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 SİBER ZORBALIK TESPİTİ DERİN ÖĞRENME MODELİ GELİŞTİRİLMESİ

Fatma ÖZ

Danışman
Doç. Dr. Akın ÖZÇİFT

MANİSA–2022

İÇİNDEKİLER

Sayfa

SİMGELER VE KISALTMALAR DİZİNİ

ML Machine Learning (Makine Öğrenmesi)

ANN Artificial Neural Network (Yapay Sinir Ağları)

NLP Natural Language Processing (Doğal Dil İşleme)

NLU Natural Language Understanding (Doğal Dil Anlama)

NLG Natural Language Generation (Doğal Dil Üretimi)

KNN K Nearest Neighbor (En Yakın Komşular)

RNN Recurent Neural Networks (Yinelemeli Sinir Ağları)

BRNN Bidirectional Recurent Neural Networks (Çift Yönlü Yinelemeli Sinir Ağları)

QA Qestion Answering(Soru Yanıtlama)

NER Named Entity Recognition (Adlandırılmış Varlık Tanımlama)

LSTM Long Short Term Memory(Uzun Kısa Vadeli Bellek Ağları)

BLSTM Bidirectional Long Short Term Memory(Çift Yönlü Uzun Kısa Vadeli Bellek Ağları)

MLM Masked Language Modelling (Maskeli Dil Modelleme)

NSP Next Sentence Prediction (Sonraki Cümle Tahmini)

SVM Support Vector Machine (Destek Vektör Makineleri)

MCC Matthews Correlation Coefficient (Matthew Korelasyon Katsayısı)

NB Naive Bayes

MAP Maksimum A Posteriori

ŞEKİLLER DİZİNİ

TEŞEKKÜR

Çalışmamın her aşamasında bana destek olan, bilgi ve deneyimleri ile yol gösteren bilgi ve tecrübesi ile lisansüstü öğrenim hayatımın tüm zorlu aşamalarında yardımcı olan, tecrübeleri ile beni aydınlatan ve desteğini hiç eksik etmeyen, kendisini tanımaktan büyük onur duyduğum sevgili danışman hocam Sayın Doç. Dr. Akın ÖZÇİFT'e, yürekten teşekkür ederim.

Fatma ÖZ
Manisa, 2022

ÖZET

Yüksek Lisans Tezi

Manisa Celal Bayar Üniversitesi
Fen Bilimleri Enstitüsü
Yazılım Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Akın Özçift

Gelişen ve değişen teknoloji ile internet kullanımının yaygınlaşması ve sosyal medya platformlarının kullanımının artması iletişimi, bilgi paylaşımını güçlendirmekle birlikte, kişileri sürekli rahatsız etme, kişilerle alay etme, tehdit, dedikodu yayma, internet üzerinden kişiye hakaret etme gibi ilişkisel saldırı davranışlarını içeren siber zorbalık olarak tanımlanan eylemin meydana gelmesine sebep olmuştur. Siber zorbalığın insan üzerinde birçok olumsuz etkisi bulunmaktadır. Bilim insanları ve araştırmacılar, günden güne artan bu eylemi tespit etmek, tespit eden sistemler geliştirmek, siber zorbalığı önlemek amacıyla çeşitli çalışmalar yapmışlardır. Bu çalışmalardan bir tanesi de Doğal Dil İşleme ile Makine Öğrenmesi algoritmaları kullanarak Siber Zorbalık tespitinin yapılmasıdır. Siber Zorbalık tespiti için kullanılan yaygın yöntemlerden biri Makine Öğrenmesi algoritmalarıdır. Girdiler, algoritmalar tarafından işleme tutulmadan önce ön işleme adımlarından geçmektedir. Ön işleme adımlarının fazlalığı ve karmaşıklığı Türkçe gibi morfolojik açıdan zengin olan dillerde Makine Öğrenmesi algoritmalarının performansını olumsuz yönde etkilemektedir.

Doğal Dil İşleme için Google geliştiricileri tarafından yayınlanan BERT algoritması ve ön eğitimli bir algoritma olup, ön işleme adımlarına ihtiyaç duymamaktadır. Yukarıda belirtilen sebeplerden dolayı, BERT algoritmasının Makine Öğrenmesi algoritmalarına karşı Doğal Dil İşleme çalışmalarında daha iyi sonuçlar vereceği düşünülmektedir. Bu çalışmada Türkçe siber zorbalık veri seti üzerinde Makine Öğrenmesi algoritmaları ve BERT algoritmasının performansı karşılaştırmalı sonuçlarla değerlendirilmiştir.

