T.C.
MANİSA CELAL BAYAR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ
YAZILIM MÜHENDİSLİĞİ ANABİLİM DALI
YAZILIM MÜHENDİSLİĞİ BİLİM DALI

ÇİFT YÖNLÜ ENKODER TRANSFORMATÖR TABANLI TÜRKÇE METİN SINIFLANDIRMA DERİN ÖĞRENME MODELİ GELİŞTİRİLMESİ

Kamil AKARSU

Danışman
Doç. Dr. Akın ÖZÇİFT

MANİSA–2022

KAMİL AKARSU	**ÇİFT YÖNLÜ ENKODER TRANSFORMATÖR TABANLI TÜRKÇE METİN SINIFLANDIRMA DERİN ÖĞRENME MODELİ GELİŞTİRİLMESİ**	**2022**

TEZ ONAYI

Kamil Akarsu tarafından hazırlanan **"Çift Yönlü Enkoder Transformatör Tabanlı Türkçe Metin Sınıflandırma Derin Öğrenme Modeli Geliştirilmesi"** adlı tez çalışması 21/06/2022 tarihinde aşağıdaki jüri üyeleri önünde Manisa Celal Bayar Üniversitesi Fen Bilimleri Enstitüsü **Yazılım Mühendisliği Anabilim Dalı**'nda **YÜKSEK LİSANS** olarak savunulmuş ve **oybirliği** ile başarılı olarak kabul edilmiştir.

Danışman	Doç. Dr. Akın ÖZÇİFT Manisa Celal Bayar Üniversitesi	.....
Jüri Üyesi	Dr. Öğr. Üyesi. Fatih YÜCALAR Manisa Celal Bayar Üniversitesi	.....
Jüri Üyesi	Dr. Öğr. Üyesi Okan ÖZTÜRKMENOĞLU İzmir Bakırçay Üniversitesi	.....

TAAHHÜTNAME

Bu tezin Manisa Celal Bayar Üniversitesi Hasan Ferdi Turgutlu Teknoloji Fakültesi Yazılım Mühendisliği Bölümü'nde, akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını beyan ederim.

Kamil AKARSU

İÇİNDEKİLER

SİMGELER VE KISALTMALAR DİZİNİ

ACC	Başarım
AL	Dikkat Katmanı
CNN	Evrişimsel Sinir Ağı
DDA	Doğal Dil Anlama
DDİ	Doğal Dil İşleme
DİM	Dil İşleme Modeli
DL	Derin Öğrenme
DT	Karar Ağaçları Algoritması
EMOT	Duygu Tanıma Veri Seti
FFNW	İleri Beslemeli Sinir Ağı
GLUE	Nesilsel Dil Anlama Değerlendirmesi
KNN	K En Yakın Komşu Algoritması
ML	Makine Öğrenmesi
MLM	Maskeli Dil Modeli
NB	Naif Bayes Algoritması
NEWS	Haber Metni Veri Seti
NN	Sinir Ağları
NSP	Sonraki Cümle Tahmini
SMS	Kısa Mesaj Servisi
SPAM	Gereksiz Kısa Mesaj Veri Seti
SquAD	Stanford Soru Yanıtlama Veri Kümesi
YSA	Yapay Sinir Ağları

ŞEKİLLER DİZİNİ

TABLO DİZİNİ

TEŞEKKÜR

Çalışmamın her aşamasında bana destek olan, bilgi ve deneyimleri ile yol gösteren danışman hocam Sayın Doç. Dr. Akın ÖZÇİFT'e teşekkür ederim.

Tüm çalışmalarım ve eğitimim boyunca bana değerli bilgi ve tecrübeleriyle yol gösteren Sayın Doç. Dr. Deniz KILINÇ, Sayın Dr. Öğretim Üyesi Fatih YÜCALAR ve Sayın Dr. Öğretim Üyesi Emin BORANDAĞ hocalarıma teşekkürü bir borç bilirim.

Ayrıca eğitim öğretim hayatım boyunca bana maddi manevi destek olan canım aileme ve sevgili eşime yürekten teşekkür ederim.

Kamil AKARSU
Manisa, 2022

ÖZET

Yüksek Lisans Tezi

Kamil AKARSU

Manisa Celal Bayar Üniversitesi
Fen Bilimleri Enstitüsü
Yazılım Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Akın ÖZÇİFT

Yazının icadından günümüz dünyasına kadar geçen sürede katlamalı olarak artan sayıda yazılı bilgi kaynakları mevcuttur. Çok büyük miktarda metin verisi beraberinde kaçınılmaz sorunları da getirmiştir. Elde edilen yazılı verilerin incelenmesi, etiketlenmesi, konularının tahmin edilmesi gibi problemler Doğal Dil İşleme, Makine Öğrenmesi ve son yıllarda adını sıkça duyduğumuz Derin Öğrenme teknikleri ile çözüme kavuşturulmaktadır.

Yapısal olarak birbirinden farklı diller üzerinde çıkarım ve inceleme yapmak ise çok daha zordur. Bu sorunun bilincinde olarak her dil için aynı teknikleri ve işlemleri kullanmak yerine dillerin kökenine ve morfolojik yapısına göre çözüm geliştirmek daha akılcı olacaktır.

Bu tez çalışmasında yapısal olarak zengin bir dil olan Türkçe üzerinde BERT modeli başarımının geleneksel Makine Öğrenmesi algoritmalarına kıyasla daha fazla olduğunun gösterilmesi hedeflenmektedir. Morfolojik olarak zengin diller üzerinde geleneksel Makine Öğrenmesi algoritmalarını uygulamak için çok yoğun ön işleme adımları gerekmektedir. Fakat BERT modeli kullanılırken bu yoğun ön işleme adımlarına gerek kalmamaktadır. Yarı eğitimli bir model olan BERT etiketlenmemiş ham veri üzerinde daha performanslı çalışmaktadır. Bu çalışmada literatürden Metin Sınıflandırma, Duygu Analizi, Spam Tespiti olmak üzere üç farklı Türkçe veri seti üzerinde deneyler yapılmıştır. Yapılan deneyler sonucunda BERT modelinin geleneksel Makine Öğrenmesi algoritmalarının birçoğundan daha iyi performansla çalıştığı kanıtlanmıştır. Deneylerden elde edilen çıktılar sonuç ve öneriler bölümünde paylaşılmıştır.

Anahtar Kelimeler: Çift Yönlü Kodlayıcı Temsilleri Transformatörler; Doğal Dil İşleme; Türkçe Metin Sınıflandırma

2022, 67 Sayfa

ABSTRACT

M.Sc. Thesis

Kamil AKARSU

Celal Bayar University
Graduate School of Applied and Natural Sciences
Department of Software Engineering

Supervisor: Assoc. Prof. Dr. Akın ÖZÇİFT

From the invention of writing to today's world, an exponentially increasing number of written information sources are available. The huge amount of text data brought with it inevitable problems. It is possible with Natural Language Processing, Machine Learning and Deep Learning techniques, which we have heard frequently in recent years, for problems such as examining, labeling, estimating the subjects of the written data obtained.

It is much more difficult to make inferences and analyzes on languages that are structurally different from each other. Being aware of this problem, instead of using the same techniques and operations for each language, it would be more rational to develop solutions according to the origin and morphological structure of the languages.

In this thesis, it is aimed to show that the performance of the BERT model on Turkish, which is a structurally rich language, is higher than traditional Machine Learning algorithms. Extensive preprocessing steps are required to implement traditional Machine Learning algorithms on morphologically rich languages. However, these intensive preprocessing steps are not required when using the BERT model. BERT, which is a semi-trained model, performs better on unlabeled raw material. In this study, experiments were conducted on three different Turkish datasets from the literature: Text Classification, Sentiment Analysis, Spam Detection. As a result of the experiments, it has been proven that the BERT model works better than most of the traditional Machine Learning algorithms. The results and findings obtained from the experiments are shared in the conclusion and recommendations section.

Keywords: Bidirectional Encoder Representations Transformers; Natural Language Processing; Turkish Text Classification

2022, 67 pages

1. GİRİŞ

İnsanlık, yazının icadından itibaren çok çeşitli yazılı eserler vermiştir. Bu eserler kimi zaman mağara duvarlarındaki resimler, kimi zaman kil tabletler üzerindeki çivi yazıları, kimi zaman hayvan derileri üzerine mürekkep dokunuşları, kimi zaman da günümüzde olduğu gibi kitap, dergi, roman veya bilgisayar çıktıları olmuştur. Bu eserlerin çeşitliliğine eser sahibi kişi ya da toplumların sosyokültürel yapıları, dini inanışları ve yaşayışları sebep olmuştur. Dil insanların birbirleriyle anlaşmasındaki en önemli ve etkili araçtır. Amerikalı ünlü Dilbilimci Noam Chomsky dili "Sözcük ve cümle birimleri aracılığıyla, düşünceyi konuşmayla ilişkilendiren çok seviyeli bir sistemdir" diyerek tanımlamıştır [1].

Teknoloji sürekli değişmekte ve gelişmektedir. Bununla birlikte yazılı eser sayıları her geçen gün büyük bir ivmeyle artmakta hatta son yirmi yıldaki elektronik ortamda bulunan yazılı eser sayısı yazının icadından itibaren var olan yazılı eser sayısını geçmektedir. Giderek artan ve çeşitlenen eserler elbette ki tek bir dil ve alfabe üzerinden üretilemez. Geçmişten günümüze kadar yaşamış tüm toplumların inanışları, yaşayışları, yaptıkları barış ve savaşlarla bile dilin çeşitlenmesini sağlamışlardır.

Hiç şüphe yoktur ki internetin keşfi; ücretsiz olarak bilginin ve dilin çok daha hızlıca bir şekilde yayılmasını sağlamıştır. Bununla birlikte refah seviyesinin artması, bilgisayar fiyatlarının ucuzlaması, bilgisayar kullanımının dünyada yaygınlaşması, elektronik yazıların-çıktıların artması beraberinde bazı sorunları getirmiştir. Bu sorunların en büyüklerinden bazıları ise şu şekildedir: Güvenli bilgiye erişim, aranılan bilgiye en kolay ve en hızlı yoldan erişim sorunlarıdır. Günümüzde kullanılan elektronik sistemlerde verilerin istatistiksel olarak incelenmesi, modellemelerinin yapılması mümkündür. Bu modeller üzerinde verinin tipine göre çalışmalar yapılabilmektedir. Bu veri tipleri; metin, ses, resim vb. olabilmektedir. Günümüz sürekli gelişen teknolojisinde veri modellemeleri, veriden anlam çıkarımları ve veriyi belli kurallar çerçevesinde sınıflandırma, kümeleme, istatistiksel inceleme işlemleri yapmak oldukça kolay ve az maliyetlidir.

Bu tezin amacı, morfolojik açıdan zengin olan Türkçe dili üzerinde Çift Yönlü Enkoder Transformatör Tabanlı Metin Sınıflandırma Derin Öğrenme

modelinin kurulması ve geleneksel olarak kullanılan Metin Sınıflandırma Algoritmalarına karşı başarımının test edilmesidir.

Bu tez çalışmasında Doğal Dil, Makine Öğrenmesi, Derin Öğrenme, Transformatörler, Doğal Dil İşleme, Derin Öğrenmeyle Doğal Dil İşleme Modelleri detaylıca anlatılmış, kullanılan veri setleri ile yapılan deneyler ve deneylerin sonuçları yazılmıştır.

2. LİTERATÜR TARAMASI

Çoban Ö. ve arkadaşları tarafından yapılan bu çalışmada Türkçe Twitter metinleri üzerinde Duygu Analizi üzerine odaklanılmıştır. Çalışma kapsamında 14777 Türkçe Twitter metni barındıran veri seti oluşturulmuş ve veri seti olumlu-olumsuz olarak iki kategoriye bölünmüştür. Destek Vektör Makineleri (SVM), Naif Bayes (Naive Bayes), Çok Terimli Naif Bayes (Multinomimal Naive Bayes) ve K En Yakın Komşu (KNN) vb. Makine Öğrenmesi algoritmaları üzerinde deneysel sonuçlar yapılmıştır. Vektör uzayı ile temsil edilen öznitelikler, N-Gram ve Bag of Words olmak üzere iki birbirinden farklı modelden elde edilmiştir. Deneysel sonuçlar, sınıflandırma yöntemlerinin etkisi üzerine araştırılmıştır. En yüksek başarım skorunu %66.06 ile Çok Terimli Naif Bayes sınıflandırıcısı vermiştir [2].

Shervin M. ve araştırma ekibi tarafından yapılan bu çalışmada çok kapsamlı veri setleri üzerinde Naive Bayes, BoW+SVM, tfIdf, Deep Pyramid CNN, BLSTM-2DCNN, Neural Semantic Encoder, GLUE ELMo baseline, BERT-large, ALBERT, RoBERTa gibi oldukça popüler Derin Öğrenme modelleri ile deneyler yapılmıştır. 2014-2020 yılları arasındaki üretilen yazılı metinlerdeki Metin Sınıflandırma problemini ele alan bu çalışmada 40 tane veri seti 150'den fazla Derin Öğrenme modeli ile test edilmiş olup, modellerin sonuçlarına göre performans karşılaştırılmaları incelenmiştir [3].

Demircan M. ve arkadaşları tarafından yayınlanan makale çalışmasında Doğal Dil İşleme ve Makine Öğrenmesi yöntemlerini kullanılarak internet medyasındaki metinler aracılığıyla ifade edilen duyguları belirlemeyi amaçlamaktadır. İlk araştırmalar sonucunda, metinlerin ve duyguların en iyi örtüştüğü durumun e-ticaret sitelerinde kullanılan ürün incelemeleri ve puanlar olduğu belirlenmiştir. Bir e-ticaret sitesinden alınan inceleme puanları ile farklı ürünlere ilişkin incelemeler, Makine Öğrenmesi tabanlı Duygu Analizi modellerinde kullanılmak üzere bir tabloya dönüştürülmüştür. İnceleme puanları kullanılarak yorumlar olumsuz, olumlu ve tarafsız olmak üzere üç gruba ayrılmıştır. SVM, RF, DT, RF ve KNN algoritmaları kullanarak Makine Öğrenmesi algoritmaları arasında performans karşılaştırmaları yapılmıştır [4].

Nergiz G. ve ekibi makalelerinde FastText modeli kullanılarak Tekrarlayan Yapay Sinir Ağlarının farklılaşmış bir türü LSTM (Kısa Uzun Vadeli Bellek) ile haber metinlerinin kategorizeleştirmesini yapmıştır. Farklı kategorilerde Türkçe haberlerin bulunduğu veri seti oluşturan araştırma ekibi veri seti üzerinde FastText, Word2Vec ve Doc2Vec tabanlı modellerin oluşturulmasını sağlamışlardır. Performans skorları incelenen deneyler sonucunda en iyi sonucu veren LSTM modelidir [5].

3. DOĞAL DİL

Dil, doğadaki canlılar arasındaki iletişimi sağlayan, sözlü, yazılı veya harekete dayalı sembollerden oluşan en temel iletişim aracıdır. Dünya üzerindeki farklı kültüre sahip ulusların sözlü veya yazılı haberleşmesinden, balina ve diğer memelilerin seslendirmelerine, bal arılarının sallanma dansı yapmalarına kadar birçok çeşit dil vardır.

Doğal Dil, bilinçli bir planlama veya amaç olmaksızın insanlar arasında kullanım ve tekrarlama yoluyla gelişen dildir. Bilgisayar kodları, yapay olarak türetilmiş olanlar, mantık çalışmak için kullanılan diller Doğal Dillerden farklıdır. İnsanlar arasındaki iletişimi herhangi bir kasıt veya amaç olmadan kademeli olarak geliştirmek için söz dizilimindeki dalgalanmalar ve kelime dağarcığı çeşitliliği kullanılmıştır [6].

3.1. Türkçe Tarihi ve Morfolojik Yapısı

Türk dili, Asya ve Avrupa kıtalarında büyük oranda kullanılan, Ural Altay dil familyasına mensup sonradan eklemeli yapıya sahip dildir. Türki lisan ailesinin Oğuz kolundan olan Garp Oğuz lisanı olan Osmanlı Türkçesinin süregelmişliğidir. Türkçe başta ülkemizde konuşulmakla birlikte Osmanlı İmparatorluğu'nun egemenlik sahalarında sıkça konuşulur. Etnolojiye göre Türkçe 95 milyon kullanıcıya sahip dünyada en fazla konuşulan 12. dildir. Türkçe, Türkiye Cumhuriyeti, Kıbrıs Türk Cumhuriyeti ve Kuzey Kıbrıs'ta ulusal resmi dil olarak kabul edilmekte ve kullanılmaktadır [7].

Türkçe, yapısı bakımından sondan eklemeli bir dil olup içerdiği büyük ve küçük ünlü uyumları gibi kuralları ile çeşitli özellikler barındırmaktadır. Cümleler yapı bakımından bir özne, nesne ve bir yüklemden oluşmaktadır. Başka ailelere ait dillerin aksine morfolojik olarak cinsiyet ayrımının bulunmadığı Türk Dili'nde içerik bakımından sözcüklerin bir kısmı Farsça'dan ve büyük çoğunlukları Arapça ve Fransızca'dan dillerinden geçmiştir [8]. Ayrıca Türkiye Türkçesi; Azerbaycan Türkçesi, Gagavuzca ve Türkmenistan Türkçesi gibi diğer dil kardeşleriyle oldukça fazla miktarda ortak kelime barındırmakta ve anlamsal olarak anlaşılabilirlilik artmaktadır.

Türkçe'de alfabe olarak 1928 yılında Mustafa Kemal Atatürk öncülüğünde yapılan Harf İnkılabı'ndan sonra Latin Alfabesi kullanılmaktadır. Türkçede ki yazım kuralları Türk Dil Kurumu'nca denetlenmektedir. Türkçe yazı dili, yöresel farklılıklar sebebiyle çeşitli ağızlar olmasına karşın İstanbul Türkçesi olarak da bilinen İstanbul ağzı temel kabul eder.

Türkçe, eski Orhun Kitabelerinin yazılı en eski kaynak olarak düşünüldüğünde 1500 senelik bir geçmişe sahiptir. Eskiden kullanılan Türkçe'nin de mensubu olarak kabul edilen Garp grubu ile Altay ailesine ait bütün Türki lisanların ortak vasisi olan bir Öncü Türkçe dilinin de var olduğu varsayılmaktadır [2].

