibi alanlarla yakından ilişkilidir (Alpaydin, 2016).

Denetimli öğrenme algoritmalarında veriler ve bu verilerin etiketleri de giriş olarak makine öğrenmesi algoritmasına verilir. Algoritma tarafından, algoritmaya verilen veriler arasında bir ilişki ağı öğrenilerek, yeni verilerin hangi çıktıya sahip olacağı tahmin edilir (Goodfellow vd., 2017).

3.4.1. Yapay sinir ağı temelli olmayan algoritmalar

Bu çalışmada K-Nearest Neighbor (KNN), Support Vector Machine (SVM) ve Random Forest (RF) yapay sinir ağı temelli olmayan denetimli öğrenme algoritmaları kullanılmıştır.

3.4.1.1. K-Nearest Neighbor (KNN) algoritması:

KNN algoritması, dışarıdan bir k hiper parametresi almaktadır. Bu k parametresi sınıflandırılacak olan bir veriye en yakın k adet komşuyu belirtir. Sınıflandırılacak olan veriye en yakın olan k adet elemanla uzaklık ölçülür. Bu uzaklıklar Öklid (Denklem 3.5), Manhattan (Denklem 3.6), Minkowski (Denklem 3.7), Hamming ve Kosinüs uzaklık metotları gibi farklı metotlar ile ölçülebilmektedir. Uzaklık hesaplama metotlarının sonucuna göre, sınıflandırılacak olan yeni veri hangi sınıfa daha yakın ise o sınıfa dahil edilir (Cunningham ve Delany, 2020).

Öklid uzaklığı:

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{3.5}$$

Manhattan uzaklığı:

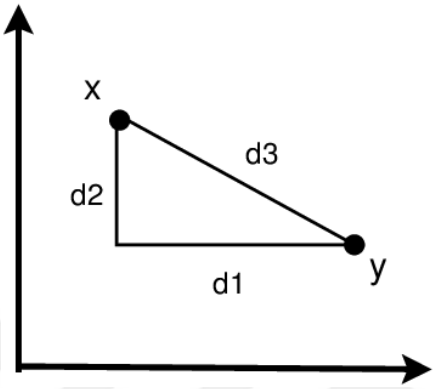
$$\sum_{i=1}^{k}|x_i - y_i| \tag{3.6}$$

Minkowski uzaklığı:

$$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q} \tag{3.7}$$

Denklem 3.7'de q değişkenine 1 değeri verildiğinde Manhattan uzaklığı, 2 değeri verildiğinde ise Öklid uzaklığı denklemine ulaşılmaktadır.

Öklid uzaklığı iki nokta arasındaki geometrik uzaklığı verir. Manhattan uzaklığı ise iki noktada geçen ve dik kesişen doğruların toplamıdır. Şekil 3.5'de, $d3$, Öklid uzaklığını, $d1 + d2$ ise Manhattan uzaklığını göstermektedir.



Şekil 3.5. Öklid ve manhattan uzaklıkları (Kiritchenko, 2005).

Bu çalışmada Sklearn (Pedregosa vd., 2012) kütüphanesinin "KNeighborsClassifier" metodu kullanılmıştır. Bu metoda k hiper parametresi $k = 2$ olarak verilmiştir. Uzaklık hesaplama metodunu belirleyen "p" parametresi ile manhattan, euclidean ve minkowski uzaklık fonksiyonları kullanılmıştır. KNeighborsClassifier metodunun p parametresi 1 ise manhattan, 2 ise euclidean ve 3 ve üzerinde ise minkowski fonksiyonları kullanılabilmektedir.

3.4.1.2. Support Vector Machines (SVM) algoritması:

SVM algoritması, eğitim verisinde bulunan vektörleri farklı sınıflara ayırmak için en optimal olan hiper düzlemleri belirler. Lineer olmayan düzlemleri belirlemek için çekirdek fonksiyonlar kullanılmaktadır (Vapnik, 1995). En çok kullanılan çekirdek fonksiyonları Doğrusal Çekirdek Fonksiyonu (Denklem 3.8), Polinom Çekirdek Fonksiyonu (Denklem 3.9),Sigmoid Çekirdek Fonksiyonu (Denklem 3.10) ve Radyal Tabanlı Çekirdek (Denklem 3.11) Fonksiyonudur (Başakın vd., 2019).

Doğrusal Çekirdek Fonksiyonu:

$$K(x_i, x_j) = {x_i}^T . x_j \tag{3.8}$$

Polinom Çekirdek Fonksiyonu:

$$K(x_i, x_j) = (\gamma {x_i}^T . x_j + r)^d, \gamma > 0 \tag{3.9}$$

Sigmoid Çekirdek Fonksiyonu:

$$K(x_i, x_j) = tanh(\gamma {x_i}^T . x_j + r)^d \tag{3.10}$$

Radyal Tabanlı Çekirdek Fonksiyonu:

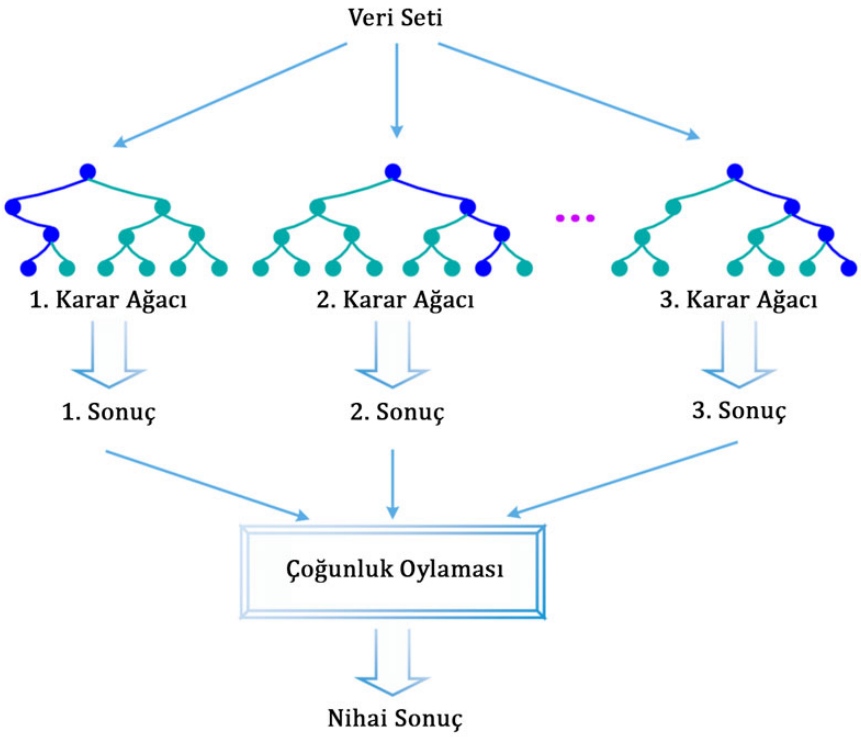
$$K(x_i, x_j) = exp(-\gamma ||x_i - x_j||^2), \gamma > 0 \tag{3.11}$$

Destek vektör makineleri algoritmasının çekirdek fonksiyonu denklemlerinde d polinomun derecesini, γ ise çekirdek boyutunu belirtmektedir.

Bu çalışmada, Sklearn (Pedregosa vd., 2012) kütüphanesinin "SVC" metodu kullanılmıştır. Lineer olmayan düzlem belirlemek için gereken çekirdek fonksiyonunu parametresi olan "kernel" ile, linear, poly (2., 3., 4 5. ve 6. dereceden polinom), RBF ve sigmoid algoritmaları kullanılmıştır.

3.4.1.3. Random Forest (RF) algoritması:

Bir karar ağacı algoritması olan RF algoritmasında, algoritma farklı alt setler seçerek farklı karar ağaçları oluşturur. Her oluşan karar ağacı tahminde bulunur. Daha sonra sınıflandırmada bu tahminler için en yüksek oyu olan değer seçilir. Şekil 3.6'da bu durum gösterilmiştir. Birden çok alt set oluşturulmasından dolayı karar ağaçlarının aşırı öğrenme problemi bu algoritmada azalmaktadır (Breiman, 2001).



Şekil 3.6. Rastgele Orman Yapısı (Cao vd., 2017).

Bu çalışmada, Sklearn (Pedregosa vd., 2012) kütüphanesinin "RandomForestClassifier" metodu kullanılmıştır. Ormandaki ağaç sayısını belirten hiper parametre olan "n_estimators" parametresi 50, 100, 500 ve 1000 olarak kullanılmıştır. Bölme kalitesi ölçütünü belirleyen "criterion" parametresi ile "gini" ve "entropy" fonksiyonları kullanılmıştır.

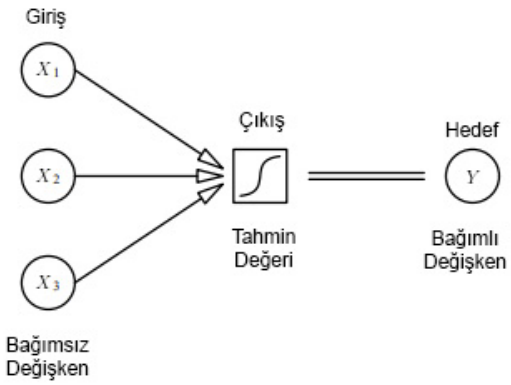
3.4.2. Derin öğrenme algoritmaları

Sinir ağları, çıktısı doğrusal olmayan problemlerin çözümünde kullanılırlar. İnsan beyninin öğrenme yapısını taklit etmektedirler. Sinir ağını taklit ettikleri için Yapay Sinir Ağları (YSA) adı verilmiştir. YSA kendisine girdi olarak verilen örnekler ile ilgili bilgileri toplayarak bu bilgilerden genel bir model çıkartır. Daha sonra hiç görmediği veri geldiğinde ise bu genel model ile bir karar verir.

YSA'nın elemanları sinir sistemindeki yapıların isimlerini almıştır. YSA'da her bir düğümü nöron olarak adlandırılır. Sinapslar ise bu nöronları birbirine bağlamaktadır. Giriş nöronlarının çıkış nöronlarını ne kadar etkileyeceği ağırlık değerleri ile hesaplanır.

Başlangıçta ağırlıklar rastgele bir değer alır. YSA öğrenme olayını gerçekleştirdikçe bu ağırlık değerleri güncellenir. Girişlerin her biri kendi ağırlık değeriyle çarpılır. Daha sonra ağırlıklarıyla çarpılan tüm giriş değerleri bias değerleri ile toplanır. Sonuçta oluşan giriş değerlerinin tümü toplanarak aktivasyon fonksiyonuna iletilir. Aktivasyon fonksiyonu ise üretilecek çıktıyı belirler. Daha sonra kayıp fonksiyonu olması gereken değerle çıktı değerine bakarak ağırlık ve bias değerlerini günceller.

Perceptron, en küçük YSA öğrenme birimidir. Perceptron sadece giriş ve çıkış nöronlarından oluşmaktadır. Şekil 3.7'de Perceptron yapısı gösterilmiştir.



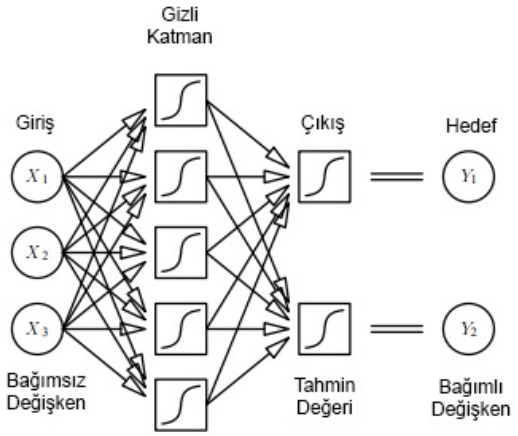
Şekil 3.7. Perceptron yapısı (Sarle, 1994).

$$Z = \sum_{i=1}^{n} (W_i.x_i + b_i) \tag{3.12}$$

Denklem 3.12'de Z Perceptron'un çıkış değerini, W her bir nöronun ağırlık değerini, x giriş nöronlarını ve b her bir nöronun bias değerini temsil etmektedir.

Multi Layer Perceptron (MLP) ise gizli katmanları bulunan algılayıcılardır. YSA da MLP'dir. Yapay Sinir Ağları bir Giriş Katmanı, bir Gizli Katman ve bir Çıkış katmanı olmak üzere 3 katmandan oluşur. Şekil 3.8'de bir YSA yapısı gösterilmiştir.

Aktivasyon fonksiyonları: Transfer fonksiyonu olarak da bilinen aktivasyon fonksiyonları, ağırlıkları ile çarpılıp bias eklenen nöronun aktif edilip edilmeyeceğine veya çıkışı ne kadar etkileyeceğine karar verir (Mercioni ve Holban, 2020). Aşağıda sık kullanılan aktivasyon fonksiyonlarından birkaçı açıklanmıştır.



Şekil 3.8. Yapay Sinir Ağı yapısı (Sarle, 1994).

- Sigmoid Fonksiyonu:

  Sigmoid fonksiyonu 0 ile 1 arasında bir değer almaktadır. Sigmoid aktivasyon fonksiyonu YSA'nın sınıflandırma problemlerinde çok sık kullanılmaktadır (Denklem 3.13).

$$S(x) = \frac{1}{1+e^{-x}} \tag{3.13}$$

- Tanh Fonksiyonu:

  Sigmoid fonksiyonuna benzeyen Tanh fonksiyonu, -1 ile 1 arasında bir değer almaktadır (Denklem 3.14).

$$tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{3.14}$$

- ReLU Fonksiyonu:

  Sigmoid fonksiyonu 0 ile 1 arasında bir değer almaktadır. Sigmoid aktivasyon fonksiyonu YSA'nın sınıflandırma problemlerinde çok sık kullanılmaktadır (Denklem 3.15). Tüm negatif değerler sıfır değerini döndürdüğü için model uygun eğitilmemektedir. Negatif değerlerin yok olmasını engellemek için negatif değerler sıfıra çok yakın 0,001 gibi bir α sayısı ile çarpılır (Denklem 3.16). Bu aktivasyon fonksiyonuna ise sızıntı ReLU aktivasyon fonksiyonu adı verilmektedir.

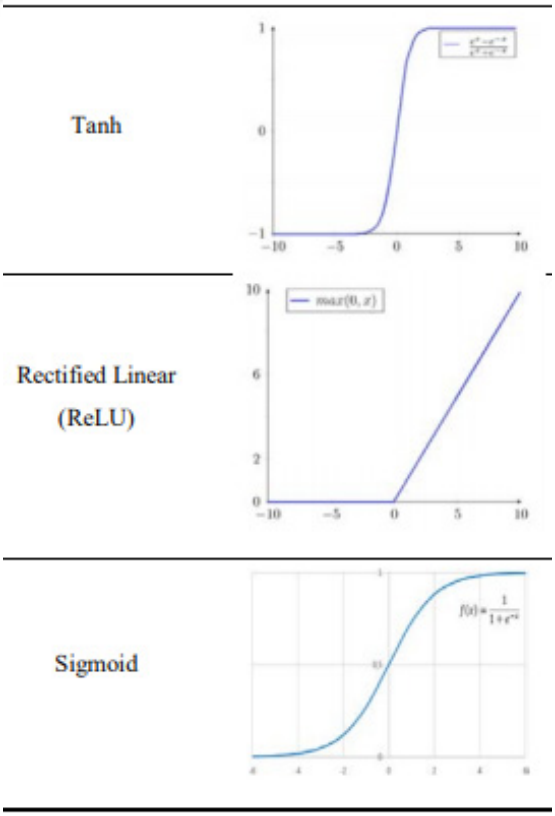
$$ReLU(x) = \begin{cases} x \leq 0 \text{ için}, & 0 \\ x > 0 \text{ için}, & x \end{cases} \tag{3.15}$$

$$ReLU(x) = \begin{cases} x \leq 0 \text{ için}, & \alpha.x \\ x > 0 \text{ için}, & x \end{cases} \tag{3.16}$$

- Softmax Fonksiyonu:

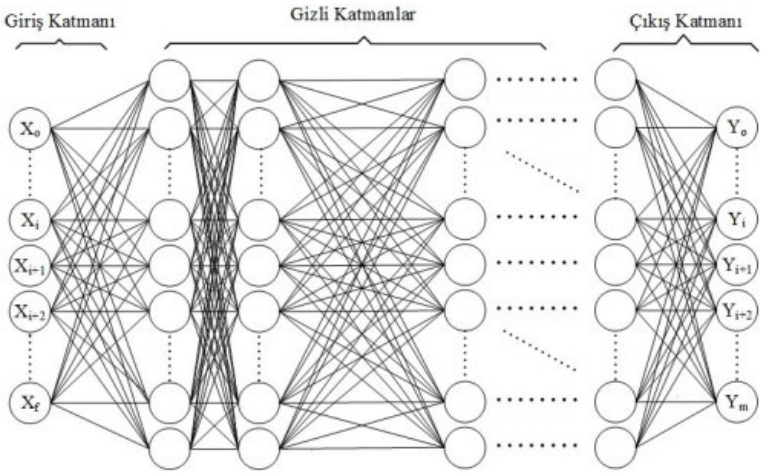
  Softmax aktivasyon fonksiyonu verileri 3 veya daha fazla sınıfa ayırmak için kullanılır (Denklem 3.17). Girişin kümeye ait olup olmadığını olasılıksal olarak veren aktivasyon fonksiyonudur. Şekil 3.9'da aktivasyon fonksiyonlarının grafikleri verilmiştir.

$$\sigma(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \tag{3.17}$$



Şekil 3.9. Aktivasyon Fonksiyonları (Ser ve Bati, 2019).

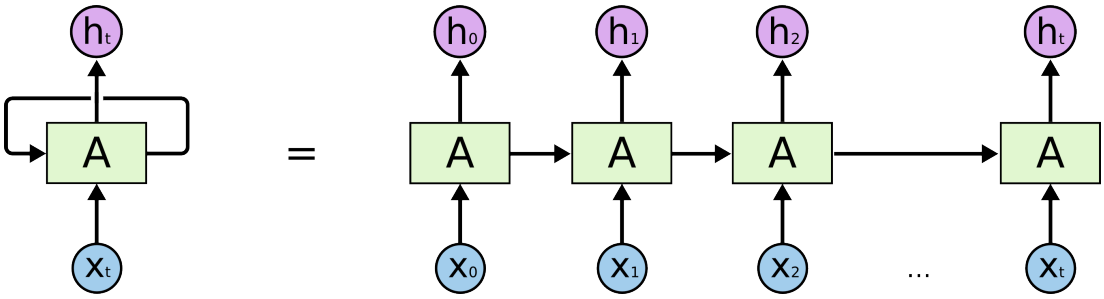
Birden fazla Gizli Katmana sahip Yapay Sinir Ağlarına Derin Sinir Ağları denir. Bu Sinir ağlarının öğrenme olayına ise Derin Öğrenme adı verilir. Şekil 3.10'da bir derin sinir ağının yapısı gösterilmiştir. Bu çalışmada kullanılan derin öğrenme yöntemleri bu başlık altında açıklanmıştır.



Şekil 3.10. Derin Sinir Ağı Yapısı.

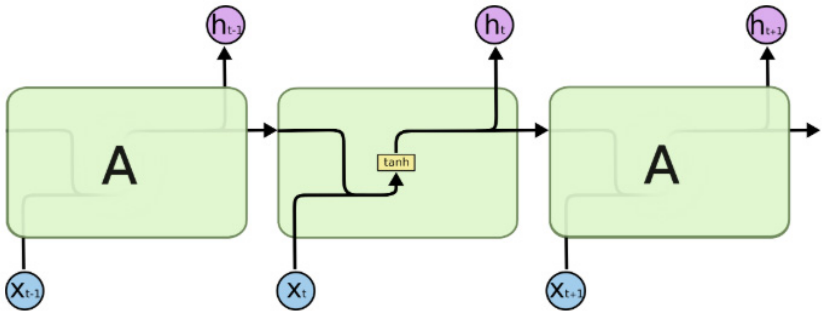
3.4.2.1. Recurrent Neural Network (RNN) algoritması:

İnsanlar bir metni okurken bir önceki kelime ve cümleleri de aklında tutarak metinden anlam çıkarır. Eskiden öğrenilmiş bilgileri, yeni öğrenilen bilgilerle güncelleyerek öğrenme olayını gerçekleştirir. İnsan doğasını taklit eden sinir ağlarından, geleneksel sinir ağlarında çıkarım yapılırken geçmişteki veriler dikkate alınmaz. RNN'lerde ise geçmişteki veriler dikkate alınarak çıkarım yapılır. Bu özelliği nedeniyle RNN, Doğal Dil İşleme (DDİ) alanında çok sık kullanılır. RNN modeli, birden fazla sinir ağının tekrarlanması ile oluşur. Her adımdaki çıktı bir sonraki sinir ağına iletilir. Bu durum Şekil 3.11'de gösterilmiştir.



Şekil 3.11. RNN yapısı (Olah, 2015b)

RNN modülünün iç yapısı Şekil 3.12'de verilmiştir. Bir önceki adımdan gelen çıktıyı alarak "tanh" aktivasyon fonksiyonundan geçirerek bir sonraki adıma iletmektedir.



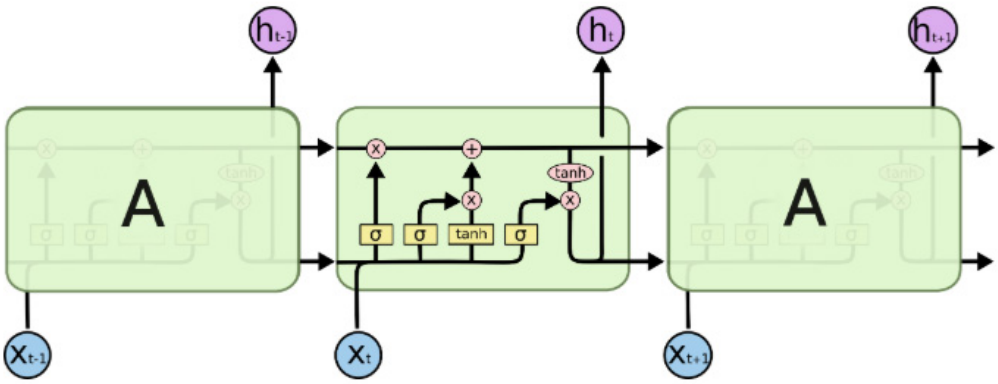
Şekil 3.12. RNN modülünün iç yapısı (Olah, 2015b)

RNN ile uzun cümlelerde veya metinlerde cümle veya metnin ilk bölümlerindeki (kelimeleri veya cümlelerindeki) veriler son bölümlere kadar taşınamamaktadır. Dolayısıyla uzun cümle ve metinlerde başlarda bulunan önemli veriler cümle veya metin sonlarında unutulmaktadır. Örneğin; "Bulutlar gökyüzündedir." cümlesinde RNN modeline "bulutlar" kelimesi girdi olarak verildiğinde "gökyüzündedir" kelimesini tahmin edebilir. Fakat; "Ben Türkiye'de büyüdüm ve akıcı Türkçe konuşurum." cümlesinde Türkçe tahmin edilmek istendiğinde, cümle başındaki "Türkiye'de" kelimesinin hatırlanması gerekmektedir. Bu nedenle RNN modellerinin hafızası kısa ve yetersiz kalmaktadır. Bu sorunun çözümü için Long-Short Term Memory (LSTM) modeli önerilmiştir (Olah, 2015b).

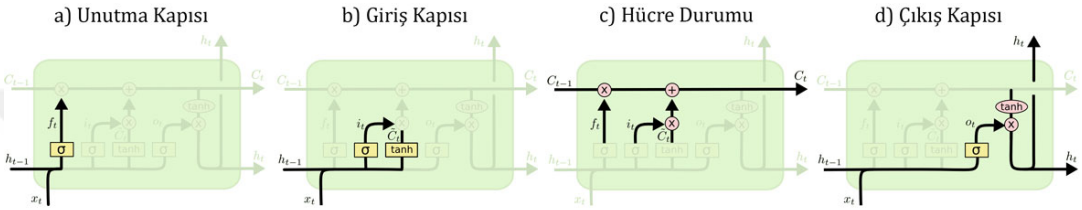
3.4.2.2. Long-Short term memory (LSTM) algoritması:

Hochreiter ve Schmidhuber (1997) tarafından önerilen LSTM modeli, RNN modelindeki uzun süreli hatırlama sorununa çözüm olarak üretilmiştir. LSTM gibi tüm RNN modelleri Şekil 3.11'de görüldüğü gibi tekrarlayan yapıdadır. Sadece modüllerin iç yapısında farklılık bulunur. Şekil 3.13'de LSTM modülünün iç yapısı görülmektedir. Bu yapının detayları sonraki bölümlerde anlatılacaktır.

LSTM modülü; "Unutma Kapısı" (Şekil 3.14 a), "Giriş Kapısı" (Şekil 3.14 b) ve "Çıkış Kapısı" (Şekil 3.14 d) olmak üzere 3 kapıdan oluşur. Ayrıca bir tür taşıma bandı görevi gören "Hücre Durumu" (Şekil 3.14 c) yapısı vardır. Ayrıca bellek yapılarında kullanılan şekillerin açıklaması Şekil 3.15'de verilmiştir.



Şekil 3.13. LSTM modülünün iç yapısı (Olah, 2015b)



Şekil 3.14. LSTM kapıları (Olah, 2015b)

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{3.18}$$

Unutma kapısı, hücre durumundaki bilginin saklanıp saklanmayacağına karar veren kapıdır. Denklem 3.18'de unutma kapısının matematiksel formülü verilmiştir. Denklemde; f_t, t zamanındaki unutma kapısı vektörünü, W_f, unutma kapısının ağırlık değerini, b_f, unutma kapısının bias değerini, h_{t-1}, bir öndeki zamandaki (t-1 zamanı) çıkış değerini ve x_t, mevcut zamandaki giriş değerini belirtmektedir.

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{3.19}$$

Denklem 3.19'da; i_t, mevcut zamandaki giriş kapısı vektörünü, W_i giriş kapısının ağırlık değerini, b_i giriş kapısının bias değerini, h_{t-1} bir önceki zamandaki çıkış değerini ve x_t ise mevcut zamandaki giriş değerini belirtir.

Şekil 3.15. Bellek yapılarında kullanılan işaretler (Olah, 2015b)

$$\widetilde{C}_t = tanh(W_C.[h_{t-1}, x_t] + b_C) \tag{3.20}$$

Denklem 3.20'de; C_t, mevcut zamandaki hücre durumuna eklenebilecek yeni aday vektörü, W_C aday değerin ağırlık değerini, b_C aday değerin bias değerini, h_{t-1} bir önceki zamandaki çıkış değerini ve x_t ise mevcut zamandaki giriş değerini belirtir.

Giriş Kapısı, hücre durumunda hangi yeni bilgilerin depolanacağına karar verir. İlk olarak, giriş kapısındaki sigmoid katman hangi değerlerin güncelleneceğine karar verir (Denklem 3.19). Daha sonra tanh katmanı, hücre durumuna eklenebilecek yeni aday değerlerin ($\widetilde{C}_t$) bir vektörünü oluşturur (Denklem 3.20).

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{3.21}$$

Hücre durumu modülü, unutma kapısı ve giriş kapısında saklanıp saklanmayacağına karar verilen bilgileri çıkış hücre durumu olarak iletir (Denklem 3.21). Denklem 3.21'de C_t, mevcut zamandaki hücre durumu vektörünü, C_{t-1}, ise bir önceki zamandaki hücre durumunu, f_t, mevcut zamandaki unutma kapısının kararını ve i_t, mevcut zamandaki giriş kapısı değerini belirtir.

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{3.22}$$

Denklem 3.22'de; o_t, mevcut zamandaki çıkış kapısı vektörünü, W_o çıkış kapısının ağırlık değerini, b_o çıkış kapısının bias değerini, h_{t-1} bir önceki zamandaki çıkış değerini ve x_t ise mevcut zamandaki giriş değerini belirtir.

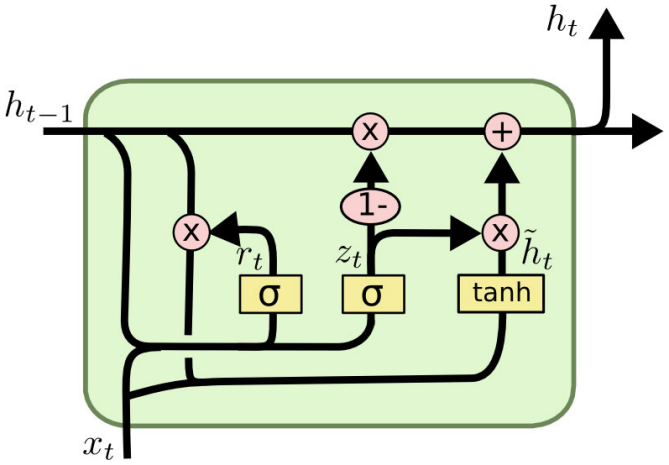
$$h_t = o_t * tanh(C_t) \tag{3.23}$$

Çıkış kapısında ilk olarak, hücre durumunun ne kadarının çıktısının verileceğine karar veren sigmoid katman çalışır (Denklem 3.22). Son olarak, hücre durumunun tanh değeri ile çıkış kapısındaki sigmoid katmanının değeri çarpılır. Sigmoid katmanı 0 ile 1 arasında değer verir bu sayede hücre değerinin ne kadarının çıkışa aktarılacağı belirlenmiş olur (Denklem 3.23).

Denklem 3.23'de; h_t, mevcut zamandaki çıkış değerini, o_t mevcut zamandaki çıkış kapısı değerini ve C_t ise mevcut zamandaki hücre durumu değerini belirtir.

3.4.2.3. Gated Recurrent Units (GRU) algoritması:

Cho vd. (2014) tarafından önerilen GRU, LSTM'nin sadeleştirilmiş halidir. LSTM'de bulunan unutma kapısı ve giriş kapısı, GRU'de sıfırlama kapısı olarak birleştirilmiştir. Bu durum Şekil 3.16'da gösterilmiştir. LSTM'e göre daha basit yapıları olduğu için GRU'in hesaplama işlemi LSTM'e göre daha hızlı gerçekleşmektedir.



Şekil 3.16. GRU modülünün iç yapısı (Olah, 2015b)

$$z_t = \sigma(W_z.[h_{t-1}, x_t] + b_z) \tag{3.24}$$

Denklem 3.24'de z_t, güncelleme kapısı vektörünü, W_z, güncelleme kapısının ağırlık değerini, b_z, güncelleme kapısının bias değerini, x_t, mevcut zamandaki giriş değerini ve h_{t-1}, bir önceki zamandaki çıkış değerini temsil etmektedir.

$$r_t = \sigma(W_r.[h_{t-1}, x_t] + b_r) \tag{3.25}$$

Denklem 3.25'de r_t, sıfırlama kapısı vektörünü, W_r, sıfırlama kapısının ağırlık değerini, b_r, sıfırlama kapısının bias değerini, x_t, mevcut zamandaki giriş değerini ve h_{t-1}, bir önceki zamandaki çıkış değerini temsil etmektedir.

$$\widetilde{h}_t = tanh(W_h.[r_t * h_{t-1}, x_t] + b_h) \tag{3.26}$$

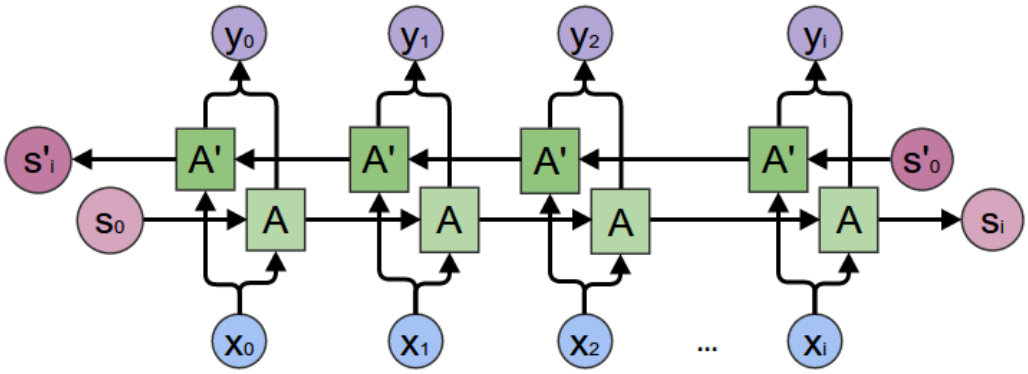
Denklem 3.26'da, $\widetilde{h}_t$, mevcut bellek içeriğini, r_t, sıfırlama kapısı vektörünü, W_h, çıkış değerinin ağırlık değerini, b_h, çıkış değerinin bias değerini, x_t, mevcut zamandaki giriş değerini ve h_{t-1}, bir önceki zamandaki çıkış değerini temsil etmektedir.