Anahtar Kelimeler: NLP, BERT, ön eğitim, Makine Öğrenmesi, Siber Zorbalık

2021, 68 sayfa

ABSTRACT

M.Sc. Thesis

Fatma ÖZ

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

Supervisor: Assoc. Doç. Dr. Akın Özçift

With the developing and changing technology, the widespread use of the internet and the increase in the use of social media platforms strengthen communication and information sharing. caused to occur. Cyberbullying has many negative effects on people. Scientists and researchers have carried out various studies in order to detect this increasing action day by day, to develop detection systems, and to prevent cyberbullying. One of these studies is the detection of Cyber Bullying using Natural Language Processing and Machine Learning algorithms. One of the common methods used for cyberbullying detection is Machine Learning algorithms. Inputs go through preprocessing steps before being processed by algorithms. The redundancy and complexity of preprocessing steps negatively affect the performance of Machine Learning algorithms in morphologically rich languages such as Turkish.

BERT algorithm published by Google developers for Natural Language Processing and it is a pre-trained algorithm and does not need pre-processing steps. For the reasons mentioned above, it is thought that BERT algorithm will give better results in Natural Language Processing studies against Machine Learning algorithms. In this study, the performance of Machine Learning algorithms and BERT algorithm on Turkish cyberbullying dataset was evaluated with comparative results.

Keywords: NLP, BERT, pre-trained, Machine Learning, Cyberbullying

2022, 67 pages

1. GİRİŞ

İnternet teknolojisinin gelişmesiyle birlikte insanlar arasındaki iletişim ve ilişki sanal ortama taşınmaya başlamıştır. Milyonlarca insan zamanını internet ortamında sosyal ağlarda geçirmektedir. Sosyal ağlar, herkesle, herhangi bir zamanda ve aynı anda birçok kişi ile iletişim kurma ve bilgi paylaşma yeteneğine sahiptir. Sosyal ağların popülerliğinin artmasında dolayı sanal ortamda milyonlarca kişisel bilgi (fotoğraf, video, mesajlar) bulunmaktadır. Sosyal ağların kullanımı ve sanal ortamdaki bilgi artışı, bilinmeyen veya algılaması zor, pek çok tehlikeyi de beraberinde getirmiştir. Bu tehlikelerden biri olan Siber Zorbalık, karalayıcı ifadeler içeren mesajlar göndermek veya diğer kişi veya kişilere çevrim içi topluluğun geri kalanının önünde sözlü olarak zorbalık yapmak anlamına gelir [1]. Siber Zorbalık, cep telefonlarının, video oyun uygulamalarının veya metin, fotoğraf veya video göndermenin başka bir kişiyi kasten yaraladığı veya utandırdığı durumlarda çevrim içi olarak kullanılabilir.

Siber zorbalık günün her saatinde, haftasında gerçekleşebilir ve internet üzerinden her yerde herkese ulaşabilirsiniz. Siber zorbalığın metinleri, fotoğrafları veya videoları gizli bir şekilde yayınlanabilir. Twitter, Instagram, Facebook, YouTube, Snapchat, Skype gibi çeşitli sosyal medya platformları internetteki en yaygın zorbalık siteleridir. Çevrimiçi sosyal ağlar, siber zorbaların daha önce erişilemeyen yerlere ve ülkelere erişmesini sağlamaktadır. Siber zorbalık, kurbana internetin sadece bir tık ötede olduğu için her yerde saldırıya uğradığını hissettirir. Mağdur üzerinde zihinsel, fiziksel ve duygusal etkiler yaratır. Bu sebeplerden dolayı, Siber Zorbalık bağlamını analiz etme ve inceleme ihtiyacı doğmuştur.

Siber zorbalık, çoğunlukla sosyal medyada metin veya görseller şeklinde gerçekleşir. Zorbalık içeren metin, zorbalık içermeyen metinden ayırt edilebiliyorsa, sistem buna göre hareket edebilir. Etkili bir siber zorbalık tespit sistemi, bu tür saldırılara karşı koymak ve siber zorbalık vakalarının sayısını azaltmak için sosyal medya internet siteleri ve diğer mesajlaşma uygulamaları için faydalı olabilir. Siber zorbalık tespit sisteminin amacı, siber zorbalık metnini tespit etmek ve anlamını da dikkate almaktır. Sistem önce belirli bir metnin çeşitli yönlerini analiz eder ve ardından metnin bağlamını bulmak için önceki bilgileri veya görselleri uygular. Böyle bir metne etkili ve verimli bir şekilde erişebilecek kişiselleştirilmiş bir sistem oluşturmaya