Türkiye, Azeri ve Türkmenistan Türkçelerinin zaman içerisinde Ogur diye adlandırılan kadim bir lisandan değişerek meydana geldiği varsayılır. Bahsedilen öncü dilin Türkiye Türkçesini meydana getirecek batı kolu, 12. ve 16. asırlar boyunca Selçuklu Devleti ve Anadolu Beylikleri tarafınca çeşitli sosyo-kültürel olay çevresinde evrilerek Eski Anadolu Türkçesi'ni meydana getirmiştir. Halk arasında kullanılan dil ise yerini 14. asırdan 19. asıra kadar sürecek Osmanlı Türkçesi'ne bırakır. Yeni Osmanlı Türkçesi, 18. ve 19. asırlar arasında evrilmiştir. Türkiye Cumhuriyeti zamanında yapılan inkılaplar neticesindeyse 20. asırla birlikte Türk dilinde çeşitli gelişmeler yaşanmıştır [3].

Türkçe kullanılarak yazılan en eski eser Orhun Kitabeleri'dir. Bu eser düşünülerek Türkçe miladı tarihinin 1300 sene öncesine dayandırılmış fakat eserlerde yer alan alfabenin ileri seviyesi bu dilin daha önceki asırlarda kullanılmış olma ihtimalini artırmıştır. Günümüz Moğolistan ülkesi Orhun nehri civarında yer alan Kül Tigin ve Bilge Kağan eserlerinden farklı olarak döneminin bilinen devlet adamı Tonyukuk Hanın da kendi için yaptırdığı Ulan Batur şehri civarında bulunan iki taş, Orhun Kitabeleri'nin en bilindik örnekleridir.

3.2. Türkçe Yazı Sistemleri

Türkler için tarihte bilinen en eski alfabe Orhun alfabesi milattan sonra 7. asırın sonlarına kadar etkinliğini devam ettiren fakat zaman içerisinde çeşitli nedenlerden dolayı yerini civardaki yerel alfabelerden etkilenmiş Uygur alfabesine 9. asır civarında bırakmıştır. Asırlar boyunca yaşnan sosyo kültürel ilişkiler, savaşlar ve

göçler neticesinde yer değiştiren Türk halkları Yunan, İbrani ve Latin alfabelerini kullanmışlardır. Türklerin islamiyet ile tanışmasından sonra Arap asıllı alfabe sistemi etkin alfabe olmuştur. Bu Arap asıllı Türk alfabesi, 1 Kasım 1928' de ilan edilen Latin alfabe temelli Türk alfabesine geçilinceye kadar, lisanın konuşanları tarafından yazı sistemini kullanmışlardır [4]. Ayrıca, Türkçe eserleri sadece Türkiye veya Türki ırklara sahip devletler değil dilin konuşulduğu devletlerde hayat süren farklı etknik kökenlere ait azınlıklar da kendi alfabelerini kullanarak vermişlerdir [5].

3.2.1. Orhun ve Eski Uygur Alfabeleri

Türkçenin tarihte ilk olduğuna inanılan alfabesi, Orhun Kitabelerinde de kullanılmış olan, daha sık kullanılan olarak ise Göktürk alfabesi olarak tanınan Orhun alfabesidir. 7. asırın sonlarına doğru olduğu düşünülen alfabe sisteminin Çin alfabe sistemlerinden esinlenilerek meydana geldiği varsayılmaktadır. Yazım olarak sağdan sola kullanılan yazı sistemi, Göktürk ve Uygur Devletleri tarafından ana dil olarak kullanılmış fakat Uygur Devleti döneminde ortadan kalkmıştır [10]. Şekil 3.1. de Kül Tigin Yazıtından örnek bir cümle gösterilmiştir.

Şekil 3.1. Kül Tigin Yazıtından Örnek Bir Cümle

3.2.2. Arap Asıllı Türk Alfabesi

Milattan sonra 8. asırdan 19. asrın başlarına kadar, neredeyse 900 yıl boyunca Türkçe, Arap yazı sistemine ait uyularak yazılmıştır. İslamın yaygınlaşmasıyla birlikte Türk Devletleri, geçen asırlar boyunca bu yazı sistemini kullanmaya başlamış ve 12. asır civarlarında bu alfabe Türk Devletleri içinde kullanılan eş alfabe olma niteliği kazanmıştır [14].

Arapça kökenli bir yazı sistemini benimseyen Osmanlı Devleti'nde alfabenin içinde çeşitli geliştirme ve değişiklikler yapılmış, Osmanlı alfabesi diye adlandırılan bir yazı sistemi kullanılmıştır. Ayrıca bu alfabe, Osmanlı Türkçesi'ne birebir olarak geçmiş Fars ve Arap dillerinden alınma kelimeleri yazmak amacıyla çok iyi olmakla birlikte, Osmanlıca'daki Türkçe kökenli kelimeleri yazarken fazla zorluk doğurmaktaydı. Arap Dili morfolojik olarak ünsüz sesler açısından zengin, ünlü

seslerin varlığı açısından ise zayıf bir dilken, Türkçe bunun tam aksi özellikler sergilemektedir. Böylece Arap asıllı Türk alfabesi Türkçe içinde olan çok fazla ses uyumuna müsait durum gösterememektedir. Türkçe içinde var olan herhangi bir ünlü ses için Arap alfabesinde 3 değişik ses varken, bazı harfler için ise harfi vurgulayacak uygun ses yoktur. Teknolojinin gelişmesiyle birlikte matbaanın 19. asırda Osmanlı Devleti'nde kullanılmasıyla Arap asıllı Türk alfabesinin çoğu Türk seslerini vurgulayamaması çok büyük bir sorun olmuştur [7].

Şekil 3.2. Osmanlı Türkçesi ile Yazılmış Siyer-i Nebi Eseri

3.2.3. Latin Asıllı Türk Alfabesi

Mustafa Kemal Atatürk tarafından 1 Kasım 1928'de gerçekleştirilen Harf İnkılabı ile birlikte eskiden Osmanlı Devleti'nde kullanılan Arap asıllı Türk alfabesinin yerini Latin yazı sisteminden Türkiye Türkçesine göre uyarlanan 29 harfli alfabe getirilmiştir. Harf İnkılabı Türkiye sınırları içerisinde bu devrimin kabul görmesi ve yeni yazı sisteminin kullanılması sürecine verilen addır [8]. Türkçe yazı sistemindeki harflerin yazılış ve okunuşları çoğu Batı yazı sistemlerinde olan harflerin okunuşlarından da değişiktir. Örnek olarak 'a' harfinin okunuşu Türkçede 'a' İngilizcede 'ey'dir.

Büyük harfler																												
A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z
Küçük harfler																												
a	b	c	ç	d	e	f	g	ğ	h	ı	i	j	k	l	m	n	o	ö	p	r	s	ş	t	u	ü	v	y	z

Şekil 3.3. Latin Alfabesi

4. MAKİNE ÖĞRENMESİ

4.1. Makine Öğrenmesi Tanım ve Temel Kavramları

Makine Öğrenmesi, insanların bir bilgiyi öğrenme şeklinden esinlenerek veri ve çeşitli yöntemlerin uygulanmasına dayanan ve kurulan modelin başarısını aşamalı şekilde yükselten Yapay Zekâ (*Artificial Intelligence*) ve bilgisayar bilimleri branşıdır. Makine Öğrenmesindeki temel hedef, veri girişleriyle eğitimi tamamlanan modelin kurulması ve bu model üzerinde çeşitli analizler yapıp tahminlerde bulunmaktır. Makine Öğrenmesi bir konu hakkındaki spesifik bir problemin o probleme ait veri seti ile modellenmesi yapan algoritmalar bütünüdür [17].

Makine Öğrenmesi tarihçesine bakacak olursak Makine Öğrenmesi kavramı hiç de yeni bir teknoloji değildir. Bir kavram olarak Makine Öğrenmesi oldukça uzun bir süredir var. Makine Öğrenmesi terimi, IBM'de bir bilgisayar bilimcisi ve bilgisayar dünyasında çığır açan Arthur Samuel tarafınca ortaya atılmıştır [9]. Samuel bilgisayarla karşılıklı dama oynamak için bir program tasarladı. Program ne kadar çok oynarsa, tahminler yapmak için algoritmalar kullanarak deneyimlerinden o kadar çok şey öğrendi. Bir disiplin olarak Makine Öğrenmesi, verilerden öğrenebilen ve veriler üzerinde tahminler yapabilen algoritmaların analizini ve oluşturulmasını araştırır. Makine Öğrenmesi, sorunları yalnızca insan zihni tarafından kopyalanamayacak bir hızda ve ölçekte çözebildiği için değerli olduğunu kanıtlamıştır. Tek bir görevin veya birden fazla belirli görevin ardındaki muazzam miktarda hesaplama yeteneği ile makineler, girdi verileri arasındaki kalıpları ve ilişkileri belirlemek ve rutin işlemleri otomatikleştirmek için eğitilebilir [10].

Makine Öğrenmesini yönlendiren algoritmalar başarı için kritik öneme sahiptir. Makine Öğrenmesi algoritmaları, doğrudan program yazmadan tahminlerde bulunmak veya karar vermek için eğitimde kullanılan veriler diye adlandırılan çeşitli örnekler temelli bir matematiksel model var eder. Bu, bilgi işletmelerinin karar verme sürecini iyileştirmek, verimliliği optimize etmek ve uygun ölçekte eyleme geçirilebilir verileri yakalamak için kullanabileceği verilerdeki eğilimleri ortaya çıkarabilir. Makine Öğrenmesi, süreçleri otomatikleştiren ve veri tabanlı iş sorunlarını özerk bir şekilde çözen Yapay Zekâ sistemleri için temel sağlar. Şirketlerin belirli insan yeteneklerini değiştirmesini veya artırmasını sağlar. Gerçek

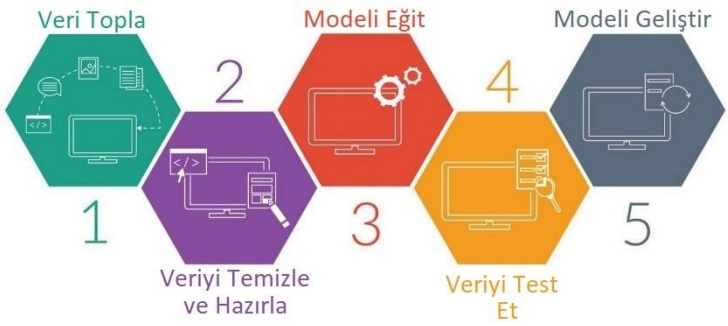
dünyada bulabileceğiniz yaygın makine öğrenimi uygulamaları arasında sohbet robotları, sürücüsüz arabalar ve konuşma tanıma yer alır. Ayrıca Veri Güvenliği, Sağlık, Finans ve Muhasebe, Dolandırıcılık tespiti gibi sayısı artırılabilir çok çeşitli sektörlerde kullanılmaktadır [11].

Makine Öğrenmesinin tarihçesi ve işlevinden bahsettikten sonra Makine Öğrenmesinin daha kolay anlaşılabilmesi için gerekli temel kavramlar şu şekildedir [21].

Tablo 4.1. Makine Öğrenmesi Terminolojisi ve Açıklamaları

Öznitelik (*Feature*)	Öznitelik, bir olgunun bireysel ölçülebilir özelliği veya özelliğidir [12].
Veri (Data)	Yorumlanmayan ve analiz edilmeyen işlenmemiş herhangi bir gerçek, değer, metin, ses veya resim olabilir.
Eğitim Verisi (Train Data)	Makine Öğrenmesi modelinin eğitimi için kullanılan ön işleme adımlarından geçmiş veridir.
Test Verisi (Test Data)	Eğitimi tamamlanmış Makine Öğrenmesi modelinin başarımını arttırmak ve hata sayısını azaltmak için kullanılan veridir.
Başarım (Accuracy)	Eğitim verilerine dayalı olarak bir veri kümesindeki değişkenler arasındaki ilişkileri ve kalıpları belirlemede hangi modelin en iyi olduğunu belirlemek için kullanılan ölçümdür [13].
Çapraz Doğrulama (Cross Validation)	Makine Öğrenmesi modelimizin görünmeyen veriler üzerinde ne kadar iyi performans sergilediğini ispatlamak üzere kullanılan bir tekniktir [14].
Aşırı Öğrenme (Overfitting)	Bir Makine Öğrenmesi modeli, eğitim verisindeki gereksizleri, gelecek veriler için başarımını negatif doğrultu derecesinde öğrendiğinde, Aşırı Öğrenme olur.
Eksik Öğrenme (Underfitting)	Aşırı Öğrenmenin tersi durumdur. Model kurulması için gerekli olan verilerin eğitilememesi ve yeni veriler üzerinde genelleme işlemi yapamayan modeldir.

Makine Öğrenmesi Yaşam Döngüsü yani akış sıralaması ise şu şekildedir. Öncelikle ham veri toplanır, veri çeşitli ön işlemlerden geçirilir. Bu ön işlemler sayesinde veri içindeki eksik, hatalı ve gürültülü veriler veri setinden temizlenir. Ön işleme adımından geçmiş veriler eğitim, test ve doğrulama verisi olarak ayrılır. Ayrılan veriler uygun görülen Makine Öğrenmesi algoritmasıyla eğitilir ve sonrasında test ve doğrulama verileriyle karşılaştırılır. Başarım istenen düzeyde ise Makine Öğrenmesi modeli *Deploy* edilir.



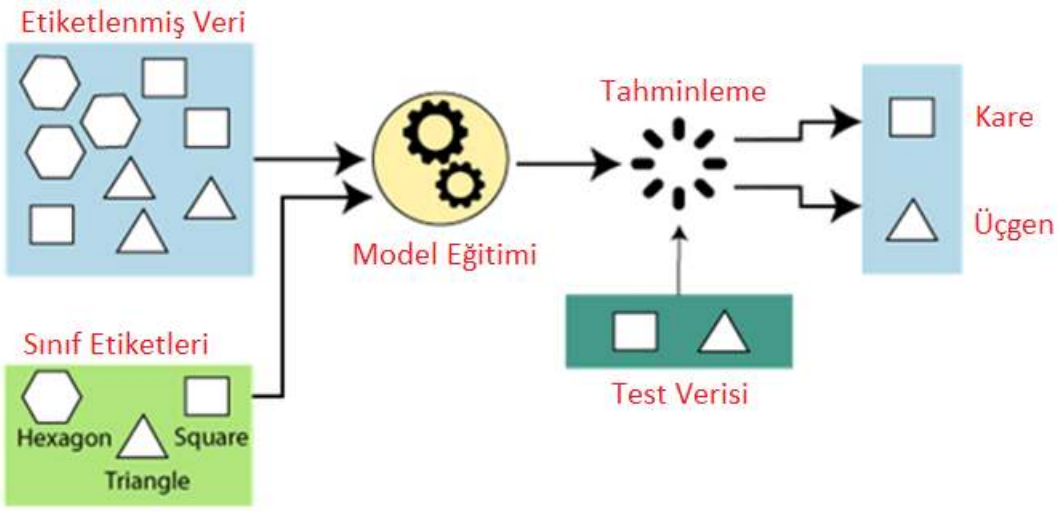
Şekil 4.1. Makine Öğrenmesi Yaşam Döngüsü

4.2. Makine Öğrenmesi Yöntemleri

Bu çalışmada Gözetimli Öğrenme, Yarı Gözetimli Öğrenme, Pekiştirmeli Öğrenme, Gözetimsiz Öğrenme olmak üzere dört Makine Öğrenmesi Yöntemi incelenmiştir.

4.2.1. Gözetimli Öğrenme

Gözetimli Öğrenme, makinelerin etiketlenmiş eğitim verileri kullanılarak eğitildiği ve bu verilere dayanarak makinelerin çıktıyı tahmin ettiği Makine Öğrenmesi yöntemlerinden biridir. Etiketli veriler, bazı girdi verilerinin zaten doğru çıktıyla etiketlendiği anlamına gelir. Gözetimli Öğrenmede, makinelere sağlanan eğitim verileri, makinelere çıktıyı doğru tahmin etmeyi öğreten mantık olarak çalışır. Öğrencinin öğretmen gözetiminde öğrendiği kavramın aynısını uygular. Gözetimli Öğrenme, Makine Öğrenmesi modeline girdi verilerinin yanı sıra doğru çıktı verilerini sağlama sürecidir [15]. Gözetimli Öğrenme algoritmasının amacı, girdi değişkenini çıktı değişkeni ile eşlemek için bir eşleme işlevi bulmaktır. Gerçek dünyada Gözetimli Öğrenme, Risk Değerlendirmesi, Görüntü sınıflandırması, Dolandırıcılık Tespiti, Spam filtreleme vb. için kullanılabilir. Gözetimli Öğrenme çok kullanılan bir Makine Öğrenmesi yöntemidir. Problemlerin sonucuna göre Sınıflandırma ve Regresyon problemleri şeklinde kullanılır. Gözetimli Öğrenmede modeller, modelin her bir veri türü hakkında öğrendiği etiketli veri kümesi kullanılarak eğitilir. Eğitim süreci tamamlandıktan sonra model, test verilerine dayanarak test edilir ve ardından çıktıyı tahmin eder [16].

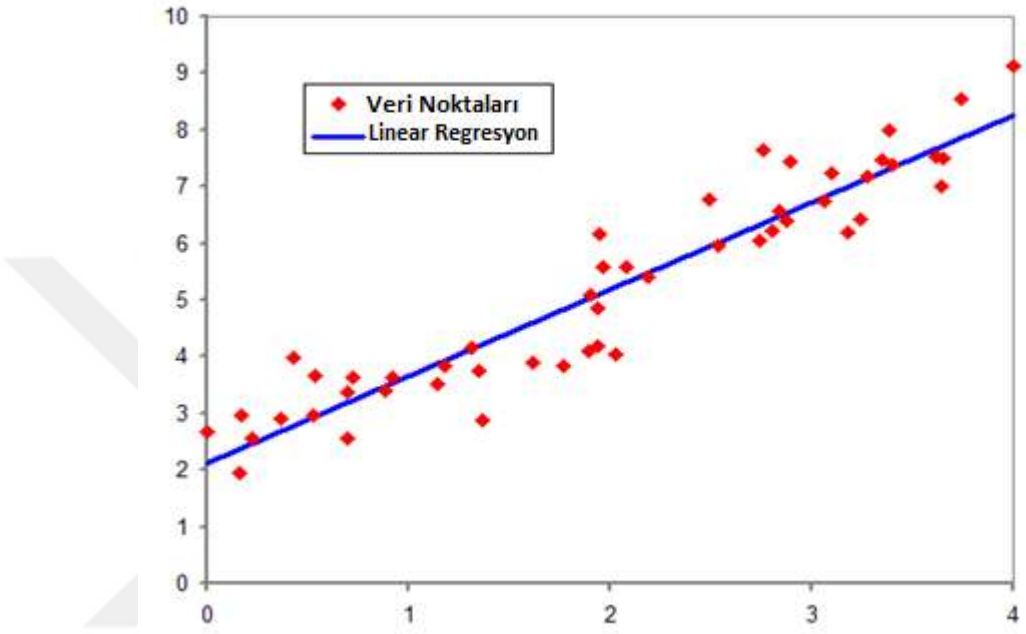


Şekil 4.2. Gözetimli Öğrenme Çalışma Mantığı

Tüm Makine Öğrenmesi algoritmaları gibi, Gözetimli Öğrenme de eğitime dayalıdır. Eğitim aşaması sırasında sistem, sisteme her bir belirli girdi değeriyle ilgili çıktının ne olduğunu bildiren etiketli veri setleri ile beslenir. Eğitilen modele daha sonra test verileri sunulur: Bu, etiketlenmiş verilerdir, ancak etiketler algoritmaya gösterilmemiştir. Test verilerinin amacı, algoritmanın etiketlenmemiş veriler üzerinde ne kadar doğru performans göstereceğini ölçmektir.