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h}_t \tag{3.27}$$

Denklem 3.27'de h_t, çıkış değerini, z_t, güncelleme kapısı vektörünü, h_{t-1}, bir önceki zamandaki çıkış değerini ve $\widetilde{h}_t$, mevcut bellek içeriğini temsil etmektedir.

3.4.2.4. Bidirectional Recurrent Neural Network (BiRNN) algoritması:

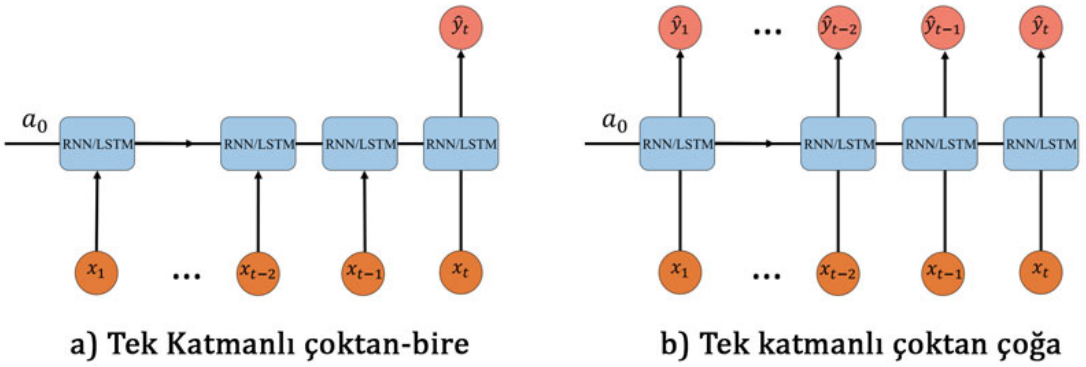
Schuster ve Paliwal (1997) tarafından önerilen modelde bir katman ileri beslemeli olarak ilerlerken bir başka katmanda sonran başlayarak geri beslemeli olarak ilerlemektedir. Hem ileri beslemeli hem de geri beslemeli katmanlar t zamanındaki çıkış değerini günceller. Bu durum Şekil 3.17'de gösterilmiştir. Bu yapı hem RNN'de hem LSTM'de hem de GRU'de aynı şekilde modellenmektedir. Şekil 3.17'de A ve A' olarak görülen birimler; RNN, LSTM veya GRU birimleri olabilir.



Şekil 3.17. BiRNN yapısı (Olah, 2015a)

BiRNN'de, doğal dil işleme alanı düşünüldüğünde; cümledeki ilk kelimeler tahmin edilirken, tahmin edilmek istenen bu kelimeler cümlede sonra gelen kelimelere bağımlı olabilmektedir. BiRNN, bir kelime tahmini yapılırken, sonraki kelimelerin de dikkate alınmasını sağlamaktadırlar.

Farklı uygulamalar için farklı RNN mimarileri bulunmaktadır. Şekil 3.18'de en çok kullanılan RNN mimarileri verilmiştir. Girişi çok boyutlu ama çıkışı tek boyutlu olan RNN mimarisine çoktan-bire (Şekil 3.18 a), Girişi de çıkışı da çok boyutlu olan RNN mimarisine ise çoktan-çoğa (Şekil 3.18 b) RNN mimarisi denir (Hao vd., 2019). Çoktan-bire RNN mimarileri genelde metin sınıflandırma ve duygu analizi gibi uygulamalarda kullanılır. Örneğin; algoritmaya bir cümle verilir ve bu cümlenin kızgın, mutlu gibi duygusu algoritmadan çıktı olarak alınır. Yine bu çalışmada yapılan tweet mesajının algoritmaya verilip bu tweet mesajının sahte veya gerçek olduğunun tahmininin yapılması çoktan-bire RNN mimarileri ile yapılır. Çoktan-çoğa RNN mimarileri ise daha çok makine çevirisi problemlerinde kullanılır. Örneğin; algoritmaya Türkçe verilen bir cümleyi, algoritma, farklı dillere çevirerek çıktı olarak üretir. Çoktan-çoğa RNN mimarilerinde çıkış boyutu, giriş boyutuyla farklı boyutta olabilir. Şekil 3.18'de görülen RNN/LSTM birimleri yerine; GRU, BiGRU ve BiLSTM birimleri ile de temsil edilebilmektedir.



Şekil 3.18. Farklı RNN mimarileri (Hao vd., 2019)

3.5. Denetimsiz Makine Öğrenmesi

Denetimsiz öğrenme algoritmalarında, etiketsiz veriler algoritmaya girdi olarak verilir. Ayrıca algoritmanın bu verileri kaç kümeye ayıracağı da bildirilir. Algoritma tarafından verilerin ilişkileri ve yapıları öğrenilerek veriler en anlamlı şekilde ve algoritmaya verilen sayıda kümeye ayrılır (Goodfellow vd., 2017).

Bu çalışmada K-means, Non-negative Matrix Factorization (NMF) ve Linear Discriminant Analysis (LDA) denetimsiz öğrenme algoritmaları kullanılmıştır.

3.5.1. K-means algoritması

En yaygın kullanılan denetimsiz öğrenme algoritması olan K-means algoritması, dışarıdan hiper parametre olarak küme sayısını almaktadır. Verilen k adet kümenin her biri için rastgele merkez nokta atanarak merkez noktaya en yakın elemanlar o kümeye dahil edilir. Böylece algoritma, k adet küme oluşturmaya çalışır.

Rastgele atanan merkez noktalar kümelemede zayıflıklar oluşturabilmektedir. Bu problemin önüne geçebilmek ve daha iyi merkez nokta seçilebilmeyi sağlamak için "K-means++" algoritması geliştirilmiştir (Arthur ve Vassilvitskii, 2006).

Bu çalışmada, Sklearn (Pedregosa vd., 2012) kütüphanesinin KMeans metodu kullanılmıştır. Küme sayısını belirten "n_clusters" parametresi sahte ve sahte olmayan etiketlerini temsil etmek üzere 2 olarak belirlenmiştir. Bu çalışmada, merkez noktalarını

belirleyen yöntemleri kullanmak için Kmeans metodunun "init" parametresi ile hem "k-means++" hem de rasgele (random) merkez noktası seçme yöntemleri kullanılmıştır.

3.5.2. Non-negative Matrix Factorization (NMF) algoritması

NMF, k-means kümeleme algoritması ve temel bileşenler analizi yöntemlerine alternatif olarak önerilmiştir (Lee ve Seung, 1999). Bu yöntemde amaç, verilen negatif olmayan bir matrise iki negatif olmayan matris çarpımı cinsinden yaklaşmaktır. Denklem 3.28'de V matrisi, negatif olmayan W ve H matrislerine ayrıştırılmıştır. V matrisinin boyutu nxm; W matrisinin boyutu nxr ve H matrisinin boyutu ise rxm'dir. r değeri ayrıştırma rankı olarak adlandırılır.

$$V = WH \tag{3.28}$$

Birçok çalışmada boyut azaltma algoritması olarak kullanılmış olsa da doküman kümeleme için kullanıldığı çalışmalarda mevcuttur (Shahnaz vd., 2006; Xu vd., 2003).

Bu çalışmada, Sklearn (Pedregosa vd., 2012) kütüphanesinin NMF metodu kullanılmıştır. Küme sayısını belirten "n_components" parametresi sahte haber ve gerçek haber etiketlerini temsil etmek üzere 2 olarak belirlenmiştir. TF-IDF, Word2vec ve Doc2vec kelime temsili yöntemleri ile algoritma eğitilmiştir.

Ayrıca, Sklearn kütüphanesinin NMF metodu kullanıldığında başlangıç matrisinin seçimi "init" parametresi ile ve çözücü seçimi ise "solver" parametresiyle belirlenebilmektedir. Bu çalışmada, "init" parametresi ile rasgele (random), NNDSVD (Nonnegative Double Singular Value Decomposition), sıfır değerlerini matris ortalamasıyla dolduran NNSVDA ve sıfır değerlerini rasgele küçük değerle dolduran NNSVDAR prosedürleri kullanılmıştır. Çözücü seçimi parametresi olan "solver" ile koordinat inişi (Coordinate Descent-CD) ve çarpmalı güncelleme (Multiplicative Update-MU) çözücüleri kullanılmıştır.

Word2Vec ve doc2Vec yöntemlerinde elde edilen vektörler pozitif ve negatif sayılardan oluşabilmektedir. NMF yöntemi ise sadece negatif olmayan sayıları girdi olarak kabul eder. Bu nedenle Word2Vec ve Doc2Vec tarafından oluşturulmuş vektörlerdeki en küçük negatif sayılar bulunup bu sayıların mutlak değeri, vektördeki tüm sayılara eklenmiştir. Bu sayede kelime vektörü, negatif olmayan sayılardan oluşmuştur.

3.5.3. Linear Discriminant Analysis (LDA) algoritması

LDA, verileri benzerliklerine göre kümelemek için bir hiperdüzlem belirler (Fisher, 1936).

Bu çalışmada, Sklearn (Pedregosa vd., 2012) kütüphanesinin "LinearDiscriminantAnalysis" metodu kullanılmıştır. Küme sayısını belirten n_components parametresi sahte ve sahte olmayan etiketlerini temsil etmek üzere 2 olarak belirlenmiştir. Çözücü algoritmasını belirlemek için kullanılan "solver" parametresi ile de Singular Value Decomposition (SVD) ve Least Squares Solution (Lsqr) algoritmaları kullanılmıştır. TF-IDF, word2vec ve doc2vec kelime temsili yöntemleri ile algoritma eğitilmiştir.

3.6. Makine Öğrenmesi Algoritmalarının Başarısını Değerlendirme Yöntemleri

Makine öğrenmesi algoritmalarının başarısının ölçülmesi için Hata matrisi (confussion matrix) kullanılmaktadır. Çizelge 3.1'de hata matrisinin yapısı gösterilmiştir. Bu matriste kaç adet doğru tahmin yapıldığı, kaç adet yanlış tahmin yapıldığı görülebilmektedir. Çizelgede bu çalışma dikkate alındığında; S sahte haberleri, G gerçek haberleri, S' sahte haber tahminini, G' ise gerçek haber tahmini göstermektedir. P sahte haberlerin toplam sayısını, N gerçek haberlerin toplam sayısını, P' tahmin edilen sahte haberlerin toplam sayısını ve N' ise tahmin edilen gerçek haberlerin toplam sayısını göstermektedir. TP, doğru tahmin edilmiş sahte haber sayısını temsil etmektedir. FP ise, sahte haber olarak tahmin edilmiş fakat gerçek haber olması gereken verilerin sayısını temsil etmektedir. TN, doğru tahmin edilmiş gerçek haber sayısını temsil etmektedir. Son olarak FN ise, gerçek olarak tahmin edilmiş fakat aslında sahte haber olan verilerin sayısını temsil etmektedir.

Çizelge 3.1. Hata Matrisi

	S'	G'	
S	TP	FN	P
G	FP	TN	N
	P'	N'	

Hata matrisinde bulunan doğru ve yanlış tahminler hakkındaki sayısal veriler kullanılarak modelin doğruluğu ölçen ölçütler hesaplanabilmektedir (Şekil 3.19). Bu ölçütlerden bazıları Denklem 3.29, 3.30, 3.31 ve 3.32'de verilmiştir.

Kesinlik değeri, algoritmanın tahmin ettiği verilerden ne kadarının ilk kümede doğru tahmin edildiğini gösterir. Sahte ve gerçek haber tespiti problemi dikkate alındığında, veri setinde bulunan sahte haberlerin kaç tanesinin sahte haber olarak seçildiğinin oranını vermektedir. Kesinlik değeri ne kadar yüksek ise ilk kümede doğru etiketlediği eleman sayısı o kadar fazladır. Denklem 3.29'da Kesinlik metrik değerinin denklemi verilmiştir.

Bir problemin çözümünde ilk kümede bulunan elemanların kesinlikle doğru tahmin edilmesi isteniyorsa, mutlaka Kesinlik değerinin yüksek olması gereklidir. Örneğin; covid-19 gibi salgın bir hastalığın tespitinde, bir kişinin hasta olduğunun yüksek oranda tahmin edilmesi istendiği durumda Kesinlik değerinin yüksek olması önemlidir.

$$Kesinlik(Precision) = \frac{TP}{TP+FP} = \frac{TP}{P'} \tag{3.29}$$

Duyarlılık değeri ise algoritmanın 1. kümede etiketlediği verilerin ne kadarının doğru etiketlendiği oranını verir. Yine sahte haber tespiti problemi düşünüldüğünde; algoritmanın sahte haber olarak işaretlediği verilerin tüm sahte haberlere oranını vermektedir. Denklem 3.30'da Duyarlılık metrik değerinin denklemi verilmiştir.

Bir problemin çözümünde, tahmin edilen verilerin iki kümede de doğruluğu önemli ise Kesinlik ve Duyarlılık değerinin birlikte yüksek tutulması önemlidir. Örneğin; sahte haber tespiti probleminde Kesinlik ve Duyarlılık metrik değerlerinin ikisinin de yüksek olması gerekmektedir.

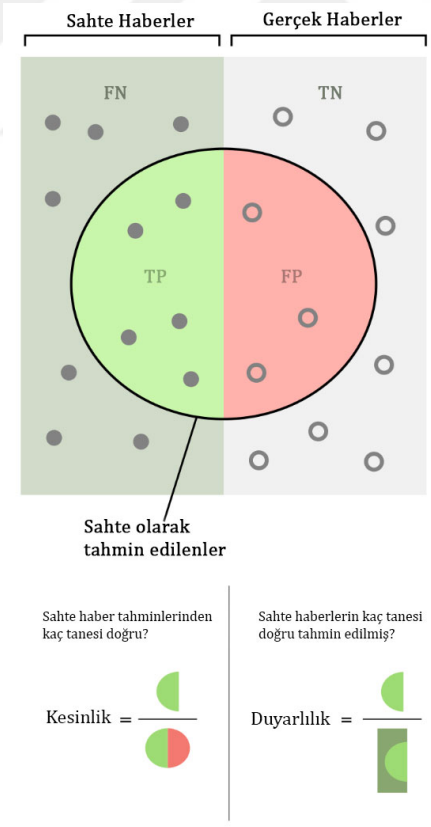
$$Duyarlılık(Recall) = \frac{TP}{TP+FN} = \frac{TP}{P} \tag{3.30}$$

Kesinlik ve Duyarlılık değerlerinin harmonik ortalaması F1-metrik değeri vermektedir.

$$F1\text{-}metrik\ (F1\text{-}score) = 2x\frac{Kesinlik\ x\ Duyarlılık}{Kesinlik\ +\ Duyarlılık} \tag{3.31}$$

Doğruluk değeri, algoritma sonucunda, doğru olarak yapılan tahminlerin tüm tahminlere oranıdır.

$$Doğruluk(Accuracy) = \frac{TP+TN}{TP+TN+FP+FN} = \frac{TP+TN}{P+N} \tag{3.32}$$



Şekil 3.19. Hata metrikleri (Wikipedia, 2006)

Makine öğrenmesi algoritmalarının başarısının ölçülmesinde bir diğer kullanılan metrik ise saflıktır (purity). Saflık, her küme için içerisindeki elemanlardan sayıca en fazla

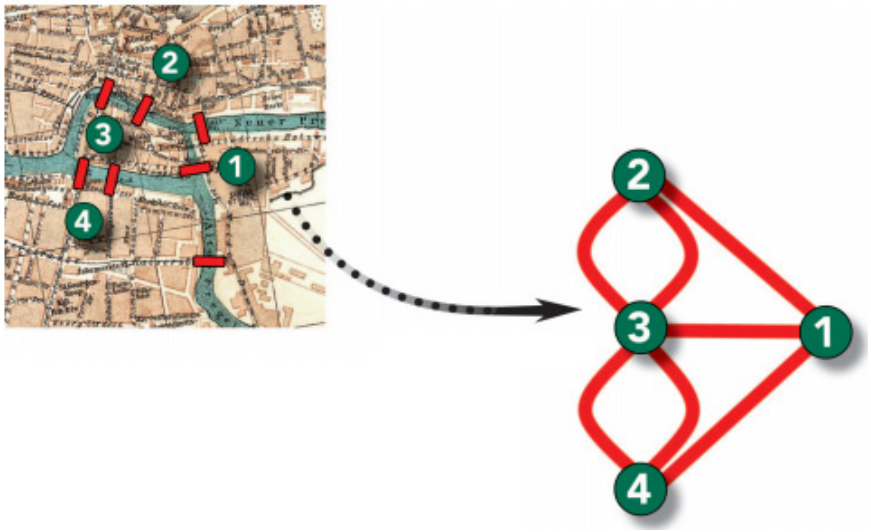
olanın tüm veri sayısına oranıdır (Senol ve Karacan, 2018). Saflık değeri ile denetimsiz öğrenme algoritmalarının kümelerinde hangi sınıftan verinin daha çok bulunduğu öğrenilir. Denklem 3.33 ile saflık değeri hesaplanır.

$$Saflık = \frac{\sum_{i=1}^{k} \frac{|c_i^d|}{|c_i|}}{K} \tag{3.33}$$

K küme sayısını, c_i^d, c_i kümesindeki baskın sınıfa ait veri sayısını ifade eder.

3.7. Graf

Leonhard Euler 1736 yılında, Konigsberg kentinde bulunan 7 köprüden yalnız ve yalnız bir kere geçerek dolaşmanın mümkün olmadığını graf teoremi ile kanıtlamıştır. Bu durum Şekil 3.20'de gösterilmiştir. Bu çalışma, graf teoreminin ilk çalışması olarak kabul edilir (Kadry ve Al-Taie, 2014).



Şekil 3.20. Konigsberg'in 7 köprüsü problemi (Shields, 2012)

Graf, en basit tanımıyla, nokta çiftlerini birbirine bağlayan bir dizi nokta ve çizgilerin oluşturduğu kümedir. Noktalar düğümler (vertex/node) olarak adlandırılırken, çizgiler ise kenarlar (edge) olarak adlandırılır. Şekil 3.20, Konigsberg'in 7 köprüsü probleminin graf ile gösterimidir.

3.8. Sosyal Ağ Analizi

Sosyal ağlar, en sık kullanılan internet servislerinden biridir. Sosyal ağlar, bağlantılar aracılığıyla diğer kullanıcılarla etkileşime girmek için kullanışlı araçlar sunan platformlardır (Galán-GarcÍa vd., 2015). Türkiye'de çok sık kullanılan sosyal ağlara; Facebook, Twitter, Instagram, LinkedIn, Pinterest ve Telegram örnek olarak gösterilebilir.

İnsanların sosyal ağ aracıyla birbiriyle etkileşiminde farklı araçlar kullanılmaktadır. Bu tez çalışması kapsamında en çok kullanılan sosyal ağlardan biri olan Twitter patformu incelenmiştir. Twitter, kullanıcıların kısa mesaj göndererek etkileşimde bulunduğu bir sosyal platformdur.

Twitter platformunun etkileşim araçlarından bazıları aşağıda açıklanmıştır.

- Tweet: Kullanıcıların paylaştıkları kısa mesajlardır. Uzun süre 140 karakter sınırı bulunan Twitter'da, şu an 280 karakter sınırında kısa mesajlar paylaşılabilmektedir. Bir tweet mesajında kullanıcıyı etiketlemek için kullanıcı adının başına @ işareti koyulur.
- Retweet: Farklı kullanıcıların tweet mesajlarını, bir kullanıcının ileterek tekrar paylaştığı kısa mesajlara retweet adı verilir.
- Yanıtla: Bir tweet mesajına insanların yorum yapması durumudur. Tweet mesajı sayfasında tweet mesajının altında yapılan yorumlar görülebilir.
- Beğeni: Bir tweet mesajının kullanıcılar tarafından kalp işaretli buton ile beğenilmesidir.
- Onaylanmış hesap: Kullanıcıların ilgisini çekebilecek bir hesabın orijinal olduğunun bilinmesini sağlar. Bir hesabın onaylanması için Twitter platformu tarafından istenen birtakım bilgilerin platforma girilmiş olması gerekmektedir. Onaylanmış hesaplar, mavi bir onay simgesi ile gösterilir.

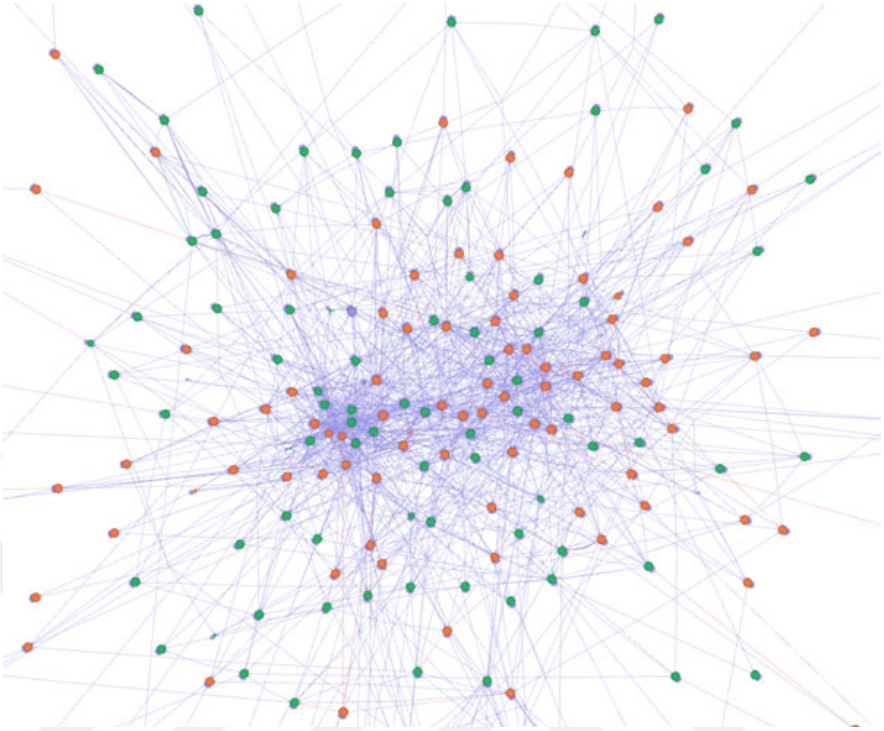
- Takipçi (Follower): Bir Twitter kullanıcısını takip eden diğer kullanıcıları temsil eder. Takipçiler takip ettikleri kullanıcının tweet mesajlarını ve güncellemelerini görebilirler.
- Takip: Bir kullanıcının başka bir kullanıcıyı takip etmesidir.
- Başlık Etiketi (Hashtag): Bir konu hakkında diyez işareti kullanılarak (#) bir başlık açılabilir. Diğer tüm kullanıcılar bu konu hakkındaki görüşlerini bu konu etiketini kullanarak paylaşabilirler.

Bu çalışmada, Twitter sosyal ağındaki kullanıcıların takipçi etkileşimi ile insanların sahte haber paylaşıp paylaşmadığı incelenmiştir. Problemimizde, "sahte haber yayan kullanıcılar, sahte haber yayan diğer kullanıcıları ne oranda takip ediyor?", "sahte haber yayan kullanıcılar, sahte haber yaymayan kullanıcıları ne oranda takip ediyor?", sorularına cevap aranmaktadır. Elde edilen veri setinden sahte haber yayan kullanıcılarla, sahte haber yaymayan kullanıcıların, twitter platformundan takipçileri toplanmıştır. Bu sayede elimizde takip takipçi ilişkisinden bir graf oluşturulmuştur.

Sosyal ağ analizi, Facebook, Twitter gibi sosyal ağların kullanıcıları ve bu kullanıcılarının birbiri ile olan etkileşimini graf teoremi ile inceleyen yöntemlerdir. Sosyal ağlarda kullanıcılar düğümler (vertex/node), birbirleri ile olan ilişkileri ise kenarlar (edge) ile temsil edilir. Facebook platformunda bir kişiyle arkadaşlık durumu çift yönlüdür. Yani bir kişi arkadaş olarak eklendiğinde ikisi de birbirinin arkadaşı olarak kabul edilmektedir. Dolayısıyla Facebook platformunda sosyal ağ grafı arkadaşlık durumu için yönsüz graf olarak kullanılabilir. Twitter platformundaki takip etme sistemi ise tek yönlü olabilmektedir. Yani bir kişi birini takip ederken takip ettiği kişi kendisini takip etmeyebilir. Bu nedenle Twitter platformunda sosyal ağ grafı, yönlü bir graf olmak zorundadır (Galán-GarcÍa vd., 2015).

Bu tez çalışmasında, sosyal ağ analizi için graf görselleştirme ve analiz yazılımı olan, Gephi uygulaması kullanılmıştır (Gephi, 2020). Gephi uygulamasından Pagerank ve HITS algoritmaları kullanılarak sosyal ağ analizi yapılmıştır. Şekil 3.21'de Gephi uygulamasından alınmış bir graf görülmektedir. Şekildeki kırmızı düğümler sahte haber

paylaşmış olan kullanıcıları, yeşil düğümler doğru haber paylaşmış olan kullanıcıları temsil etmektedir.



Şekil 3.21. Gephi uygulamasından alınan graf örneği.

Link sıralama algoritmaları arama motorları tarafından aranan kelimeye en uygun web sitesini bulmak amacıyla oluşturulmuştur. Sosyal ağ analizinde ise bu algoritmalar etkili kişileri bulmak için kullanılabilmektedirler.

3.5.2.1. Pagerank algoritması:

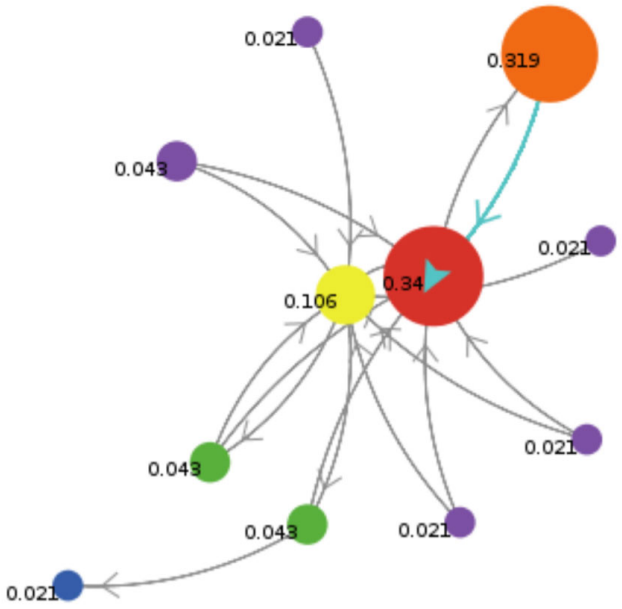
Google'ın kurucularından olan, Brin ve Page (1998), Pagerank adını verdikleri bir link sıralama algoritması önermişlerdir. Bu algoritma uzun süre Google arama motoru tarafından da kullanılmıştır. Bu algoritmada, internet üzerindeki tüm web sayfaları gezilerek, bir graf yapısı olarak saklanmaktadır. Arama motoru üzerinden bir arama yapıldığında, pagerank algoritması ile, yapılan arama için en etkili olan web sayfasından başlanarak bir liste oluşturulur. Kullanıcıya arama sonucu olarak bu web sayfalarının listesi gösterilir.

Her bir düğümün pagerank değerinin hesaplanması Denklem 3.34'de gösterilmiştir.

$$PR(u) = c \sum_{v \in B_u} \frac{R'(v)}{N_v} \tag{3.34}$$

Denklem 3.34'de, u, bir web sitesini, $B(u)$, u web sitesinin linkini barındıran sitelerin kümesini, $PR(u)$ ve $PR(v)$, sırasıyla u ve v web sitelerinin rank değerini, N_v, v sayfasındaki linklerin sayısını ve c normalizasyon faktörünü temsil eder.

Bir düğümü çok fazla düğüm işaret ediyorsa o düğümün pagerank puanı yüksek olacaktır. Aynı zamanda bir düğümü az kişi takip ediyor fakat pagerank değeri yüksek olan bir düğüm işaret ediyorsa yine o düğümün pagerank değeri yüksek olacaktır. Şekil 3.22'de, kırmızı düğüme çok fazla düğüm işaret ettiği için daha büyük boyuttadır. Turuncu renk düğüme bakıldığında ise sadece kırmızı düğüm tarafından takip edildiği halde diğer düğümlerden boyut olarak daha büyük olduğu görülür.



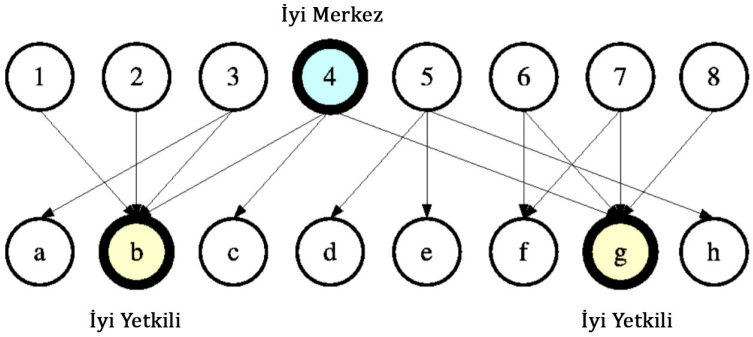
Şekil 3.22. Pagerank değerinin alt linklere dağılımı (Stonedahl vd., 2011).

3.5.2.2. Hipermetin kaynaklı konu araması-HKKA (hypertext-induced topic search-HITS) algoritması:

Kleinberg (1999) tarafından, arama motoru aramalarında, hem bağlantıları hem de içerikleri dikkate alarak puanlama yapan bir model önerilmiştir.

HKKA algoritmasında, arama sonucu ile doğrudan ilişkili olan web siteleri dışında, önerilen siteleri de getirilir. Arama sonucu ile doğrudan ilişkili olan web sitelerine merkezler (hubs), önerilen web sitelerine ise yetkililer (authorities) adı verilir (Kadry ve Al-Taie, 2014). Otomobil ile ilgili bir aramada otomobil siteleri merkezleri temsil ederken, otomobil ile ilgili forum siteleri de yetkilileri temsil edebilir.

Diğer merkez düğümlerden daha çok sayıda iyi yetkiliyi gösteren merkezlere "iyi merkez" denir. Diğer yetkili düğümlerden daha fazla sayıda iyi merkez tarafından gösterilen yetkililere "iyi yetkili" denmektedir. Şekil 3.23'de bu durum gösterilmiştir. Bu şekilde etkili merkez ve yetkililer bulunmaktadır.