ihtiyaç vardır. Siber Zorbalık tespitinin yapılabilmesi için, metinlerde kullanılan insan dilinin makineler tarafından anlaşılır hale getirilmesi gerekmektedir. İnsan dilinin, makineler tarafından anlaşılabilir, yorumlanabilir ve analiz edilebilir hale getirilmesi, Yapay Zekanın bir alt kategorisi olan Doğal Dil İşleme ile mümkün olmaktadır. İncelenecek olan metinler üç ana başlık altında analiz edilmektedir.

Morfolojik analiz; doğal dil işlemenin ana süreçlerinin başlangıcıdır. Köklerin analizi olarak da ifade edilmektedir. Bu aşamada verilen cümle boşluklar dikkat alınarak kelime ayrılmaktadır. Daha sonra her bir kelime köklerine göre ve aldıkları eklere ayrılmaktadır.

Söz dizimsel analiz; morfolojik analiz sonuç çıktıları kullanılarak cümle içerisindeki kelimelerin ve kelimeler arasında ilişkinin analizinin yapıldığı süreçtir. Cümle içerisindeki kelimelerin dizilişinin incelendiği aşama olarak ifade edilebilmektedir. Özne, yüklem, nesne ve benzeri cümle ögelerinin ayrıldığı aşamadır.

Anlamsal analiz aşamasında kelime ve cümle aşamalarından geçmiş olan veriler üzerinden cümlelerin tek bir bütün olarak hangi anlama geldiğinin tespit edildiği aşamadır.

Siber Zorbalık tespitinde veriler arasında birden fazla özelliğin bulunması ve özellikler arasında da doğru ilişkilerin kurulabilmesi nedeniyle hedef verilerin tespitinde iyi bir tahmin yürütülebilmesi gerekmektedir. Makineler tarafından anlaşılır hale getirilmiş verilerin, yorumlanıp analiz edilebilmesi için kullanılan en yaygın yöntemlerden birisi de Makine Öğrenmesidir. Makine Öğrenmesi kullanılarak Siber Zorbalık tespiti için birçok çalışma yapılmıştır. Siber Zorbalık tespitinde yaygın olarak kullanılan Makine Öğrenmesi algoritmalarına KNN, Destek Vektör Makineleri, Naive Bayes, Karar Ağaçları örnek olarak verilebilir.

Siber Zorbalık tespitinde kullanılan diğer bir yöntem de Yapay Zekanın bir alt kategorisi olan ve Evrişimsel Sinir Ağlarına dayanan Derin Öğrenmedir. Derin öğrenme hazırlanmış verilerden kendi kendine öğrenme yeteneğine sahip bir Makine Öğrenmesi tekniği şeklinde tanımlanabilir. Derin öğrenme, verilerden özellikleri otomatik olarak çıkararak ve yüksek boyutlu verilerde daha iyi sonuçlar elde ederek makine öğreniminin önüne geçmektedir. Derin Öğrenme ile Siber Zorbalık tespit

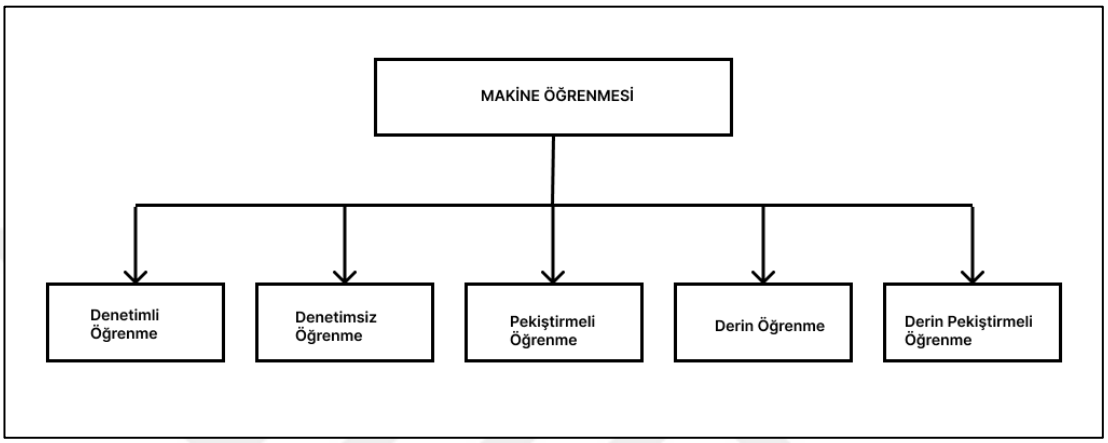
süreci; veri seti oluşturma, veri ön işleme, derin öğrenme yöntemleri oluşturma, sınıflandırma ve sonuç olarak ilerlemektedir.