Girdi değişkeni ile çıktı değişkeni arasında bir ilişki varsa, Regresyon algoritmaları kullanılır. Hava tahmini, Piyasa Trendleri vb. gibi sürekli değişkenlerin tahmini için kullanılır. Regresyon Bağımlı Değişken ile Bağımsız Değişken arasındaki ilişkinin kuvvetini ölçmeye odaklanır ve aradaki kuvvete göre değer tahmini yapmaya çalışır [17]. Regresyon modellerine örnek olarak, posta koduna dayalı olarak emlak fiyatlarının tahmin edilmesi, günün saatine göre çevrimiçi reklamlardaki tıklama oranlarının tahmin edilmesi, müşterilerin yaşlarına göre belirli bir ürün için ne kadar ödeme yapmak isteyebileceğinin belirlenmesi, hava durumu tahminleri, borsadaki finansal değişimleri tahmin etmek ve karar almak verilebilir. Doğrusal Regresyon (*Linear Regression*), Lojistik Regresyon (*Logistic Regression*), Sinir Ağları (*Neural Networks*), Doğrusal Diskriminant Analizi (*Linear Discriminant Analysis*), Karar Ağaçları (*Decision Trees*), Benzerlik Öğrenme (*Smilitary Learning*), Bayes Mantığı (*Bayesian Logic*), Destek Vektör Makineleri (*Support Vector Machines*) yaygın olarak kullanılan Regresyon algoritmalarıdır [18]. Regresyon algoritması seçerken göz önünde bulundurulması gereken birkaç şey vardır. Birincisi,

yeterince esnek olmakla çok esnek olmak arasında ince bir çizgi olduğundan, algoritma içinde var olan önyargı ve varyanstır. Bir diğeri, sistemin öğrenmeye çalıştığı modelin veya işlevin karmaşıklığıdır. Belirtildiği gibi bir algoritma seçmeden önce verilerin heterojenliği, doğruluğu, fazlalığı ve doğrusallığı da analiz edilmelidir.

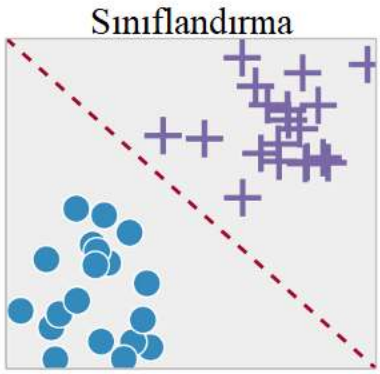


Şekil 4.3. Regresyon Temsili Gösterimi

Sınıflandırma, belirli bir veri kümesini sınıflara ayırma işlemler bütünüdür [29]. Hem yapılandırılmış hem de yapılandırılmamış veriler üzerinde gerçekleştirilebilir. Süreç, verilen veri noktalarının sınıfını tahmin etmekle başlar. Sınıflara genellikle hedef, etiket veya kategoriler denir.

Sınıflandırma kestirimci modelleme, girdi değişkenlerinden ayrık çıktı değişkenlerine eşleme fonksiyonunun yaklaşma görevidir. Temel amaç, yeni verilerin hangi sınıfa-kategoriye gireceğini belirlemektir [30]. Örneğin, e-posta servis sağlayıcılarında istenmeyen e-posta tespiti, bir sınıflandırma sorunu olarak tanımlanabilir. Spam olarak değil, spam olarak yalnızca 2 sınıf olduğundan ikili sınıflandırmadır. Bir sınıflandırıcı, verilen girdi değişkenlerinin sınıfla nasıl ilişkili olduğunu anlamak için bazı eğitim verilerini kullanır. Bu durumda, bilinen spam ve spam olmayan e-postalar eğitim verisi olarak kullanılmalıdır. Sınıflandırıcı doğru bir şekilde eğitildiğinde, bilinmeyen bir e-postayı tespit etmek için kullanılabilir.

Sınıflandırma, hedeflerin girdi verileriyle de sağlandığı Gözetimli Öğrenme kategorisine aittir. Kredi onayı, tıbbi teşhis, hedef pazarlama vb. birçok alanda sınıflandırmada birçok uygulama bulunmaktadır [19]. Şu anda birçok Sınıflandırma algoritması mevcut ancak hangisinin diğerinden üstün olduğu sonucuna varmak mümkün değildir. Mevcut veri setinin uygulamasına ve doğasına bağlıdır. Örneğin, sınıflar doğrusal olarak ayrılabilir ise, Lojistik regresyon gibi doğrusal sınıflandırıcılar kullanılmalıdır. En sık kullanılan Sınıflandırma algoritmaları; Naif Bayes (Naive Bayes), Karar Ağaçları (Decision Trees), Support Vector Machines (Destek Vektör Makineleri), J48, IBk, Yapay Sinir Ağları (Artificial Neural Networks), Rastgele Orman (Random Forest) şeklindedir [20].



Şekil 4.4. Sınıflandırma Temsili Gösterimi

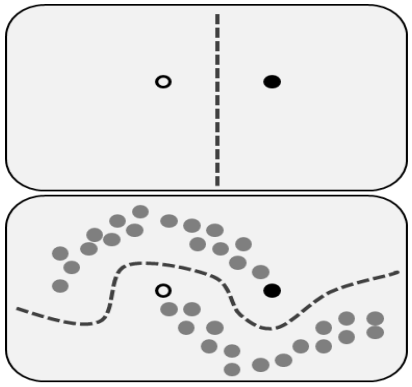
4.2.2. Yarı Gözetimli Öğrenme

Yarı Gözetimli Öğrenme, az sayıda etiketlenmiş örnek ve çok sayıda etiketlenmemiş örnek içeren bir öğrenme problemidir [21]. Bu tür öğrenme problemleri ne Gözetimli ne de Gözetimsiz Öğrenme algoritmaları etiketli ve anlaşılmaz verilerin karışımlarını etkili bir şekilde kullanamadığından zordur. Bu nedenle, özel Yarı Gözetimli Öğrenme algoritmaları gereklidir. Bir modelin öğrenmesi ve yeni örnekler üzerinde tahminlerde bulunması gereken etiketli örneklerin küçük bir bölümünü ve çok sayıda etiketlenmemiş örneği içeren bir öğrenme problemini ve öğrenme problemi için tasarlanmış algoritmaları ifade eder. Etiketleme örneklerinin zor veya pahalı olduğu durumlarda verilerle çalışırken Yarı Gözetimli Öğrenme algoritmalarına ihtiyacımız vardır.

Yarı Gözetimli Öğrenme, Gözetimli ve Gözetimsiz Makine Öğrenmesi arasında kalan önemli bir kategoridir. Yarı Gözetimli Öğrenme, Gözetimli ve

Gözetimsiz Öğrenme arasındaki orta yol olmasına ve birkaç etiketten oluşan veriler üzerinde çalışmasına rağmen, çoğunlukla etiketlenmemiş verilerden oluşur. Etiketler maliyetli olduğundan, ancak kurumsal amaç için birkaç etikete sahip olabilir. Gözetimli Öğrenmenin temel dezavantajı, Makine Öğrenmesi uzmanları veya veri bilimcileri tarafından elle etiketlemeyi gerektirmesi ve ayrıca işleme maliyetinin yüksek olmasıdır [22]. Daha fazla Gözetimsiz Öğrenme, uygulamaları için sınırlı bir spektruma sahiptir. Gözetimli Öğrenme ve Gözetimsiz Öğrenme algoritmalarının bu dezavantajlarının üstesinden gelmek için Yarı Gözetimli Öğrenme kavramı tanıtılmıştır. Bu algoritmada eğitim verileri hem etiketlenmiş hem de etiketlenmemiş verilerin bir kombinasyonudur. Ancak etiketlenmiş veriler çok az miktarda bulunurken çok büyük miktarda etiketlenmemiş veriden oluşur. Başlangıçta, benzer veriler Gözetimsiz bir öğrenme algoritması ile birlikte kümelenir ve ayrıca etiketlenmemiş verilerin etiketlenmiş verilere etiketlenmesine yardımcı olur. Bu nedenle etiket verileri, etiketlenmemiş verilere göre nispeten daha pahalı bir kazanımdır [23].

Yarı Gözetimli Öğrenme, modeli Gözetimli Öğrenmeye göre daha az etiketlenmiş eğitim verisi ile eğitmek için sözde etiketlemeyi kullanır. Süreç, çeşitli Sinir Ağı modellerini ve eğitim yollarını birleştirebilir. Yarı Gözetimli Öğrenmenin tüm işleyişi şöyle açıklanmaktadır: İlk olarak, modeli Gözetimli Öğrenme modellerine benzer şekilde daha az miktarda eğitim verisi ile eğitir. Model doğru sonuçlar verene kadar eğitim devam eder. Algoritmalar, bir sonraki adımda sözde etiketlerle etiketlenmemiş veri kümesini kullanır ve sonuç doğru olmayabilir. Etiketli eğitim verilerinden gelen etiketler ve sözde etiket verilerinden gelen etiketler birbirine bağlanmıştır [24]. Etiketli eğitim verilerindeki ve etiketsiz eğitim verilerindeki girdi verileri de bağlantılıdır. Son olarak, modeli ilk adımda olduğu gibi yeni birleştirilmiş girdiyle yeniden eğitiriz. Hatalar azalacak ve modelin doğruluğu artıracaktır.

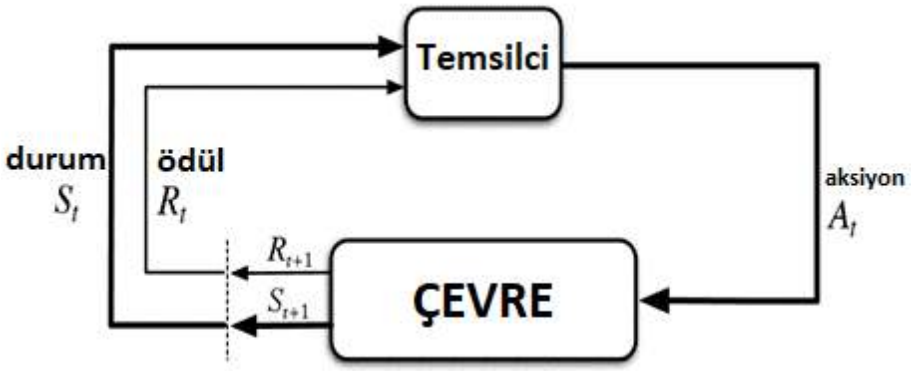


Şekil 4.5. Yarı Gözetimli Öğrenme Temsili Gösterimi

4.2.3. Pekiştirmeli Öğrenme

Pekiştirmeli Öğrenme, MÖ'nin bir alanıdır. Belirli bir durumda ödülü en üst düzeye çıkarmak için uygun eylemi yapmakla ilgilidir. Belirli bir durumda izlemesi gereken olası en iyi davranışı veya yolu bulmak için çeşitli yazılımlar ve makineler tarafından kullanılır [25]. Pekiştirmeli Öğrenme, Gözetimli Öğrenmede eğitim verilerinin cevap anahtarına sahip olması ve böylece modelin doğru cevabın kendisi ile eğitilmesi, Pekiştirmeli Öğrenmede cevap olmaması, ancak takviye ajanının ne yapacağına karar vermesi bakımından farklıdır. Verilen görevi yerine getirmek asıl amaçtır. Bir eğitim veri kümesinin yokluğunda, deneyimlerinden öğrenmesi zorunludur [26].

Hem Gözetimli hem de Pekiştirmeli Öğrenme, girdi ve çıktı arasındaki eşlemeyi kullansa da aracıya sağlanan geribildirimin bir görevi yerine getirmek için doğru eylemler dizisi olduğu Gözetimli Öğrenmenin aksine, Pekiştirmeli Öğrenme, olumlu ve olumsuz davranış için sinyaller olarak ödülleri ve cezaları kullanır. Gözetimsiz Öğrenmeye kıyasla, Pekiştirmeli Öğrenme hedefler açısından farklıdır. Gözetimsiz öğrenmede amaç, veri noktaları arasındaki benzerlikleri ve farklılıkları bulmak iken, Pekiştirmeli Öğrenme durumunda amaç, temsilcinin toplam kümülatif ödülünü maksimize edecek uygun bir eylem modeli bulmaktır. Aşağıdaki şekil, genel bir Pekiştirmeli Öğrenme modelinin eylem-ödül geribildirim döngüsünü göstermektedir [27].



Şekil 4.6. Pekiştirmeli Öğrenme Eylem-Ödül Geribildirim Döngüsü

Tablo 4.2. Pekiştirmeli Öğrenme Anahtar Kelimeleri ve Açıklamaları

Anahtar Kelimeler	Açıklamalar
Girdi *(Input)*	Girdi, modelin başlayacağı bir başlangıç durumu olmalıdır.
Çıktı *(Output)*	Belirli bir problem için çeşitli çözümler olduğu için birçok olası çıktı vardır.
Eğitim *(Training)*	Eğitim girdiye dayalıdır, Model bir durum döndürür ve kullanıcı, modeli çıktısına göre ödüllendirmeye veya cezalandırmaya karar verir.
Çevre *(Environment)*	Temsilcinin faaliyet gösterdiği fiziksel dünya.
Durum *(State)*	Temsilcinin mevcut durumu.
Ödül *(Reward)*	Çevreden geri bildirim.
Politika *(Policy)*	Aracının durumunu eylemlerle eşleştirme yöntemi.
Değer *(Value)*	Bir temsilcinin belirli bir durumda bir eylemde bulunarak alacağı gelecekteki ödül.

Pekiştirmeli Öğrenme; Endüstriyel otomasyon için Robotik alanında, Makine Öğrenmesi ve Veri İşleme problemlerinde, öğrencilerin gereksinimlerine göre özel eğitim ve materyaller sağlayan eğitim sistemleri oluşturmak için kullanılmaktadır [28].

En sık kullanılan Pekiştirmeli Öğrenme modeli ise Q modelidir. Rastgele bir çevrede en iyi politika ile sistemin nasıl öğrenebileceğini hedefler. Herhangi eğitim verisi içermeyen sistemin hangi politikayı seçeceği çok zor bir sorudur [29]. Sistem eğitim verisi olarak kendisine ödülleri kabul eder. Böylece sistem her ödül veya cezadan sonra bir takım matematiksel işlemler sayesinde en iyi politikayı öğrenmeye

çalışır. Asıl amaç ise geçmişteki hareketlerin incelenerek gelecekteki hareketlere doğru ve kesin bir yol vermektir.

En sık kullanılan alanlar ise oyun sektörü, kaynak yönetimi, kişisel tavsiye sistemleri ve robotik alanlarıdır. Oyun, Pekiştirmeli Öğrenme için muhtemelen sık kullanım alanıdır. Çokça oyun çeşitinde fazladan başarı yakalamaktadır. En iyi örneği ise Pac-Man oyunudur [30].

4.2.4. Gözetimsiz Öğrenme

Gözetimsiz Öğrenme adından da anlaşılacağı gibi modellerin eğitim veri kümesi kullanılarak denetlenmediği bir Makine Öğrenmesi yöntemidir [41]. Bunun yerine, modellerin kendisi, verilen verilerden gizli kalıpları ve iç görüleri bulur. Yeni şeyler öğrenirken insan beyninde gerçekleşen öğrenmeye benzetilebilir.

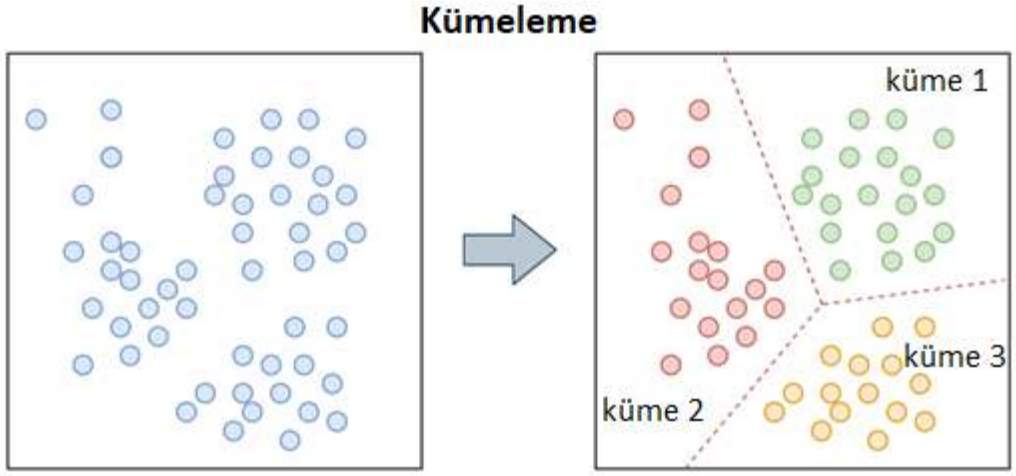
Gözetimsiz Öğrenme, bir regresyon veya sınıflandırma problemine doğrudan uygulanamaz çünkü Denetimli Öğrenmenin aksine, girdi verilerine sahibiz, ancak karşılık gelen çıktı verisine sahip değiliz. Gözetimsiz Öğrenmenin amacı, veri kümesinin temel yapısını bulmak, bu verileri benzerliklere göre gruplandırmak ve bu veri kümesini sıkıştırılmış bir biçimde temsil etmektir [43].