Şekil 3.23. HKKA: iyi merkez ve iyi yetkililer örneği (Castillo, 2005).

Her düğümün merkez ve yetki puanları hesaplanmadan önce iterasyon sayısını belirten k değerini belirlemek gerekmektedir. Daha sonra her düğümün merkez ve yetki puanını hesaplamak için, başlangıçta, her düğümün merkez ve yetki puanı 1 olarak alınır. Her düğümün yetki puanı, graftaki o düğümü gösteren tüm düğümlerin merkez puanlarının toplamıdır. Her düğümün merkez puanı ise, graftaki o düğümün gösterdiği tüm düğümlerin yetki puanlarının toplamıdır. Hesaplanan merkez ve yetki puanları 0 ile 1 arasında olacak şekilde normalleştirilir.

4. ARAŞTIRMA BULGULARI

Çalışmada Twitter üzerinden toplanan ve etiketlenmiş üç konu ile ilgili tweet mesajları, TF-IDF özellik çıkarımı kullanılarak YSA temelli olmayan denetimli öğrenme ve denetimsiz öğrenme algoritmaları ile; word2vec kelime vektörünü kullanarak da derin öğrenme algoritmaları ile sahte ve gerçek olarak tahmin edilmeye çalışılmıştır. Algoritmaların ön yargısını kaldırmak için her bir algoritma 100 defa çalıştırılmış, her adımda konulardaki tweet mesajları rastgele karıştırılmış ve tüm mesajların hem eğitim setinde hem de test setinde bulunabilmesi sağlanmış ve F1-metrik değerlerinin ortalamaları ile F1-metrik değerlerinin standart sapmaları incelenmiştir. Ayrıca sahte ve gerçek haber yayan kullanıcılar ve bu kullanıcıları takip eden kullanıcılar toplanarak sosyal ağ analizi yapılmıştır.

Bu çalışmada kullanılan bilgisayar, Intel i7-6700 işlemci, 24 gb DDR4 RAM bellek, 512 GB SSD disk, 4 TB HDD disk ve 768 adet cuda çekirdeğine sahip Nvidia GeForce 1050Ti 128 bit 4 GB bellekli ekran kartına sahiptir.

4.1. Veri Seti

Belirlenen üç konuda toplanan tweet mesajları metin halinde veritabanına kaydedilmiştir. Daha sonra oluşturulan web uygulama arayüzü ile mesajlar etiketlenerek sahte ve gerçek haberler etiketlenmiştir. Denetimli öğrenme ve denetimsiz öğrenme kullanılması için ham metin halindeki tweet mesajlarının vektörler ile gösterilmesi gerekmektedir. Metin halindeki tweet mesajlarının vektörlere dönüştürmek için TF-IDF, word2vec ve doc2vec yöntemleri kullanılmıştır.

Metin tweet mesajlarını, TF-IDF ile vektörlere dönüştürmeden önce, metin üzerinde ön işlem çalışmaları yapılmıştır. Tüm harfler küçük harflere dönüştürülmüştür. Sayısal ifadeler, noktalama işaretleri ve metinsel olmayan ifadeler tweet mesajlarından temizlenmiştir. Daha sonra NLTK (Natural Language Toolkit) kütüphanesinde (Bird vd., 2009) bulunan Türkçe durak kelimeler (stopwords) kütüphanesi kullanılarak, tweet mesajlarından bu durak kelimeler atılmıştır. Daha sonra Sklearn (Scikit-Learn)

kütüphanesinin (Pedregosa vd., 2012) TfidfVectorizer yöntemi kullanılarak metinler TF-IDF vektörlere çevrilmişlerdir.

Metin tweet mesajlarını, word2vec ve doc2vec ile vektörlere dönüştürmeden önce, aynı TF-IDF vektörlerine dönüştürmeden önce yapıldığı gibi, metin üzerinde ön işlem çalışmaları yapılmıştır. Tüm harfler küçük harflere dönüştürülmüştür. Sayısal ifadeler, noktalama işaretleri ve metinsel olmayan ifadeler tweet mesajlarından temizlenmiştir. Daha sonra NLTK kütüphanesinde bulunan Türkçe durak kelimeler (stopwords) kütüphanesi kullanılarak, tweet mesajlarında bu durak kelimeler atılmıştır. NLTK (Bird vd., 2009) kütüphanesinin "word_tokenize" yöntemi ile tweet mesajları kelimelerine ayrılarak kelime vektörleri oluşturulmuştur. Algoritmaların başarısını arttırmak için önceden eğitilmiş word2vec modeli kullanılmıştır. Oslo Üniversitesi Dil Teknoloji Grubu tarafından oluşturulan platformda, birçok araştırmacı tarafından yüklenmiş, önceden eğitilmiş kelime temsili modelleri bulunmaktadır (Language Technology Group at the University of Oslo, 2018).

Bu çalışmada, Türkçe dilinde 3633786 kelimeden oluşan, wikipedia ve genel bir tarayıcı tarafından toplanan metinlerden oluşan CoNLL (Computational Natural Language Learning) veri seti kullanılarak ve skip-gram yöntemi ile eğitilmiş word2vec modeli kullanılmıştır. Daha sonra kelime vektörü haline dönüştürülen tweet mesajlarının her bir kelimesine karşılık gelen word2vec vektörleri ile her bir tweet mesajı için 50x100 boyutunda matris oluşturulmuştur. 50 kelimeden kısa olan tweet mesajları, 50x100 boyutuna tamamlamak için 1x100 boyutundaki sıfır vektörleri ile doldurulmuştur. Derin öğrenme algoritmalarına bu 50x100 boyutundaki matris girdi olarak verilmektedir. YSA temeli olmayan denetimli öğrenme algoritmaları ve denetimsiz öğrenme algoritmaları girdi olarak vektörleri kabul etmektedir. Bu nedenle 50x100 word2vec vektörlerinden oluşan matrisi, vektörle temsil etmek gerekmektedir. Birçok makalede bu işlem için word2vec matrisinin ortalama vektörü alınmaktadır (Yang vd., 2016; Sahin, 2017; Elsaadawy vd., 2018; Bilgin, 2019; Karcioglu ve Aydin, 2019). Vektörleri alt alta ekleyerek tek bir vektör ile temsil eden çalışmalar da bulunmaktadır. Bir çalışmada da TF-IDF ve Word2vec vektörlerinin birlikte kullanımı algoritmalara girdi olarak verilmiştir (Lilleberg vd., 2015). Bu çalışmada vektörlerin alt alta eklenmesi ve ortalama vektör yöntemleri denenmiştir. Ortalama vektör yönteminin, alt alta ekleme

yöntemine göre oldukça başarılı olduğu görülmüştür. Bu nedenle, bu çalışmada, YSA temelli olmayan denetimli öğrenme ve denetimsiz öğrenme algoritmalarına girdi olarak ortalama vektör verilmiştir.

Doc2vec kelime temsili yöntemi için, gensim kütüphanesinin doc2vec kütüphanesi kullanılmıştır. Türkçe dilinde, önceden eğitilmiş Doc2vec modeli bulunmadığı için veri setinde bulunan tweet mesajları ile model eğitilmiştir. Sonuç olarak, her bir tweet mesajı için 1x20 boyutunda vektör elde edilmiştir. Önceden eğitilmiş bir Doc2vec modeli bulunmadığından ve veri setimizdeki tweet mesajları yetersiz kaldığından dolayı algoritmalara verilen doc2vec vektörleri yeterli başarıyı gösterememiştir. Tahmin edilen verilerin hepsini ya sahte haber veya hepsini doğru haber olarak sınıflandırmış yada kümelendirmiştir. Bu nedenle, çalışmada sonuçlarına yer verilmemiştir.

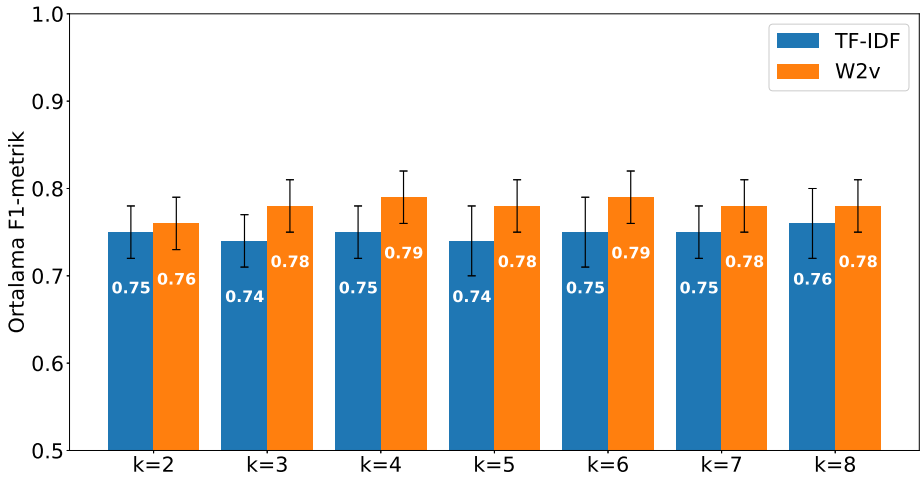
4.2. YSA Temelli Olmayan Denetimli Öğrenme Algoritmaları

Bu çalışmada K-Nearest Neighbor (KNN), Support Vector Machine (SVM) ve Random Forest (RF) YSA temelli olmayan denetimli öğrenme algoritmaları kullanılmıştır. Denetimli öğrenme algoritmalarında, tüm tweet mesajlarının bulunduğu, Konu-1 tweet mesajlarının bulunduğu, Konu-2 tweet mesajlarının bulunduğu ve Konu-3 tweet mesajlarının bulunduğu veri seti olmak üzere 4 farklı veri seti ile çalışılmış ve tweet mesajlarının vektörler ile temsil edilmesi için TF-IDF ve word2vec yöntemleri kullanılmıştır. Algoritmaların ön yargısını kaldırmak için her algoritma 100 defa çalıştırılmıştır. Her adımda, konulardaki tweet mesajları rastgele karıştırılmış ve tüm mesajların hem eğitim setinde hem de test setinde bulunabilmesi sağlanmıştır. Bu, her adımda karıştırılan veri setinin, %70'i modeli eğitmek için eğitim seti, %30'u ise bu algoritmaların başarısını ölçmek için test seti olarak kullanılmıştır.

KNN algoritması, farklı uzaklık fonksiyonları ve farklı komşu sayıları ile test edilmiştir. Algoritmada kullanılan k ölçütü bir verinin sınıfını belirlerken bakılacak olan en yakın komşu sayısını belirtmektedir. Tüm konuların tweet mesajlarının bulunduğu, Konu-1 tweet mesajlarının bulunduğu, Konu-2 tweet mesajlarının bulunduğu ve Konu-3 tweet mesajlarının bulunduğu veri seti olmak üzere 4 farklı veri seti ile

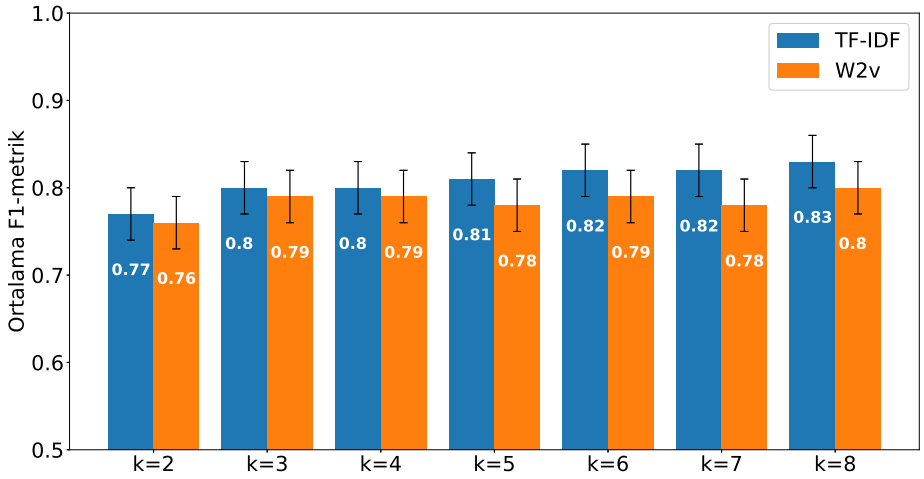
algoritma çalıştırılmıştır. TF-IDF ve Word2vec kelime temsili yöntemlerinin sonuçları karşılaştırılmıştır.

Şekil 4.1'de, KNN algoritmasının manhattan uzaklık algoritması ve farklı komşu sayıları ile tüm konuların bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. KNN algoritması, manhattan uzaklık fonksiyonu kullanıldığında ve uzaklık belirlenirken 4 komşuluğa ve 6 komşuluğa bakıldığında, 0,79 F1-metrik değeri ile en başarılı sonucu, word2vec kelime temsili yöntemi ile vermiştir. TF-IDF kelime temsili yöntemiyle ise, 8 komşuluğa bakıldığında 0,76 F1-metrik değeri gösterebilmiştir.



Şekil 4.1. Tüm konuların bulunduğu veri seti kullanılarak KNN algoritmasının manhattan uzaklık algoritması ve farklı komşu sayıları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

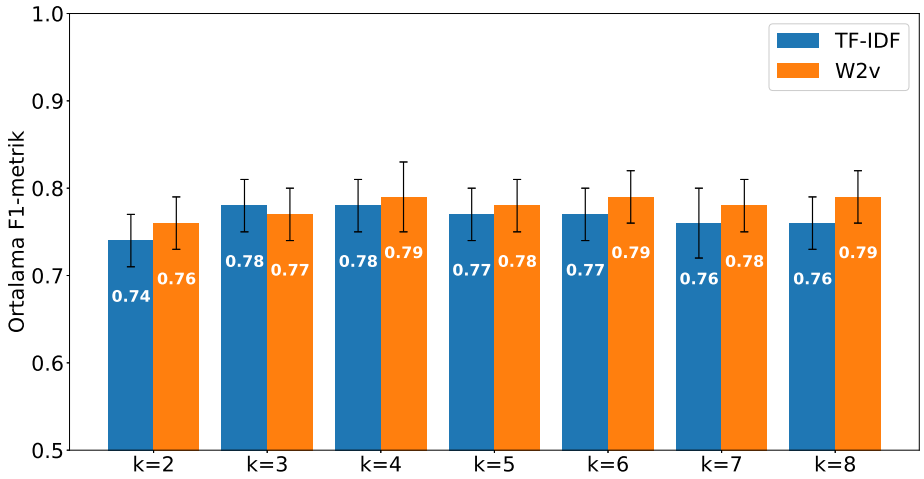
Şekil 4.2'de KNN algoritmasının öklid uzaklık algoritması ve farklı komşu sayıları ile tüm konuların bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. KNN algoritması, öklid uzaklık fonksiyonu kullanıldığında ve uzaklık belirlenirken 8 komşuluğa bakıldığında, 0,83 F1-metrik değeri ile en başarılı sonucu, TF-IDF kelime temsili yöntemi ile vermiştir. Word2vec kelime temsili yöntemiyle ise, 8 komşuluğa bakıldığında 0,80 F1-metrik değeri gösterebilmiştir.



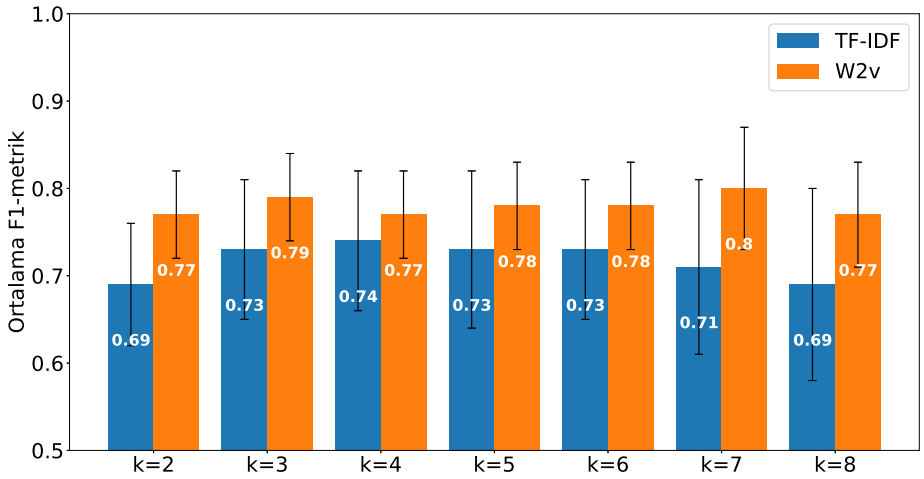
Şekil 4.2. Tüm konuların bulunduğu veri seti kullanılarak KNN algoritmasının öklid uzaklık algoritması ve farklı komşu sayıları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

Şekil 4.3'de KNN algoritmasının minkowski uzaklık algoritması ve farklı komşu sayıları ile tüm konuların bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. KNN algoritması, minkowski uzaklık fonksiyonu kullanıldığında ve uzaklık belirlenirken 4 komşuluğa, 6 komşuluğa ve 8 komşuluğa bakıldığında, 0,79 F1-metrik değeri ile en başarılı sonucu, word2vec kelime temsili yöntemi ile vermiştir. TF-IDF kelime temsili yöntemiyle ise, 3 ve 4 komşuluğa bakıldığında 0,79 F1-metrik değeri gösterebilmiştir.

Şekil 4.4'de, KNN algoritmasının manhattan uzaklık algoritması ve farklı komşu sayıları ile Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. KNN algoritması, manhattan uzaklık fonksiyonu kullanıldığında ve uzaklık belirlenirken 7 komşuluğa bakıldığında, 0,80 F1-metrik değeri ile en başarılı sonucu, word2vec kelime temsili yöntemi ile vermiştir. TF-IDF kelime temsili yöntemiyle ise, 4 komşuluğa bakıldığında 0,74 F1-metrik değeri gösterebilmiştir. Her modelde, word2vec kelime temsili yöntemi, TF-IDF yöntemine göre açık bir şekilde başarılı olmuştur.

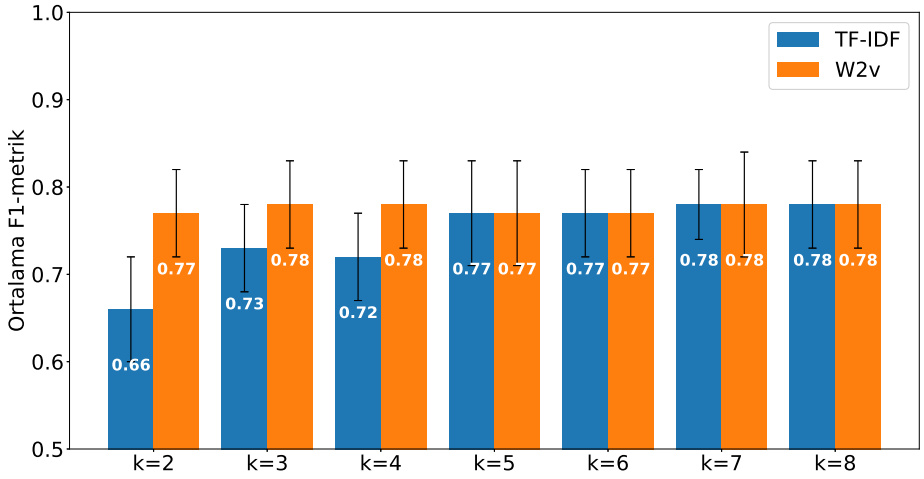


Şekil 4.3. Tüm konuların bulunduğu veri seti kullanılarak KNN algoritmasının minkowski uzaklık algoritması ve farklı komşu sayıları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.



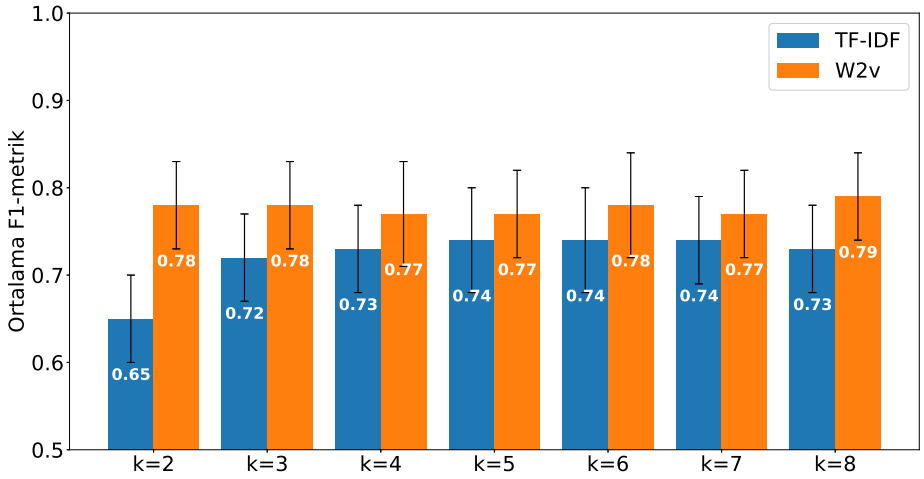
Şekil 4.4. Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak KNN algoritmasının manhattan uzaklık algoritması ve farklı komşu sayıları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

Şekil 4.5'de, KNN algoritmasının öklid uzaklık algoritması ve farklı komşu sayıları ile Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. KNN algoritması, öklid uzaklık fonksiyonu kullanıldığında ve uzaklık belirlenirken 3, 4, 7 ve 8 komşuluğa bakıldığında, 0,78 F1-metrik değeri ile en başarılı sonucu, word2vec kelime temsili yöntemi ile vermiştir. TF-IDF kelime temsili yöntemiyle ise, 7 ve 8 komşuluğa bakıldığında word2vec yöntemiyle aynı sonucu vermiştir.



Şekil 4.5. Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak KNN algoritmasının öklid uzaklık algoritması ve farklı komşu sayıları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

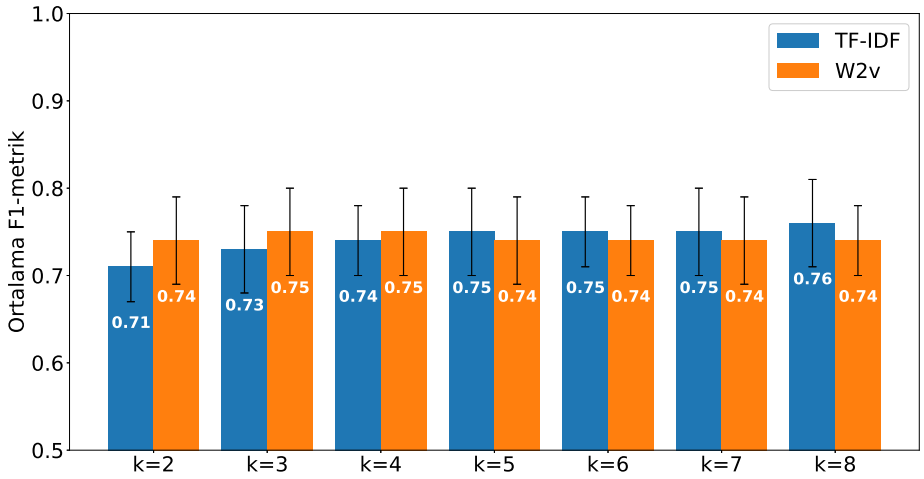
Şekil 4.6'da, KNN algoritmasının minkowski uzaklık algoritması ve farklı komşu sayıları ile Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. KNN algoritması, minkowski uzaklık fonksiyonu kullanıldığında ve uzaklık belirlenirken 8 komşuluğa bakıldığında, 0,79 F1-metrik değeri ile en başarılı sonucu, word2vec kelime temsili yöntemi ile vermiştir. TF-IDF kelime temsili yöntemiyle ise, 5, 6 ve 7 komşuluğa bakıldığında 0,74 F1-metrik değeri gösterebilmiştir. Her modelde, word2vec kelime temsili yöntemi, TF-IDF yöntemine göre açık bir şekilde başarılı olmuştur.



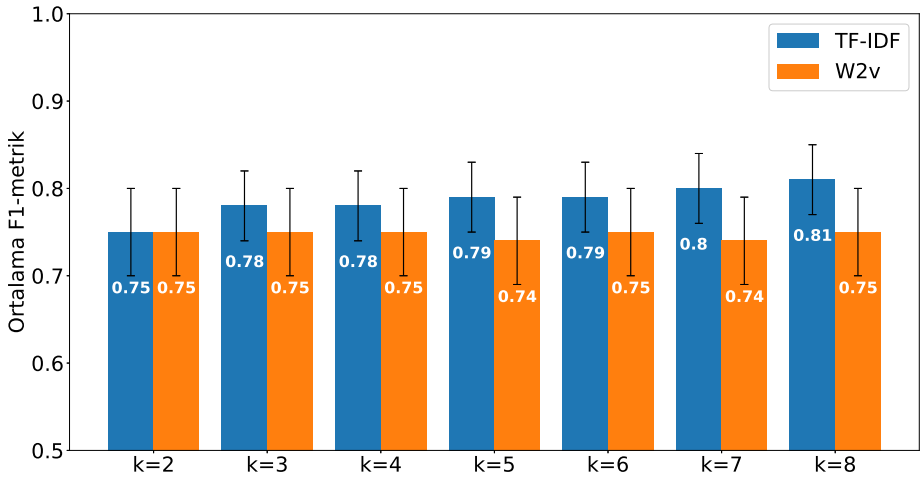
Şekil 4.6. Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak KNN algoritmasının minkowski uzaklık algoritması ve farklı komşu sayıları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

Şekil 4.7'de, KNN algoritmasının manhattan uzaklık algoritması ve farklı komşu sayıları ile Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. KNN algoritması, manhattan uzaklık fonksiyonu kullanıldığında ve uzaklık belirlenirken 8 komşuluğa bakıldığında, 0,76 F1-metrik değeri ile en başarılı sonucu, TF-IDF kelime temsili yöntemi ile vermiştir. Word2vec kelime temsili yöntemiyle ise, 3 ve 4 komşuluğa bakıldığında 0,75 F1-metrik değeri gösterebilmiştir. 2, 3 ve 4 komşuluk dikkate alındığında word2vec, 5 ve üzeri komşuluk dikkate alındığında ise TF-IDF kelime temsili yöntemi, word2vec yönteminden sadece %1 daha başarılı olmuştur.

Şekil 4.8'de, KNN algoritmasının öklid uzaklık algoritması ve farklı komşu sayıları ile Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. KNN algoritması, öklid uzaklık fonksiyonu kullanıldığında ve uzaklık belirlenirken 8 komşuluğa bakıldığında, 0,81 F1-metrik değeri ile en başarılı sonucu, TF-IDF kelime temsili yöntemi ile vermiştir. Word2vec kelime temsili yöntemiyle ise, 0,75 F1-metrik değerine ulaşabilmiştir.

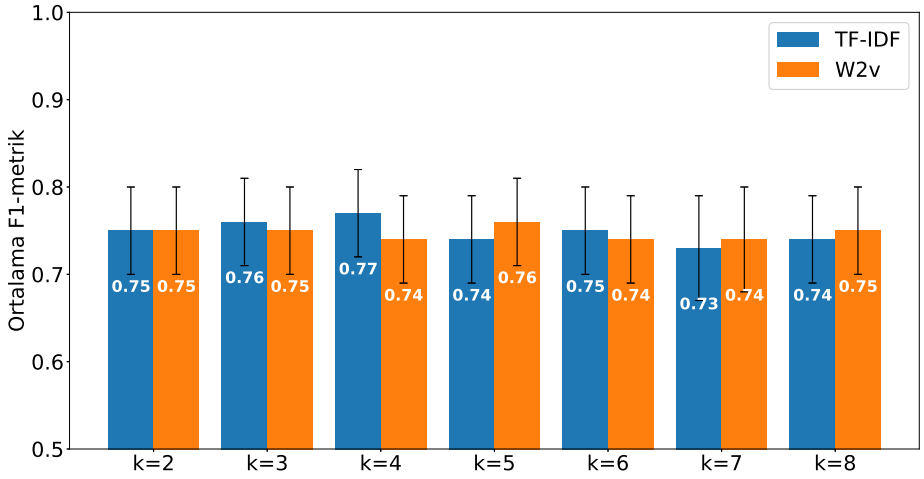


Şekil 4.7. Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak KNN algoritmasının manhattan uzaklık algoritması ve farklı komşu sayıları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.



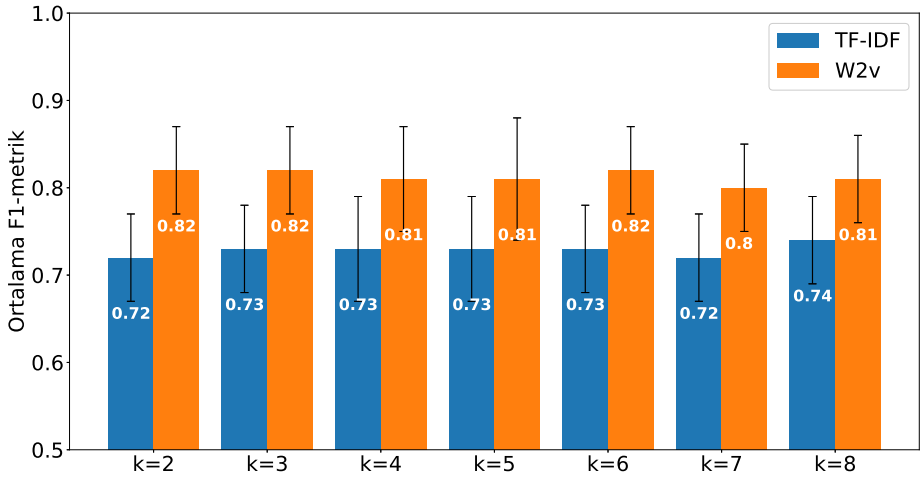
Şekil 4.8. Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak KNN algoritmasının öklid uzaklık algoritması ve farklı komşu sayıları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

Şekil 4.9'da, KNN algoritmasının minkowski uzaklık algoritması ve farklı komşu sayıları ile Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. KNN algoritması, minkowski uzaklık fonksiyonu kullanıldığında ve uzaklık belirlenirken 8 komşuluğa bakıldığında, 0,81 F1-metrik değeri ile en başarılı sonucu, TF-IDF kelime temsili yöntemi ile vermiştir. Word2vec kelime temsili yöntemiyle ise, 0,75 F1-metrik değerine ulaşabilmiştir.



Şekil 4.9. Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak KNN algoritmasının minkowski uzaklık algoritması ve farklı komşu sayıları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

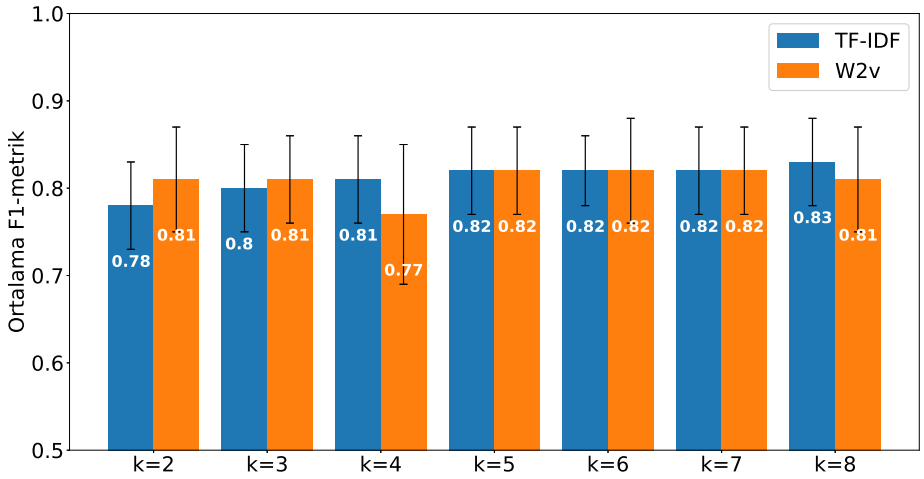
Şekil 4.10'da, KNN algoritmasının manhattan uzaklık algoritması ve farklı komşu sayıları ile Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. KNN algoritması, manhattan uzaklık fonksiyonu kullanıldığında ve uzaklık belirlenirken 2, 3 ve 6 komşuluğa bakıldığında, 0,82 F1-metrik değeri ile en başarılı sonucu, word2vec kelime temsili yöntemi ile vermiştir. TF-IDF kelime temsili yöntemiyle ise, 8 komşuluğa bakıldığında 0,74 F1-metrik değeri gösterebilmiştir. Her modelde, word2vec kelime temsili yöntemi, TF-IDF yöntemine göre açık bir şekilde başarılı olmuştur.



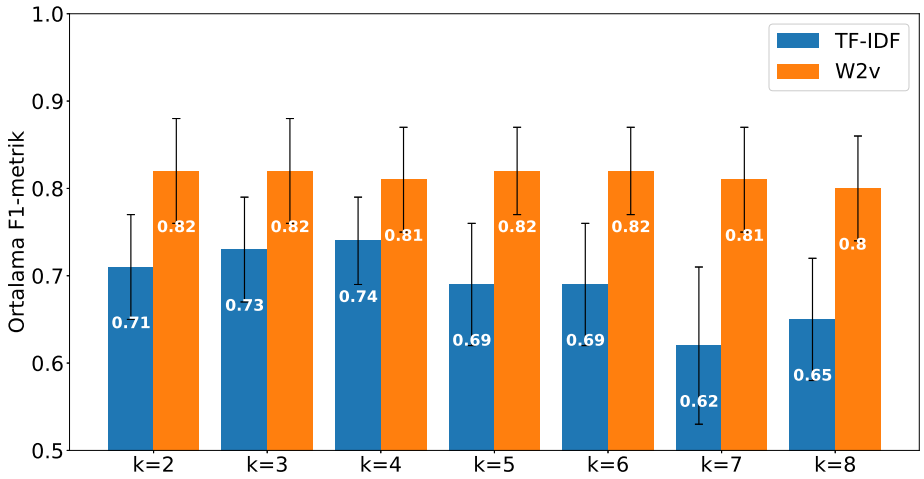
Şekil 4.10. Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak KNN algoritmasının manhattan uzaklık algoritması ve farklı komşu sayıları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

Şekil 4.11'de, KNN algoritmasının öklid uzaklık algoritması ve farklı komşu sayıları ile Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. KNN algoritması, öklid uzaklık fonksiyonu kullanıldığında ve uzaklık belirlenirken 8 komşuluğa bakıldığında, 0,83 F1-metrik değeri ile en başarılı sonucu, TF-IDF kelime temsili yöntemi ile vermiştir. Word2vec kelime temsili yöntemiyle ise, 5, 6 ve 7 komşuluğa bakıldığında 0,82 F1-metrik değeri göstermiştir. TF-IDF ve word2vec kelime temsili yöntemleri birbirine çok yakın sonuç vermiştir.

Şekil 4.12'de, KNN algoritmasının minkowski uzaklık algoritması ve farklı komşu sayıları ile Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. KNN algoritması, minkowski uzaklık fonksiyonu kullanıldığında ve tüm komşuluk değerlerinde en başarılı sonucu, word2vec kelime temsili yöntemi ile vermiştir. En yüksek F1-metrik değeri 0,82 olmuştur. Tf-IDF kelime temsili yöntemiyle ise, 4 komşuluğa bakıldığında 0,74 F1-metrik değeri göstermiştir. Her modelde, word2vec kelime temsili yöntemi, TF-IDF yöntemine göre açık bir şekilde başarılı olmuştur.



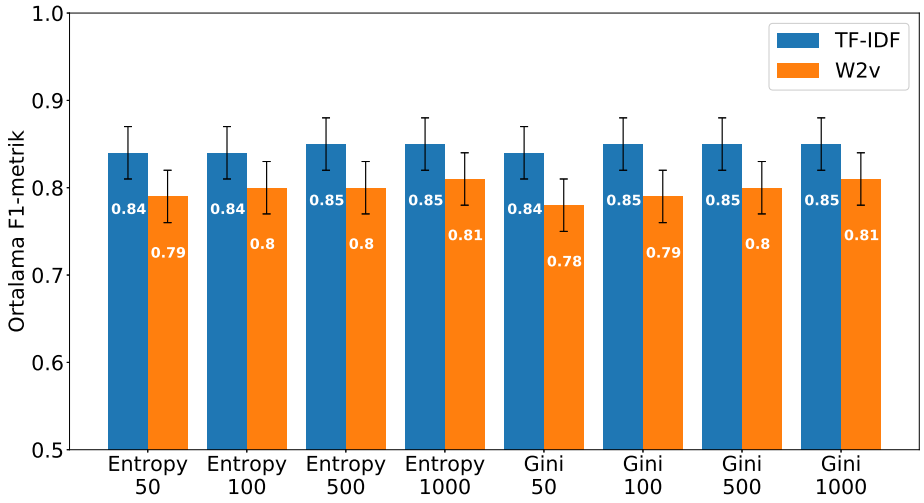
Şekil 4.11. Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak KNN algoritmasının öklid uzaklık algoritması ve farklı komşu sayıları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.



Şekil 4.12. Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak KNN algoritmasının minkowski uzaklık algoritması ve farklı komşu sayıları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

RF algoritması, farklı bölme kalitesini ölçme algoritmaları ve ağaç sayıları ile test edilmiştir. Etropy ve gini algoritmaları kullanılmıştır. Ayrıca, 50, 100, 500 ve 1000 ağaç sayısı ile algoritma test edilmiştir. Tüm konuların tweet mesajlarının bulunduğu, Konu-1 tweet mesajlarının bulunduğu, Konu-2 tweet mesajlarının bulunduğu ve Konu-3 tweet mesajlarının bulunduğu veri seti olmak üzere 4 farklı veri seti ile algoritma çalıştırılmıştır. TF-IDF ve Word2vec kelime temsili yöntemlerinin sonuçları karşılaştırılmıştır.

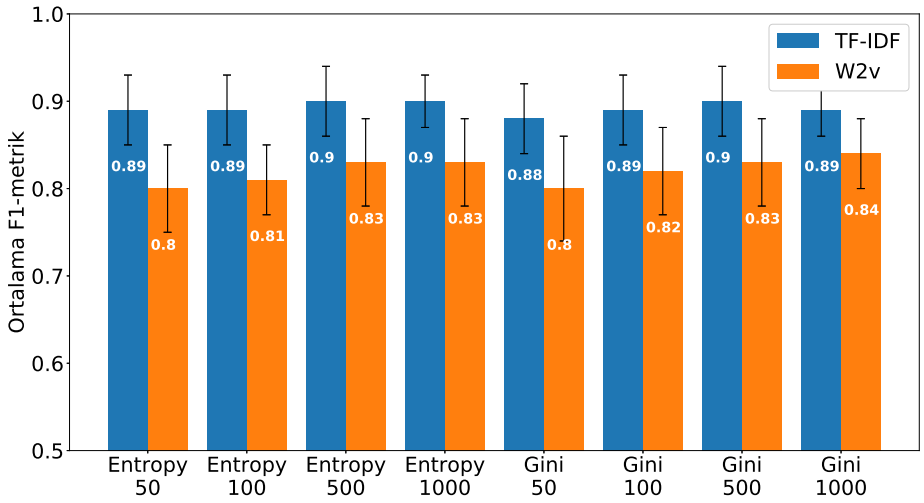
Şekil 4.13'de RF algoritmasının entropy ve gini algoritmaları ve 50, 100, 500 ve 1000 ağaç sayıları ile tüm konuların bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. RF algoritması, TF-IDF kelime temsili yöntemiyle word2vec kelime temsili yöntemine göre daha başarılı olmuştur. Her iki kelime temsili yönteminde de sonuçlar bölme algoritması ve ağaç sayısı değişse de çok fazla farklılık göstermemiştir.



Şekil 4.13. Tüm konuların bulunduğu veri seti kullanılarak RF algoritmasının entropy ve gini algoritmaları ve 50, 100, 500 ve 1000 ağaç sayıları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

Şekil 4.14'de RF algoritmasının entropy ve gini algoritmaları ve 50, 100, 500 ve 1000 ağaç sayıları ile Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. RF algoritması, TF-IDF kelime temsili yöntemiyle word2vec kelime

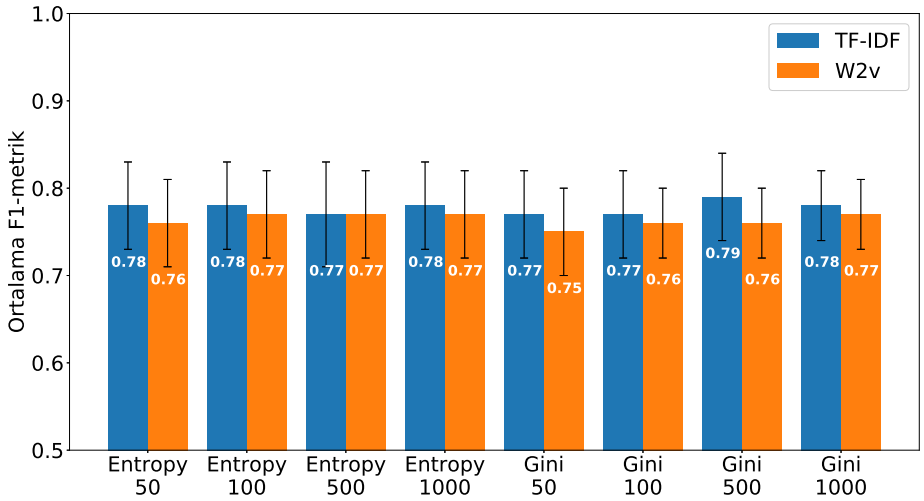
temsili yöntemine göre daha başarılı olmuştur. Her iki kelime temsili yönteminde de sonuçlar bölme algoritması ve ağaç sayısı değişse de çok fazla farklılık göstermemiştir.



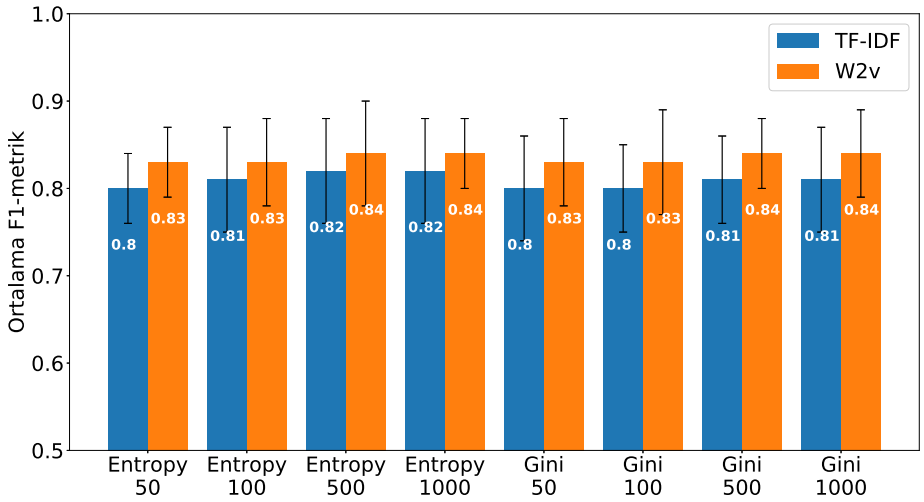
Şekil 4.14. Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak RF algoritmasının entropy ve gini algoritmaları ve 50, 100, 500 ve 1000 ağaç sayıları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

Şekil 4.15'de RF algoritmasının entropy ve gini algoritmaları ve 50, 100, 500 ve 1000 ağaç sayıları ile Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. RF algoritması, TF-IDF kelime temsili yöntemiyle word2vec kelime temsili yöntemine göre daha başarılı olmuştur. Her iki kelime temsili yönteminde de sonuçlar bölme algoritması ve ağaç sayısı değişse de çok fazla farklılık göstermemiştir.

Şekil 4.16'da RF algoritmasının entropy ve gini algoritmaları ve 50, 100, 500 ve 1000 ağaç sayıları ile Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. RF algoritması, TF-IDF kelime temsili yöntemiyle word2vec kelime temsili yöntemine göre daha başarılı olmuştur. Her iki kelime temsili yönteminde de sonuçlar bölme algoritması ve ağaç sayısı değişse de çok fazla farklılık göstermemiştir.



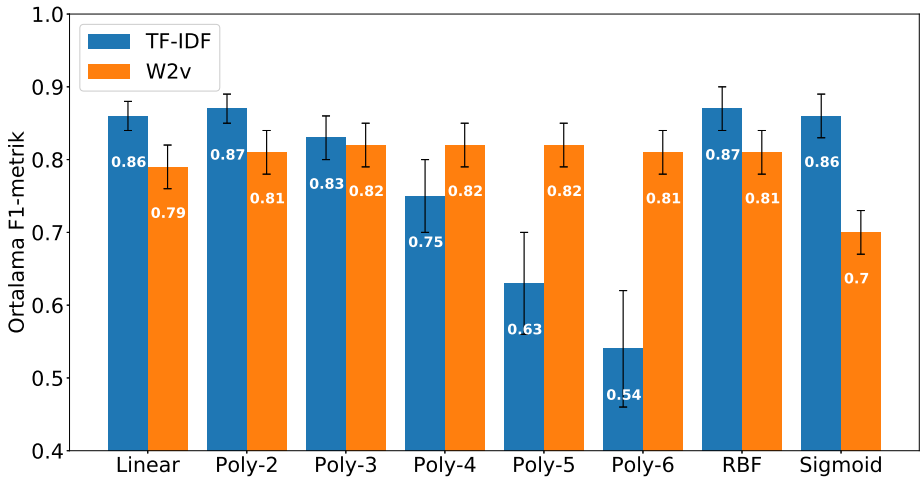
Şekil 4.15. Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak RF algoritmasının entropy ve gini algoritmaları ve 50, 100, 500 ve 1000 ağaç sayıları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.



Şekil 4.16. Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak RF algoritmasının entropy ve gini algoritmaları ve 50, 100, 500 ve 1000 ağaç sayıları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

SVM algoritması, lineer, radyal tabanlı fonksiyon, sigmoid ve polinom algoritmaları ile test edilmiştir. Aynı zamanda polinom algoritmasında, 2,3,4,5 ve 6 derece ile test edilmiştir. Etropy ve gini bölme kalitesi ölçme algoritmaları kullanılmıştır. Ayrıca, 50, 100, 500 ve 1000 ağaç sayısı ile algoritma test edilmiştir. Tüm konuların tweet mesajlarının bulunduğu, Konu-1 tweet mesajlarının bulunduğu, Konu-2 tweet mesajlarının bulunduğu ve Konu-3 tweet mesajlarının bulunduğu veri seti olmak üzere 4 farklı veri seti ile algoritma çalıştırılmıştır. TF-IDF ve Word2vec kelime temsili yöntemlerinin sonuçları karşılaştırılmıştır.

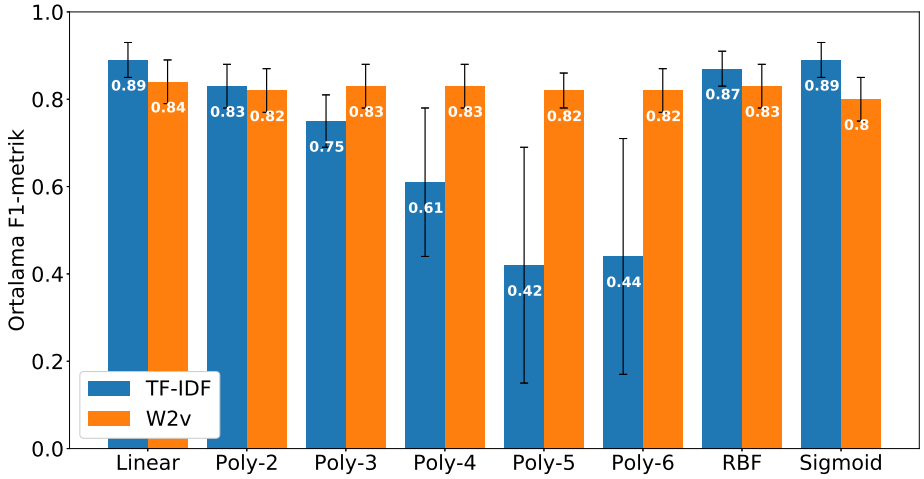
Şekil 4.17'de SVM algoritmasının lineer, polinom (2, 3, 4, 5 ve 6. dereceden), radyal tabanlı fonksiyon ve sigmoid algoritmaları ile tüm konuların bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. SVM algoritması, TF-IDF kelime temsili yöntemiyle word2vec kelime temsili yöntemine göre daha başarılı olmuştur. TF-IDF kelime temsili yöntemi ile 2. derece polinom ve radyal tabanlı fonksiyon algoritmaları ile 0,87 F1-metrik değeri ile diğer modellerden başarılı olmuştur.



Şekil 4.17. Tüm konuların bulunduğu veri seti kullanılarak SVM algoritmasının lineer, polinom (2, 3, 4, 5 ve 6. dereceden), radyal tabanlı fonksiyon ve sigmoid algoritmaları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

Şekil 4.18'de SVM algoritmasının lineer, polinom (2, 3, 4, 5 ve 6. dereceden), radyal tabanlı fonksiyon ve sigmoid algoritmaları ile Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları

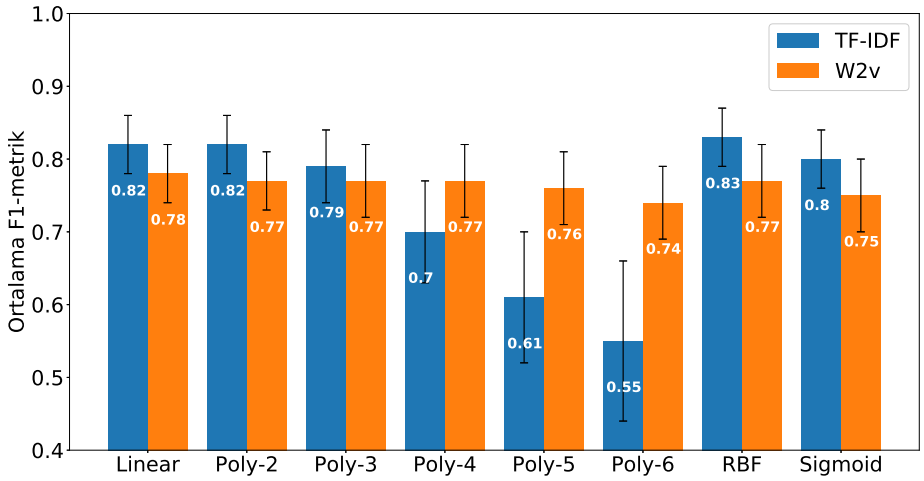
hata çubukları grafiğinde gösterilmiştir. SVM algoritması, TF-IDF kelime temsili yöntemiyle word2vec kelime temsili yöntemine göre daha başarılı olmuştur. TF-IDF kelime temsili yöntemi ile 2. derece polinom ve radyal tabanlı fonksiyon algoritmaları ile 0,87 F1-metrik değeri ile diğer modellerden başarılı olmuştur.



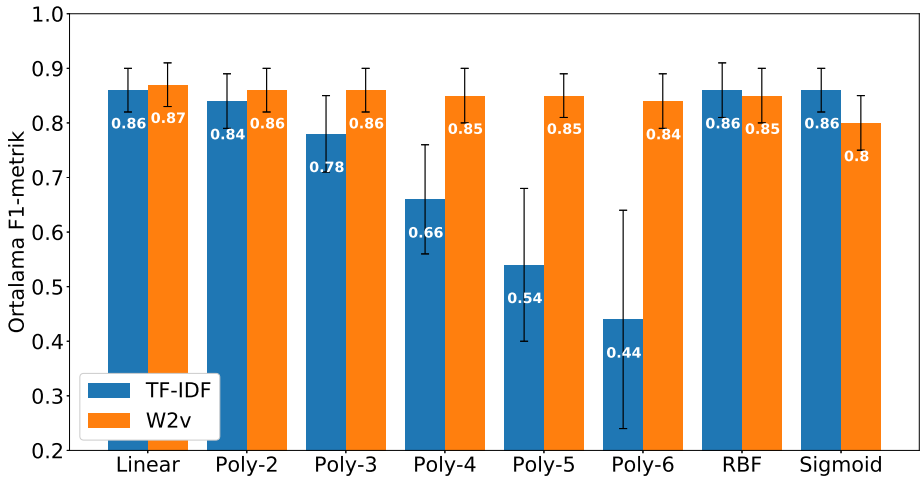
Şekil 4.18. Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak SVM algoritmasının lineer, polinom (2, 3, 4, 5 ve 6. dereceden), radyal tabanlı fonksiyon ve sigmoid algoritmaları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

Şekil 4.19'da SVM algoritmasının lineer, polinom (2, 3, 4, 5 ve 6. dereceden), radyal tabanlı fonksiyon ve sigmoid algoritmaları ile Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. SVM algoritması, TF-IDF kelime temsili yöntemiyle word2vec kelime temsili yöntemine göre daha başarılı olmuştur. TF-IDF kelime temsili yöntemi ile 2. derece polinom ve radyal tabanlı fonksiyon algoritmaları ile 0,87 F1-metrik değeri ile diğer modellerden başarılı olmuştur.

Şekil 4.20'de SVM algoritmasının lineer, polinom (2, 3, 4, 5 ve 6. dereceden), radyal tabanlı fonksiyon ve sigmoid algoritmaları ile Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. SVM algoritması, TF-IDF kelime temsili yöntemiyle word2vec kelime temsili yöntemine göre daha başarılı olmuştur. TF-IDF kelime temsili yöntemi ile 2. derece polinom ve radyal tabanlı fonksiyon algoritmaları ile 0,87 F1-metrik değeri ile diğer modellerden başarılı olmuştur.



Şekil 4.19. Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak SVM algoritmasının lineer, polinom (2, 3, 4, 5 ve 6. dereceden), radyal tabanlı fonksiyon ve sigmoid algoritmaları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.



Şekil 4.20. Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak SVM algoritmasının lineer, polinom (2, 3, 4, 5 ve 6. dereceden), radyal tabanlı fonksiyon ve sigmoid algoritmaları ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

4.3. Derin Öğrenme Algoritmaları

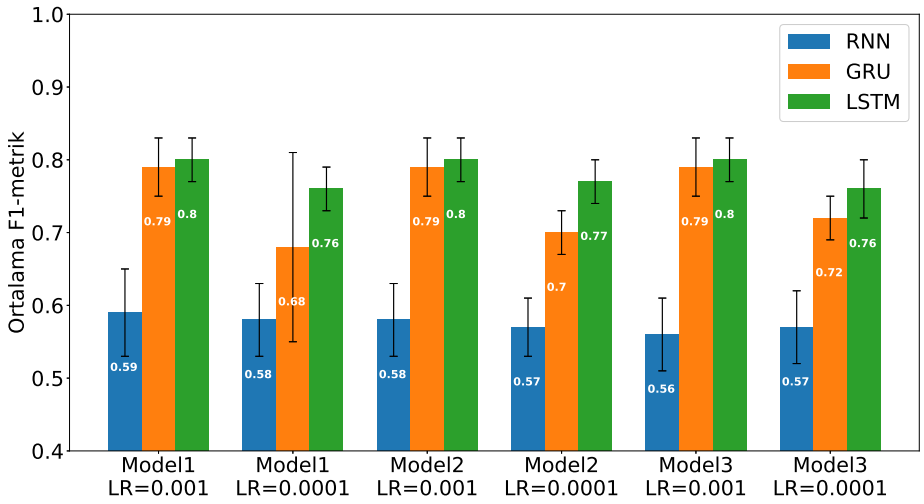
Bu çalışmada, Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Long-Short Term Memory (LSTM) ve bu algoritmaların çift yönlü algoritmaları (BiRNN, BiGRU ve BiLSTM) olmak üzere 6 farklı denetimli derin öğrenme algoritmaları kullanılmıştır. Algoritmaları oluşturmak ve modellemek için Python programlama dili kullanılmıştır. Modellerin çalıştırılması için ise Tensorflow kütüphanesinin Keras Dizi Modeli (The sequence Model) uygulama programlama arayüzü kullanılmıştır (TensorFlow, 2019).

Denetimli derin öğrenme algoritmalarında, tüm tweet mesajlarının bulunduğu, Konu-1 tweet mesajlarının bulunduğu, Konu-2 tweet mesajlarının bulunduğu ve Konu-3 tweet mesajlarının bulunduğu veri seti olmak üzere 4 farklı veri seti ile çalışılmış ve tweet mesajlarının vektörler ile temsil edilmesi için word2vec kelime temsili yöntemi kullanılmıştır. Algoritmaların ön yargısını kaldırmak için her algoritma 100 defa çalıştırılmıştır. Her adımda, konulardaki tweet mesajları rastgele karıştırılmış ve tüm mesajların hem eğitim setinde hem de test setinde bulunabilmesi sağlanmıştır. Her adımda karıştırılan veri setinin, %70'i modeli eğitmek için eğitim seti, %30'u ise bu algoritmaların başarısını ölçmek için test seti olarak kullanılmıştır.

Denetimli derin öğrenme algoritmalarının her biri 3 farklı model ve 2 farklı öğrenme katsayısı (learning rate-LR) ile çalıştırılmış ve sonuçlar test edilmiştir. Giriş katmanı, 50 birimden oluşan gizli katman, 10 birimden oluşan gizli katman ve 1 birimlik çıkış katmanına sahip modele Model-1, giriş katmanı, 50 birimden oluşan gizli katman, 25 birimden oluşan gizli katman, 10 birimden oluşan gizli katman ve 1 birimlik çıkış katmanına sahip modele Model-2 ve son olarak giriş katmanı, 100 birimden oluşan gizli katman, 50 birimden oluşan gizli katman, 25 birimden oluşan gizli katman, 10 birimden oluşan gizli katman ve 1 birimlik çıkış katmanına sahip modele Model-3 adı verilmiştir. Bu 3 model 0,001 ve 0,0001 öğrenme katsayıları ile denenmiştir. Aktivasyon fonksiyonu olarak sigmoid aktivasyon fonksiyonu kullanılmıştır. Optimizasyon algoritması olarak Adam optimizasyon algoritması kullanılmıştır. Kayıp fonksiyonu için ise ikili çapraz entropi (binary-crossentropy) kayıp fonksiyonu kullanılmıştır.

Toplanan tüm tweet mesajlarındaki kelimeler 1x100 boyutlu word2vec vektörlerine dönüştürülmüştür. Tweet mesajlarının büyük çoğunluğunun 50 kelime ve daha az kelimeden oluştuğu görülmüştür. Bu nedenle bir tweet mesajı için 50x100'lük (kelime sayısı x word2vec boyutu) bir vektör oluşturulmuştur. Kelime sayısı 50'den az olan tweet mesajlarının matris boyutunu 50x100 boyutuna tamamlamak için 50x100 boyutlu matris olana kadar 1x100 boyutlu sıfır vektörü eklenmiştir.

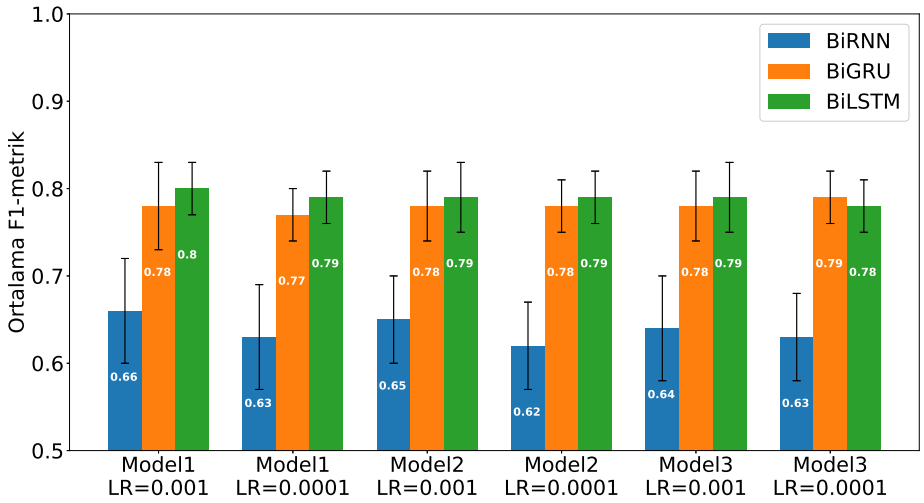
Tüm konuların tweet mesajlarından oluşan veri seti ile elde edilen tek yönlü denetimli derin öğrenme algoritmalarının sonuçları Şekil 4.21'de gösterilmiştir. Tek yönlü denetimli öğrenme algoritmalarının tüm modellerinde LSTM algoritması başarılı olmuştur. LSTM algoritması, LR değerinin 0,001 olarak kullanıldığı modellerde 0,8 F1-metrik değeri ile en başarılı algoritma olmuştur. RNN algoritması ise en fazla 0,59 F1-metrik değeri ile her modelde en başarısız sonucu göstermiştir.



Şekil 4.21. Tüm konuların tweet mesajlarının bulunduğu veri seti kullanılarak 3 farklı model ve 2 farklı LR değeri ile elde edilen tek yönlü derin öğrenme algoritmalarının F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

Tüm konuların tweet mesajlarından oluşan veri seti ile elde edilen çift yönlü denetimli derin öğrenme algoritmalarının sonuçları Şekil 4.22'de gösterilmiştir. Çift yönlü denetimli öğrenme algoritmalarının Model-3 ve LR 0,0001 olarak kullanıldığı model hariç, tüm modellerinde BiLSTM algoritması en başarılı algoritma olmuştur. BiLSTM algoritması, Model-1 ve LR 0,001 değerinde kullanıldığında 0,80 F1-metrik değeri

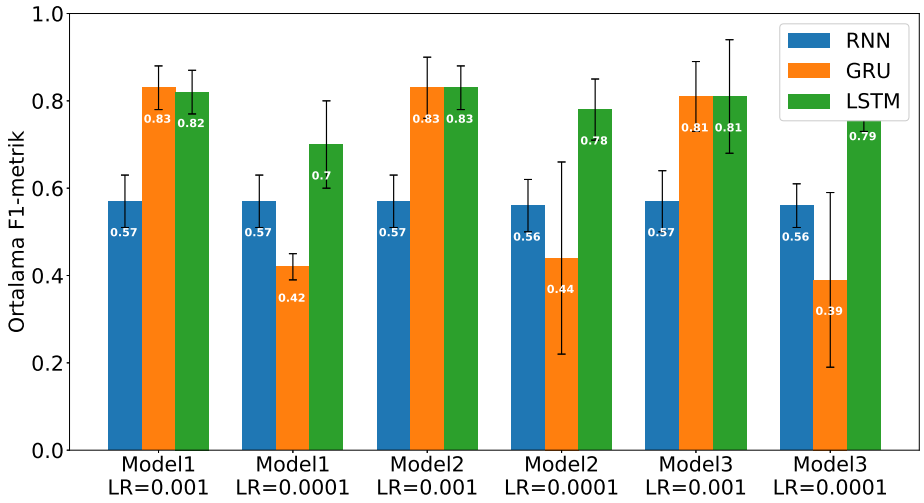
göstererek en başarılı algoritma olmuştur. BiRNN algoritması ise en fazla 0,66 F1-metrik değeri ile her modelde en başarısız sonucu göstermiştir.