Örneğin Gözetimsiz Öğrenme algoritmasına, farklı kedi ve köpek türlerinin görüntülerini içeren bir girdi veri kümesi verildiğini varsayalım. Algoritma hiçbir zaman verilen veri kümesi üzerinde eğitilmez, bu da veri kümesinin özellikleri hakkında hiçbir fikri olmadığı anlamına gelir. Gözetimsiz Öğrenme algoritmasının görevi, görüntü özelliklerini kendi başlarına tanımlamaktır. Gözetimsiz Öğrenme algoritması, görüntü veri setini görüntüler arasındaki benzerliklere göre gruplara ayırarak bu görevi gerçekleştirecektir. Gözetimsiz Öğrenme, verilerden yararlı iç görüler bulmak için yararlıdır. Gözetimsiz Öğrenme, bir insanın kendi deneyimleriyle düşünmeyi öğrenmesine çok benzer, bu da onu gerçek Yapay Zekâya daha yakın hale getirir. Gözetimsiz Öğrenme, Gözetimsiz Öğrenmeyi daha önemli hale getiren etiketlenmemiş ve kategorize edilmemiş veriler üzerinde çalışır. Gerçek dünyada, her zaman karşılık gelen çıktıya sahip girdi verisine sahip değiliz, bu nedenle bu tür durumları çözmek için Gözetimsiz Öğrenmeye ihtiyacımız vardır. Gözetimsiz öğrenme algoritması ayrıca iki tür probleme ayrılır.

Kümeleme, nesneleri kümeler halinde gruplandırma yöntemidir, öyle ki en çok benzerliğe sahip nesneler bir grup içinde kalır ve başka bir grubun nesneleriyle daha az benzerliği vardır veya hiç benzerliği yoktur. Kümeleme, veri nesneleri arasındaki ortak noktaları bulur ve bunları bu ortak noktaların varlığına ve yokluğuna göre sınıflandırır.

Birliktelik kuralı, büyük veri setindeki değişkenler arasındaki ilişkileri bulmak için kullanılan Gözetimsiz Öğrenme yöntemidir. Veri kümesinde birlikte oluşan öğelerin kümesini belirler. Birliktelik kuralı, pazarlama stratejisini daha etkili hale getirir. Örneğin X ürünü (bir ekmek varsayalım) alan kişiler, Y (Tereyağı/Reçel) ürününü de satın alma eğilimindedir. Birliktelik kuralının tipik bir örneği Pazar Sepeti Analizidir.

En sık kullanılan Gözetimsiz Öğrenme algoritmaları şu şekildedir: K-Means Kümeleme Algoritması, KNN (K En Yakın Komşu) Algoritması, Hiyerarşik *(Hierarchal)* Algoritması, Apriori Algoritmasıdır.



Şekil 4.7. Kümeleme Algoritması Temsili Gösterimi

5. DERİN ÖĞRENME

5.1. Derin Öğrenme Tanım ve Temel Kavramları

DÖ, YSA kısaltmasıyla isimlendirilen insan beyninin işlev ve yapısından yola çıkan kavramlar bütünüdür [44]. Sinir Ağları *(Neural Networks)*, insandaki beyin yapı ve işlevlerini taklit etmeye çalışır, çok fazla veriden model kurulmasına olanak sağlar. Bir katmandan oluşan NN beklenen başarıma yakın tahminler yapıyorken, gizli katmanlar başarımı ayarlamaya ve yükseltmeye çalışırlar. DÖ, otomasyonu geliştiren ve çeşitli görevleri insan müdahalesi olmadan gerçekleştiren YZ uygulamalarını barındırır. DÖ teknolojileri, sesli asistanlar, sanal alışveriş dolandırıcılığı gibi gelişmekte olan teknolojilerin arkasında yatar. DÖ, birlikte çalıştığı veri türü ve öğrendiği yöntemlerle kendisini klasik Makine Öğrenmesinden ayırır.

Makine Öğrenmesi algoritmaları, tahminler yapmak için kurulmuş, verileri kullanır. Belirli özniteliklerin kurulacak modelde girdi olarak kullanılacak verilerin tanımlandığı çeşitli saklama teknikleriyle bir arada tutulması demektir.

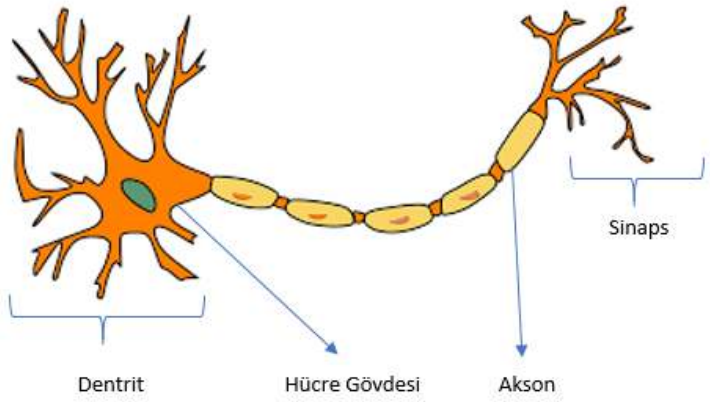
Derin Öğrenmeyi kullanan bilgisayar programları, yürümeye başlayan çocuğun köpeği tanımlamayı öğrenmesiyle hemen hemen aynı süreçten geçer. Çoğu algoritma girdi olarak kullandığı verilere çeşitli kurallar implemente eder ve model kullanıma hazır hale getirilir. Model istenen başarıma gelene kadar yinelemeler sürer. Verilerin geçmesi gereken işleme katmanlarının sayısı, etikete derinden ilham veren şeydir.

5.2. Derin Öğrenme Yöntemleri

5.2.1. Yapay Sinir Ağları

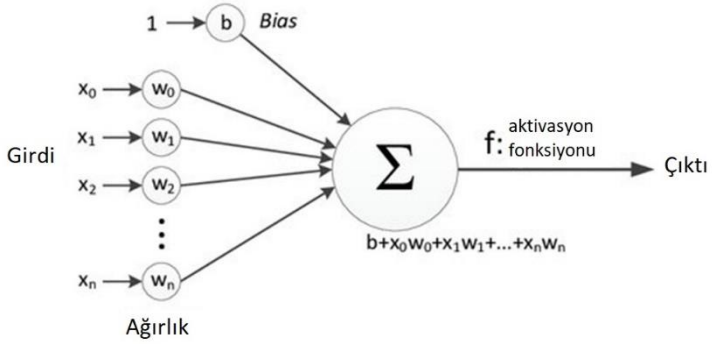
YSA, işleyişi biyolojik nöronlardan oluşan bir ağa benzeyen bir veri analiz yöntemidir [45]. YSA' lar ağırlıklı bağlantılar ile birbirine bağlanan bir düğüm sisteminden oluşur. YSA' nın sonucu, bağlantıların ağırlıklarındaki değişikliklerle değiştirilir. Veriler giriş katmanına beslenir ve ağın sonucu çıkış katmanı tarafından görüntülenir. Giriş düğümleri, bağımlı değişkenleri, yani çıkış nöronlarını tahmin etmek için kullanılan bağımsız veya tahmin edici değişkenleri temsil eder. Yapay

Sinir Ağları teorik açıdan biyolojik sinir ağlarıyla üretilen yapay sinir ağlarından meydana gelirler. Çoğu yapay sinir ağlarının girişleri mevcuttur ayrıca diğer sinir ağlarına iletilecek sadece tek çıkış barındırırlar. Girdiler, resim gibi metin gibi bir örnek veya başka sinir ağlarının çıktı değerleri olabilmektedir. Sinir ağının son output nöronlarının çıktıları, herhangi bir resimdeki objeyi tespit etmek görevini yapabilir.



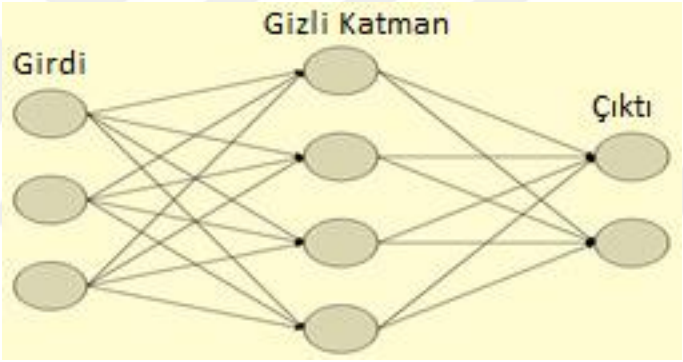
Şekil 5.1. Bir Sinir Ağının Biyolojik Gösterimi

Bir Yapay Sinir Ağı, birbirine bağlı üç veya daha fazla katmana sahiptir. Giriş katmanı, giriş sinir ağlarından oluşur. Daha derin katmanlara veri iletilir ve böylece nihai çıktı verilerini son çıktı katmanına gönderir. Tüm iç katmanlar gizlidir ve bir dizi dönüşüm yoluyla katmandan katmana alınan bilgileri uyarlamalı olarak değiştiren birimler tarafından oluşturulur. Her katman, YSA' nın daha karmaşık nesneleri anlamasını sağlayan hem girdi hem de çıktı katmanı görevi görür. Toplu olarak, bu iç katmanlara Nöral Katman denir.



Şekil 5.2. Nöronun Formüle Gösterilişi

Nöral katmanındaki birimler, toplanan bilgileri YSA' nın iç sistemine göre tartarak öğrenmeye çalışırlar. Bu yönergeler, birimlerin bir sonraki katmana çıktı olarak sağlanan dönüştürülmüş bir sonuç oluşturmasına izin verir. Ek bir öğrenme kuralları seti, ANN' nin hataları hesaba katarak çıktı sonuçlarını ayarlayabildiği bir süreç olan geri yayılımı kullanır. Geri yayılım yoluyla, denetimli eğitim aşamasında çıktının bir hata olarak etiketlendiği her seferde, bilgi geriye doğru gönderilir. Her ağırlık, hatadan ne kadar sorumlu olduklarıyla orantılı olarak güncellenir. Bu nedenle hata, istenen sonuç ile gerçek sonuç arasındaki farkı hesaba katmak için YSA' nın birim bağlantılarının ağırlığını yeniden kalibre etmek için kullanılır. Zamanla YSA, hata ve istenmeyen sonuç olasılığını nasıl en aza indireceğini öğrenecektir. YSA' lar, var olan algoritmaların başarımını artırmak için gayet temel düzeyde model tabiriyle bilinir. İş zekasında tahmine dayalı analiz, istenmeyen e-posta algılama, sohbet robotlarında doğal dil işleme ve daha pek çok pratik uygulama için kullanılabilirler.



Şekil 5.3. 3 Katmanlı Yapay Sinir Ağı Modeli

Hiper parametre, model kurulma aşamasından önce bilinen değişmez elemandır. Değerler model kurulma aşamasında öğrenilir. Kimi hiper parametreler, başka hiper parametrelere bağımlıdır. Mesela, katmanlar katman toplam adedine dayalı olabilmektedir. Belirlenecek hiper parametreler modelin başarımında ciddi rol oynamaktadır. Modelin yüksek başarım oranı yakaladığı çok çeşitli hiper parametre grupları olabilmektedir. Hiper parametre seçimi YSA tasarlanırken seçiminde dikkat edilmesi gereken en önemli konulardan birisidir. En yaygın kullanılan hiper parametreler ise şu şekildedir.

Öğrenme *(Learning)*, çeşitli bulguları gözeterek işin fazla başarımda olması için kullanılan parametredir. Learning, başarımı artırmak amacıyla sinr ağındaki katman etkilerinin ve çeşitli threshold parametlerinin düzenlenmesini barındırır.

Fazladan bulguları değerlendirirken öğrenme sona erer. Hiçbir zaman hata oranımız sıfır değerine ulaşamaz. Öğrenme işleminden sonra çıkan hata değeri hala fazlaysa, sinir ağı baştan kurulmalıdır. Teoride bunun adı maliyet diye adlandırılır. Maliyet değeri sadece yaklaşık değerle bulunabilir. Learning işlemi bulgular içindeki eksikliklerin tamamını indirgemeye amaçlar. Birçok öğrenme algoritması, dengeleme kuramı veya matematiksel bulgu tahmin aşaması olarak kabul edilir.

Geri Yayılım *(Back Propagation)*, öğrenme aşamasında yer alan hatayı gidermek amacıyla katmanlar arasında kullanılan ağırlıkları ayarlama işlemidir. Hata oranı katmanlar arasında etkili bir şekilde pay edilir.

Model kurma becerileri sayesinde, YSA disiplinler arası uygulamalarda kullanılmaktadır. Sistem Tanımlama, Kuantum Kimyası, Örüntü Tanıma, 3D rekonstrüksiyon, Nesne Tanıma, Sensör Veri Analizi, Tıbbi Teşhis, Finans, Veri Madenciliği, Görselleştirme, Makine Çevirisi, Sosyal Ağ Filtreleme gibi kullanım alanları bulunmaktadır. Ayırca tıp alanında kötü huylu kanser hücrelerini tespit amacıyla kullanılmışlardır.

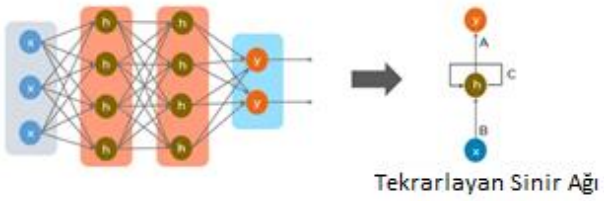
5.2.2. Tekrarlayan Sinir Ağları

RNN' ler Tekrarlayan Sinir Ağları anlamına gelir; bu, sıralı verileri işleyebilen, kalıpları tanıyabilen ve nihai çıktıyı tahmin edebilen bir tür Yapay Sinir Ağıdır. Bu Sinir Ağı, bir dizi girdi üzerinde aynı görevi veya işlemi tekrar tekrar gerçekleştirebildiği için Tekrarlayan olarak adlandırılır [46]. Bir RNN, aldığı girdinin bilgilerini hatırlamasını veya ezberlemesini sağlayan bir dahili belleğe sahiptir ve bu, sistemin bağlam kazanmasına yardımcı olur. Bu nedenle, zaman serisi gibi sıralı verileriniz varsa, bu verileri işlemek için bir RNN iyi bir seçim olacaktır. Bu, bir CNN veya İleri Beslemeli Sinir Ağları tarafından yapılamaz, çünkü önceki giriş ile sonraki giriş arasındaki korelasyonu sıralayamazlar. Google'ın sesli araması ve Apple'ın Siri'si gibi popüler ürünler, kullanıcılarından gelen girdileri işlemek ve çıktıyı tahmin etmek için RNN'yi kullanır.

Bir RNN'nin arkasındaki mantık, belirli katmanın çıktısını kaydetmek ve katmanın çıktısını tahmin etmek için onu girdiye geri beslemektir. Aşağıda, İleri Beslemeli Sinir Ağını Tekrarlayan Sinir Ağına (RNN) nasıl dönüştürebileceğinize dair basit bir örnek verilmiştir.

RNN' ler katmanların liste doğrultusunda yönlendirilmiş veya yönlendirilmemiş bir şekil çıkarttığı YSA koludur. İleri Beslemeli Sinir Ağından *(FFNN)* gelen RNN' ler, çeşitli boyuttaki giriş parametlerini kullanmak amacıyla konumlarını kullanabilirler. Tekrarlayan Sinir Ağları teorik olarak Turing'i tamamlamıştır ve rastgele girdi dizilerini işlemek için rastgele programlar çalıştırabilir.

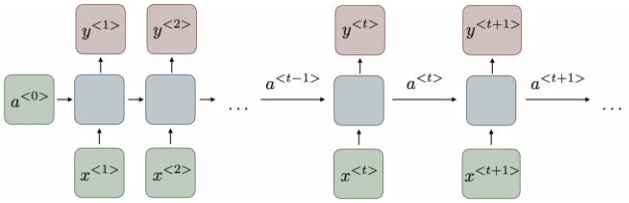
Tekrarlayan sinir ağı terimi, sonsuz bir dürtü yanıtı olan ağların sınıfını belirtmek için kullanılırken, Konvolüsyonel Sinir Ağı, sonlu dürtü yanıtı sınıfını ifade eder. Sınıflar zamansal hareketli olurlar. SDY' ler açılabilen ve kesinlikle FRNN' le değiştirilebilen yönlendirilmiş tekrarsızlardır, SDRNN ise açılamayan tekrarlıdır.



Şekil 5.4. İleri Beslemeli Sinir Ağının RNN Modeline Dönüştürülmesi

Tablo 5.1. Tekrarlayan Sinir Ağları Modelinin Avantaj ve Dezavantajları

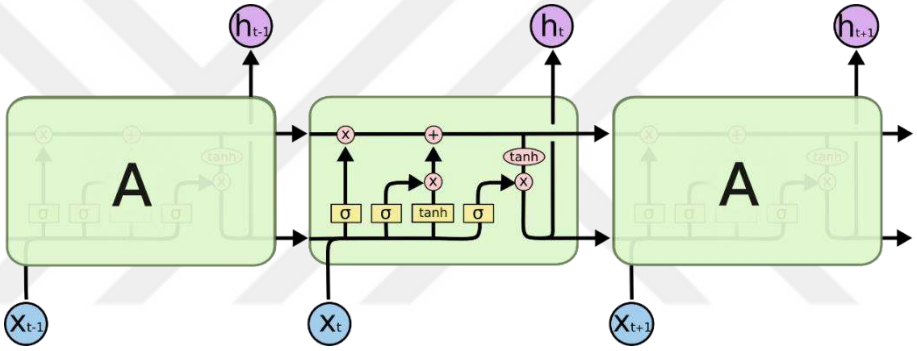
Avantajları	Dezavantajları
İstenilen miktarda giriş verisi işlenebilir	Nispeten az hızlı işlem gücü
Giriş verisine göre değişmeyen model performansı	Eski verilere erişimde sıkıntılar
Eski verilere dayalı kurulan istatistikler	Gelecek verilere göre dizayn edilememe



Şekil 5.5. Geleneksel Tekrarlayan Sinir Ağı Modeli Gösterimi

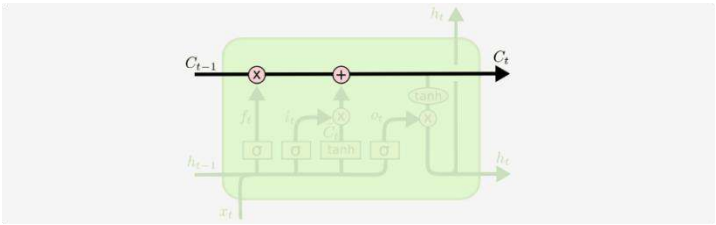
5.2.3. Uzun Kısa Süreli Bellek

Uzun Kısa Süreli Bellek Ağı *(LSTM)*, bilginin kalıcı olmasını sağlayan sıralı bir ağ olan gelişmiş bir RNN' dir. RNN' nin karşılaştığı kaybolan gradyan problemini çözebilir. Kalıcı bellek için kullanılan RNN olarak da bilinen tekrarlayan bir sinir ağıdır [47]. Diyelim ki bir video izlerken önceki sahneyi hatırlıyorsunuz veya bir kitap okurken önceki bölümde neler olduğunu biliyorsunuz. Benzer şekilde RNN' ler de çalışır, önceki bilgileri hatırlar ve mevcut girişi işlemek için kullanırlar. RNN' nin eksikliği, kaybolan gradyan nedeniyle uzun vadeli bağımlılıkları hatırlayamamalarıdır. Klasik RNN mimarisinde yinelenen katman tek gizli kısım gibi fazlaca basit kurgudadır. Fakat LSTM için yapı doğrusaldır, tekrarlayan katman daha değişik mimaridedir. LSTM' ler dört özel katmana sahiptir.



Şekil 5.6. LSTM Mimarisi Gösterimi

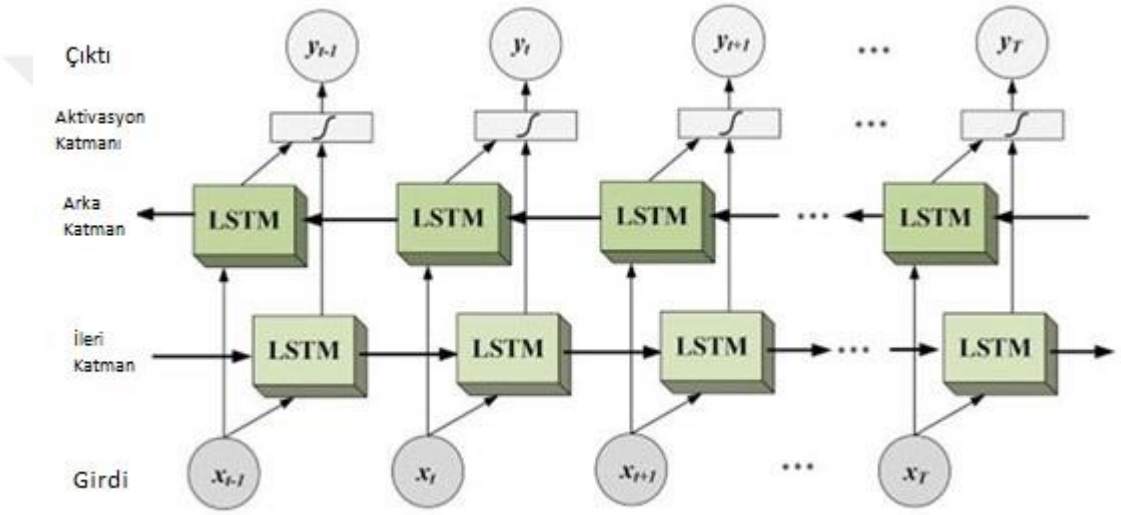
LSTM' lerin asıl mantığı hücre durumudur, diyagramın üstünden geçen yatay çizgidir. Hücre durumu bir tür taşıma bandı gibidir. Sadece bazı küçük doğrusal etkileşimlerle, tüm zincir boyunca dosdoğru çalışır. Bilginin değişmeden akması çok kolaydır.



Şekil 5.7. LSTM En Temel Birim Gösterimi

5.2.4. Çift Yönlü Uzun Kısa Süreli Bellek

Çift Yönlü LSTM veya biLSTM, iki LSTM' den oluşan bir dizi işleme modelidir: biri girdiyi ileri yönde, diğeri ise geri yönde alır. BiLSTM' ler, model için mevcut olan bilgi miktarını etkili bir şekilde arttırır, algoritma için mevcut olan bağlamı iyileştirir. Eğitilmiş iki modelimiz olduğundan, ikisini birleştirmek için bir mekanizma oluşturmamız gerekiyor. Genellikle Birleştirme adımı olarak adlandırılır. Birleştirme adımları, Sum (Toplam), Multiplication (Çarpma), Averaging (Ortalama), Concatenation (Birleştirme) olabilmektedir [48].



Şekil 5.8. Bidirectional LSTM Mimarisi

İleri ve geri katmanlardan bilgi akışını görebiliriz. biLSTM genellikle sıralamadan sıralamaya görevlerin gerekli olduğu durumlarda kullanılır. Bu tür bir model, Metin Sınıflandırma, Konuşma Tanıma ve Tahmin modellerinde kullanılabilir.

Genellikle normal LSTM ağında doğrudan çıktı alıyoruz ancak biLSTM ağında da ileri ve geri katmanın her aşamadaki çıktısı bir sinir ağı olan aktivasyon katmanına verilir ve bu aktivasyon katmanının çıktısı dikkate alınır. Bu çıktı, geçmiş ve gelecek sözcüğün bilgisini veya ilişkisini de içerir.

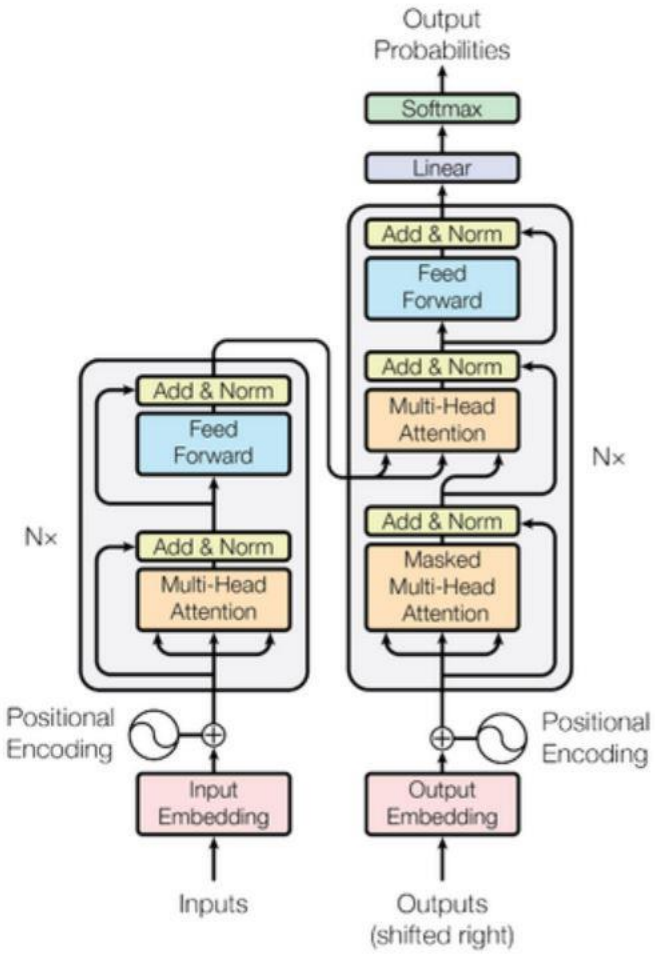
6. TRANSFORMATÖRLER

Transformatör, giriş verilerinin her bir bölümünün önemini farklı şekilde ağırlıklandıran, kendi kendine dikkat mekanizmasını benimseyen bir Derin Öğrenme modelidir [49]. Öncelikle Doğal Dil İşleme *(DDİ)* ve Bilgisayarlı Görü alanlarında kullanılır. Tekrarlayan Sinir Ağları (RNN'ler) gibi, Transformatörler, çeviri ve metin özetleme gibi görevler için doğal dil gibi sıralı giriş verilerini işlemek üzere tasarlanmıştır. Ancak, RNN'lerin aksine, Transformatörler verileri sadece sırayla işlemez. Bunun yerine, dikkat mekanizması girdi dizisindeki herhangi bir konum için bağlam sağlar. Örneğin, giriş verileri bir doğal dil cümlesi ise, Transormatörün cümlenin başlangıcını bitmeden önce işlemesi gerekmez. Bunun yerine, cümledeki her bir kelimeye anlam veren bağlamı tanımlar. Bu özellik, RNN'lerden daha fazla paralelleştirmeye izin verir ve bu nedenle eğitim sürelerini azaltır. Transformatörlerden önce, en son teknoloji ürünü DDİ sistemleri, LSTM ve Kapılı Tekrarlayan Birimler (GRU'lar) gibi kapılı RNN'lere ve ek dikkat mekanizmalarına dayanıyordu. Transformatörler, bir RNN yapısı kullanılmadan bu dikkat teknolojileri üzerine inşa edilmiştir ve dikkat mekanizmalarının tek başına RNN'lerin performansını dikkatle eşleştirebileceği gerçeğini vurgulamaktadır [50].

Geçitli RNN' ler (Gated RNNs), her belirteçten sonra görülen verilerin bir temsilini içeren bir durum vektörünü koruyarak belirteçleri sırayla işler. N'inci belirteci işlemek için, model, cümleye kadar olan durumu temsil eden durumu N-1 yeni belirteç bilgisiyle birleştirerek, cümleyi belirtecine kadar temsil eden yeni bir durum oluşturur N. Teorik olarak, eğer durum her noktada belirteç hakkındaki bağlamsal bilgiyi kodlamaya devam ederse, bir belirteçten gelen bilgi, sıranın aşağısına keyfi olarak yayılabilir. Pratikte bu mekanizma kusurludur: Gradient Descent Loss, modelin durumunu uzun bir cümlenin sonunda, önceki belirteçler hakkında kesin, çıkarılabilir bilgi olmadan bırakır. Belirteç hesaplamalarının önceki belirteç hesaplamalarının sonuçlarına bağımlılığı, modern Derin Öğrenme donanımında hesaplamayı paralelleştirmeyi de zorlaştırır. Bu, RNN'lerin eğitimini verimsiz hale getirebilir.

Bu sorunlar dikkat mekanizmaları tarafından ele alınmaktadır. Dikkat mekanizmaları (Attention Layer), bir modelin dizi boyunca herhangi bir önceki noktada durumdan çekilmesine izin verir. Dikkat katmanı, önceki tüm durumlara

erişebilir ve bunları öğrenilmiş bir alaka düzeyine göre tartabilir ve uzaktaki belirteçler hakkında ilgili bilgiler sağlar. Dikkatin değerinin açık bir örneği, bir cümledeki bir kelimenin anlamını atamak için bağlamın gerekli olduğu dil çevirisidir. İngilizce-Türkçe çeviri sisteminde, Türkçe çıktının ilk kelimesi büyük olasılıkla büyük ölçüde İngilizce girdinin ilk birkaç kelimesine bağlıdır. Ancak klasik bir LSTM modelinde, Türkçe çıktının ilk kelimesini üretmek için modele sadece son İngilizce kelimenin durum vektörü verilir. Teorik olarak, bu vektör, modele gerekli tüm bilgileri vererek, tüm İngilizce cümle hakkındaki bilgileri kodlayabilir. Uygulamada, bu bilgi genellikle LSTM tarafından yetersiz şekilde korunur. Bu sorunu çözmek için bir dikkat mekanizması eklenebilir: kod çözücüye, yalnızca sonuncusuna değil, her İngilizce giriş kelimesinin durum vektörlerine erişim izni verilir ve her bir İngilizce giriş durum vektörüne ne kadar katılacağını belirleyen dikkat ağırlıklarını öğrenebilir. RNN'lere eklendiğinde dikkat mekanizmaları performansı artırır. Transformatör mimarisinin gelişimi, dikkat mekanizmalarının kendi içinde güçlü olduğunu ve RNN'lerin kalite kazanımlarını dikkatle elde etmek için sıralı tekrarlayan veri işlemenin gerekli olmadığını ortaya koydu. Transformatörler, RNN'siz bir dikkat mekanizması kullanır, tüm belirteçleri aynı anda işler ve birbirini izleyen katmanlarda aralarındaki dikkat ağırlıklarını hesaplar. Dikkat mekanizması yalnızca alt katmanlardaki diğer belirteçler hakkındaki bilgileri kullandığından, tüm belirteçler için paralel olarak hesaplanabilir, bu da eğitim hızının artmasına neden olur.



Şekil 6.1. Transformatör Mimarisi

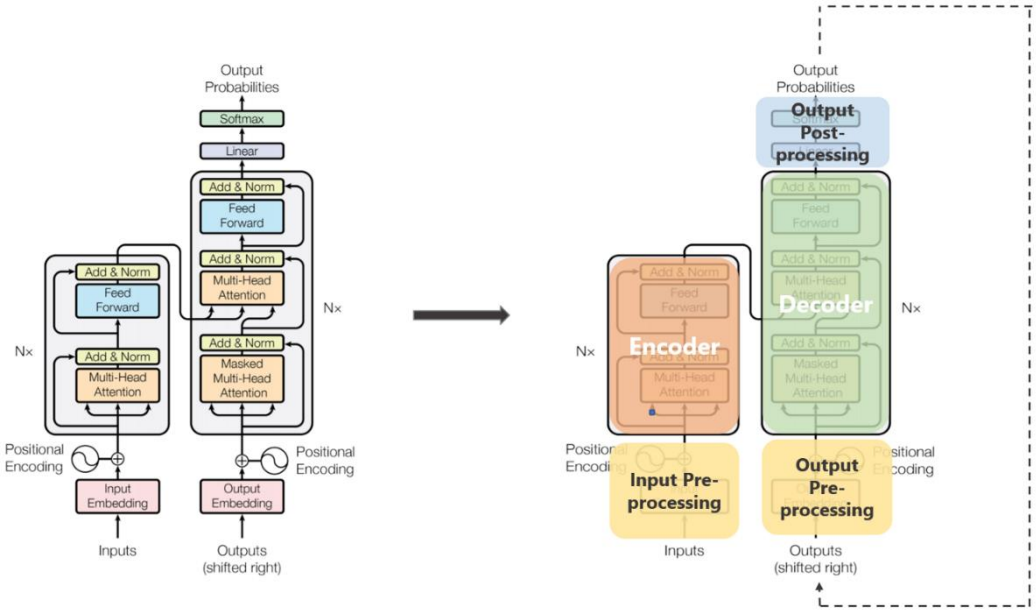
Transformatörler, hiç şüphesiz RNN tabanlı seq2seq modellerine göre büyük bir gelişmedir. Ancak kendi sınırlama payıyla birlikte gelmektedir. Dikkat yalnızca sabit uzunluktaki metin dizileriyle ilgilenebilir. Metin, girdi olarak sisteme beslenmeden önce belirli sayıda parçaya veya parçaya bölünmelidir. Bir metin veriseti, bağlamın parçalanmasına neden olur. Örneğin, bir cümle ortadan bölünürse, önemli miktarda bağlam kaybolur. Başka bir deyişle, metin, cümleye veya başka herhangi bir anlamsal sınıra saygı gösterilmeden bölünür.

6.1. Transformatör Mimarisi

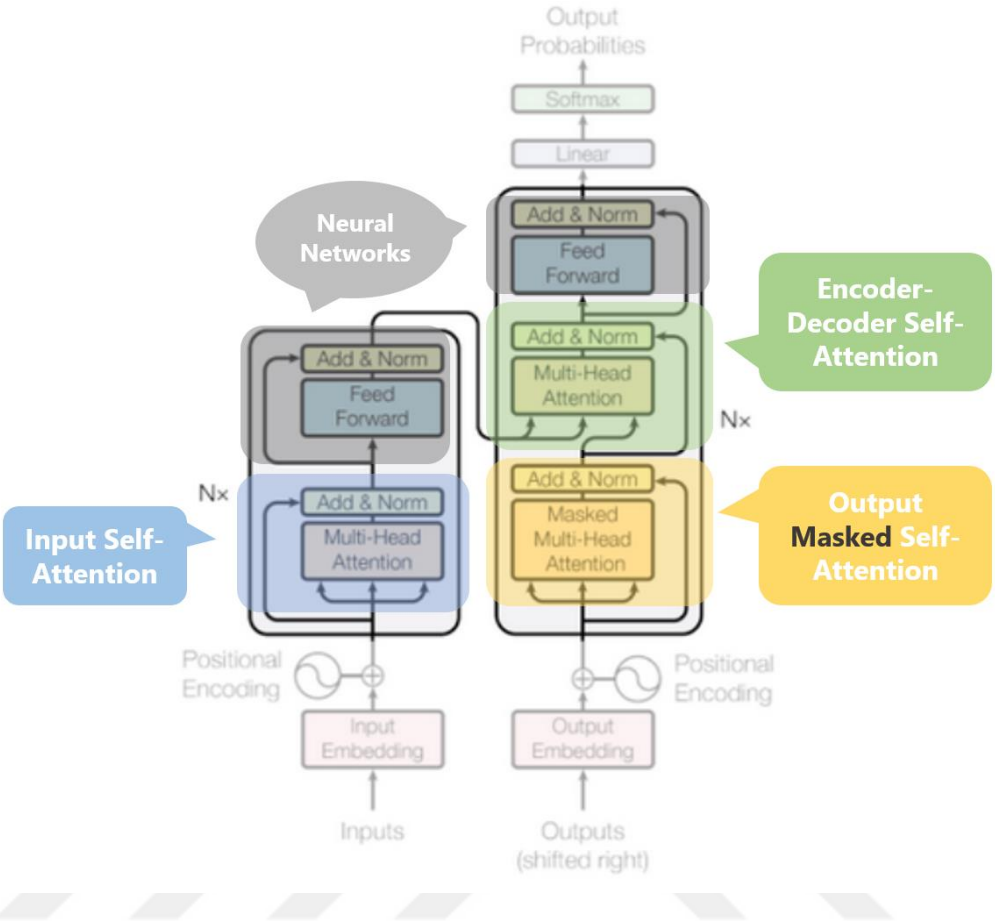
Kodlayıcı (Encoder), iki ana bileşenden oluşur: kendi kendine dikkat mekanizması ve İleri Beslemeli Sinir Ağı. Öz-dikkat mekanizması, önceki kodlayıcıdan gelen girdi kodlamalarını kabul eder ve çıktı kodlamalarını oluşturmak için bunların birbiriyle olan ilişkisini tartar. İleri Beslemeli Sinir Ağı, her çıktı

kodlamasını ayrı ayrı işler. Bu çıktı kodlamaları, daha sonra, kod çözücülerin yanı sıra girdisi olarak bir sonraki kodlayıcıya iletilir. İlk kodlayıcı, giriş olarak kodlama yerine konum bilgisini ve giriş dizisinin yerleştirmelerini alır. Transformatörün sıranın sırasını kullanması için konum bilgisi gereklidir, çünkü transformatörün başka hiçbir parçası bunu kullanmaz.