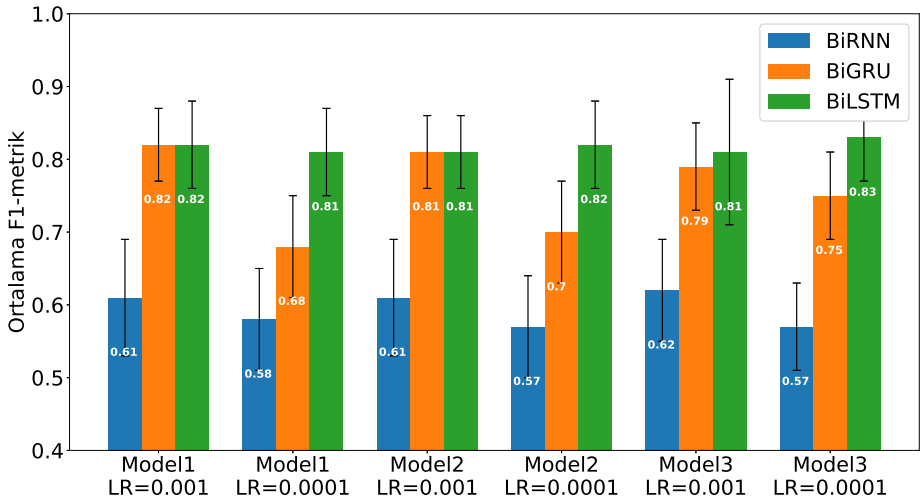
Şekil 4.22. Tüm konuların tweet mesajlarının bulunduğu veri seti kullanılarak 3 farklı model ve 2 farklı LR değeri ile elde edilen çift yönlü derin öğrenme algoritmalarının F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

Konu-1 tweet mesajlarından oluşan veri seti ile elde edilen tek yönlü denetimli derin öğrenme algoritmalarının sonuçları Şekil 4.23'de gösterilmiştir. Tek yönlü denetimli öğrenme algoritmalarının LR değerinin 0,001 olarak kullanıldığı modellerinde LSTM ve GRU algoritmaları başarılı olmuştur. Model-1 ve LR değerinin 0,001 olarak kullanıldığı modelde GRU algoritması, Model-2 ve LR değerinin 0,001 olarak kullanıldığı modelde ise LSTM ve GRU algoritmaları 0,83 F1-metrik değeri ile en başarılı algoritma olmuştur. LR değerinin 0,0001 olarak kullanıldığı modellerde ise en başarısız sonucu GRU algoritması göstermiştir. Model-3 ve LR değerinin 0,0001 olarak kullanıldığı modelde GRU algoritması 0,39 F1-metrik değeri ile en düşük F1-metrik değerini göstermiştir.

Konu-1 tweet mesajlarından oluşan veri seti ile elde edilen çift yönlü denetimli derin öğrenme algoritmalarının sonuçları Şekil 4.24'de gösterilmiştir. Çift yönlü denetimli öğrenme algoritmalarının Model-3 ve LR 0,0001 olarak kullanıldığı modelde 0,83 F1-metrik değeri ile BiLSTM algoritması en başarılı algoritma olmuştur. BiLSTM algoritması tüm modellerde en başarılı algoritma olmuştur. BiRNN algoritması ise en fazla 0,62 F1-metrik değeri ile her modelde en başarısız sonucu göstermiştir.

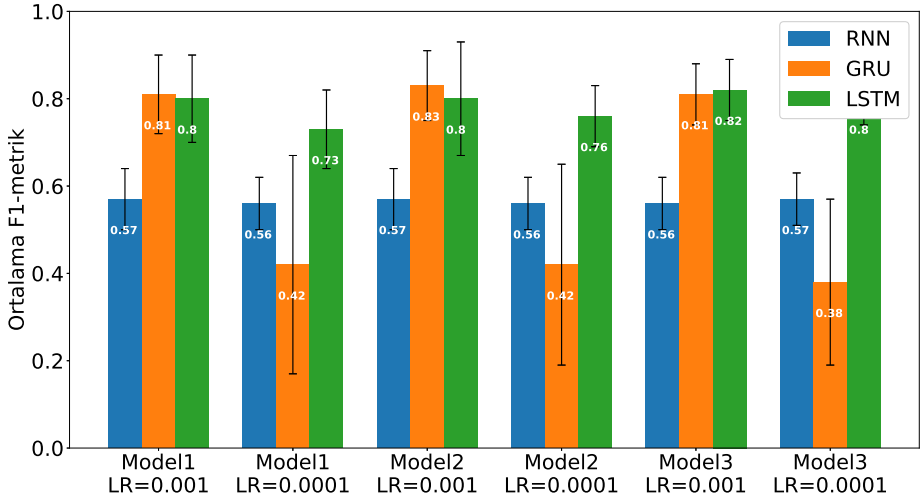


Şekil 4.23. Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak 3 farklı model ve 2 farklı LR değeri ile elde edilen tek yönlü derin öğrenme algoritmalarının F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.



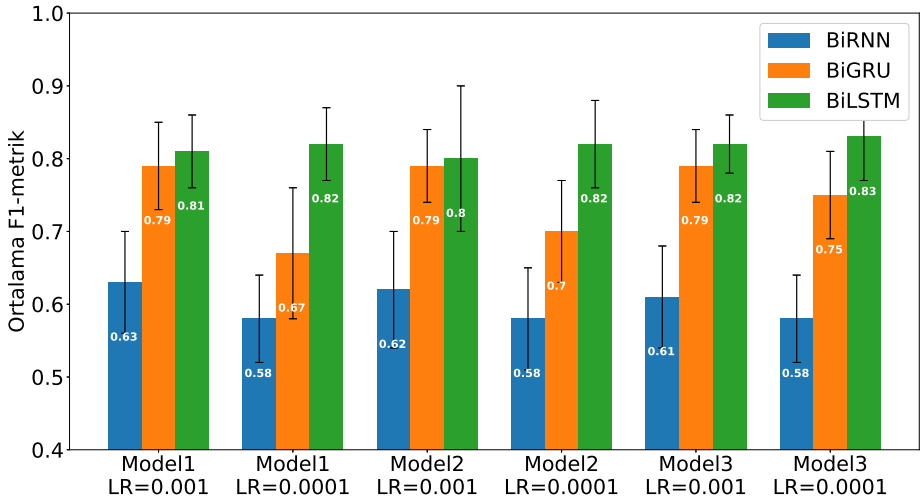
Şekil 4.24. Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak 3 farklı model ve 2 farklı LR değeri ile elde edilen çift yönlü derin öğrenme algoritmalarının F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

Konu-2 tweet mesajlarından oluşan veri seti ile elde edilen tek yönlü denetimli derin öğrenme algoritmalarının sonuçları Şekil 4.25'de gösterilmiştir. Tek yönlü denetimli öğrenme algoritmalarının LR değerinin 0,001 olarak kullanıldığı modellerinde LSTM ve GRU algoritmaları başarılı ve birbirlerine yakın F1-metrik değerleri vermişlerdir. Model-2 ve LR değerinin 0,001 olarak kullanıldığı modelde GRU algoritması 0,83 F1-metrik değeri ile en başarılı algoritma olmuştur. LR değerinin 0,0001 olarak kullanıldığı modellerde ise en başarısız sonucu GRU algoritması, en başarılı sonucu ise LSTM algoritması göstermiştir. Model-3 ve LR değerinin 0,0001 olarak kullanıldığı modelde GRU algoritması 0,38 F1-metrik değeri ile en düşük F1-metrik değerini göstermiştir.



Şekil 4.25. Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak 3 farklı model ve 2 farklı LR değeri ile elde edilen tek yönlü derin öğrenme algoritmalarının F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

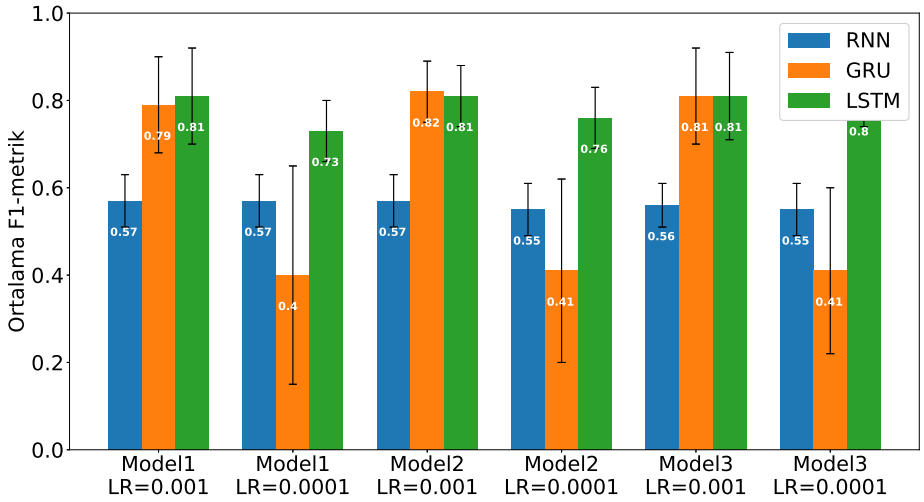
Konu-2 tweet mesajlarından oluşan veri seti ile elde edilen çift yönlü denetimli derin öğrenme algoritmalarının sonuçları Şekil 4.26'da gösterilmiştir. Bu veri seti kullanıldığında her modelde BiLSTM algoritması diğer algoritmalardan daha başarılı olmuştur. Çift yönlü denetimli öğrenme algoritmalarının Model-3 ve LR 0,0001 olarak kullanıldığı modelde 0,83 F1-metrik değeri ile BiLSTM algoritması en başarılı algoritma olmuştur. BiRNN algoritması ise en fazla 0,62 F1-metrik değeri ile her modelde en başarısız sonucu göstermiştir.



Şekil 4.26. Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak 3 farklı model ve 2 farklı LR değeri ile elde edilen çift yönlü derin öğrenme algoritmalarının F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

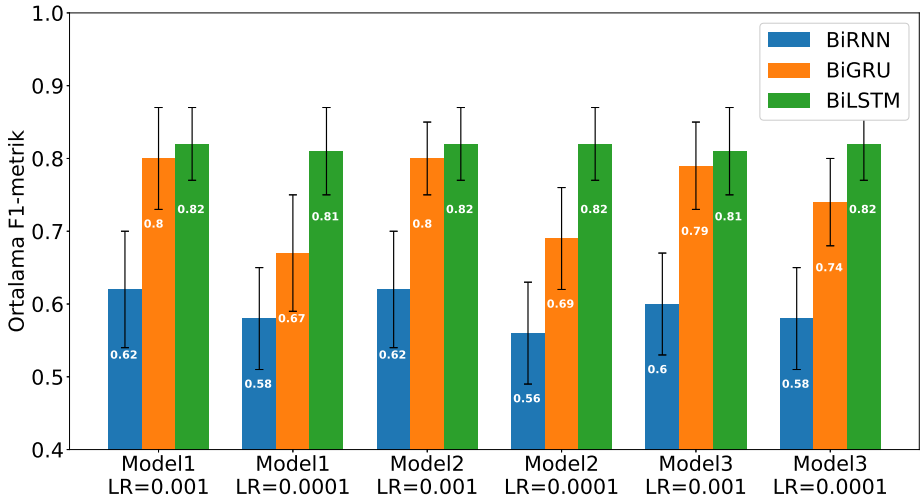
Konu-3 tweet mesajlarından oluşan veri seti ile elde edilen tek yönlü denetimli derin öğrenme algoritmalarının sonuçları Şekil 4.27'de gösterilmiştir. Tek yönlü denetimli öğrenme algoritmalarının LR değerinin 0,001 olarak kullanıldığı modellerinde LSTM ve GRU algoritmaları başarılı ve birbirlerine yakın F1-metrik değerleri vermişlerdir. Model-2 ve LR değerinin 0,001 olarak kullanıldığı modelde GRU algoritması 0,82 F1-metrik değeri ile en başarılı algoritma olmuştur. LR değerinin 0,0001 olarak kullanıldığı modellerde ise en başarısız sonucu GRU algoritması en başarılı sonucu ise LSTM algoritması göstermiştir. Model-1 ve LR değerinin 0,0001 olarak kullanıldığı modelde GRU algoritması 0,40 F1-metrik değeri ile en düşük F1-metrik değerini göstermiştir.

Konu-3 tweet mesajlarından oluşan veri seti ile elde edilen çift yönlü denetimli derin öğrenme algoritmalarının sonuçları Şekil 4.28'de gösterilmiştir. Bu veri seti kullanıldığında her modelde BiLSTM algoritması diğer algoritmalardan daha başarılı olmuştur. Çift yönlü denetimli öğrenme algoritmalarından 0,82 F1-metrik değeri ile BiLSTM algoritması, en başarılı algoritma olmuştur. LR değerinin 0,001 olarak kullanıldığı modellerde BiGRU algoritması BiLSTM algoritmasına yakın F1-metrik değerleri gösterirken, LR değerinin 0,0001 olarak kullanıldığı modellerde F1-metrik



Şekil 4.27. Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak 3 farklı model ve 2 farklı LR değeri ile elde edilen tek yönlü derin öğrenme algoritmalarının F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

değeri düşmüştür. BiRNN algoritması ise en fazla 0,62 F1-metrik değeri ile her modelde en başarısız sonucu göstermiştir.



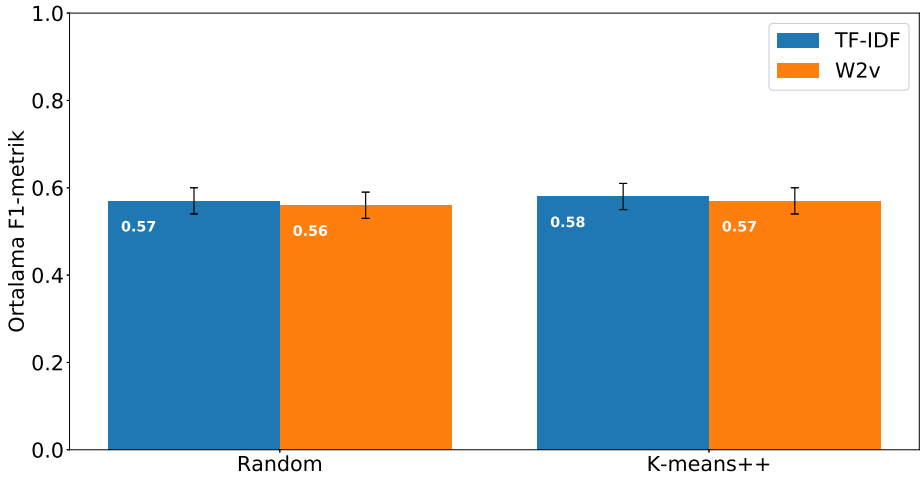
Şekil 4.28. Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak 3 farklı model ve 2 farklı LR değeri ile elde edilen çift yönlü derin öğrenme algoritmalarının F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

4.4. Denetimsiz Öğrenme Algoritmaları

Bu çalışmada, K-means, Non-negative Matrix Factorization (NMF) ve Linear Discriminant Analysis (LDA) denetimsiz öğrenme algoritmaları kullanılmıştır. Denetimsiz öğrenme algoritmalarında, tüm tweet mesajlarının bulunduğu, Konu-1 tweet mesajlarının bulunduğu, Konu-2 tweet mesajlarının bulunduğu ve Konu-3 tweet mesajlarının bulunduğu veri seti olmak üzere 4 farklı veri seti ile çalışılmış ve tweet mesajlarının vektörler ile temsil edilmesi için TF-IDF ve word2vec yöntemleri kullanılmıştır. Algoritmaların ön yargısını kaldırmak için her algoritma 100 defa çalıştırılmıştır. Her adımda, konulardaki tweet mesajları rastgele karıştırılmış ve tüm mesajların hem eğitim setinde hem de test setinde bulunabilmesi sağlanmıştır. Her adımda karıştırılan veri setinin, %70'i modeli eğitmek için eğitim seti, %30'u ise bu algoritmaların başarısını ölçmek için test seti olarak kullanılmıştır.

K-means algoritması, merkez noktaları belirleme ölçütü olarak rastgele (random) belirleme yöntemi ve k-means++ yöntemi ile test edilmiştir. Tüm konuların tweet mesajlarının bulunduğu, Konu-1 tweet mesajlarının bulunduğu, Konu-2 tweet mesajlarının bulunduğu ve Konu-3 tweet mesajlarının bulunduğu veri seti olmak üzere 4 farklı veri seti ile algoritma çalıştırılmıştır. TF-IDF ve Word2vec kelime temsili yöntemlerinin sonuçları karşılaştırılmıştır.

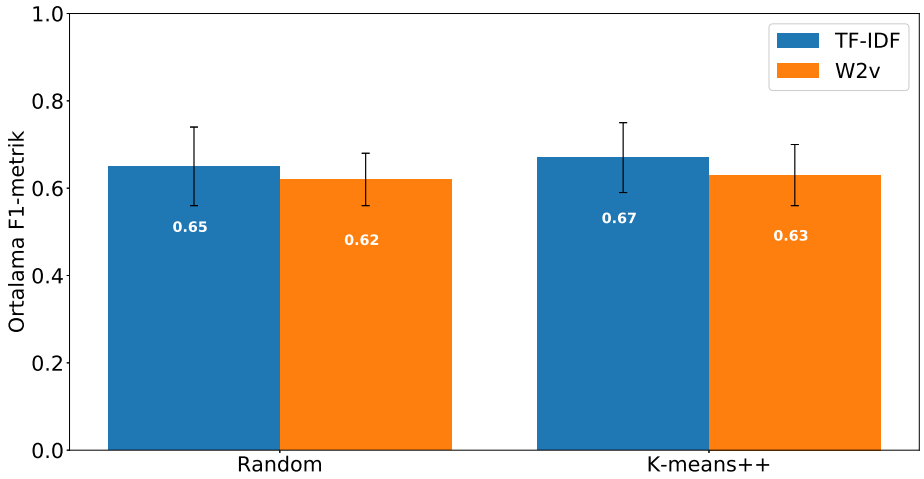
Şekil 4.29'da, K-means algoritmasının, merkez noktaları belirleme ölçütü olarak rastgele (random) belirleme yöntemi ve k-means++ yöntemi ile tüm konuların bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. K-means algoritması, k-means++ merkez belirleme yöntemi kullanıldığında, 0,58 F1-metrik değeri ile en başarılı sonucu, TF-IDF kelime temsili yöntemi ile vermiştir. Word2vec kelime temsili yöntemiyle ise, k-means++ merkez belirleme yöntemi kullanıldığında, 0,57 F1-metrik değeri ile en başarılı sonucu göstermiştir. Hem TF-IDF, hem de word2vec kelime temsili yöntemleri birbirine çok yakın sonuçlar vermiştir.



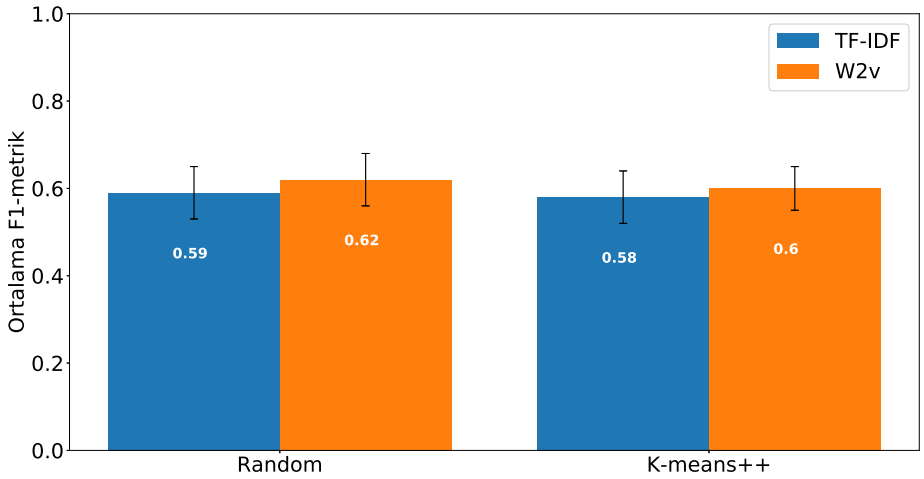
Şekil 4.29. Tüm konuların bulunduğu veri seti kullanılarak K-means algoritmasının rastgele merkez noktası belirleme ve k-means++ merkez noktası belirleme yöntemi ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

Şekil 4.30'da, K-means algoritmasının, merkez noktaları belirleme ölçütü olarak rastgele (random) belirleme yöntemi ve k-means++ yöntemi ile Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. K-means algoritması, k-means++ merkez belirleme yöntemi kullanıldığında, 0,67 F1-metrik değeri ile en başarılı sonucu, TF-IDF kelime temsili yöntemi ile vermiştir. Word2vec kelime temsili yöntemiyle ise, k-means++ merkez belirleme yöntemi kullanıldığında, 0,63 F1-metrik değeri ile en başarılı sonucu göstermiştir.

Şekil 4.31'de, K-means algoritmasının, merkez noktaları belirleme ölçütü olarak rastgele (random) belirleme yöntemi ve k-means++ yöntemi ile Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. K-means algoritması, k-means++ merkez belirleme yöntemi kullanıldığında, 0,62 F1-metrik değeri ile en başarılı sonucu, Word2vec kelime temsili yöntemi ile vermiştir. TF-IDF kelime temsili yöntemiyle ise, rastgele merkez belirleme yöntemi kullanıldığında, 0,63 F1-metrik değeri ile en başarılı sonucu göstermiştir.

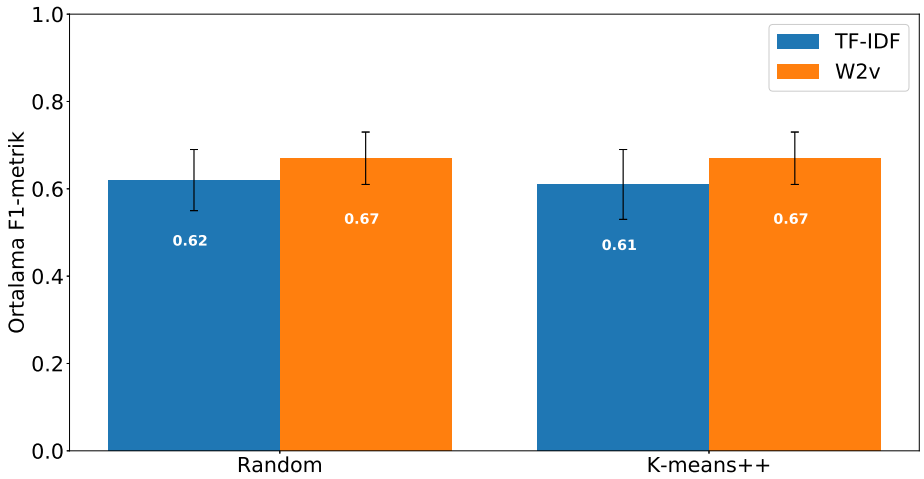


Şekil 4.30. Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak K-means algoritmasının rastgele merkez noktası belirleme ve k-means++ merkez noktası belirleme yöntemi ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.



Şekil 4.31. Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak K-means algoritmasının rastgele merkez noktası belirleme ve k-means++ merkez noktası belirleme yöntemi ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

Şekil 4.32'de, K-means algoritmasının, merkez noktaları belirleme ölçütü olarak rastgele (random) belirleme yöntemi ve k-means++ yöntemi ile Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. K-means algoritması hem rastgele hem de k-means++ merkez belirleme yöntemi kullanıldığında, 0,67 F1-metrik değeri ile en başarılı sonucu, Word2vec kelime temsili yöntemi ile vermiştir. TF-IDF kelime temsili yöntemiyle ise, rastgele merkez belirleme yöntemi kullanıldığında, 0,62 F1-metrik değeri ile en başarılı sonucu göstermiştir.

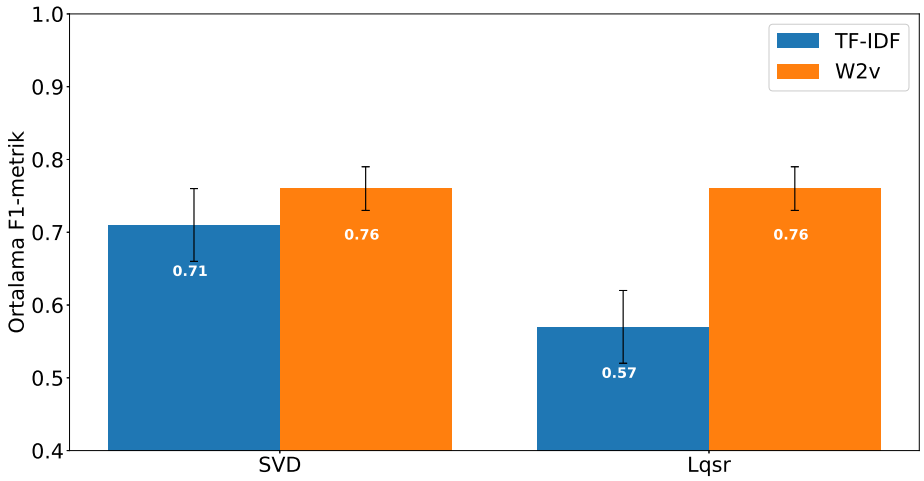


Şekil 4.32. Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak K-means algoritmasının rastgele merkez noktası belirleme ve k-means++ merkez noktası belirleme yöntemi ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

LDA algoritması, singular value decomposition (SVD) ve least squares (Lsqr) yöntemleri ile test edilmiştir. Tüm konuların tweet mesajlarının bulunduğu, Konu-1 tweet mesajlarının bulunduğu, Konu-2 tweet mesajlarının bulunduğu ve Konu-3 tweet mesajlarının bulunduğu veri seti olmak üzere 4 farklı veri seti ile algoritma çalıştırılmıştır. TF-IDF ve Word2vec kelime temsili yöntemlerinin sonuçları karşılaştırılmıştır.

Şekil 4.33'de, LDA algoritmasının, SVD ve Lsqr yöntemleri ile tüm konuların bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. LDA algoritması hem SVD hem de Lsqr yöntemleri kullanıldığında, 0,76 F1-metrik değeri ile en başarılı sonucu,

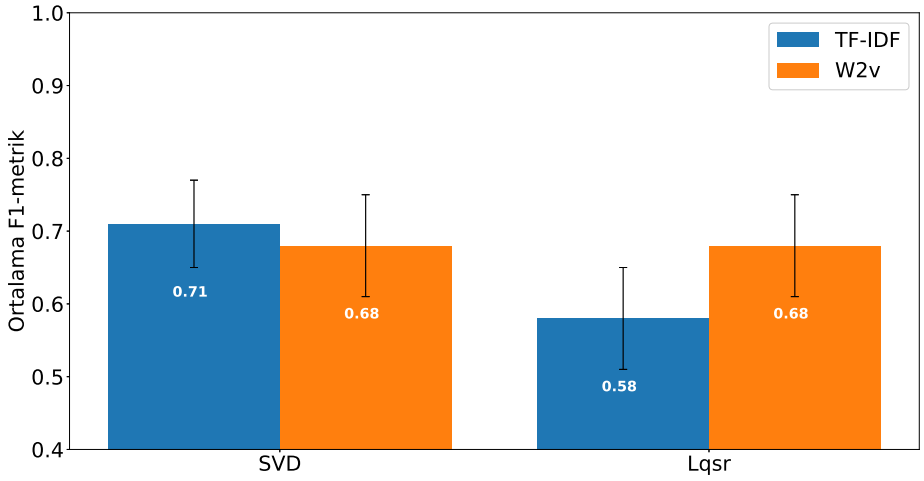
word2vec kelime temsili yöntemi ile vermiştir. TF-IDF kelime temsili yöntemiyle ise SVD yöntemi kullanıldığında, 0,71 F1-metrik değeri ile en başarılı sonucu göstermiştir.



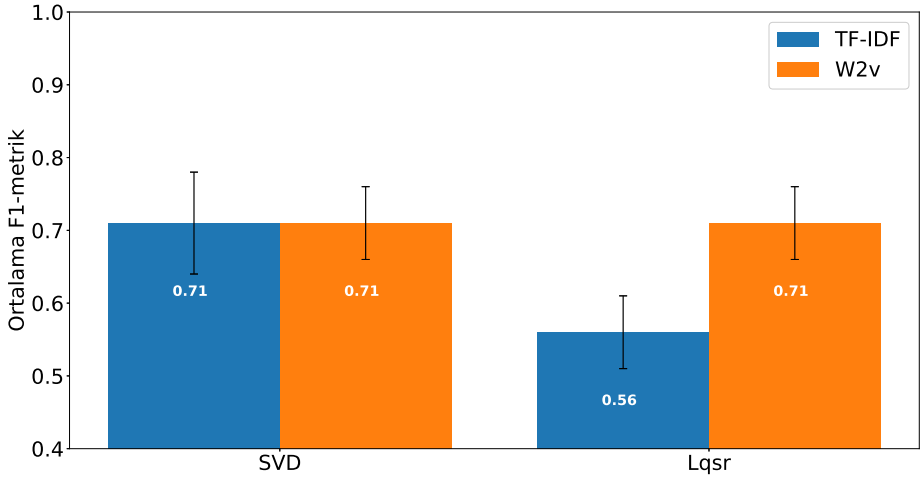
Şekil 4.33. Tüm konuların bulunduğu veri seti kullanılarak LDA algoritmasının SVD ve Lsqr yöntemleri ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

Şekil 4.34'de, LDA algoritmasının, SVD ve Lsqr yöntemleri ile Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. LDA algoritması, SVD yöntemi kullanıldığında, 0,71 F1-metrik değeri ile en başarılı sonucu, TF-IDF kelime temsili yöntemi ile vermiştir. Word2vec kelime temsili yöntemiyle ise, hem SVD hem de Lsqr yöntemleri kullanıldığında, 0,68 F1-metrik değeri ile en başarılı sonucu göstermiştir.

Şekil 4.35'de, LDA algoritmasının, SVD ve Lsqr yöntemleri ile Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. LDA algoritması, SVD yöntemi kullanıldığında, 0,71 F1-metrik değeri ile en başarılı sonucu, hem TF-IDF hem de word2vec kelime temsili yöntemi ile vermiştir. Ayrıca Lsqr yöntemi kullanıldığında da word2vec kelime temsili yöntemiyle, yine 0,71 F1-metrik değerini vermiştir. Lsqr yöntemi ile TF-IDF kelime temsili yöntemi kullanıldığında ise 0,56 metrik değeri gösterebilmiştir.

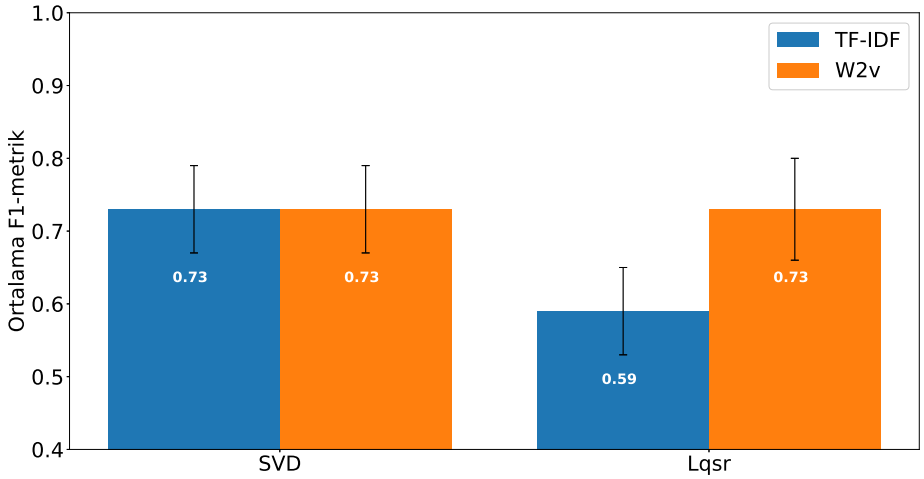


Şekil 4.34. Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak LDA algoritmasının SVD ve Lsqr yöntemleri ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.