Kod Çözücü *(Decoder)*, üç ana bileşenden oluşur: bir öz-dikkat mekanizması, kodlamalar üzerinde bir dikkat mekanizması ve İleri Beslemeli bir Sinir Ağı. Kod çözücü, kodlayıcıya benzer şekilde çalışır, ancak bunun yerine kodlayıcılar tarafından oluşturulan kodlamalardan ilgili bilgileri çeken ek bir dikkat mekanizması eklenir. Birinci kodlayıcı gibi, birinci kod çözücü de girdi olarak kodlama yerine konum bilgisini ve çıktı dizisinin yerleştirmelerini alır. Transformatör, bir çıkışı tahmin etmek için mevcut veya gelecekteki çıkışı kullanmamalıdır, bu nedenle bu ters bilgi akışını önlemek için çıkış dizisi kısmen maskelenmelidir. Son kod çözücüyü, kelime dağarcığı üzerinden çıktı olasılıklarını üretmek için son bir doğrusal dönüşüm ve Softmax katmanı takip eder [51].



Şekil 6.2. Encoder Decoder Mimarisi



Şekil 6.3. İleri Beslemeli NN ile Encoder-Decoder Yığınlarındaki Dikkatler

6.2. Transformatör Eğitimi ve Uygulama Alanları

Transformatörler Denetimsiz ön eğitimi ve ardından Denetimli ince ayarı içeren Yarı Denetimli Öğrenmeden geçer. Etiketli eğitim verilerinin sınırlı mevcudiyeti nedeniyle, ön eğitim tipik olarak ince ayardan daha büyük bir veri kümesinde yapılır. Ön eğitim ve ince ayar görevleri ile Dil Modelleme (Language Modelling), Sonraki Cümle Tahmini (Next Sentence Prediction), Soru Cevaplama (Question Answering), Okuduğunu Anlama (Reading Compheresion), Duygu Analizi (Sentiment Analysis) yapılabilmektedir.

GPT-2, GPT-3, BERT, XLNet ve RoBERTa gibi önceden eğitilmiş birçok model, transformatörlerin DDİ ile ilgili çok çeşitli görevleri yerine getirme yeteneğini gösterir. 2020'de transformatör mimarisinin, daha spesifik olarak GPT-2'nin satranç oynamak için ayarlanabileceği gösterildi [52]. Transformatörler,

Evrişimli Sinir Ağlarıyla (CNN) rekabet edebilecek sonuçlarla görüntü işlemeye uygulanmıştır.

7. DOĞAL DİL İŞLEME

Son zamanlarda Bilgisayar Teknolojilerinin yaygın olarak kullanılmasıyla Doğal Dillerin kurallı yapısının çözümlenerek bilgisayarların anlayacağı formata çevrilmesi ve yeniden türetilmesi amaçlanmaktadır. Bilgisayarların konuştuğumuz yani Doğal Dili anlaması, yorum yapması, işlemesi ve hatta cümle kurabilmesine Doğal Dil İşleme denir [53]. Doğal Dil İşleme; Dil Bilimi (Linguistic) ve Hesaplamalı Bilimlerin (Matematik, İstatistik, Makine Öğrenmesi) ortak kullandığı disiplinler arası bir çalışma alanıdır. Doğal Dil İşleme kullanım alanlarının başlıcaları ise; Duygu Analizi (Sentiment Analysis), Dil Çevirisi (Language Translation), Konuşma Tanıma (Speech Recognition), Yazı Denetimi (Spell Check), Metin Özetleme (Text Summarization), Bilgi Çıkarımı (Information Extraction), Metin Sınıflandırma (Text Classification)' dır.

Şekil 7.1. Doğal Dil İşleme Temsili Görseli

7.1. Doğal Dil İşlemeye Genel Bakış

Yakın gelecekte farklı toplumlara ait insanların doğal dil farklılığından dolayı konuşmalarını anlayamama sorunu, makine-insan iletişimindeki sınırların fazlalığı büyük ölçüde kalkacaktır. Bunu sağlayacak olan ise Yapay Zekânın alt dalı olan Doğal Dil İşlemedir. Yapay Zekâda olduğu gibi Doğal Dil İşleme teknikleri de matematiksel ve istatistiksel temellere dayanmaktadır. İncelenecek olan verilerin türlerine, sözlü ifadelerdeki matematiksel çıkarımlara, yazılı ifadelerde ise dillerin ait olduğu ailelere göre bu inceleme süreçleri değişkenlik gösterebilecektir.

7.2. Doğal Dil İşleme Teknikleri

Doğal Dil İşlemenin veriden bilgi çıkarımı için kullandığı metotlara Doğal Dil İşleme Teknikleri denir. Bu teknikler uygulanırken dikkat edilecek hususlardan bazıları şu şekildedir; Konuşma Dili, Yazım Dili, Sesbilim (phonology), Biçimbilim (morphology), Sözdizim (syntax) ve Anlamsallıktır (semantic).

7.2.1. Duygu Analizi

Girdi olarak kullanılan metin verisinden; olumlu, olumsuz, etkisiz ve ara değer sonuçlarının çıkarımı için kullanılan Doğal Dil İşleme tekniğidir.

Şekil 7.2. Duygu Analizi Temsili Gösterimi

Duygu Analizi, metindeki fikirlerin, duyguların ve öznelliğin hesaplamalı bir sonucudur.

7.2.2. Dil Çevirisi

Farklı toplum ve insan kaynaklarına ait olan yazılı veya sözlü eserlerin istenilen dilin yapı ve kurallarına uygun şekilde çevrilmesine Dil Çevirisi denmektedir. Birçok farklı Dil Çevirisi yapan sistem günlük hayatta kullanılmaktadır. Bunlardan başlıcaları; Google Translate, Yandex Translate' dir.

Şekil 7.3. Dil Çevirisi Temsili Gösterimi

7.2.3. Konuşma Tanıma

Konuşma Tanıma; İnsanların konuşmalarındaki kelimeleri tanımlama ve bunları okunabilir metne dönüştürme işlemidir. İlk Konuşma Tanıma teknikleri oldukça sınırlı kelime havuzuna sahip olduğu için yalnızca net bir şekilde konuşulduğunda kelimeler ve cümleleri tanımlayabilirken Derin Öğrenme tabanlı geliştirilen Konuşma Tanıma yazılımları doğal konuşmaları, farklı aksanları ve çeşitli dilleri tanımlayabilmektedir.

Şekil 7.4. Konuşma Tanıma Temsili Gösterimi

Konuşma tanıma sistemleri, konuşulan kelimeleri işlemek ve yorumlamak ve bunları metne dönüştürmek için bilgisayar algoritmalarını kullanır. Bir yazılım programı, bir mikrofonun kaydettiği sesi, şu adımları izleyerek bilgisayarların ve insanların anlayabileceği bir yazılı dile dönüştürür; Sesi analiz etmek, Bilgisayar tarafından okunabilir formatta sayısallaştırmak, En uygun metin temsiliyle eşleştirmek için uygun algoritmayı seçmek.

7.2.4. Yazı Denetimi

Yazı Denetimi, yazımı düzeltmek için kullanılan tekniktir. Kelime işlem programlarında, e-posta programlarında, cep telefonlarında, mesajlaşma uygulamalarında, bloglar ve forumlar gibi diğer çeşitli tüm uygulamalarda bulunur. Yazım Denetimi, sözcüklerin yanlış yazılıp yazılmadığını kullanıcıya bildirir ve kullanıcı yazıyı yazarken yanlış yazılan sözcükleri düzeltir, yanlış yazılmış sözcükler için tüm belgede arama yapılmasına olanak sağlar.

En popüler Yazı Denetim aracı Microsoft Office Word uygulamasında yer almaktadır. Kullanıcı tarafından istenen tüm denetimler seçime özeldir. Kullanıcı

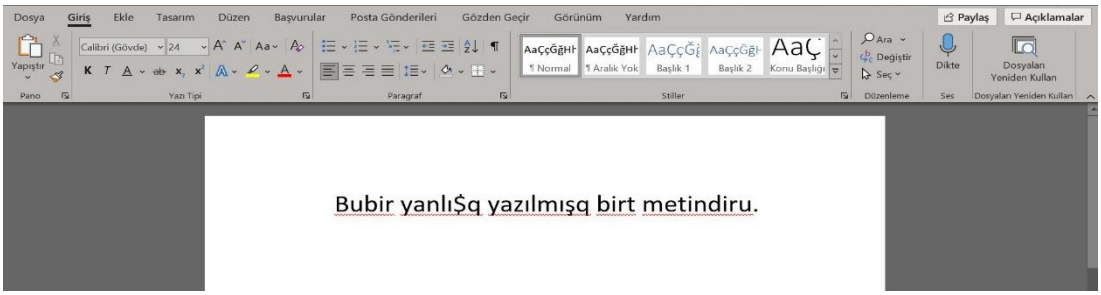
yazısını yazarken hatalı yazılan kelimeleri kırmızı çizgi işaretlerken, dilbilgisi hataları için yeşil ve bağlamsal yazım hataları için mavi gibi renk tonlarıyla kelimelerin altını çizer.

Şekil 7.5. Yazım Denetimi Temsili Gösterimi

7.2.5. Metin Özetleme (Text Summarization)

Metin özetleme, daha uzun bir metin belgesinin kısa, doğru ve akıcı bir özetini oluşturma tekniğidir. Hem ilgili bilgilerin daha iyi keşfedilmesine yardımcı olmak hem de ilgili bilgileri daha hızlı tüketmek için sürekli artan miktarda metin verisini ele alarak Metin Özetleme yöntemlerine büyük ölçüde ihtiyaç vardır.

Web sayfaları, haber makaleleri, durum güncellemeleri, bloglar ve çok daha fazlasını içeren internet dünyası düşünüldüğünde her geçen gün artan çok büyük miktarda metin verisinden söz edilmektedir. Çoğu yapılandırılmamış bu verilerde gezinmek için yapılacak en iyi şey metnin özetini çıkartmaktır. Metin verisinden özet çıkartmanın yani Metin Özetlemenin başlıca yararları şu şekildedir; Okuma süresinin kısalması, belge seçim süresinin kısalması, indeksleme etkinliğinin artması, insan özetlerine kıyasla daha az önyargıdır.



Şekil 7.6. Metin Özetleme Temsili Gösterimi

7.2.6. Bilgi Çıkarımı

Bilgi Çıkarımı, metin verilerinden önceden belirlenmiş bilgilerin çıkarılması tekniğidir. En yaygın örneklerden biri, gelen e-postalardaki tarihsel verilerin e-posta konu ve içeriği ile ilişkilendirilerek olası olayları takviminize eklemesidir. Buna ek kullanım alanları olarak; Yasal işlemler, tıbbi kayıtlar, sosyal medya etkileşimleri ve akışları, online haberler, kurumsal raporlar verilebilir.

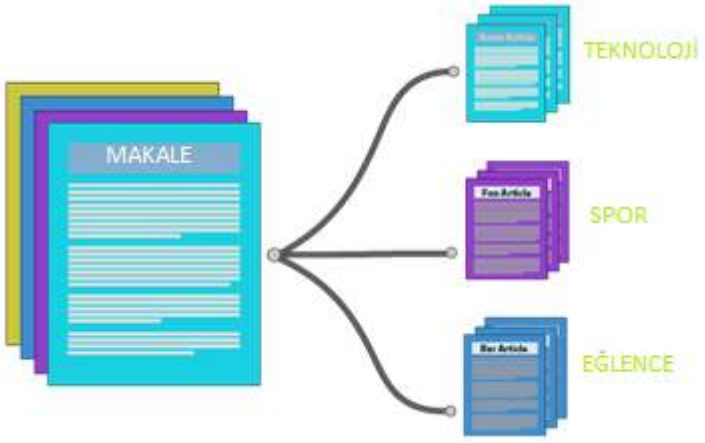
Şekil 7.7. Bilgi Çıkarımı Temsili Gösterimi

7.2.7. Metin Sınıflandırma

Metin Sınıflandırma, doğal dil metinlerini önceden tanımlanmış sınıf kümelerinden ilgili olana etiketleme tekniğidir. Layman terimiyle Metin Sınıflandırma; Yapılandırılmamış metinden genel etiketleri çıkarma işlemidir. Bu genel etiketler bir dizi önceden tanımlanmış kategoriden gelir. İçeriğinizi ve ürünlerinizi kategoriler halinde sınıflandırmak, kullanıcıların web sitesi veya uygulama içinde kolayca arama yapmasına ve gezinmesine yardımcı olur.

Uzun zamandır Metin Sınıflandırma tekniklerini gündelik hayatlarımızda farkında olmasak da kullanmaktayız. Örnek olarak kütüphanelerdeki kitapların sınıflandırılması, haberlerdeki makalelerin bölümlere ayrılması, gereksiz maillerin

ayıklanması, dolandırıcılık, küfür ve çevrimiçi kötüye kullanım tespitleri, duygu analizleri, dil algılama işlemleri, müşteri geri bildiriminde trendleri tespit etme işlemleri verilebilecek örneklerden sadece birkaçıdır.



Şekil 7.8. Metin Sınıflandırma Temsili Gösterimi

8. DERİN ÖĞRENMEYLE DOĞAL DİL İŞLEME MODELLERİ

8.1. WORD2Vec

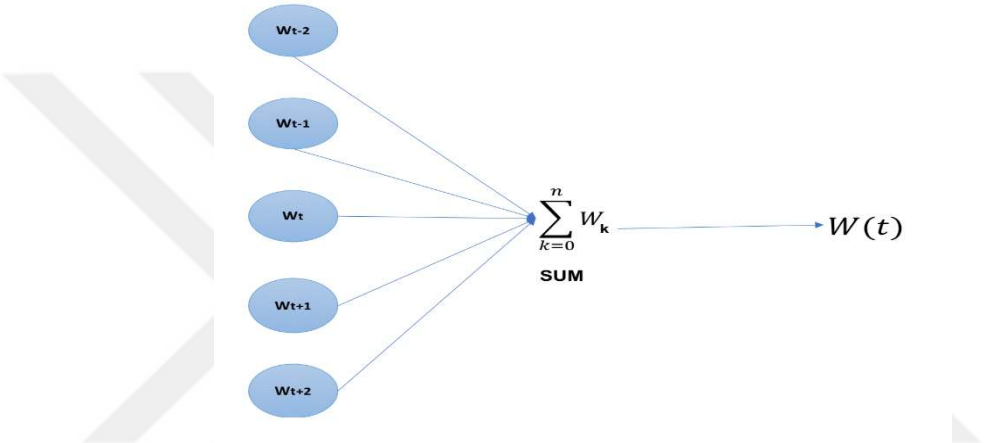
Word2Vec; kestirim mantığıyla çalışan kelime temsil yöntemdir. 2013 senesinde Google şirketi mühendis ve araştırmacısı olan Thomas Mikolov ve ekip arkadaşlarıyla beraber özünde Yapay Sinir Ağı ve iki farklı D.Ö. modeli kullanarak kelimelerin eğitiminin tamamlanmasını amaçlayarak geliştirilmiştir [54]. Kelimelerin vektörel ifadeleri kelimeler üzerinde matematiksel ve istatistiksel çıkarımların yapılmasını sağlamaktadır. Edinilen vektör uzayları ve sözcükler arasında ilişkilendirmeler yapılabilecektir. Bu ilişkilendirmeler kelime vektörleri arasındaki mesafenin hesaplanmasıyla mümkün olmaktadır. Word2Vec modeli aslında insanların düşünerek, yorumlayarak cümledeki eksik kelimeyi tamamlaması olayını makinelerin yapması hedefidir. Birtakım modeller, matematiksel ve istatistiksel çıkarımlardan oluşan Word2Vec modeli cümledeki veya kullanılan metin verisindeki her sözcüğe uygulanmalıdır. Böylece her kelime vektörü vektör uzayında daha iyi temsil edilecek ve model daha iyi sonuçlar verecektir.

Şekil 8.1. Word2Vec Örneği Gösterimi

Özünde Word2Vec, Kelime Temsil yöntemidir. Kelime Temsil (Word Embedding) yöntemleri bir dil modelleme tekniğidir. Kelimeleri veya cümleleri vektörize edip yani sayısallaştırıp metin verisinin Vektör Uzayında temsil edilmesini sağlar. Kelime Temsilleri Yapay Sinir Ağları veya olasılık modelleri gibi yöntemler kullanılarak elde edilebilir. Word2vec modeli, kelime temsillerini oluşturmak için iki farklı mimariye sahiptir.

8.2. CBOW(Continuous Bag of Words)

CBOW modeli, kelimelerin bağlamını anlamaya çalışır ve bunu girdi olarak alır. Daha sonra bağlamsal olarak doğru olan kelimeleri tahmin etmeye çalışır. Girdi olarak kullanılan kelimeler için One Hot Encoding *(OHE)* işlemi uygulanır. OHE, kategorik parametlerin binary şekilde gösterilmesi demektir. OHE için önce parametreler nümerik ifadelere dönüştürülür. OHE işlemi uygulanan kelimeler ile hedef kelime arasında hata oranları ölçülür. Bunu yapmak, çıktıyı en az hatayla kelimeye dayalı olarak tahmin etmemize yardımcı olacaktır [55].



Şekil 8.2. CBOW Mimarisi Gösterimi

Şartlı Olasılık Formülü kullanılarak vektörler arasındaki benzeme oranları hesaplanır:

$$P(W_t+j|\ W_t) = \exp(U_{t+j}^{T}\ V_{wt}) / \sum_{w=1}^{v} \exp(U_{wt}^{t})$$

Şekil 8.3. Şartlı Olasılık Formülü

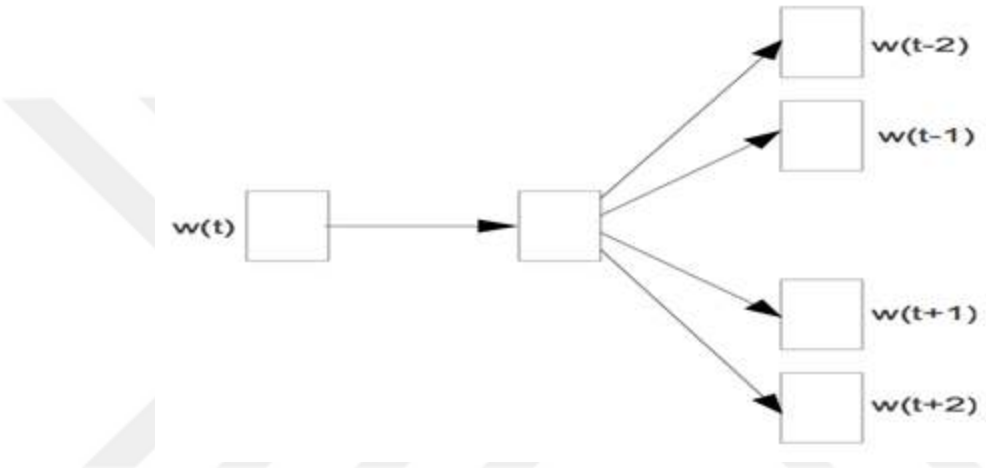
Eğitimi optimize etmek için kullanılan hata fonksiyonu:

$$J = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m} logP(wt+j \mid wt)$$

Şekil 8.4. CBOW için optimize edilecek hata fonksiyonu

8.3. Skip Gram

Skip Gram modeli kelimeleri pencere boyutunun merkezinde kabul ederek girdi parametresi olarak alır ve pencere boyutunun merkezinde var olmayan kelimeleri çıktı şeklinde kestirmeye çalışmaktadır. Bahsedilen konu ise şekilde gösterilmek istenmiştir. Skip Gram ve CBOW modelleri arasındaki fark Skip Gram dil modelinin CBOW modelinin tam zıttı olmasıdır. NN' de girdi ve çıktı yerleri değişmektedir [56].