Şekil 4.35. Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak LDA algoritmasının SVD ve Lsqr yöntemleri ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

Şekil 4.36'da, LDA algoritmasının, SVD ve Lsqr yöntemleri ile Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. LDA algoritması, SVD yöntemi kullanıldığında, 0,73 F1-metrik değeri ile en başarılı sonucu, hem TF-IDF hem de word2vec kelime temsili yöntemi ile vermiştir. Ayrıca Lsqr yöntemi kullanıldığında da word2vec kelime temsili yöntemiyle, yine 0,73 F1-metrik değerini vermiştir. Lsqr yöntemi ile TF-IDF kelime temsili yöntemi kullanıldığında ise 0,59 metrik değeri gösterebilmiştir.

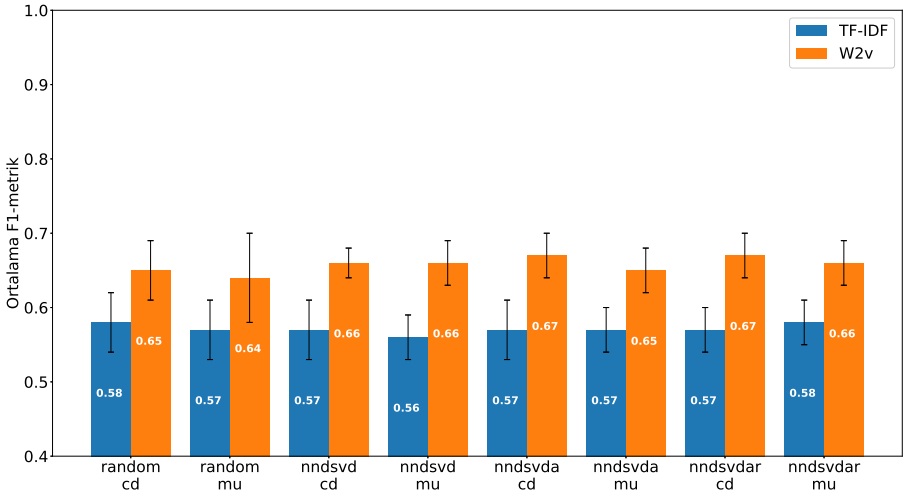


Şekil 4.36. Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak LDA algoritmasının SVD ve Lsqr yöntemleri ile elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

NMF algoritması, singular value decomposition (SVD) ve least squares (Lsqr) yöntemleri ile test edilmiştir. Tüm konuların tweet mesajlarının bulunduğu, Konu-1 tweet mesajlarının bulunduğu, Konu-2 tweet mesajlarının bulunduğu ve Konu-3 tweet mesajlarının bulunduğu veri seti olmak üzere 4 farklı veri seti ile algoritma çalıştırılmıştır. TF-IDF ve Word2vec kelime temsili yöntemlerinin sonuçları karşılaştırılmıştır.

Şekil 4.37'de, NMF algoritmasının, farklı parametrelerle tüm konuların tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının, F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. NMF algoritması, farklı parametrelerle kullanıldığında, her modelde en başarılı sonucu,

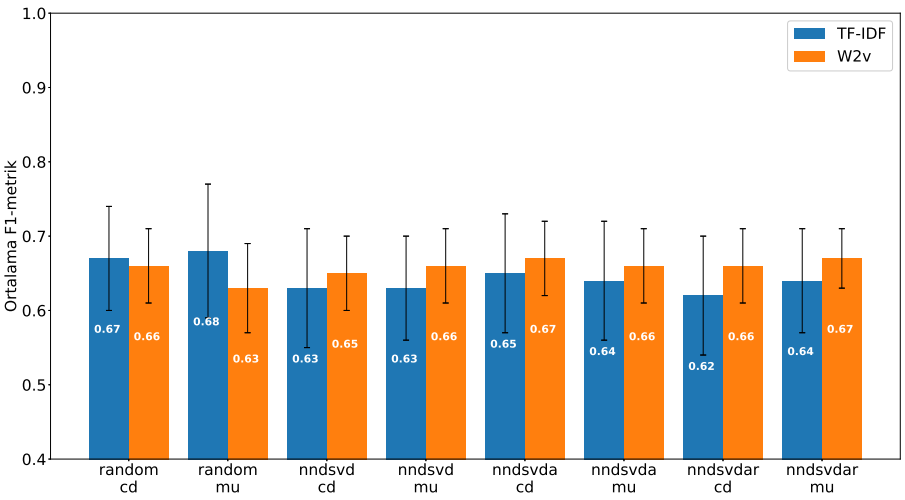
word2vec kelime temsili yöntemi ile vermiştir. NMF algoritması bu veri setinde, başlangıç prosedürü olarak NNDSVDA ile NNDSVDAR ve çözücü olarak CD çözücüsü kullanıldığında 0,67 F1-metrik değeri ile en başarılı sonucu word2vec kelime temsili yöntemi ile vermiştir. NMF algoritmasının tüm parametreleriyle TF-IDF kelime temsili yöntemi yeterli başarıyı gösterememiştir. TF-IDF kelime temsili yöntemiyle, başlangıç prosedürü olarak NNDSVDAR ve çözücü olarak MU çözücüsü kullanıldığında, 0,58 F1-metrik değeri ile en yüksek sonucu gösterebilmiştir.



Şekil 4.37. Tüm konuların bulunduğu veri seti kullanılarak NMF algoritmasının farklı parametrelerle elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

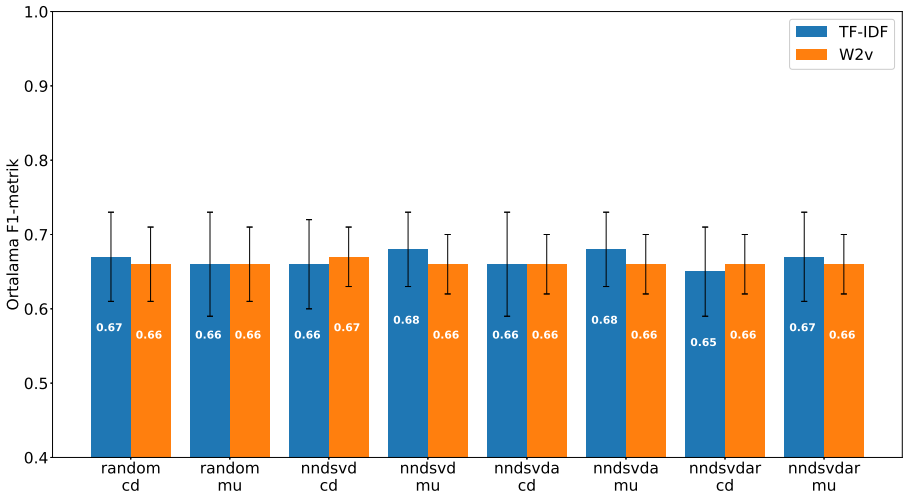
Şekil 4.38'de, NMF algoritmasının, farklı parametrelerle Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının, F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. NMF algoritması, farklı parametrelerle kullanıldığında, başlangıç prosedürü olarak rasgele prosedürü kullanıldığı model hariç diğer modellerde en başarılı sonucu, word2vec kelime temsili yöntemi ile vermiştir. NMF algoritması bu veri setinde, başlangıç prosedürü olarak rastgele prosedür ve çözücü olarak MU çözücüsü kullanıldığında 0,68 F1-metrik değeri ile en başarılı sonucu TF-IDF kelime temsili yöntemi ile vermiştir. Word2vec kelime temsili yöntemiyle, başlangıç prosedürü olarak NNDSVDAR ve çözücü olarak MU çözücüsü kullanıldığında, 0,67 F1-metrik değeri ile en yüksek sonucu göstermiştir. TF-IDF ve word2vec kelime temsili yöntemlerinin sonuçları birbirine yakın çıkmıştır. Word2vec ve TF-IDF kelime temsili yöntemlerinin en başarılı

sonuçlarına bakıldığında TF-IDF kelime temsili yönteminin F1-metrik değerinin, word2vec yönteminin F1-metrik değerinden 0,01 daha başarılı olsa da, word2vec yönteminin standart sapmasının daha düşük olduğu görülmektedir.



Şekil 4.38. Konu-1 tweet mesajlarının bulunduğu veri seti kullanılarak NMF algoritmasının farklı parametrelerle elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

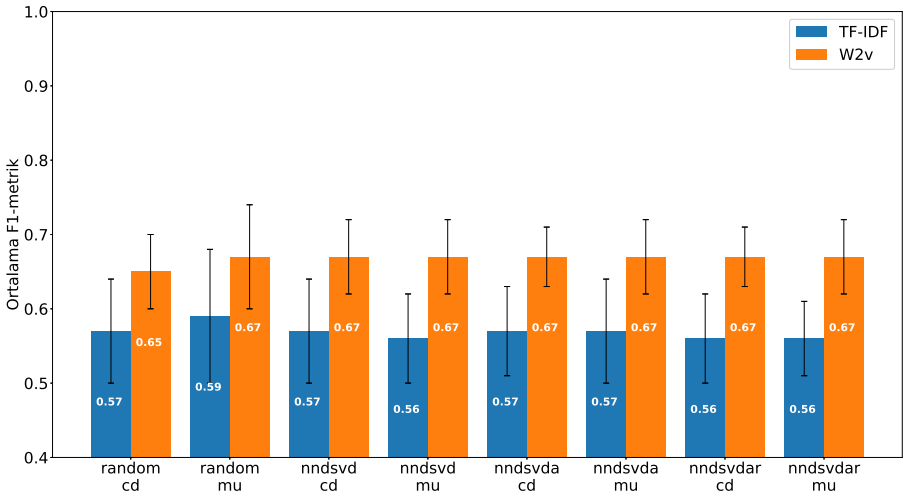
Şekil 4.39'da, NMF algoritmasının, farklı parametrelerle Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının, F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. NMF algoritması, farklı parametrelerle kullanıldığında, her modelde en başarılı sonucu, hem word2vec kelime temsili yöntemi ile hem de TF-IDF kelime temsili yöntemi ile benzer sonuçlar vermiştir. NMF algoritması bu veri setinde, başlangıç prosedürü olarak NNDSVD ile NNDSVDA ve çözücü olarak MU çözücüsü kullanıldığında 0,68 F1-metrik değeri ile en başarılı sonucu TFIDF kelime temsili yöntemi ile vermiştir. NMF algoritmasının tüm parametreleriyle TF-IDF kelime temsili yöntemi yeterli başarıyı gösterememiştir. Word2vec kelime temsili yöntemiyle, başlangıç prosedürü olarak NNDSVD ve çözücü olarak CD çözücüsü kullanıldığında, 0,67 F1-metrik değeri ile en yüksek sonucu göstermiştir. Word2vec kelime temsili yöntemi ile kullanılan modellerin standart sapmasının daha düşük olduğu görülmektedir.



Şekil 4.39. Konu-2 tweet mesajlarının bulunduğu veri seti kullanılarak NMF algoritmasının farklı parametrelerle elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

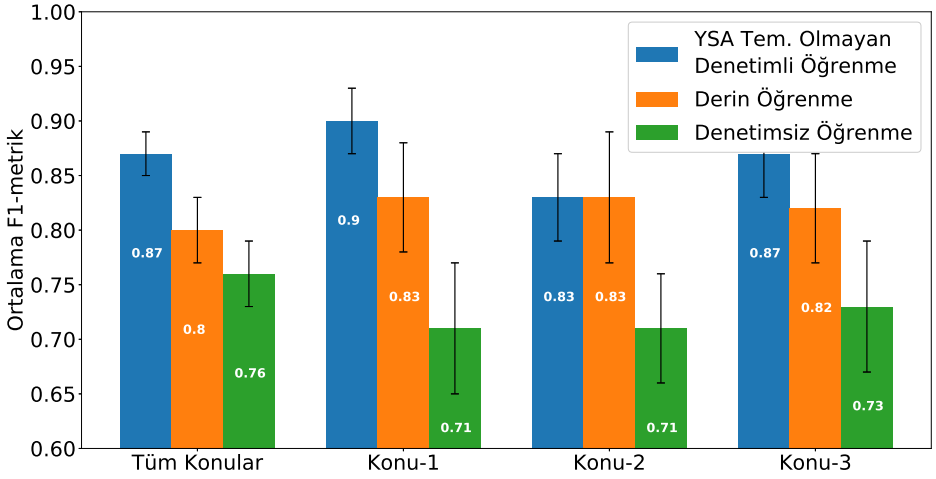
Şekil 4.40'da, NMF algoritmasının, farklı parametrelerle Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak elde edilen sonuçlarının, F1-metrik ortalamaları ve standart sapmaları hata çubukları grafiğinde gösterilmiştir. NMF algoritması, farklı parametrelerle kullanıldığında, her modelde en başarılı sonucu, word2vec kelime temsili yöntemi ile vermiştir. NMF algoritması bu veri setinde, başlangıç prosedürü olarak rastgele prosedürü kullanıldığı model hariç diğer modellerde 0,67 F1-metrik değeri ile en başarılı sonucu word2vec kelime temsili yöntemi ile vermiştir. NMF algoritmasının tüm parametreleriyle TF-IDF kelime temsili yöntemi yeterli başarıyı gösterememiştir. TF-IDF kelime temsili yöntemiyle, başlangıç prosedürü olarak rasgele prosedür ve çözücü olarak MU çözücüsü kullanıldığında, 0,59 F1-metrik değeri ile en yüksek sonucu gösterebilmiştir.

Tüm veri setleri için YSA temelli olmayan denetimli öğrenme algoritmaları, derin öğrenme algoritmaları ve denetimsiz öğrenme algoritmalarının en başarılı F1-metrik değerleri Şekil 4.41'de gösterilmiştir. Tüm veri setlerinde YSA temelli olmayan denetimli öğrenme algoritmaları diğer algoritmalardan daha başarılı olmuştur. Konu-2 tweet mesajlarının bulunduğu veri setinde ise derin öğrenme algoritmaları, YSA temelli olmayan denetimli öğrenme algoritmaları ile aynı F1-metrik ortalama değerini gösterse de standart sapma değeri daha yüksek olduğu görülmektedir. YSA temelli olmayan derin



Şekil 4.40. Konu-3 tweet mesajlarının bulunduğu veri seti kullanılarak NMF algoritmasının farklı parametrelerle elde edilen F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

öğrenme algoritmalarından SVM algoritması, Konu-1 veri seti hariç tüm veri setlerinde en yüksek F1-metrik değerini vermiştir. Konu-1 veri setinde ise RF algoritması en yüksek F1-metrik değerini vermiştir. Derin öğrenme algoritmalarından BiLSTM algoritması tüm veri setlerinde en başarılı algoritma olmuştur. Denetimsiz öğrenme algoritmalarının tümünde LDA algoritması en başarılı algoritma olmuştur.



Şekil 4.41. Tüm veri setlerinde, YSA temelli olmayan denetimli öğrenme algoritmaları, derin öğrenme algoritmaları ve denetimsiz öğrenme algoritmalarının F1-metrik ortalamalarını ve standart sapmalarını gösteren hata çubukları grafiği.

4.5. Sosyal Ağ Analizi

Twitter kullanıcılarının sahte ve gerçek haber yayan kullanıcılarla olan etkileşimini analiz etmek için sosyal ağ analizi yöntemlerinden Bölüm 3.8'de anlatılan Pagerank ve Bölüm 3.8'de anlatılan HKKA algoritmaları kullanılmıştır. Bu algoritmaların sonuçları Gephi uygulamasında hesaplatılmıştır. Sahte ve gerçek haber yayan kullanıcıların takipçileri Bölüm 3.1.3'de anlatıldığı gibi toplanmıştır.

Sahte ve gerçek haber yayan kullanıcılar ve bu kullanıcıları takip eden 7.996.407 adet kullanıcı toplanmıştır. 846 Kullanıcı sahte haber yaymış, 441 kullanıcı ise gerçek haber yaymıştır. Sahte ve gerçek haber yayan kullanıcıların takipçi grafı için 20.483.954 adet veri toplanmıştır. Sahte haber yayan kullanıcıların 5.034.826, gerçek haber yayan kullanıcıların ise 4.325.433 adet takipçisi bulunmuştur.

Toplanan kullanıcıların verileri, takipçi ilişkileri veri tabanında "FollowersGraph" tablosunda toplanmıştır. Veri tabanından sahte ve gerçek haber tweet mesajı gönderen kullanıcıların takipçileri için bir saklı yordam yazıldı (Şekil 4.42). Bu saklı yordam dışarıdan "@ara" ve "@maxFollowerSayisi" parametrelerini almaktadır. "@ara" parametresi hangi konu ile ilgili sahte ve gerçek tweet atan kullanıcıların getirileceğini belirler. "@maxFollowerSayisi" parametresi ise getirilen bir kullanıcın en fazla kaç adet takipçisi olacağını belirler. Belirlenen sayıdan daha az takipçisi olan kişilerin tüm takipçileri getirilir. Ayrıca bu saklı yordam, sahte ve gerçek haber yayan kullanıcıları eşit sayıda getirir.

Kenarları getiren SQL saklı yordamı ".csv" uzantılı dosya olarak kaydedilerek Gephi uygulamasına verilmiştir. Daha sonra kenarları alan uygulama, düğümleri de tespit eder. Tespit edilen bu düğümlerin, sahte haber yayan kullanıcı, gerçek haber yayan kullanıcı veya sadece takipçi olması durumunun belirlenmesi gerekmektedir. Bunun için gephi uygulamasından düğümler ".csv" uzantılı dosya olarak indirilmiştir. Yine bir SQL saklı yordamı yardımıyla, bu düğümlere, sahte haber yayanlara 's', gerçek haber yayanlara 'g' ve herhangi bir haber yaymamış takipçilere ise 'f' etiketi verildi (Şekil 4.43). Bu saklı yordam dışarıdan "@ara" ve "@userList" parametrelerini alır. "@ara" parametresi, kullanıcıların hangi konu ile ilgili tweet attıklarına bakılması gerektiğini

```
ALTER PROCEDURE [dbo].[GephiEdgeSP]
@ara nvarchar(max),
@maxFollowerSayisi int
AS
BEGIN

DECLARE @FakeTweetCount int;
DECLARE @GercekTweetCount int;
DECLARE @maxMesajTuruSayisi int;

SET @FakeTweetCount=(SELECT COUNT(*) FROM Tweet Where AranankKelime like '@ara' and MesajTuruID=1);
SET @GercekTweetCount=(SELECT COUNT(*) FROM Tweet Where AranankKelime like '@ara' and MesajTuruID=2);

IF (@FakeTweetCount<@GercekTweetCount)
BEGIN
   SET @maxMesajTuruSayisi=@FakeTweetCount
END
ELSE
BEGIN
   SET @maxMesajTuruSayisi=@GercekTweetCount
END

print '@maxMesajTuruSayisi = ' + Convert(varchar(50), @maxMesajTuruSayisi);
print '@FakeTweetCount = ' + Convert(varchar(50), @FakeTweetCount);
print '@GercekTweetCount = ' + Convert(varchar(50), @GercekTweetCount);

    Select FollowersGraphMainID,FollowersGraphFollowerID from (Select *,

row_number() over (partition by FollowersGraphMainID order by FollowersGraphMainID asc) as rnk

 from dbo.FollowersGraph


 where
   FollowersGraphMainID in

(Select rankedTweets.CreatedByID from
(Select CreatedByID,row_number() over (partition by MesajTuruID order by MesajTuruID asc) as rnk2
from Tweet Where AranankKelime like '%'+@ara+'%' and (MesajTuruID=1 or MesajTuruID=2)) as rankedTweets
where rnk2<=@maxMesajTuruSayisi)
```

Şekil 4.42. Kenarları getiren SQL saklı yordamı.

belirler. "@userList" parametresi ise gephi uygulamasından alınan düğümlerin virgülle ayrılmış listesini alır. Saklı yordam çalıştırıldığında artık düğümler ve etiketleri belirlenmiş olur. Daha sonra, bu düğümler ".csv" uzantılı dosya olarak kaydedilerek Gephi uygulamasına verilmiştir.

Her üç konu için de kenarlar ve düğümler belirlendikten sonra, Gephi uygulamasında her konunun Pagerank ve HKKA puanları hesaplatılmıştır.

4.5.1. HKKA algoritması

Gephi uygulamasında, HKKA algoritması çalıştırıldığında Bölüm 3.8'de anlatıldığı gibi merkezler ve yetkililer için puanlar hesaplanmıştır. Bu bölümde merkez puanları en yüksek olan kullanıcılarla, yetki puanları en yüksek olan kullanıcılar incelenmiştir.

```
ALTER PROCEDURE [dbo].[GephiNodeSP]
@ara nvarchar(max),
    @userList nvarchar(max)
AS
BEGIN
    Select rnks.Id as Idm, sahte,(Select Top(1) t2.MesajTuruID from Tweet as t2
        where t2.CreatedByID=rnks.Id and AranankKelime like '%'+@ara +'%'
            and (t2.MesajTuruID=1 or t2.MesajTuruID =2)) as aa from (
Select Id,
row_number() over (partition by Id order by Id asc) as rnk ,
CAST(

CASE


WHEN (Select Top(1) t2.MesajTuruID from Tweet as t2 where t2.CreatedByID=uu.Id
        and AranankKelime like '%'+@ara+'%' and (t2.MesajTuruID=1 or t2.MesajTuruID =2))=1 THEN 's'
WHEN (Select Top(1) t2.MesajTuruID from Tweet as t2 where t2.CreatedByID=uu.Id
        and AranankKelime like '%'+@ara+'%' and (t2.MesajTuruID=1 or t2.MesajTuruID =2))=2 THEN 'g'
WHEN (Select Top(1) t2.MesajTuruID from Tweet as t2 where t2.CreatedByID=uu.Id
        and AranankKelime like '%'+@ara+'%' and (t2.MesajTuruID=1 or t2.MesajTuruID =2)) is null THEN 'f'
WHEN (Select Count(*) from Tweet as t2 where t2.CreatedByID=uu.Id) =0 THEN 'f'
ELSE 'f' END AS nvarchar) as sahte

 from dbo.[User] as uu

where Id IN (SELECT value FROM STRING_SPLIT( REPLACE(REPLACE(@userList,CHAR(13),''), CHAR(10), ''), ',')) as rnks

 where rnk=1
END
```

Şekil 4.43. Düğümleri getiren SQL saklı yordamı.

4.5.1.1. Merkez puanına göre:

Merkez puanı yüksek olan kullanıcılar sosyal ağda bulunan yetki puanı yüksek olan birçok kişiyi takip eden kullanıcılardan oluşmaktadır. Sahte ve gerçek haber yayan kullanıcılar veri setinde bu algoritma çalıştırıldığında merkez puanı en yüksek olan kullanıcıların, sahte ve gerçek haber yaymamış fakat sahte ve gerçek haber yayan etkili kullanıcıları takip ettikleri görülmüştür. Bu nedenle merkez puanı; Pagerank puanı ve Yetki puanı gibi kesin bir noktada diğer kullanıcılardan ayrışmamaktadır. Bu nedenle merkez puanına göre sosyal ağ analizinde, merkez puanı en yüksek olan 50 kullanıcı dikkate alınmıştır.

Her üç konuya ait veri setinin merkez puanları SQL veri tabanında bir tabloda tutulmaktadır. Bu tabloda "HubScoreUserID" alanında kullanıcıların Twitter kullanıcı numaraları, "HubScoreUserMesajTuru" alanında sahte haber göndermiş bir kullanıcıysa "s", gerçek haber göndermiş bir kullanıcıysa "g" ve herhangi bir mesaj göndermemiş fakat takip etkileşiminde bulunmuşsa "f" harfleri, "HubScoreValue" alanında merkez puan değerleri, "HubScoreFakeFollowCount" alanında bu kullanıcının kaç adet sahte haber gönderen kullanıcıyı takip ettiği tutulmaktadır. "HubScoreNotFakeFollowCount" alanında bu kullanıcının gerçek haber gönderen kaç kullanıcıyı takip ettiği tutulmaktadır.

"HubScoreMoreFollow" alanında ise kullanıcı daha çok sahte haber yayan kullanıcıları takip ediyorsa "1", gerçek haber yayan kullanıcıları takip ediyorsa "2" rakamları tutulmaktadır. "HubScoreMoreFollow" alanı Denklem 4.1 ile hesaplanabilir. Şekil 4.44'de Konu-1 tweet mesajlarında merkez puanı en yüksek olan 50 kullanıcı gösterilmektedir.

	HubScoreUserID	HubScoreUserMesajTuru	HubScoreValue	HubScoreFakeFollowCount	HubScoreNotFakeFollowCount	HubScoreMoreFollow
1	429xxx415	f	0,079325944	55	26	1
2	312xxx986	f	0,07925332	37	23	1
3	068xxx785	f	0,07821585	68	21	1
4	082xxx974	f	0,07810716	50	23	1
5	963xxx960	f	0,073327176	60	29	1
6	505xxx211	f	0,072754815	51	19	1
7	480xxx703	f	0,071029335	72	27	1
8	396xxx270	f	0,07098424	40	23	1
9	073xxx765	f	0,06929474	48	20	1
10	034xxx401	f	0,068936616	41	17	1
11	486xxx113	f	0,06694473	52	18	1
12	947xxx367	f	0,06680945	46	17	1
13	787xxx970	f	0,06654013	47	20	1
14	374xxx138	f	0,06645524	45	24	1
15	116xxx359	f	0,06508066	44	22	1
16	952xxx215	f	0,0644264	30	15	1
17	815xxx626	f	0,06398547	28	17	1
18	002xxx563	f	0,063849606	45	21	1
19	557xxx296	f	0,062343493	43	16	1
20	410xxx117	f	0,060452893	38	18	1
21	776xxx115	s	0,059586797	72	17	1
22	253xxx356	f	0,059050206	28	15	1
23	342xxx487	f	0,0568588	32	18	1
24	387xxx621	f	0,055765197	41	20	1
25	936xxx825	f	0,05505454	30	12	1
26	024xxx207	f	0,051280275	41	11	1
27	711xxx980	f	0,04888302	39	16	1
28	898xxx823	f	0,048645128	40	20	1
29	279xxx248	f	0,048389893	23	14	1
30	976xxx827	f	0,047462124	28	16	1
31	844xxx954	f	0,04714334	32	15	1
32	418xxx192	f	0,046777118	31	11	1
33	716xxx163	f	0,046705343	28	12	1
34	646xxx146	f	0,045736548	42	13	1
35	851xxx264	f	0,04571237	27	4	1
36	817xxx128	f	0,04513783	30	13	1
37	920xxx110	f	0,04495467	40	7	1
38	453xxx896	f	0,04384657	35	13	1
39	546xxx172	f	0,043146078	33	14	1
40	266xxx570	f	0,041251764	24	9	1
41	060xxx234	f	0,04026518	38	13	1
42	159xxx331	f	0,039337765	41	6	1
43	952xxx859	f	0,039276645	35	6	1
44	940xxx171	f	0,03897717	31	18	1
45	017xxx329	f	0,038943913	46	9	1
46	221xxx462	f	0,038615275	13	12	1
47	546xxx820	f	0,03793067	16	11	1
48	343xxx130	f	0,037593156	24	13	1
49	076xxx193	f	0,037480734	21	7	1
50	840xxx248	f	0,037316956	24	11	1

Şekil 4.44. Konu-1 veri setinde merkez puanı en yüksek 50 kullanıcı.

$$[htb!] f_i^k = \begin{cases} s \geq g \text{ için}, & 1 \\ x < 0 \text{ için}, & 2 \end{cases} \tag{4.1}$$

Denklem 4.1'de; f, "HubScoreMoreFollow" alanının yeni değerini, i, i kullanıcısını, k, k konusunu, s, i kullanıcısının k konusundaki takip ettiği sahte haber yayan kullanıcı sayısını ve g, i kullanıcısının k konusundaki takip ettiği gerçek haber yayan kullanıcı sayısını temsil etmektedir.

Konu-1 tweet mesajlarında, merkez puanı en yüksek olan 50 kişinin 1 tanesi sahte haber yaymış, geri kalan 49 diğer kullanıcı ise bu konuda herhangi bir haber paylaşmamıştır. Bu 50 kullanıcının tümü, daha çok sahte haber yayan kullanıcıları takip etmişlerdir.

Konu-2 tweet mesajlarında, merkez puanı en yüksek olan 50 kişinin 4 tanesi gerçek haber yaymış, geri kalan 46 diğer kullanıcı ise bu konuda herhangi bir haber paylaşmamıştır. Bu 50 kullanıcının 43 tanesi daha çok sahte haber yayan kullanıcıları, 7 tanesi ise daha çok gerçek haber yayan kullanıcıları takip etmişlerdir.

Konu-3 tweet mesajlarında, merkez puanı en yüksek olan 50 kişinin 1 tanesi gerçek haber yaymış, geri kalan 49 diğer kullanıcı ise bu konuda herhangi bir haber paylaşmamıştır. Bu 50 kullanıcının 44 tanesi daha çok sahte haber yayan kullanıcıları, 6 tanesi ise daha çok gerçek haber yayan kullanıcıları takip etmişlerdir.

Merkez puanı dikkate alındığında; merkez puanı en yüksek olan 50 kişinin büyük bir çoğunluğunun sahte haber yayan kullanıcıları, gerçek haber yayan kullanıcılardan daha çok takip ettiği görülmektedir.

4.5.1.2. Yetki puanına göre:

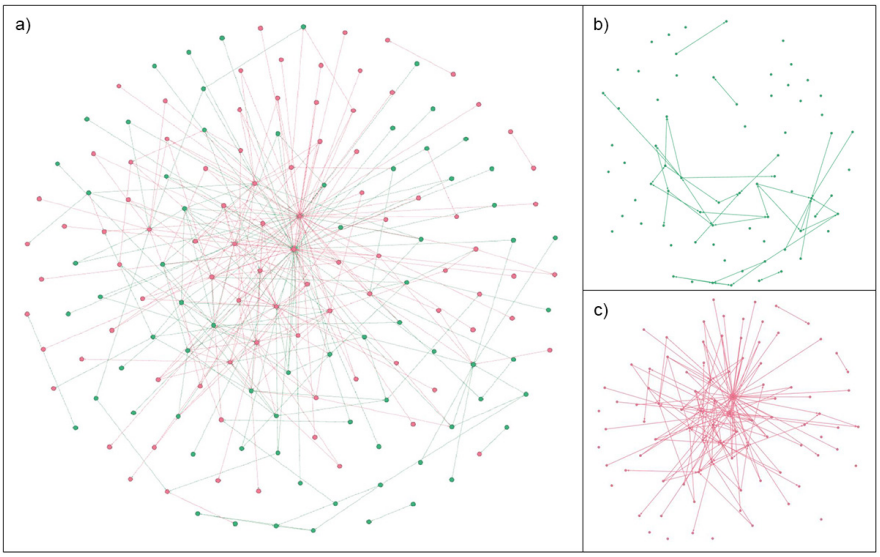
Yetki puanı yüksek olan kullanıcılar sosyal ağda bulunan merkez puanı yüksek olan kişiler tarafından takip edilen kullanıcılardır. Sahte ve gerçek haber yayan kullanıcılar veri setinde, bu algoritma çalıştırıldığında yetki puanı en yüksek olan kullanıcılar sahte ve gerçek haber yaymış kullanıcılardan oluşmaktadır.