Şekil 8.5. CBOW optimizasyonu için kullanılan hata fonksiyonu gösterimi

Eğitim esnasında Skip Gram optimizasyonu için kullanılacak olan hata fonksiyonu şu şekildedir:

$$J = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m} logP(wt \mid wt+j)$$

Şekil 8.6. CBOW optimizasyonu için kullanılan hata fonksiyonu

Word2Vec dil modelindeki sağlanan kâr, sözcüklerin cümle içindeki konumlarına göre şartlı olasılık fonksiyonu işleme mantığından ötürü sözcükler içindeki yakınlıkların yitirilmemesidir.

8.4. FastText

2016 yılında Facebook Yapay Zekâ Araştırmacıları tarafından geliştirilen kullanıcıların metin temsillerini ve metin sınıflandırıcılarını öğrenmelerini sağlayan

açık kaynaklı, ücretsiz kütüphanedir [57]. FastText metin veya kelimeleri herhangi bir dil ile ilgili görevde kullanılabilecek sürekli vektörlere dönüştürür. Her bir kelimeyi YSA' na bir bir parametre olarak geçirmek yerine kelime öbeklerine kadar N-gram yardımıyla ayrıştırır. Kullanılan n tekrar sayısını bildirmektedir.

Bu kütüphanede öğretici olarak Skip Gram ve CBOW modelleri kullanılmaktadır. Kelime vektörleri kelimeye ait N-Gram vektörlerinin toplamından meydana gelir. FastText modeli eğitim setinde olmayan kelimelerin temsil edilmesine olanak sağlamaktadır. Daha az kullanılan kelimelerin N-Gram vektör sayıları da daha az çıkacağı için kelime temsil doğruluk oranları artacaktır. N-Gram miktarı sözcük adetinden çok fazla olacağından learning süreside uzayacaktır.

8.5. Glove (Global Vectors for Word Representation)

Gözetimsiz Öğrenme algoritmalarının ana fikrinde verilerin matematiksel hesaplamaları bulunmaktadır. Diğer çeşitli modeller insan düzeyinde anlamı olan verileri bulabilir fakat kelimelerin birarada kullanılma durumlarını kaçırırlar. Bunlar gibi modeller için anlam kaygısı taşınmamaktadır [58].

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T w_j + b_i + b_j - \log X_{ij})^2$$

Şekil 8.7. Glove Modeli Formülü

Formüldeki "wi" Glove modelindeki matrisin toplam satır sayısını, "wj" ise sütun sayısını "f" ağırlık fonksiyonunu ve "V" vektör matrisinin boyutunu göstermektedir.

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

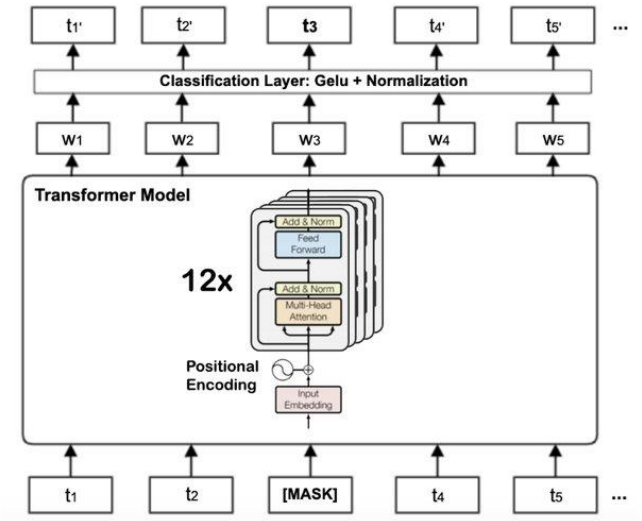
```
[('tahran', 0.6414337754249573),
 ('bağdat', 0.5921595096588135),
 ('ırak', 0.5417447090148926),
 ('kahire', 0.5351825952529907),
 ('erbil', 0.5221700072288513),
 ('suriye', 0.5216017365455627),
 ('washington', 0.5198656320571899),
 ('başkenti', 0.5075976252555847),
 ('arap', 0.4841397702693939),
 ('york', 0.47895127534866333)]
```

Şekil 8.8. Eğitilmiş Glove Modeli Ülke-Başkent İlişkisi

8.6. BERT (Bidirectional Encoder Representations from Transformers)

Transformatör kavramı ilk olarak Google tarafından 2017' de tanıtıldı. Tanıtıldıkları sırada, dil modelleri Doğal Dil İşleme görevlerini yerine getirmek için öncelikle Tekrarlayan Sinir Ağlarını (RNN) ve Evrişimsel Sinir Ağlarını (CNN) kullanıyordu. Bu modeller yetkin olmasına rağmen, Transformatör modeli önemli bir gelişme olarak kabul edilmektedir çünkü RNN ve CNN mimarileri herhangi bir sabit sırada işlenecek veri dizilerine ihtiyaç duymazlar yani sıralı veri beklerler. Transformatörler verileri herhangi bir sırayla işleyebildiğinden, var olmalarından önce mümkün olandan daha fazla miktarda veri üzerinde eğitime olanak tanırlar. Bu da, piyasaya sürülmeden önce büyük miktarda dil verisi üzerinde eğitilmiş olan BERT gibi önceden eğitilmiş modellerin oluşturulmasını kolaylaştırmaktadır [31].

Metin Sınıflandırma, Duygu Analizi, Anlamsal Rol Etiketleme, Cümle Tanıma, Çok Anlamlı Kelimelerin Anlam Ayrımı gibi toplamda 11 Doğal Dil İşleme Tekniğine sahip olan BERT 2018 yılında Google Mühendisleri tarafından duyuruldu. Böylece BERT, bağlam ve çok anlamlı sözcükleri yorumlarken sınırlı yeterliliğe sahip olan Word2vec ve Glove gibi önceki dil modellerinden avantajlı konuma geçti. BERT Doğal Dil alanındaki araştırmacı bilim adamlarına göre Doğal Dil İşlemenin en büyük zorluğu olan belirsizliği etkin bir şekilde ele almaktadır. Nispeten insan benzeri bir sağduyu ile dili ayrıştırma yeteneğine sahiptir.



Şekil 8.9. Temel BERT Modeli Mimarisi

BERT, bir dilin cümlesindeki bir sonraki kelimeyi tahmin etmek için önemli kelimeleri belirlemek için 2017 yılında geliştirilen ve benimsenen çığır açan bir model olan Transformer'a dayanmaktadır. Daha küçük veri kümeleriyle sınırlı olan önceki DDİ çerçevelerini deviren Transformer, daha büyük bağlamlar oluşturabilir ve metnin belirsizliği ile ilgili sorunları çözebilir. Bunu takiben, BERT çerçevesi, derin öğrenme tabanlı DDİ görevlerinde olağanüstü performans gösterir. BERT, DDİ modelinin bir cümlenin anlamsal anlamını anlamasını sağlar.

Cümle çifti, tek cümle sınıflandırması, tek cümle etiketleme ve soru cevaplama gibi görevlerde, BERT çerçevesi oldukça kullanışlıdır ve etkileyici bir doğrulukla çalışır. BERT iki aşamalı uygulamaları içerir ayrıca denetimsiz ön eğitim ve denetimli ince ayar. MLM ve NSP konusunda önceden eğitilmiştir. MLM görevi, çerçevenin maskelenmiş belirteçlerin maskesini kaldırarak bağlamı sağ ve sol katmanlarda kullanmayı öğrenmesine yardımcı olurken; NSP görevi, iki cümle arasındaki ilişkiyi yakalamaya yardımcı olur. Çerçeve için gerekli olan teknik özellikler açısından, önceden eğitilmiş modeller Temel (12 katman, 786 gizli katman, 12 öz dikkat başlığı ve 110 m parametresi) ve Büyük (24 katman, 1024 gizli katman, 16 öz olarak mevcuttur, dikkat başlığı ve 340 m parametreleri).

BERT, bağlamı bulmak ve ilişkilendirmek için bir kelimenin etrafında birden çok yerleştirme oluşturur. BERT'nin girdi yerleştirmeleri belirteç, segment ve konum bileşenlerini içerir [60].

2018' den bu yana, BERT çerçevesinin çeşitli NLP modelleri için ve derin dil öğrenme algoritmalarında yaygın olarak kullanıldığı bilinmektedir. BERT açık kaynak olduğundan, genellikle ALBERT, HUBERT, XLNet, VisualBERT, RoBERTA, MT-DNN, vb. gibi temel çerçeveden daha iyi sonuçlar veren, kullanımda olan çeşitli varyantları vardır.

Google, BERT çerçevesini tanıtıp açık kaynaklı hale getirdiğinde, duygu analizi, birden çok anlama sahip kelimeler ve cümle sınıflandırması gibi görevleri basitleştiren 11 dilde son derece doğru sonuçlar üretti. Yine 2019' da Google, arama motorundaki arama sorgularının amacını anlamak için bu çerçeveyi kullandı. Bunu takiben, ürün incelemesine dayalı ürün tavsiyesi, kullanıcı amacına dayalı daha derin

duyarlılık analizi için SquAD, GLUE ve NQ veri kümeleriyle ilgili soruları yanıtlama gibi görevlerde yaygın olarak uygulanmaktadır.

2019'un sonunda, çerçeve, farklı Yapay Zekâ programlarında kullanılan yaklaşık 70 dil için kabul edilmiştir. BERT, insanlar tarafından konuşulan doğal dillere odaklanılarak oluşturulan DDİ modellerinin çeşitli karmaşıklıklarının çözülmesine yardımcı oldu. Büyük etiketlenmemiş veri havuzlarında eğitim almak için önceki DDİ tekniklerinin gerekli olduğu durumlarda, BERT önceden eğitilmiştir ve bağlamlar oluşturmak ve tahmin etmek için iki yönlü olarak çalışır. Bu mekanizma, verilerin sıralanmasını ve düzenli bir şekilde düzenlenmesini gerektirmeden verileri çalıştırabilen DDİ modellerinin kapasitesini daha da artırır. Buna ek olarak, BERT modeli, diziden diziye dil geliştirme ve DDA görevlerini çevreleyen DDİ görevleri için olağanüstü performans gösterir.

BERT, sıfırdan bir DİM oluşturmak yerine tek kolaylaştırıcı olarak ortaya çıkarak çok zaman, maliyet, enerji ve altyapı kaynaklarından tasarruf edilmesine yardımcı oldu. Açık kaynak olmasıyla, önceki Word2Vec ve Glove dil modellerinden çok daha verimli ve ölçeklenebilir olduğunu kanıtlamıştır. BERT, insan doğruluk seviyelerini %2 oranında geride bıraktı ve GLUE skorunda %80 ve SquAD 1.1' de neredeyse %93,2 doğruluk elde etti. BERT, herhangi bir içerik hacmine uyarlanabilirken, kullanıcı spesifikasyonuna göre ince ayar yapılabilmektedir [61].

BERT, önceden eğitilmiş dil modellerini tanıtarak DDİ görevlerine değerli bir katkı sağlarken, çoklu varyantlarının çıkmasıyla DDA görevlerini yürütmek için güvenilir bir kaynak olduğunu kanıtladı.

9. MATERYAL VE YÖNTEMLER

9.1. Kullanılan Veri Setleri

9.1.1. Haber Metni Sınıflandırma Veri Seti

Metin Sınıflandırma Türkçe dilinde çalışılmış bir konudur ve ekonomi, kültür-sanat, sağlık, siyaset, spor ve teknoloji kategorileri ile güncel TTC-3600 haber veri setini seçtik. Yazarlar, Makine Öğrenmesi sınıflandırıcılarından önce verileri işlemek için stop-word filtreleme, kök çıkarma ve özellik eleme yaklaşımlarını kullandılar. Veri setini değerlendirmek için Naif Bayes (Naive Bayes), Destek Vektör Makineleri (Support Vector Machines), J48 Ağacı (J48-tree) ve Rastegele Orman (Random Forest) kullanmışlar ve en iyi performansları için Rastgele Orman sınıflandırıcı ile %91.03 doğruluk elde etmişlerdir.

Veri seti 6 Sınıf ve toplamda 3600 örnekten oluşmaktadır.

9.1.2. Duygu Tanıma Veri Seti

Duygu Tanıma veri seti Mansur Alp Toçoğlu ve arkadaşları tarafından 2018 yılında yayınlanan “TREMO: A dataset for emotion analysis in Turkish” adlı çalışmadan alınmıştır. Bu veri seti Destek Vektör Makineleri (Support Vector Machines), J48 Ağacı ve Naive Bayes ile mutluluk, korku, öfke, üzüntü, iğrenme ve şaşkınlığı tespit etmek için kullanılmıştır.

Veri seti 6 Sınıf ve toplamda 18000 örnekten oluşmaktadır.

9.1.3. Gereksiz Kısa Mesaj Tespiti Veri Seti

Bir göndericiden birden çok kullanıcıya çoğunlukla ticari amaçlarla istenmeyen mesajlar gönderilmesi spam SMS olarak bilinir. Bu veri seti tez kapsamında kullanıcılardan gelen SMS’lerin spam olup olmadıklarını tespit etmek amacıyla kullanılmışır.

Veri seti 2 Sınıf ve toplamda 517 örnekten oluşmaktadır.

9.2. Kullanılan Veri Setleri Üzerinde Ön İşleme Adımları

Kullanılan 6 veri seti için; Noktalama İşaretleri Kaldırılması (Stop Words Removing), Simgeleştirme (Tokenization), N-Gram Gösterimi (N-Gram Representation), Köke İndirgeme (Stemming), Öznitelik Mühendisliği (Feature Engineering), TF-IDF Gösterimi (TF-IDF Representation), Sözlük Temelli İşleme (Lexicon-Based Processing) ön işleme adımları kullanılmıştır.

Tablo 9.1. Veri Setleri Üzerinde Uygulanan Ön İşleme Adımları

Ön İşleme Tekniği	NEWS	EMOT	SPAM
Noktalama İ. Kaldırma	+		
Simgeleştirme		+	+
N-Gram Gösterimi			
Köke İndirgeme	+	+	
Öznitelik Mühendisliği	+	+	+
TF-IDF Gösterimi	+	+	
Sözlük Temelli İşleme			

9.3. BERT Modeli Deneyleri

9.3.1. BERT İnce Ayar Parametre Seçimleri

BERT Modeli geliştiricileri tarafından kullanılması mutlaka tavsiye edilen parametreler şu şekildedir. Batch-Size (Yığın Sayısı), Learning Rate (Öğrenme Oranı) ve Epoch-Size (Tekrar Sayısı). Önerilen sayısal değerler ise şu şekildedir: Batch-Size için 16 ya da 32; Learning Rate için 5e-5 veya 3e-5 veya 2e-5; Epoch-Size için 2 veya 3 veya 4'tür. Bu tez kapsamında kullanacağımız değerler ise şu şekilde olmuştur. Batch-Size 16, Learning Rate 2e-5, Epoch-Size 4'tür. Deneysel çalışmamızda pre-trained BERT-base-multi-language (Yarı Eğitilmiş BERT Temel Çoklu Dil) modeli, 12 Transformatör bloğu, 768 Hidden Layer (Gizli-Ara Katman) kullanılmıştır [62].

9.3.2. Deneylerin Gerçekleştirildiği Cihaz ve Özellikleri

Deneyleri yaparken Manisa Celal Bayar Üniversitesi Hasan Ferdi Turgutlu Teknoloji Fakültesi Bilgisayar Laboratuvarında bulunan 128 GB RAM, Intel Core i9 9900X İşlemci ve 11 GB belleğe sahip GeForce RTX 2080 Ti Ekran Kartına sahip bilgisayardan yararlanılmıştır.

9.3.3 BERT Algoritması ile Çalışma ve Deney Sonuçları

Morfolojik olarak zengin dillerle ilgili çalışmaların çoğunun, yeterli bir Makine Öğrenmesi modeli elde etmek için bir şekilde bahsettiğimiz ön işleme adımlarından yararlandığı görülmektedir. Ancak, BERT bakış açısından, böyle ön işleme adımları gerekli değildir ve çoğu görevin analizi en son sonuçlarla yapılır.

NEWS (News Text Classification), EMOT (Emotion Recognition), SPAM (Spam Sms), veri setleri üzerinde BERT algoritması uygulanmış ve elde edilen ACC geleneksel Makine Öğrenmesi algoritmalarına göre çoğunlukla daha yüksek çıkmıştır. Gelenek Makine Öğrenmesi algoritmaları olarak: ANN, RF, SVM, LR, NB kullanılmıştır [63].