Gephi uygulaması üzerinde link sıralama algoritmaları hesaplatıldıktan sonra yetki puanı en yüksek olandan en düşük olana doğru sıralama yapılmıştır. Sıralama sonucunda, her konu için 200 adet kullanıcıdan sonra yetki puanı sıfırlanmıştır. Bu kullanıcılar yetki puanı bakımından etkisiz kullanıcılar oldukları için dikkate alınmamıştır. Yetki puanı sıfır olmayan kullanıcılar incelenmiştir.

Konu-1 tweet mesajlarında sahte ve gerçek haber yayan kullanıcılar ve takipçilerinden, yetki puanı en yüksek olan 254 kişi alınmıştır. Bu 254 kişiden sonraki kişilerin puanının sıfır olduğu görülmüştür. Yetki puanı en yüksek olan bu kullanıcıların 127 tanesi sahte haber, 127 tanesi ise gerçek haber paylaşan kullanıcılardan oluşmaktadır. 127 tane sahte haber yayan kullanıcı; 302 tane sahte haber yayan hesabı; 52 tane ise gerçek haber yayan hesabı takip etmektedir. 127 tane gerçek haber yayan kullanıcı ise 177 tane sahte haber yayan kullanıcıları, 47 tane ise gerçek haber yayan kullanıcıları takip etmektedir. Yetki puanı yüksek olan 254 kişi, sahte haberde veya gerçek haber yaydığına bakılmaksızın, sahte haber yayan kullanıcıları daha çok takip etmişlerdir (Şekil 4.45).

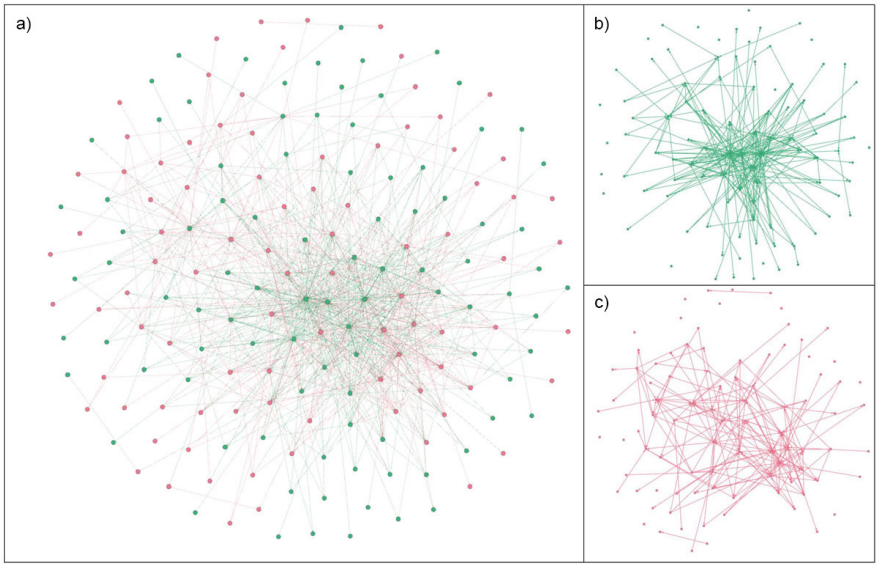
Şekil 4.45'de, kırmızı düğümlerin her biri Konu-1 ile ilgili sahte haber yaymış ve yetki puanı en yüksek kullanıcıları, yeşil düğümlerin her biri ise Konu-1 ile ilgili gerçek haber paylaşmış ve yetki puanı en yüksek olan kişileri temsil etmektedir. Şekilden de görüleceği üzere yetki puanı yüksek olan kişiler sahte haber yayan kişileri takip etmişlerdir.

Konu-2 tweet mesajlarında sahte ve gerçek haber yayan kullanıcılar ve takipçilerinden, Pagerank puanı en yüksek olan 247 kişi alınmıştır. Bu 247 kişiden sonraki kişilerin puanının sıfır olduğu görülmüştür. Pagerank puanı en yüksek olan bu 247 kullanıcının 120 tanesi sahte haber, 127 tanesi ise gerçek haber paylaşan kullanıcılardan oluşmaktadır. 120 tane sahte haber yayan kullanıcı; 304 tane sahte haber yayan hesabı; 319 tane ise gerçek haber yayan hesabı takip etmektedir. 127 tane gerçek haber yayan kullanıcı ise 423 tane sahte haber yayan kullanıcıları, 395 tane ise gerçek haber yayan kullanıcıları takip etmektedir. Konu-2 tweet mesajları dikkate alındığında, Pagerank puanı yüksek olan 247 kişinin, hemen hemen eşit sayıda sahte ve gerçek haber yayan kullanıcıları takip ettiği görülmüştür (Şekil 4.46).



Şekil 4.45. a) Yetki puanı yüksek olan Konu-1 kullanıcıları. b) Yetki puanı yüksek olan gerçek haber paylaşmış Konu-1 kullanıcıları. c) Yetki puanı yüksek olan sahte haber paylaşmış Konu-1 kullanıcıları.

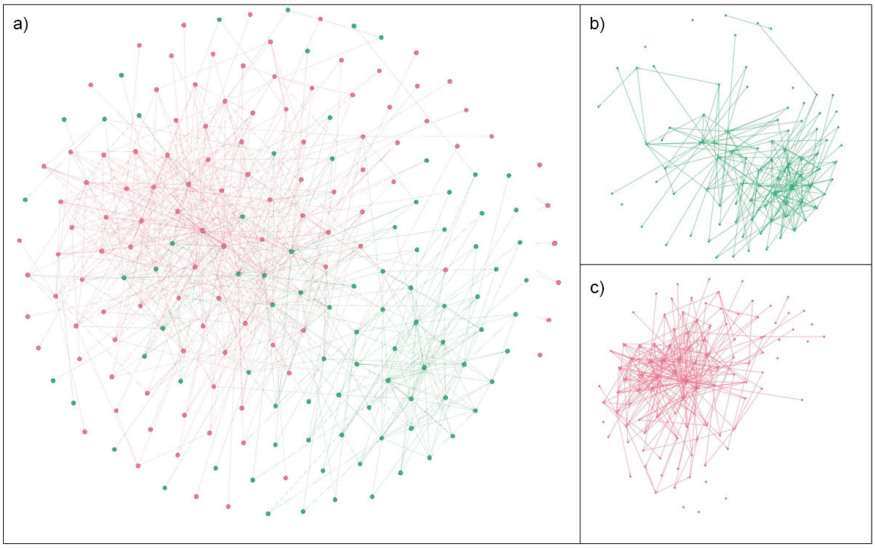
Şekil 4.46'da, kırmızı düğümlerin her biri Konu-2 ile ilgili sahte haber yaymış ve yetki puanı en yüksek kullanıcıları, yeşil düğümlerin her biri ise Konu-2 ile ilgili gerçek haber paylaşmış ve yetki puanı en yüksek olan kişileri temsil etmektedir. Şekilden de görüleceği üzere yetki puanı yüksek olan kişiler sahte ve gerçek haber yayan kişileri eşit oranda takip etmektedirler.



Şekil 4.46. a) Yetki puanı yüksek olan Konu-2 kullanıcıları. b) Yetki puanı yüksek olan gerçek haber paylaşmış Konu-2 kullanıcıları. c) Yetki puanı yüksek olan sahte haber paylaşmış Konu-2 kullanıcıları.

Konu-3 tweet mesajlarında sahte ve gerçek haber yayan kullanıcılar ve takipçilerinden, Pagerank puanı en yüksek olan 246 kişi alınmıştır. Bu 246 kişiden sonraki kişilerin puanının sıfır olduğu görülmüştür. Pagerank puanı en yüksek olan bu 246 kullanıcının 137 tanesi sahte haber, 109 tanesi ise gerçek haber paylaşan kullanıcılardan oluşmaktadır. 137 tane sahte haber yayan kullanıcı; 768 tane sahte haber yayan hesabı; 167 tane ise gerçek haber yayan hesabı takip etmektedir. 109 tane gerçek haber yayan kullanıcı ise 184 tane sahte haber yayan kullanıcıları, 315 tane ise gerçek haber yayan kullanıcıları takip etmektedir. Pagerank puanı yüksek olan 246 kişiden sahte haber yayan 137 kullanıcının, yine sahte haber yayan kullanıcıları; gerçek haber yayan 109 kullanıcı ise gerçek haber yayan kullanıcıları takip ettiği görülmüştür (Şekil 4.46).

Şekil 4.47'de, kırmızı düğümlerin her biri Konu-3 ile ilgili sahte haber yaymış ve yetki puanı en yüksek kullanıcıları, yeşil düğümlerin her biri ise Konu-3 ile ilgili gerçek haber paylaşmış ve yetki puanı en yüksek olan kişileri temsil etmektedir. Şekilden de görüleceği üzere yetki puanı yüksek olan kişilerden sahte haber yayan kişiler sahte haberleri, gerçek haber yayan kişiler de gerçek haber yayan kişileri takip etmektedirler.



Şekil 4.47. a) Yetki puanı yüksek olan Konu-3 kullanıcıları. b) Yetki puanı yüksek olan gerçek haber paylaşmış Konu-3 kullanıcıları. c) Yetki puanı yüksek olan sahte haber paylaşmış Konu-3 kullanıcıları.

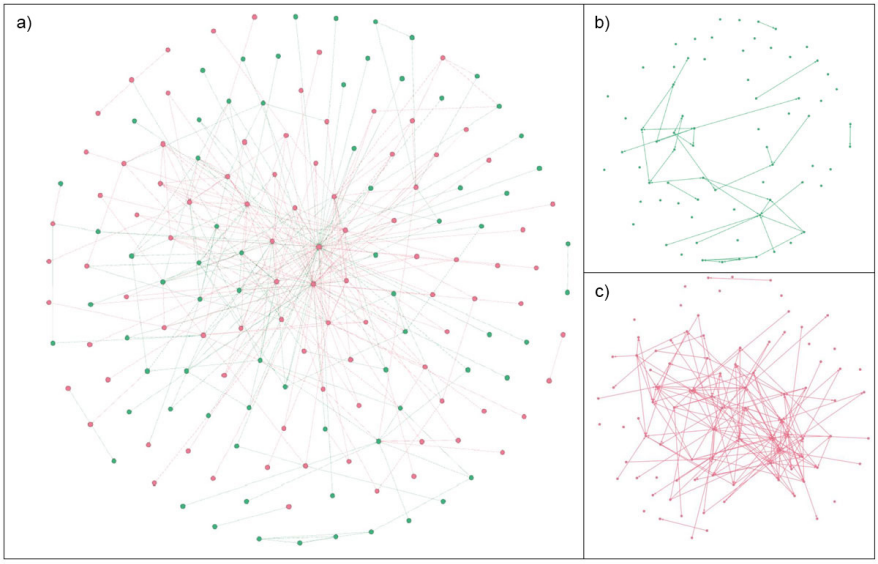
4.5.2. Pagerank algoritması

Gephi uygulaması üzerinde link sıralama algoritmaları hesaplatıldıktan sonra Pagerank puanı en yüksek olandan en düşük olana doğru sıralama yapılmıştır. Sıralama sonucunda, her konu için belirli bir kullanıcıdan sonra Pagerank puanı tekrar etmeye başlamaktadır. Pagerank puanının tekrar eden bölümüne kadar olan kullanıcılar alınıp incelenmiştir.

Konu-1 tweet mesajlarında sahte ve gerçek haber yayan kullanıcılar ve takipçilerinden, Pagerank puanı en yüksek olan 257 kişi alınmıştır. Bu 257 kişiden sonraki kişilerin puanı 0,00003 olarak tekrar ettiği görülmüştür. Pagerank puanı en yüksek olan bu 257 kullanıcının 130 tanesi sahte haber, 127 tanesi ise gerçek haber paylaşan kullanıcılardan oluşmaktadır. 130 tane sahte haber yayan kullanıcı; 302 tane sahte haber yayan hesabı; 52 tane ise gerçek haber yayan hesabı takip etmektedir. 127 tane gerçek haber yayan kullanıcı ise 177 tane sahte haber yayan kullanıcıları, 47 tane ise gerçek haber yayan kullanıcıları takip etmektedir. Pagerank puanı yüksek olan 257 kişi, sahte haberde veya gerçek haber yaydığına bakılmaksızın, sahte haber yayan kullanıcıları daha çok takip etmişlerdir.

Şekil 4.48'de, kırmızı düğümlerin her biri Konu-1 ile ilgili sahte haber yaymış ve yetki puanı en yüksek kullanıcıları, yeşil düğümlerin her biri ise Konu-1 ile ilgili gerçek haber paylaşmış ve yetki puanı en yüksek olan kişileri temsil etmektedir. Şekilden de görüleceği üzere, yetki puanında olduğu gibi pagerank puanında da puanı yüksek olan kişiler daha çok sahte haber yayan kişileri takip etmişlerdir.

Konu-2 tweet mesajlarında sahte ve gerçek haber yayan kullanıcılar ve takipçilerinden, Pagerank puanı en yüksek olan 258 kişi alınmıştır. Bu 258 kişiden sonraki kişilerin puanı 0,00004 olarak tekrar ettiği görülmüştür. Pagerank puanı en yüksek olan bu 258 kullanıcının 128 tanesi sahte haber, 130 tanesi ise gerçek haber paylaşan kullanıcılardan oluşmaktadır. 128 tane sahte haber yayan kullanıcı; 307 tane sahte haber yayan hesabı; 323 tane ise gerçek haber yayan hesabı takip etmektedir. 130 tane gerçek haber yayan kullanıcı ise 423 tane sahte haber yayan kullanıcıları, 395 tane ise gerçek haber yayan kullanıcıları takip etmektedir. Konu-2 tweet mesajları dikkate alındığında, Pagerank

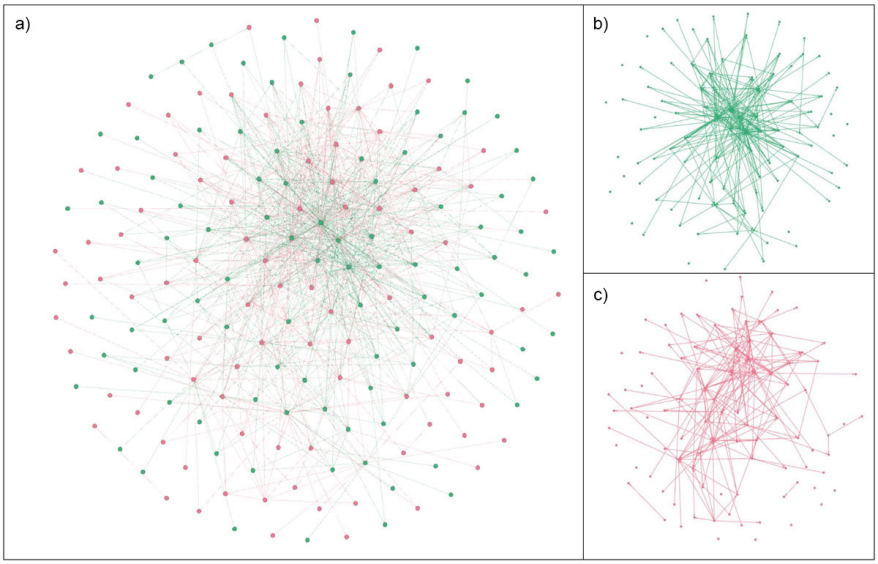


Şekil 4.48. a) Pagerank puanı yüksek olan Konu-1 kullanıcıları. b) Pagerank puanı yüksek olan gerçek haber paylaşmış Konu-1 kullanıcıları. c) Pagerank puanı yüksek olan sahte haber paylaşmış Konu-1 kullanıcıları.

puanı yüksek olan 258 kişinin, hemen hemen eşit sayıda sahte ve gerçek haber yayan kullanıcıları takip ettiği görülmüştür.

Şekil 4.49'da, kırmızı düğümlerin her biri Konu-2 ile ilgili sahte haber yaymış ve pagerank puanı en yüksek kullanıcıları, yeşil düğümlerin her biri ise Konu-2 ile ilgili gerçek haber paylaşmış ve pagerank puanı en yüksek olan kişileri temsil etmektedir. Şekilden de görüleceği üzere yetki puanı yüksek olan kişilerde olduğu gibi pagerank puanı yüksek olan kişilerde de sahte ve gerçek haber yayan kişileri eşit oranda takip etmektedirler.

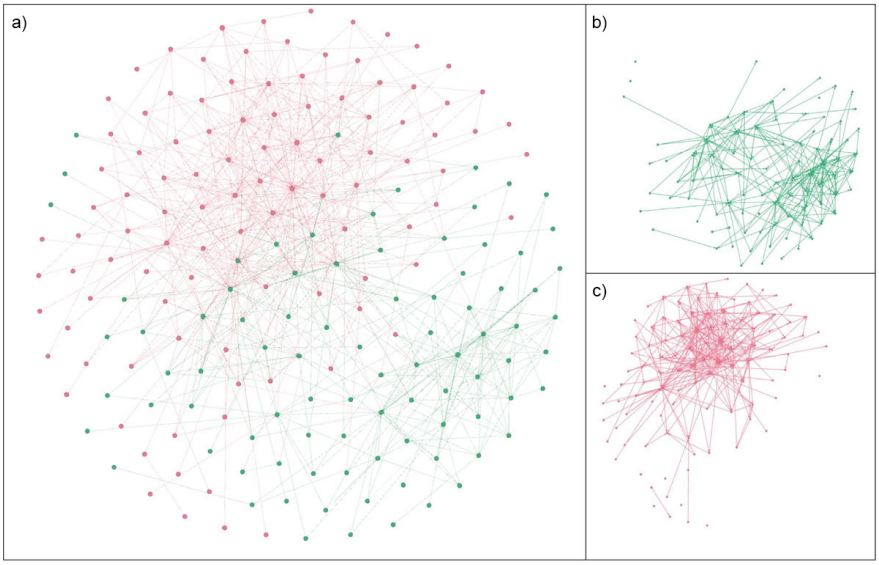
Konu-3 tweet mesajlarında sahte ve gerçek haber yayan kullanıcılar ve takipçilerinden, Pagerank puanı en yüksek olan 250 kişi alınmıştır. Bu 250 kişiden sonraki kişilerin puanı 0,00003 olarak tekrar ettiği görülmüştür. Pagerank puanı en yüksek olan bu 250 kullanıcının 138 tanesi sahte haber, 112 tanesi ise gerçek haber paylaşan kullanıcılardan oluşmaktadır. 138 tane sahte haber yayan kullanıcı; 768 tane sahte haber yayan hesabı; 167 tane ise gerçek haber yayan hesabı takip etmektedir. 112 tane gerçek haber yayan kullanıcı ise 184 tane sahte haber yayan kullanıcıları, 316 tane ise gerçek haber yayan kullanıcıları takip etmektedir. Pagerank puanı yüksek olan 250 kişiden sahte haber



Şekil 4.49. a) Pagerank puanı yüksek olan Konu-2 kullanıcıları. b) Pagerank puanı yüksek olan gerçek haber paylaşmış Konu-2 kullanıcıları. c) Pagerank puanı yüksek olan sahte haber paylaşmış Konu-2 kullanıcıları.

yayan 138 kullanıcının, yine sahte haber yayan kullanıcıları; gerçek haber yayan 112 kullanıcı ise gerçek haber yayan kullanıcıları takip ettiği görülmüştür.

Şekil 4.50'de, kırmızı düğümlerin her biri Konu-3 ile ilgili sahte haber yaymış ve pagerank puanı en yüksek kullanıcıları, yeşil düğümlerin her biri ise Konu-3 ile ilgili gerçek haber paylaşmış ve pagerank puanı en yüksek olan kişileri temsil etmektedir. Şekilden de görüleceği üzere yetki puanı yüksek olan kişilerde, sahte haber yayan kullanıcılar, diğer sahte haber yayan kullanıcıları, gerçek haber yayan kullanıcılar ise diğer gerçek haber yayan kullanıcıları takip etmektedirler.



Şekil 4.50. a) Pagerank puanı yüksek olan Konu-3 kullanıcıları. b) Pagerank puanı yüksek olan gerçek haber paylaşmış Konu-3 kullanıcıları. c) Pagerank puanı yüksek olan sahte haber paylaşmış Konu-3 kullanıcıları.

5. TARTIŞMA VE SONUÇ

Yapılan çalışmalara bakıldığında, Türkiye'de yaşayan insanların sahte habere maruz kalma oranının oldukça yüksek ve sahte haberi ayırt edenlerin oranının da düşük olduğu görülmektedir. Ayrıca sahte haberlerin ilk 2 saatlik dilimde çok fazla paylaşıldığı düşünüldüğünde, sahte haberlerin tespitinde otomatik tespit sistemlerinin kullanılması büyük bir öneme sahiptir.

Bu çalışmada, YSA temelli olmayan denetimli öğrenme, derin öğrenme ve denetimsiz öğrenme algoritmaları kullanılarak sahte haber tespiti yapılmıştır. YSA temelli olmayan denetimli öğrenme algoritmaları ve denetimsiz öğrenme algoritmaları TF-IDF ve word2vec kelime temsili yöntemleri ile denenmiştir. Ayrıca sahte ve gerçek haber kullanıcıların etkileşimini incelemek amacıyla sosyal ağ analizi yöntemleri kullanılmıştır. Tüm kullanıcıların takipçi ilişkileri toplanmıştır. Daha sonra HITS algoritması ve pagerank algoritması ile etkili kişiler bulunmuş ve görselleştirilmiştir.

5.1. Veri Seti

Kullanılan veri seti üç farklı konudan toplanan tweet mesajlarından oluşmaktadır. Üç konunun da birbirinden farklı özellikler gösterdiği görülmüştür.

Konu-1 tweet mesajları incelendiğinde sahte ve gerçek haberi ayıran sadece 1 kelime vardır. "Bayburt Havalimanı'nın 2 milyon yolcu **garantili** yaptırılması" sahte haberi, "Bayburt Havalimanı'nın 2 milyon yolcu **kapasiteli** yaptırılması" ise gerçek haberi temsil etmektedir. Kullanıcı grubu olarak incelendiğinde, sahte haber veya gerçek haber gönderenlerin Bayburt'a havalimanı yapılmasını eleştirdikleri ve aynı görüşte oldukları görülmektedir.

Konu-2 tweet mesajları incelendiğinde sahte ve gerçek haber tweet mesajları birden fazla olabilmektedir. Örneğin; "Falcao Galatasaray'a hayırlı olsun.", "Falcao Galatasaray'a geldi.", "Falcao transferi Kamuyu Aydınlatma Platformu'na (KAP) bildirildi" gibi haberler sahte haberi, "Falcao Galatasaray'a gelmedi.", "Falcao'yu Galatasaray'da görmek isteriz.", "Falcao Galatasaray'a gel." gibi mesajlar ise gerçek haberi temsil

etmektedir. Bu tweet mesajlarındaki kullanıcı grubunun ise tamamı Galatasaray Futbol Kulübünün taraftarlarıdır. Ayrıca, Radamel Falcao isimli futbolcunun transferi daha sonraki günlerde gerçekleştiği için sahte haber olmaktan çıkmıştır.

Konu-3 tweet mesajları incelendiğinde ise, sahte haberler, benzer mesajlardan oluşurken, gerçek haberler birden fazla mesajla ifade edilebilmektedir. "Kuleli Askeri Lisesi satıldı." tweet mesajı sahte haberi temsil ederken, "Kuleli Askeri Lisesi'nin satıldığını Kültür ve Turizm Bakanı Mehmet Nuri Ersoy yalanladı.", "Kuleli askeri lisesinin satıldığı yalan haberdir." gibi tweet mesajları ise gerçek haberleri temsil etmektedir. Buradaki kullanıcı grubunun ise tamamen birbirine zıt görüşlü kişilerden oluştuğu görülmüştür.

5.2. Makine Öğrenmesi Yöntemleri

YSA temelli olmayan denetimli öğrenme algoritmaları ve denetimsiz öğrenme algoritmalarının girdisi vektör olmak zorundadır. Word2vec kelime temsili yönteminde ise bir tweet mesajında geçen tüm kelimeler bir vektörle temsil edilmektedir. Bu algoritmalarda word2vec kelime temsili yönteminin kullanılması için tweet matrisinin vektör olarak ifade edilmesi gerekmiştir. Bu nedenle, tweet mesajlarının tek bir vektör ile temsil edilmesi için iki yöntem kullanılmıştır. Birinci yöntem, veri setindeki tweet mesajlarında geçen tüm kelimelerin, word2vec vektörlerinin ortalamasının alınması ve ikinci yöntem ise tüm vektörlerin alt alta eklenmesi ile vektör elde edilmesidir. İki yöntem incelendiğinde ortalama vektör yönteminin daha başarılı olduğu görülmüş ve YSA temelli olmayan denetimli öğrenme algoritmaları ve denetimsiz öğrenme algoritmalarında word2vec kelime temsili yöntemi kullanmak için ortalama vektör alınmıştır.

Tüm konulara ait tweet mesajlarından oluşan ve Konu-1, Konu-2 ve Konu-3 tweet mesajlarından oluşan veri seti, TF-IDF kelime temsili yöntemi, word2vec kelime temsili yöntemi ve word2vec yöntemini YSA temelli olmayan denetimli öğrenme algoritmaları ve denetimsiz öğrenme algoritmalarında kullanmak üzere word2vec ortalama vektörü ile vektörlere çevrilerek algoritmalara giriş olarak verilmiştir. Veri setlerinin %70'i eğitim seti olarak %30'u ise algoritmaların başarısını ölçmek için test seti olarak kullanılmıştır.

5.2.1. YSA Temelli Olmayan Denetimli Öğrenme Algoritmaları

Tüm konular hakkında tweet mesajlarından oluşan veri seti ile YSA temelli olmayan denetimli öğrenme algoritmaları ile eğitilip test edildiğinde, en başarılı F1-metrik değerini SVM algoritması, TF-IDF kelime temsili yöntemi kullanıldığında ve RBF ve 2. dereceden polinom algoritmaları ile vermiştir. İki farklı parametrede SVM algoritması, 0,87 F1-metrik değeri vermiştir. Fakat 2. dereceden polinom algoritması ile kullanılan SVM algoritmasının standart sapması 0,2 ile RBF algoritması ile kullanılan SVM algoritmasından daha iyi sonuç vermiştir. 0,87 F1-metrik değerini veren 2. dereceden polinom ile kullanılan SVM algoritmasının standart sapmasının 0,3 olduğu görülmüştür. Genel olarak TF-IDF kelime temsili yöntemi word2vec kelime temsili yöntemine göre daha başarılı olmuştur.

Konu-1 tweet mesajlarından oluşan veri seti ile YSA temelli olmayan denetimli öğrenme algoritmaları ile eğitilip test edildiğinde, en başarılı F1-metrik değerini RF algoritması, TF-IDF kelime temsili yöntemi kullanıldığında ve 500 ağaç sayısı ile gini ve entropy algoritmaları, 1000 ağaç sayısı ile ise entropy algoritması kullanılarak vermiştir. SVM algoritması da RF algoritmasına çok yakın sonuçlar vermiştir. Farklı parametrelerde RF algoritması, 0,90 F1-metrik değeri vermiştir. Fakat 1000 ağaç sayısı ve entropy bölme algoritması ile kullanılan RF algoritmasının standart sapması, 0,3 ile diğer parametrelerle kullanılan RF algoritmalarından daha başarılı olduğu görülmüştür. 0,90 F1-metrik değerini veren diğer parametrelerle çalıştırılan RF algoritmasının standart sapmasının 0,4 olduğu görülmüştür. Genel olarak bu veri setinde de TF-IDF kelime temsili yöntemi, word2vec kelime temsili yöntemine göre daha başarılı olmuştur.

Konu-2 tweet mesajlarından oluşan veri seti ile YSA temelli olmayan denetimli öğrenme algoritmaları ile eğitilip test edildiğinde, en başarılı F1-metrik değeri olan 0,83 değerini, SVM algoritması, TF-IDF kelime temsili yöntemi kullanıldığında ve RBF algoritması ile kullanıldığında vermiştir. 0,83 F1-metrik değerini veren RBF algoritması ile kullanılan SVM algoritmasının standart sapmasının 0,4 olduğu görülmüştür. Genel olarak TF-IDF kelime temsili yöntemi word2vec kelime temsili yöntemine göre daha başarılı olmuştur.

Konu-3 tweet mesajlarından oluşan veri seti ile YSA temelli olmayan denetimli öğrenme algoritmaları ile eğitilip test edildiğinde, en başarılı F1-metrik değeri olan 0,87 değerini, SVM algoritması, word2vec kelime temsili yöntemi kullanıldığında ve Linear algoritması ile kullanıldığında vermiştir. 0,87 F1-metrik değerini veren Linear algoritması ile kullanılan SVM algoritmasının standart sapmasının 0,4 olduğu görülmüştür. Diğer veri setlerinden farklı olarak Konu-3 tweet mesajlarından oluşan veri setinde word2vec kelime temsili yöntemi ile kullanılan modeller, TF-IDF kelime temsili yöntemleri ile kullanılan modellerden daha başarılı olsalar da çok büyük bir fark görülmemiştir.

YSA temelli olmayan denetimli öğrenme algoritmalarında, Konu-3 veri setinde word2vec kelime temsili yöntemi diğer veri setlerinde ise TF-IDF kelime temsili yöntemi daha başarılı olmuştur. Her iki kelime temsili yönteminde de F1-metrik değerleri arasındaki fark azdır. Konu-3 veri setinde ise veri setinin özelliğinden dolayı kelimelerin önceki ve sonraki kelimelerle olan ilişkilerini dikkate alan word2vec yöntemi TF-IDF yöntemine göre biraz daha başarılı olmuştur.

YSA temelli olmayan denetimli öğrenme algoritmalarından, SVM algoritması hem konu bazlı veri setleri ile hem de tüm konuları içeren veri setiyle olan eğitimlerde diğer algoritmalardan daha başarılı olmuş veya başarılı olan algoritmaya yakın sonuç vermiştir. Sahte ve gerçek haber metinlerindeki farklılıkları ayırt eden ve en iyi sonucu veren algoritma SVM algoritması olmuştur.

5.2.2. Denetimsiz Öğrenme Algoritmaları

Bu çalışmada; K-means, Non-Negative Matrix Factorization (NMF) ve Linear Discriminant Analysis (LDA) denetimsiz öğrenme algoritmaları kullanılmıştır. Denetimsiz öğrenme algoritmalarının F1-metrik değerlerine bakıldığında tüm veri setlerinde (Tüm konular, Konu-1, Konu-2 ve Konu-3 tweet mesajlarından oluşan) SVD yöntemi ile kullanılan LDA algoritmasının diğer algoritmalardan daha başarılı olduğu görülmüştür. Tüm konuların tweet mesajlarından oluşan veri setinde word2-vec kelime temsili yöntemi TF-IDF kelime temsili yönteminden daha başarılı olmuştur. Diğer veri setlerinde ise TF-IDF ve word2vec yöntemleri birbiriyle hemen hemen aynı sonuçları

vermişlerdir. LDA algoritması, tüm konuların tweet mesajlarından oluşan veri setinde 0,76 F1-metrik ve 0,03 standart sapma, Konu-1 tweet mesajlarından oluşan veri setinde 0,71 F1-metrik ve 0,06 standart sapma, Konu-2 tweet mesajlarından oluşan veri setinde 0,71 F1-metrik ve 0,05 standart sapma ve Konu-3 tweet mesajlarından oluşan veri setinde 0,73 F1-metrik ve 0,06 standart sapma değerini göstermiştir.

5.2.3. Derin Öğrenme Algoritmaları

Derin öğrenme algoritmalarından Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Long-Short Term Memory (LSTM), Bidirectional RNN (BiRNN), Bidirectional GRU (BiGRU) ve Bidirectional LSTM (BiLSTM) algoritmaları sahte haber tespiti problemimizde kullanılmıştır. Kelimeler arasındaki ilişkileri, yapılarındaki hafıza elemanları ile geçmiş kelimelerden de etkilenerek tespit ettiklerinden dolayı GRU, LSTM, BiGRU ve BiLSTM algoritmaları tüm konuların verildiği veri setinde başarılı olmuşlardır. RNN ve BiRNN ise daha basit yapıda oldukları için kelimeler arasındaki anlam ilişkisi kuramamışlardır.

Tüm konulardan oluşan veri seti ile algoritmalar eğitilip sonuçları incelendiğinde, en iyi sonucu veren, LSTM, GRU, BiLSTM ve BiGRU algoritmalarının sonuçları birbirlerine çok yakın çıkmıştır. RNN ve BiRNN algoritmaları ise yapısı gereği cümlede geçen önceki ve sonraki kelimeleri dikkate almadığı için başarısız sonuçlar vermiştir. Başarılı olan algoritmaların Kesinlik, Duyarlılık, F1-metrik ve doğruluk değerleri 0,79-0,80 arasında değişmektedir. Yani Kesinlik değeri dikkate alındığında; sahte olarak tahmin ettiği verilerin %80'ini doğru tahmin etmiş, Duyarlılık değeri dikkate alındığında, veri setindeki tüm sahte haberlerin %80'ini tespit edebilmiştir.

Derin öğrenme algoritmalarında, tüm konuların bulunduğu veri setinde çift yönlü olmayan LSTM algoritması başarılı olmuştur. Diğer veri setlerinde ise sınıflar arasındaki ayrımı daha iyi yapan çift yönlü algoritmalar diğer algoritmalara göre daha başarılı olmuştur. Konu bazlı veri setlerinde, çift yönlü algoritmalardan sonra ise saklama hafızasına sahip olan yapay sinir ağları algoritmaları başarılı olmuştur. RNN ve BiRNN algoritmaları ise bir hafızasının bulunmamasından dolayı sahte haber tespiti probleminde başarısız sonuçlar vermiştir.

Sonuç olarak etiketli veriler ile sahte haber tespitinde denetimli öğrenme algoritmalarının, denetimsiz öğrenme algoritmalarına göre daha başarılı olduğu görülmüştür. Bunun nedeni; Denetimsiz Öğrenme Algoritmalarının verileri kümelemek için bir etikete ihtiyaç duymamasından kaynaklanmaktadır. Denetimsiz öğrenme algoritmaları veriler arasındaki benzerliklere göre verileri kümelemektedir. Denetimli Öğrenme algoritmalarında ise modele etiketli veri verildiği için aynı etiketler arasındaki benzerlikleri çözmekte denetimsiz öğrenme algoritmalarına göre daha başarılı olmuşlardır.

YSA temelli olmayan denetimli öğrenme algoritmalarında, TF-IDF kelime temsili yönteminin kullanılması, word2vec kelime temsili yöntemlerinin kullanılmasına göre daha başarılı olmuştur. Bunun nedeni, önceden eğitilmiş (pre-trained) word2vec veri setinin, eğitim için kullanılan verilerinin düzenli makalelerden oluşuyor olmasıdır. Tweet mesajları ise resmi yazım kurallarından bağımsız kişilerin paylaştıkları mesajlardan oluşmaktadır. Örneğin; Konu-2 veri setinde; "gel be falcao", insallah gelir", Konu-3 veri setinde; "peşkeş çekilme" gibi sokak dilinde sıkça kullanılan kelime ve kelime grupları bulunmaktadır. Yeterince tweet mesajları verisiyle eğitilmiş bir ön-eğitimli (pre-trained) word2vec modeli bulunmamaktadır. Bu nedenle tweet mesajları veri seti ile eğitilmiş ön eğitimli modeller, TF-IDF kelime temsili yöntemine göre daha başarısız olmuştur. Ayrıca; YSA temelli olmayan denetimli ve denetimsiz öğrenme algoritmalarına veriler vektör olarak verilmektedir. Bu nedenle word2vec kelime temsili ile tweet mesajından bulunan tüm kelime vektörlerinin ortalamasının alınması yine kelime temsilinin başarısını etkilemiştir.

5.3. Sosyal Ağ Analizi

Sosyal ağ analizi için sahte ve gerçek haber yayan tweet kullanıcıları ve bu kullanıcıların takipçileri "Twitter Search API" kullanılarak toplanmıştır. Daha sonra sosyal ağ analizi yapmak için toplanan bu kişiler konu bazlı ayrılarak Gephi uygulamasına düğümler ve kenarlar olarak girilmiştir. Gephi uygulamasında sosyal ağ analizi yöntemlerinden; HKKA ve pagerank algoritmaları kullanılarak kullanıcıların, merkez, yetki ve pagerank puanları hesaplatılmıştır. Merkez ve pagerank puanları yüksek olan kullanıcıların aynı kullanıcılardan oluştuğu görülmüştür.

Merkez ve pagerank puanları yüksek çıkan Konu-1 kullanıcılarının büyük bir çoğunluğu Şekil 4.45'de de görüldüğü üzere sahte haber yayan kullanıcıları takip etmektedirler. Bunun nedeni Konu-1 hakkında tweet mesajı gönderen kullanıcıların büyük çoğunluğu benzer düşüncelere sahip kişilerden oluşmaktadır. Bayburt Havalimanı'nın 2 milyon yolcu garantili yaptırılması sahte haber, Bayburt Havalimanı'nın 2 milyon yolcu kapasiteli yaptırılması ise gerçek haber verisidir. Kullanıcılar "garantili" ve "kapasiteli" kelimelerinden hangisini kullandığına bakılmaksızın Bayburt'a havalimanı yapılmasını eleştiren tweet mesajları göndermişlerdir. Bu nedenle düşünce olarak benzer yapıya sahip kullanıcılar birbirlerini takip etmişlerdir. Diğer görüşe sahip kullanıcılar ise tweet mesajlarının daha küçük bir bölümünü oluşturdukları için gerçek haber kullanıcılarının daha az birbirleriyle takip etkileşiminde bulunduğu görülmüştür.

Merkez ve pagerank puanları yüksek çıkan Konu-2 kullanıcılarında ise sahte ve gerçek haber yayanların birbirleri ile takip etkileşiminin daha homojen olduğu Şekil 4.46 görülmektedir. Konu-2 de Radamel Falcao isimli futbolcunun Galatasaray Futbol Kulübüne transferinin yapıldığı iddiası sahte haber, Radamel Falcao transferinin gerçekleşmemesi ve Galatasaray futbol takımına transferin gerçekleşmesini isteyen taraftarların gönderileri gerçek haber olarak görülmektedir. Burada tüm mesaj atan kullanıcılar Galatasaray futbol takımının taraftarı oldukları için sahte haber yayanlarla gerçek haber yayanlar tamamen homojen bir şekilde takip ilişkisinde bulunmuşlardır.

Merkez ve pagerank puanları yüksek çıkan Konu-2 kullanıcılarında ise Şekil 4.47'de görüldüğü gibi sahte haberler çoğunlukla sahte haber yayanları, gerçek haber yayanlar ise genellikle gerçek haber yayan kullanıcıları takip etmektedirler. Bu konuda ise Kuleli Askerî Lisesi'nin satıldığı iddiası sahte, Kuleli Askerî Lisesi'nin satılmadığını belirten mesaj atan kullanıcıların mesajları ise gerçek mesaj olarak görülmektedir. Burada iki farklı görüşteki kullanıcılar keskin bir şekilde ayrılmaktadırlar. Bu nedenle sahte haber yayan kullanıcı grubu kendi arasında, gerçek haber yayan kullanıcı grubu ise kendi arasında takip ilişkisinde bulunmuşlardır.

Makine öğrenmesi algoritmalarının sonuçlarından sonra algoritmanın yanlış tahmin ettiği tweet mesajlarının gönderen kişilerin takipçi ilişkileri incelenmiştir. Algoritma tarafından yanlış etiketlenen verilerin bir kısmının bu şekilde doğru etiketlenebilmesi

sağlanmıştır. Örnek olarak, en yüksek F1-metrik değerini veren SVM algoritmasının sonuçları incelenmiştir. Makine öğrenmesi tarafından doğru olarak tahmin edilmiş birçok veriye bakıldığında, sadece takip etkileşimi ile doğru sonuçlar elde edilememiştir. Bu nedenle makine öğrenmesi sonucunda yanlış tahmin edilen veriler üzerinde takipçi ilişkisi incelenmiştir. Çizelge 5.1'de Konu-3 tweet mesajları ile kullanılan SVM algoritmasının yanlış tahmin ettiği verilere ait takipçi analizi bulunmaktadır. 9 tane yanlış tahmin edilen verinin 4 tanesi daha düzeltilmiştir.

Çizelge 5.1. Makine öğrenmesi algoritması tarafından yanlış tahmin edilen verilerin takipçi ilişkileri analizi.

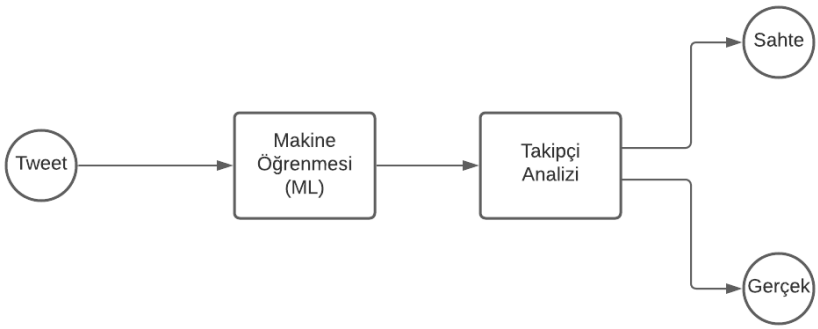
Kullanıcı	**Etiket**	**Sahte Takip**	**Gerçek Takip**	**Sahte Yüzde**	**Gerçek Yüzde**	**Tahmin**
169xxx538	Sahte	0	0	%0	%0	?
338xxx886	**Sahte**	**44**	**6**	**%88**	**%12**	**Sahte**
102xxx951	**Gerçek**	**0**	**4**	**%0**	**%100**	**Gerçek**
280xxx362	**Gerçek**	**0**	**2**	**%0**	**%100**	**Gerçek**
335xxx933	**Sahte**	**6**	**2**	**%75**	**%25**	**Sahte**
112xxx680	Gerçek	0	0	%0	%0	?
418xxx695	Sahte	0	0	%0	%0	?
111xxx074	Gerçek	15	4	%79	%21	Sahte
324xxx959	Gerçek	8	3	%73	%27	Sahte

Konu-3 tweet mesajları ile kullanılan SVM algoritmasının yanlış tahmin ettiği verilere ait takipçi analizi sonrasında oluşan yeni değerlendirme metrikleri Çizelge 5.2'de verilmiştir. Makine öğrenmesi yöntemleri ile elde edilen, 0,84 F1-metrik değeri, takipçi analizi sonrasında 0,91'e yükseltilmiştir. Sadece takipçi analizi ile sahte haber tespiti yapılmak istendiğinde ise, Çizelge 5.2'de görüldüğü üzere, 0,54 F1-metrik değeri elde edilebilmektedir. Önerilen yöntem Şekil 5.1'de gösterilmiştir.

Çizelge 5.2. Makine öğrenmesi, takipçi analizi ve hibrid modellerin değerlendirme ölçütleri.

	TP	**FP**	**TN**	**FN**	**Precision**	**Recall**	**F1-metrik**	**Accuracy**
ML	23	5	28	4	0,82	0,85	0,84	0,85
Takipçi	14	13	22	11	0,52	0,56	0,54	0,60
Hibrid	25	3	26	2	0,89	0,93	0,91	0,91

Literatürdeki çalışmalar incelendiğinde, İngilizce dilinde çok fazla çalışma olduğu görülmüştür. Türkçe dilinde ise detaylı bir çalışmaya rastlanmamıştır. Sosyal medya



Şekil 5.1. Önerilen yöntem.

üzerinde Türkçe dilinde yapılan ilk çalışma olduğu görülmüştür. Ayrıca veri seti bu çalışma için toplanmış özgün bir veri setidir.

Yapılan bu çalışmada sahte haberlerin tespiti otomatik bir sistem tarafından yaptırılmış ve çok kısa sürede sahte haberin tespit edilmesine olanak sağlamıştır. Bu sayede kısa sürede çok fazla paylaşım yapılmadan ve çok fazla kullanıcı sahte habere maruz kalmadan, sahte haberin tespiti ve önlenmesi sağlanabilir.

Sonraki çalışmalarda, sahte veya gerçek haber yayan kullanıcıların, takipçilerinin takipçileri de incelenerek farklı sonuçlar elde edilebilir. Ayrıca, makine öğrenmesi algoritmalarına takipçi analizi girdi olarak verilebilir. Düzenli metinlerle eğitilmiş ön eğitimli word2vec modelleri yerine, yeterince tweet mesajı ile eğitilmiş word2vec modeli kullanarak algoritma başarısı arttırılabilir. Ayrıca, Türkçe dilinde yeterince kelime sayısına sahip ön eğitimli bir model bulunmadığı için kullanamadığımız, eş sesli kelimeler için birden fazla farklı vektörler oluşturan ELMo (Peters vd., 2018), BERT (Devlin vd., 2019) ve GPT-3 (Brown vd., 2020) gibi yeni modeller kullanılarak algoritma başarısı arttırılabilir.

KAYNAKLAR

Ågren, A., Ågren, C., 2018. Combating Fake News with Stance Detection using Recurrent Neural Networks. Yüksek lisans tezi, University of Gothenburg.

Ahmed, H., 2017. Detecting Opinion Spam and Fake News Using N-gram Analysis and Semantic Similarity. Yüksek lisans tezi, University of Ahram Canadian.

Alpaydin, E., 2016. Machine Learning: The New AI. The MIT Press, Cambridge, MA.

Altunbey Özbay, F., 2020. Sürü Zekası Tabanlı Yöntemler Kullanılarak Çevrimiçi Sosyal Ağlarda Sahte Haber Tespiti. Doktora tezi, Fırat Üniversitesi.

Arthur, D., Vassilvitskii, S., 2006. k-means++: The Advantages of Careful Seeding. Teknik rapor, Stanford.

Bajaj, S., 2017. "The Pope Has a New Baby!" Fake News Detection Using Deep Learning. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2710385.pdf (Erişim tarihi: 17.08.2018).

Başakın, E.E., EKMEKCİOĞLU, Ö., Ozger, M., 2019. Drought Analysis with Machine Learning Methods. Pamukkale University Journal of Engineering Sciences, 25(8), 985–991.

Bhatt, G., Sharma, A., Sharma, S., Nagpal, A., Raman, B., Mittal, A., 2017. On the Benefit of Combining Neural, Statistical and External Features for Fake News Identification.

Bilgin, M., 2019. Kelime Vektörü Yöntemlerinin Model Oluşturma Sürelerinin Karşılaştırılması. Bilişim Teknolojileri Dergisi, 141–146.

Bird, S., Klein, E., Loper, E., 2009. Natural language processing with Python: analyzing text with the natural language toolkit. O'Reilly Media, Inc.

Breiman, L., 2001. Random Forests. Machine Learning, Springer, chapter 45. (ss. 5–32).

Brin, S., Page, L., 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks, 30, 107–117.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language Models are Few-Shot Learners.

Cao, H., Wachowicz, M., Renso, C., Carlini, E., 2017. An edge-fog-cloud platform for anticipatory learning process designed for Internet of Mobile Things.

Castillo, C., 2005. Effective web crawling. ACM SIGIR Forum, 39(1), 55–56.

Chen, Y.R., Chen, H.H., 2015. Opinion Spam Detection in Web Forum: A Real Case Study. Proceedings of the 24th International Conference on World Wide Web - WWW '15. ACM Press, New York, New York, USA, (ss. 173–183).

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Stroudsburg, PA, USA, (ss. 1724–1734).

Cunningham, P., Delany, S.J., 2020. k-Nearest Neighbour Classifiers – 2nd Edition.

Del Vicario, M., Bessi, A., Zollo, F., Petroni, F., Scala, A., Caldarelli, G., Stanley, H.E., Quattrociocchi, W., 2016. The spreading of misinformation online. Proceedings of the National Academy of Sciences, 113(3), 554–559.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North. Association for Computational Linguistics, Stroudsburg, PA, USA, (ss. 4171–4186).

Elsaadawy, A., Torki, M., Ei-Makky, N., 2018. A Text Classifier Using Weighted Average Word Embedding. 2018 International Japan-Africa Conference on Electronics, Communications and Computations (JAC-ECC). IEEE, (ss. 151–154).

Fang, Y., Gao, J., Huang, C., Peng, H., Wu, R., 2019. Self Multi-Head Attention-based Convolutional Neural Networks for fake news detection. PLOS ONE, 14(9), e0222713.

Fisher, R.A., 1936. The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics, 7(2), 179–188.

Galán-GarcÍa, P., Puerta, J.G.D.L., Gómez, C.L., Santos, I., Bringas, P.G., 2015. Supervised machine learning for the detection of troll profiles in twitter social network: application to a real case of cyberbullying. Logic Journal of IGPL, jzv048.

Gephi, 2020. Gephi-open source graph visualization software. https://gephi.org/features/ (Erişim tarihi: 10.10.2019).

Girgis, S., Amer, E., Gadallah, M., 2018. Deep Learning Algorithms for Detecting Fake News in Online Text. 2018 13th International Conference on Computer Engineering and Systems (ICCES). IEEE, (ss. 93–97).

Github, 2019. TweetScraper. https://github.com/jonbakerfish/TweetScraper (Erişim tarihi 22.03.2019).

Goodfellow, I., Bengio, Y., Courville, A., 2017. Deep learning. The MIT Press, Cambridge, MA.

Granik, M., Mesyura, V., 2017. Fake news detection using naive Bayes classifier. 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON). IEEE, (ss. 900–903).

Hao, S., Lee, D.H., Zhao, D., 2019. Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system. Transportation Research Part C: Emerging Technologies, 107, 287–300.

Hochreiter, S., Schmidhuber, J., 1997. Long Short-Term Memory. Neural Computation, 9(8), 1–32.

Ihlas Haber Ajansi, 2020. Eskişehir'de sahte hesaptan kar tatili mesajı atıldı. https://www.iha.com.tr/haber-eskisehirde-sahte-hesaptan-kar-tatili-mesaji-atildi-821170/ (Erişim tarihi: 06.01.2020).

Kadry, S., Al-Taie, M.Z., 2014. Başlık: Social Network Analysis : An Introduction with an Extensive Implementation to a Large-scale Online Network Using Pajek. eBook Collection (EBSCOhost).

Karcioglu, A.A., Aydin, T., 2019. Sentiment Analysis of Turkish and English Twitter Feeds Using Word2Vec Model. 2019 27th Signal Processing and Communications Applications Conference (SIU). IEEE, (ss. 1–4).

Kayes, I., Kourtellis, N., Quercia, D., Iamnitchi, A., Bonchi, F., 2015. The Social World of Content Abusers in Community Question Answering. Proceedings of the 24th International Conference on World Wide Web - WWW '15. ACM Press, New York, New York, USA, (ss. 570–580).

Kim, J., Tabibian, B., Oh, A., Schölkopf, B., Gomez-Rodriguez, M., 2018. Leveraging the Crowd to Detect and Reduce the Spread of Fake News and Misinformation. Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining - WSDM '18. ACM Press, New York, New York, USA, (ss. 324–332).

Kiritchenko, S., 2005. Hierarchical Text Categorization and Its Application to Bioinformatics. Dokotra tezi, University of Ottawa.

Kleinberg, J.M., 1999. Authoritative sources in a hyperlinked environment. Journal of the ACM (JACM), 46(5), 604–632.

Language Technology Group at the University of Oslo, 2018. NLPL word embeddings repository.

Lazer, D., Baum, M.A., Grinberg, N., Friedland, L., Joseph, K., Hobbs, W., Mattsson, C., 2017. Combating Fake News: An Agenda for Research and Action. Teknik rapor, Harvard University.

Le, Q.V., Mikolov, T., 2014. Distributed Representations of Sentences and Documents.

Lee, D.D., Seung, H.S., 1999. Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755), 788–791.

Levenshtein, V.I., 1965. , (Binary Codes Capable of Correcting Deletions, Insertions, and Reversals). , 163(4), 845–848.

Lilleberg, J., Zhu, Y., Zhang, Y., 2015. Support vector machines and Word2vec for text classification with semantic features. 2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC). IEEE, (ss. 136–140).

Mercioni, M.A., Holban, S., 2020. The Most Used Activation Functions: Classic Versus Current. 2020 International Conference on Development and Application Systems (DAS). IEEE, (ss. 141–145).

Mertoğlu, U., Genç, B., Sever, H., Sağlam, F., 2019. Auto-Tagging Model For Turkish News. International Ankara Conference on Scientific Researches. Ankara, (ss. 615–623).

Mertoğlu, U., Sever, H., Genç, B., 2018. Savunmada Yenilikçi Bir Dijital Dönüşüm Alanı. Savtek 2018, 9. Savunma Teknolojileri Kongresi. ODTÜ, Ankara, (ss. 771–778).

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient Estimation of Word Representations in Vector Space.

Monti, F., Frasca, F., Eynard, D., Mannion, D., Bronstein, M.M., 2019. Fake News Detection on Social Media using Geometric Deep Learning.

Newman, N., Fletcher, R., Kalogeropoulos, A., Nielsen, R., 2018. Digital News Report 2018. Teknik rapor.

Newman, N., Fletcher, R., Kalogeropoulos, A., Nielsen, R., 2019. Digital News Report 2019. Teknik rapor.

Olah, C., 2015a. Neural Networks, Types, and Functional Programming. https://colah.github.io/posts/2015-09-NN-Types-FP/ (Erişim tarihi: 13.05.2019).

Olah, C., 2015b. Understanding LSTM Networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (Erişim tarihi: 13.05.2019).

Ott, M., Cardie, C., Hancock, J.T., 2013. Negative Deceptive Opinion Spam. Proceedings of NAACL-HLT 2013. Atlanta, Georgia, (ss. 497–501).

Patel, M., 2017. Detection of Maliciously Authored News Articles. Yüksek lisans tezi, The Cooper Union For The Advancement of Science and Art.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É., 2012. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825—-2830.

Pérez-Rosas, V., Kleinberg, B., Lefevre, A., Mihalcea, R., 2017. Automatic Detection of Fake News. Proceedings of the 27th International Conference on Computational Linguistics. (ss. 3391–3401).

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep Contextualized Word Representations. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Association for Computational Linguistics, Stroudsburg, PA, USA, (ss. 2227–2237).

Pomerleau, D., Rao, D., 2015. Fake News Challange 1. http://www.fakenewschallenge.org (Erişim: 17.08.2018).

Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., Stein, B., 2018. A Stylometric Inquiry into Hyperpartisan and Fake News. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Stroudsburg, PA, USA, (ss. 231–240).

Pratiwi, I.Y.R., Asmara, R.A., Rahutomo, F., 2017. Study of hoax news detection using naïve bayes classifier in Indonesian language. 2017 11th International Conference on Information & Communication Technology and System (ICTS). IEEE, (ss. 73–78).

Qian, F., Gong, C., Sharma, K., Liu, Y., 2018. Neural User Response Generator: Fake News Detection with Collective User Intelligence. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence Organization, California, (ss. 3834–3840).

Rajendran, G., Chitturi, B., Poornachandran, P., 2018. Stance-In-Depth Deep Neural Approach to Stance Classification. Procedia Computer Science, 132, 1646–1653.

Rubin, V., Conroy, N., Chen, Y., Cornwell, S., 2016. Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News. Proceedings of the Second Workshop on Computational Approaches to Deception Detection. Association

for Computational Linguistics, Stroudsburg, PA, USA, (ss. 7–17).

Ruchansky, N., Seo, S., Liu, Y., 2017. CSI: A Hybrid Deep Model for Fake News Detection. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. ACM, New York, NY, USA, (ss. 797–806).

Sahin, G., 2017. Turkish document classification based on Word2Vec and SVM classifier. 2017 25th Signal Processing and Communications Applications Conference (SIU). IEEE, (ss. 1–4).

Sarle, W.S., 1994. Neural networks and statistical models. Proceedings of the Nineteenth Annual SAS Users Group International Conference. SAS Institute, Cary, (ss. 1538–1550).

Schuster, M., Paliwal, K., 1997. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11), 2673–2681.

Senol, A., Karacan, H., 2018. Akan Veri Kümeleme Teknikleri Üzerine Bir Derleme. European Journal of Science and Technology, 17–30.

Ser, G., Bati, C.T., 2019. Derin Sinir Ağları ile En İyi Modelin Belirlenmesi: Mantar Verileri Üzerine Keras Uygulaması. Yüzüncü Yıl Üniversitesi Tarım Bilimleri Dergisi, 406–417.

Shahnaz, F., Berry, M.W., Pauca, V., Plemmons, R.J., 2006. Document clustering using nonnegative matrix factorization. Information Processing & Management, 42(2), 373–386.

Shields, R., 2012. Cultural Topology: The Seven Bridges of Königsburg, 1736. Theory, Culture & Society, 29(4-5), 43–57.

Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H., 2017. Fake News Detection on Social Media. ACM SIGKDD Explorations Newsletter, 19(1), 22–36.

Singhania, S., Fernandez, N., Rao, S., 2017. 3HAN: A Deep Neural Network for Fake News Detection. 24th International Conference on Neural Information Processing (ICONIP 2017). (ss. 572–581).

Stonedahl, F., Wilkerson-Jerde, M., Wilensky, U., 2011. MAgICS: Toward a multi-agent introduction to computer science. Multi-Agent Systems for Education and Interactive Entertainment, IGI Global. (ss. 1–25).

Tacchini, E., Ballarin, G., Della Vedova, M.L., Moret, S., de Alfaro, L., 2017. Some Like it Hoax: Automated Fake News Detection in Social Networks. Proceedings of the Second Workshop on Data Science for Social Good. Skopje, Macedonia.

TensorFlow, 2019. The Sequential model- Keras. https://www.tensorflow.org/guide/keras/sequential_model (Erişim tarihi: 13.05.2019).

Teyit.org, 2016. teyit.org. https://teyit.org/ (Erişim tarihi: 08.01.2018).

Tschiatschek, S., Singla, A., Gomez Rodriguez, M., Merchant, A., Krause, A., 2018. Fake News Detection in Social Networks via Crowd Signals. Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18. ACM Press, New York, New York, USA, (ss. 517–524).

Twitter, 2006. Twitter Inc. https://twitter.com (Erişim tarihi: 01.03.2018).

Twitter, 2020a. KAMUOYUNA DUYURU İletişim Başkanlığı, vatandaşlardan hiçbir şekilde kredi kartı bilgilerini talep etmez. Kurumumuzun adı ve logosu ile yayılan "elektrik ve doğal gaz fatura iadesi" bildirimi, dolandırıcıların milletimizin devletimize olan güvenini [Tweet]. https://twitter.com/iletisim/status/1213530046733979649 (Erişim tarihi: 04.01.2020).

Twitter, 2020b. Yoğun kar yağışı,buzlanma ve soğuk nedeniyle, 07 Ocak 2020 Salı günü, il merkezi dışında kalan resmi ve özel tüm okul ve kurumlarımızda (okul öncesi, ilkokul, ortaokul, lise ve yaygın eğitim kurumları) eğitim öğretime bir gün ara verilmiştir. [Tweet].

Twitter Search API, 2018. Twitter Search API. https://developer.twitter.com/en/docs/basics/getting-started (Erişim tarihi: 10.06.2018).

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer.

Vo, K., Nguyen, T., Pham, D., Nguyen, M., Truong, M., Mai, T., Quan, T.T., 2019. Combination of domain knowledge and deep learning for sentiment analysis of short and informal messages on social media. International Journal of Computational Vision and Robotics, 9(5), 458.

Volkova, S., Shaffer, K., Jang, J.Y., Hodas, N., 2017. Separating Facts from Fiction: Linguistic Models to Classify Suspicious and Trusted News Posts on Twitter. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, Stroudsburg, PA, USA, (ss. 647–653).

Vosoughi, S., 2015. Automatic Detection and Verification of Rumors on Twitter. Massachusetts Institute of Technology, (2008), 147.

Vosoughi, S., Roy, D., Aral, S., 2018. The spread of true and false news online. Science, 359(6380), 1146–1151.

Wang, W.Y., 2017. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, Stroudsburg, PA, USA, (ss. 422–426).

Wikipedia, 2006. F1 score. https://en.wikipedia.org/wiki/F1_score (Erişim tarihi: 24.08.2018).

Xu, W., Liu, X., Gong, Y., 2003. Document clustering based on non-negative matrix factorization. Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval - SIGIR '03. ACM Press, New York, New York, USA, (s. 267).

Yang, S., Shu, K., Wang, S., Gu, R., Wu, F., Liu, H., 2019. Unsupervised Fake News Detection on Social Media: A Generative Approach. Proceedings of the AAAI Conference on Artificial Intelligence, 33, 5644–5651.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E., 2016. Hierarchical Attention Networks for Document Classification. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Stroudsburg, PA, USA, (ss. 1480–1489).

Zhao, W.X., Jiang, J., Weng, J., He, J., Lim, E.P., Yan, H., Li, X., 2011. Comparing Twitter and Traditional Media Using Topic Models. (ss. 338–349).

ÖZGEÇMİŞ

Adı Soyadı	: Süleyman Gökhan TAŞKIN
Doğum Yeri ve Yılı	: Konya, 1985
Medeni Hali	: Evli
Yabancı Dili	: İngilizce
E-posta	: taskinster@gmail.com

Eğitim Durumu

Lise	: Bursa Osmangazi Gazi Anadolu Lisesi, 2004
Önlisans	: SDÜ, Uluborlu S. Karasoy M.Y.O., Bilgisayar Programcılığı
Lisans	: Uludağ Ü., Eğitim Fakültesi, Bilg. Ve Öğret. Tek. Öğretmenliği
Lisans	: SDÜ, Mühendislik Fakültesi, Bilgisayar Mühendisliği
Yüksek Lisans	: SDÜ, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği A.B.D.

Mesleki Deneyim

AIT Bilgisayar Sist. San. Tic. Ltd. Şti.	: 2011-2012
SDÜ, Senirkent M.Y.O.	: 2014-2015
SDÜ, Bilgi İşlem Daire Bşk.	: 2014-2015
Bayburt Üniv., Bilgi İşlem Daire Bşk.	: 2015-2016
Bandırma Onyedi Eylül Üniv., Bilgi İşlem D.B.	: 2016-. . . . . (halen)

Yayınlar

Taskin, S.G., Kesgin, K., Arucu, M. (2019). LoraWan Teknolojisinin Kullanımı:Endüstri 4.0 Örneği. 10th International Non-Governmental Organizations Congress, 1(1), 42.

Taskin, S. G., Kucuksille, E. U. (2018). Recovering Data Using MFT Records in NTFS File System. Academic Perspective Procedia, 1(1), 448-457.

Taskin, S.G., Kesgin, K. (2018). Akıllı Ulasım Sistemlerinde LoRaWAN Teknolojisinin Kullanımı. 1. Uluslararası Akıllı Ulaşım Sistemleri Konferansı, 1(1), 297-298.

Kesgin, K., Taskin, S.G. (2018). Akıllı Ulastırma Sistemlerinde Kullanılan Sistem Odalarının ve Bilgilerin Yedekliligi. 1. Uluslararası Akıllı Ulaşım Sistemleri Konferansı, 1(1), 153-154.