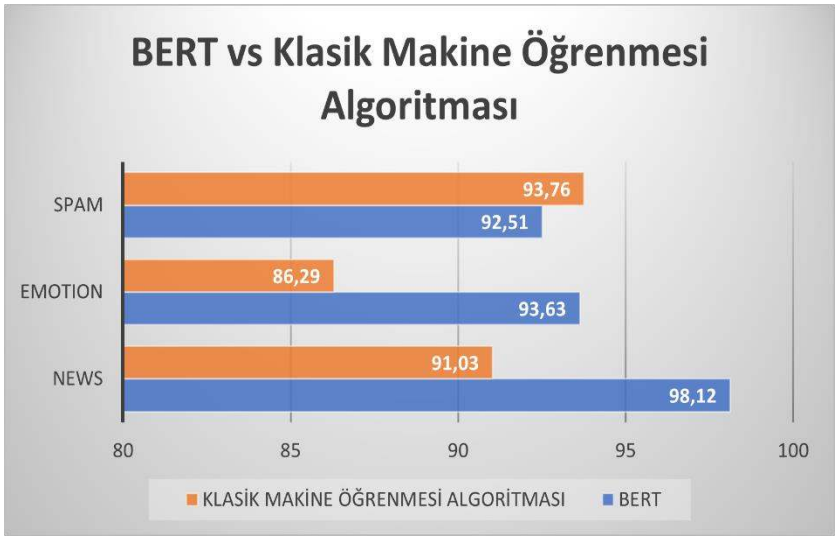
Tablo 9.2. BERT Deneylerinin Başarım Ölçüt Skorları [63]

Veri Seti	Acc	F1	Precision	Recall	Kappa
NEWS	0.9812	0.9829	0.9811	0.9848	0.9620
EMOT	0.9363	0.9401	0.9350	0.9453	0.8720
SPAM	0.9251	0.9300	0.9384	0.9217	0.8500

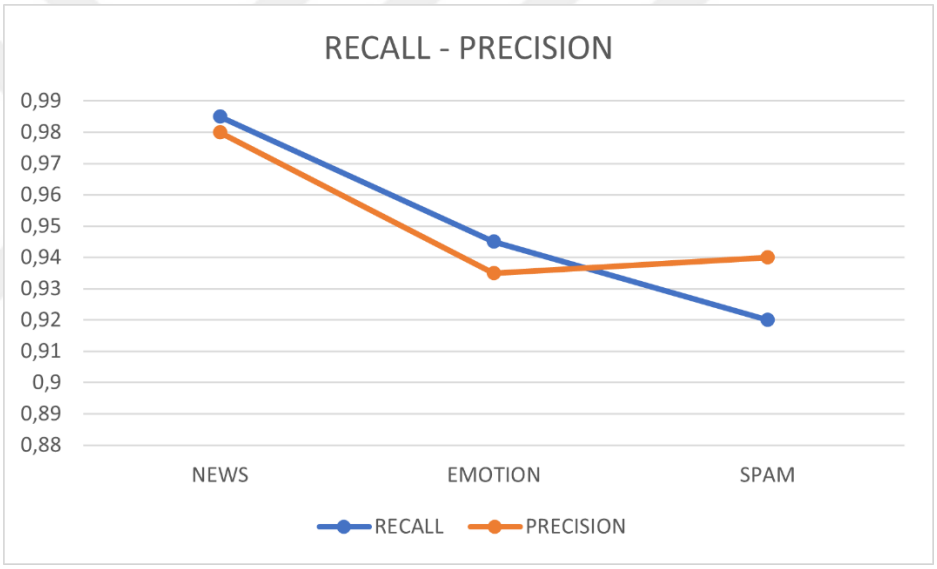
Tablo 9.3. BERT ve Makine Öğrenmesi Algoritmalarının Performans Karşılaştırması [63]

Veri Seti	M.L. Alg.	M.L.A. Acc	Bert Acc	Performans Karşılaştırması
NEWS	RF	91.03	0.98	+7.09
EMOT	SVM	86.29	0.93	+7.34
SPAM	LR	93.76	0.92	-1.25

BERT'in üç veri setinden ikisinde daha yüksek tahmin performansları ürettiğini görmekteyiz. Özellikle genel performans artışı Spam SMS verileri hariç ortalama %7,5 civarındadır. Elde edilen kıyaslamayı görsel olarak gösterecek olursak;



Şekil 9.1. BERT ve Geleneksel M.Ö. Algoritmalarının Başarım Karşılaştırması [63]



Şekil 9.2. Tüm Veri Setleri Üzerindeki Recall-Precision Eğrisi Performansı [63]

Tablo 9.4. BERT Deneyleri Kapsamında Veri Setlerine Göre Algoritma Çalışma Süresi [63]

Veri Seti	Çalışma Süresi
NEWS	2 Saat 34 Dakika
EMOT	1 Saat 46 Dakika
SPAM	23 Dakika

10. SONUÇ VE ÖNERİLER

Morfolojik olarak zengin diller, elde edilen Makine Öğrenmesi modelinin performansını etkileyen çeşitli ön işleme teknikleri kullanılarak analiz edilir. Bu adımların kombinasyonları veya varyasyonları, tahmin verimliliği açısından elde edilen sonuçları değiştirir. Bahsedilen ön işleme görevleri, yoğun yazılım tabanlı adımlara ihtiyaç duyar ve kesin olarak iyi tanımlanmış bir ardışık düzende uygulanmazlar. Öte yandan, BERT modeli, çeşitli dil görevlerini çözmek için nispeten basit bir uygulamaya sahiptir. Ayrıca BERT, çoklu dil versiyonu ile bile DDİ problemlerinde dikkat çekici sonuçlar üretebilmektedir.

Bu tez çalışmasında Türk dili alanından üç tür problemi ve üç veri setini analiz etmek için BERT modeli kullandık ve elde edilen sonuçları literatürdeki temel Makine Öğrenmesi algoritmalarının en iyi tahminleriyle karşılaştırdık. BERT' in Spam SMS veri seti dışındaki tüm verilerde tahmin performanslarını artırdığı deneysel olarak gösterilmiştir. Ayrıca BERT tahminlerindeki performans artışının temel algoritmalara göre dikkat çekici olduğu gözlemlenmiştir. Performans sayısal olarak %6,75'ten %8,60'a yükselmiştir.

Türkçe DDİ alanından çeşitli veri kümelerinin ampirik değerlendirmesi, morfolojik olarak zengin dillerin BERT çok dilli sinir modelinden iki önemli noktada yararlanabileceğini göstermektedir: İlk olarak göz korkutucu ön işleme dil adımları ihtiyacının ortadan kaldırılması ve tahminin güçlendirilmesi kabiliyetidir. Bir ince ayar görevi için bile gerekli eğitim süresi daha büyük olsa da geçen süreyi BERT' in hafif sürümlerinin veya bazılarının kullanımıyla azaltmak mümkün olabilir.

Bu çalışmada sadece birkaç Türkçe dil problemi üzerinde çalışılmış olmasına rağmen, BERT çoklu dil modelinin benzer şekilde Arapça, Çekçe, Macarca, Almanca ve Fince gibi morfolojik olarak zengin ve düşük kaynak dillerine uygulanabileceği beklenmektedir.

İlerleyen çalışmalarımızda tahmin performanslarını yüksek tutarken eğitim sürelerini azaltmak için BERT benzeri mimarilerin daha hafif sürümlerini kullanmayı planlıyoruz.

KAYNAKLAR

[1] Chomsky, N., Ferebee, A. S. Aspects of the Theory of Syntax. The Journal of Symbolic Logic. 1970, 35(1), 167-180.

[2] Coban, O., Ozyer, B., Ozyer, G. T. Sentiment analysis for Turkish Twitter feeds. 23rd Signal Processing and Communications Applications Conference. 2015, 2388–2391.

[3] Minaee, S., Cambria, E., Gao, J. Deep Learning Based Text Classification: A Comprehensive Review. Automatika. 2020, 12(2), 150-165.

[4] Demircan, M., Seller, A., Abut, F., Akay, M. F. Developing Turkish sentiment analysis models using machine learning and e-commerce data. International Journal of Cognitive Computing in Engineering. 2021, 2(1), 202–207.

[5] Nergiz, G., Safali, Y., Avaroglu, E., Erdogan, S. Classification of Turkish News Content by Deep Learning Based LSTM Using Fasttext Model. 2019 International Conference on Artificial Intelligence and Data Processing Symposium. 2019, DOI: 10.1109/IDAP.2019.8875949.

[6] Lyons, J. Natural language and universal grammar. Cambridge University Press. 1991, 290-307.

[7] Karademir, F. Türk Dilinin Tarihî Dönemlerini Adlandirma Sorunu. Uluslararası Türkçe Edebiyat Kültür Eğitim (TEKE) Dergisi. 2016, 5(2), 545-545.

[8] Berbercan, M. T. Türk Yazi Dilinin Tarihî Dönemleri ve Orta Türkçenin Yeri Meselesi. Journal of History School. 2014, 765–783.

[9] Bingol, Y. Language, Identity and Politics in Turkey: Nationalist Discourse on Creating a Common Turkic Language. Alternatives: Turkish Journal of International Relations. 2009, 8(2), 75-84.

[10] Gerhard, D. The Older Mongolian Layer in Ancient Turkic. Türk Dilleri Araştırmaları 3. 1993, 79-86.

[11] Selvelli, G. The Clash Between Latin And Arabic Alphabets Among The Turkish Community In Bulgaria In The Interwar Period. Journal of Balkan Research Institute Cilt. 2018, 7(1), 367–390.

[12] Aytürk, I. The First Episode of Language Reform in Republican Turkey: The Language Council from 1926 to 1931. Journal of the Royal Asiatic Society. 2008, 18(3), 281-297.

[13] Doğan, F., Tükoğlu, İ. Derin Öğrenme Algoritmalarının Yaprak Sınıflandırma Başarımlarının Karşılaştırılması. Sakarya University Journal Of Computer And Information Sciences. 2018, 1(1), 45-46.

[14] Açıdan, E., Kaya, İ. Critical Examination of the Alphabet and Language Reforms Implemented in the Early Years of the Turkish Republic Türkiye Cumhuriyeti'nin İlk Yıllarında Yapılan Alfabe ve Dil Devrimlerinin. Journal of Social Studies Education Research Sosyal Bilgiler Eğitimi Araştırmaları Dergisi. 2011, 59–82.

[15] Burak, S. The Reflections Of Turkish Revolution On The Language Between 1930-1938. Atatürk ve Türkiye Cumhuriyeti Tarihi Dergisi. 2019, 2(4), 127-154.

[16] Aytürk, I. The First Episode of Language Reform in Republican Turkey: The Language Council from 1926 to 1931. Journal of the Royal Asiatic Society. 2008, 18(3), 281-292.

[17] Atalay, M., Çelik, E. Büyük Veri Analizinde Yapay Zekâ ve Makine Öğrenmesi Uygulamaları. Artificial Intelligence And Machine Learning Applications In Big Data Analysis. 2017, 9(22), 155-172.

[18] İnternet, Kızrak, M. A. Derine Daha Derine: Evrişimli Sinir Ağları. https://ayyucekizrak.medium.com/deri%CC%87ne-daha-deri%CC%87ne-evri%C5%9Fimli-sinir-a%C4%9Flar%C4%B1-2813a2c8b2a9

[19] Borandağ, E., Yücalar, F., Akarsu, K. Software Fault Prediction in Object Oriented Software Systems Using Ensemble Classifiers. Celal Bayar Üniversitesi Fen Bilimleri Dergisi. 2018, 14(3), 297–302.

[20] Tzanis, G., Katakis, I., Partalas, I., Vlahavas, I. Modern Applications of Machine Learning. Artificial Intelligence And Machine Learning Applications In Big Data Analysis. 2022, 7(2), 44-63.

[21] Liu, Y., Zhou, Y., Wen, S., Tang, C. A Strategy on Selecting Performance Metrics for Classifier Evaluation. International Journal of Mobile Computing and Multimedia Communications. 2014, 6(4), 20–35.

[22] Bishop, C. M., Pattern Recognition and Machine Learning. Information Science and Statistics. 2006, 3(2), 12-23.

[23] Walker, S., Khan, W., Katic, K., Maassen, W., Zeiler, W. Accuracy of different machine learning algorithms and added-value of predicting aggregated-level energy performance of commercial buildings. Energy and Buildings. 2020, 209(1), 189-205.

[24] Vabalas, A., Gowen, E., Poliakoff, E., Casson, A. J. Machine learning algorithm validation with a limited sample size. Plos One. 2019, 14(11), 43-65.

[25] Nasteski, V. An overview of the supervised machine learning methods. Horizons. 2017, 4(1), 51-62.

[26] Jiang, T., Gradus, J. L., Rosellini, A. J. Supervised Machine Learning: A Brief Primer. Behavior Therapy 2002, 51(5), 675–687.

[27] Edelman, E. R., Van Kuijk, S. M. J., Hamaekers, A. E. W., De Korte, M. J. M., Van Merode, G. G., Buhre, W. F. F. A. Improving the prediction of total surgical procedure time using linear regression modeling. Frontiers in Medicine. 2017, 4(2), 85-97.

[28] Gupta, A., Sharma, A., Goel, A. Review of Regression Analysis Models. Ijert Science and Engineering, 2022, 1(3), 78-92.

[29] Rivest, M., Vignola-Gagné, E., Archambault, É. Article-level classification of scientific publications: A comparison of deep learning, direct citation and bibliographic coupling. Plos One. 2021, 16(5), 41-53.

[30] Gulzat, T., Lyazat, N., Siladi, V., S. Gulbakyt, Maxatbek, S. Research on predictive model based on classification with parameters of optimization. Neural Network World. 2002, 30(5), 295–308.

[31] Romano Jr, N. C., Fjermestad, J. Electronic Commerce Customer Relationship Management (ECCRM). International Journal of Electronic Commerce. 2011, 6(2), 34-42.

[32] Akpan, U. I., Starkey, A. Review of classification algorithms with changing inter-class distances. Machine Learning with Applications. 2021, 4(2), 10-16.

[33] Van Engelen, J. E., Hoos, H. H. A survey on semi-supervised learning. Machine Learning. 2020, 109(2), 373–440.

[34] Padmanabha, Y., Viswanath, P., Reddy, B. Semi-supervised learning: a brief review. International Journal of Engineering & Technology. 2018, 7(1), 81-87.

[35] Chapelle, O., Schölkopf, B., Zien, A. Semi-Supervised Learning edited by. Neural Network World. 2011, 42(3), 76-92.

[36] Ouali, Y., Hudelot, C., Tami, M. An Overview of Deep Semi-Supervised Learning. Frontiers in Medicine. 2017, 13(3), 27-38.

[37] Botvinick, M., Ritter, S., Wang, J. X., Kurth-Nelson, Z., Blundell, C., Hassabis, D. Reinforcement Learning, Fast and Slow. Trends in Cognitive Sciences. 2019, 23(5), 408–422.

[38] Labash, A., Aru, J., Matiisen, T., Tampuu, A., Vicente, R. Perspective Taking in Deep Reinforcement Learning Agents. Frontiers in Computational Neuroscience. 2020, 14(4), 69-80.

[39] Nasir, Y., He, J., Hu, C., Tanaka, S., Wang, K., Wen, X. H. Deep Reinforcement Learning for Constrained Field Development Optimization in

Subsurface Two-phase Flow. Frontiers in Applied Mathematics and Statistics. 2021, 7(2), 54-65.

[40] İnternet, Du, Y. Chen, W. Branching Reinforcement Learning. Feb. 2022, http://arxiv.org/abs/2202.07995

[41] İnternet, Halperin, I., Liu, J., Zhang, X. Combining Reinforcement Learning and Inverse Reinforcement Learning for Asset Allocation Recommendations. http://arxiv.org/abs/2201.01874

[42] İnternet, Dickson B. Reinforcement learning improves game testing, AI team finds | VentureBeat. https://venturebeat.com/2021/10/07/reinforcement-learning-improves-game-testing-ai-team-finds/

[43] Khanum, M., Mahboob, T., Imtiaz, W., Abdul Ghafoor, H., Sehar, R. A Survey on Unsupervised Machine Learning Algorithms for Automation, Classification and Maintenance. International Journal of Computer Applications. 2015, 119(13), 34–39.

[44] Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S., Dehmer, M. An Introductory Review of Deep Learning for Prediction Models With Big Data. Frontiers in Artificial Intelligence. 2020, 3(4), 33-49.

[45] Grossi, E. Buscema, M. Introduction to artificial neural networks. European Journal of Gastroenterology and Hepatology. 2007, 19(12), 1046–1054.

[46] Sherstinsky, A. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. Physica D: Nonlinear Phenomena. 2018, 404(2), 23-36.

[47] Hochreiter, S. Schmidhuber, J. Long Short-Term Memory. Neural Computation. 1997, 9(8), 1735–1780.

[48] Thireou, T. Reczko, M. Bidirectional long short-term memory networks for predicting the subcellular localization of eukaryotic proteins. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2007, 4(3), 441–446.

[49] İnternet. Adaloglou, N. How Transformers work in deep learning and NLP: an intuitive introduction | AI Summer. https://theaisummer.com/transformer/ (accessed Mar. 29, 2022)

[50] İnternet. Transformers for Machine Learning: A Simple Explanation | by François St-Amant | Towards Data Science. https://towardsdatascience.com/transformers-a-simple-explanation-5b17aeb587e6

[51] Nayak, T., Ng, H. T. Effective Modeling of Encoder-Decoder Architecture for Joint Entity and Relation Extraction. Net Decoding For Java. 2022, 3(5), 52-73.

[52] Brown, T. B. Language Models are Few-Shot Learners. OpenAI. 2020, 4(1), 22-97.

[53] Nadkarni, P. M., Ohno-Machado, L., Chapman, W. W. Natural language processing: an introduction. 2011, 18(5), 544-551.

[54] Jang, B., Kim, I., Kim, J. W. Word2vec convolutional neural networks for classification of news articles and tweets. Plos One. 2019, 14(8), 92-107.

[55] Fesseha, A., Xiong, S., Emiru, E. D., Diallo, M., Dahou, A., Text classification based on convolutional neural networks and word embedding for low-resource languages: Tigrinya. Information (Switzerland). 2021, 12(2), 1-17.

[56] İnternet. Lendave, V. Guide To Word2vec Using Skip Gram Model. https://analyticsindiamag.com/guide-to-word2vec-using-skip-gram-model

[57] Amalia, A., Sitompul, O. S., Nababan, E. B., Mantoro, T. An Efficient Text Classification Using fastText for Bahasa Indonesia Documents Classification. 2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics. 2020, 69–75.

[58] Pennington, J., Socher, R., Manning, C. D. GloVe: Global Vectors for Word Representation. OpenAI. 2022, 4(7), 115-142.

[59] İnternet. Devlin, J., Chang, M.-W., Lee, K. A. I. Language, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.". https://github.com/tensorflow/tensor2tensor

[60] İnternet. BERT Explained: State of the art language model for NLP | by Rani Horev | Towards Data Science. https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

[61] Qasim, R., Bangyal, W. H., Alqarni, M. A., Almazroi, A. A. A Fine-Tuned BERT-Based Transfer Learning Approach for Text Classification. Journal Of Healthcare Engineering. 2022, 1(3), 23-38.

[62] Szczepański, M., Pawlicki, M., Kozik, R., Choraś, M. New explainability method for BERT-based model in fake news detection. Scientific Reports. 2021, 11(4), 245-280.

[63] Özçift, A., Akarsu, K., Yumuk, F., Söylemez, C. Advancing natural language processing (NLP) applications of morphologically rich languages with bidirectional encoder representations from transformers (BERT): an empirical case study for Turkish. Automatika. 2021, 62(2), 226–238.