Son zamanlarda, derin sinir ağı tabanlı modeller, siber zorbalığın tespit edilmesinde geleneksel modeller üzerinde önemli bir iyileşme göstermiştir. Ayrıca, çeşitli Doğal Dil İşleme görevlerinde yararlı olduğunu kanıtlayan yeni ve daha karmaşık derin öğrenme mimarileri geliştirilmektedir. Google, bağlamsal eklemeler üretebilen ve sınıflandırma için göreve özgü eklemeler üretebilen Bert adlı bir dil öğrenme modeli geliştirdi. Sosyal medya platformlarında siber zorbalık tespiti için, mevcut sonuçları geliştiren bir sınıflandırıcı olarak üstte tek bir doğrusal sinir ağ katmanı ile önceden eğitilmiş yeni BERT modelini kullanarak yeni bir yaklaşım önerilmiştir. Bu çalışmada belirli Makine Öğrenmesi algoritmaları ile BERT algoritmasının aynı veri seti üzerinde performans kıyaslaması yapılacaktır. Bu bağlamada Siber Zorbalık tespiti için Makine Öğrenme tabanlı algoritmalar siber zorbalık veri setinde uygulanmıştır. Ardından önerilen BERT algoritması aynı veri seti üzerinde uygulanacak ve karşılaştırmalar yapılacaktır.

2. MATERYAL VE YÖNTEM

2.1. Makine Öğrenmesi Türleri

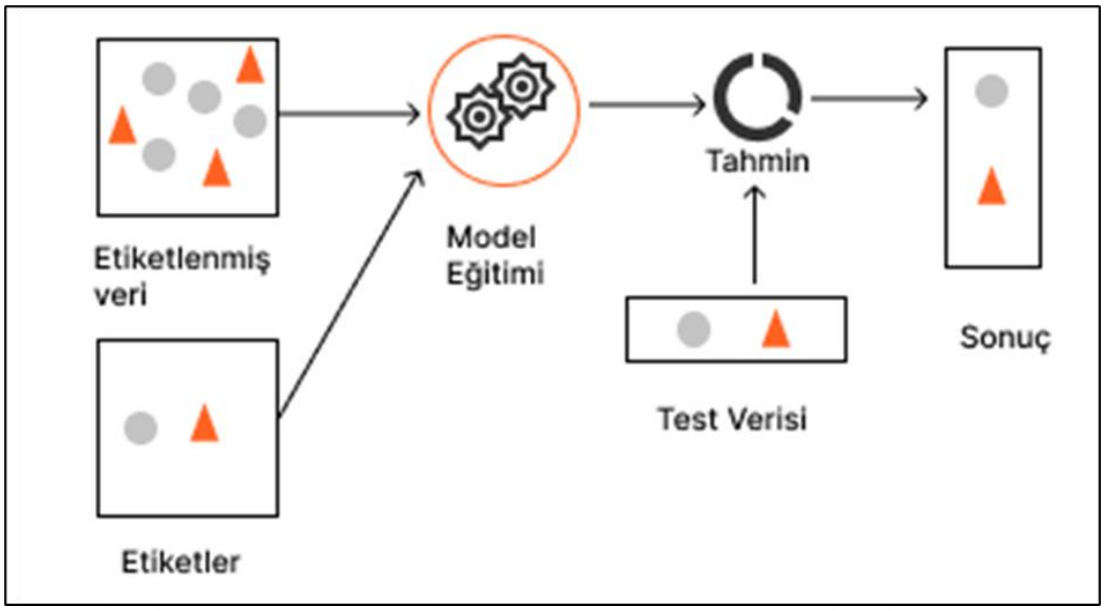
Makine öğrenimi, geniş anlamda bir makinenin akıllı insan davranışını taklit etme yeteneği olarak tanımlanan yapay zekanın bir alt alanıdır. Makine öğrenmesi alt alanları Şekil 2.1'de gösterilmiştir.



Şekil 2.1. Makine Öğrenmesi Alt Alanları [2]

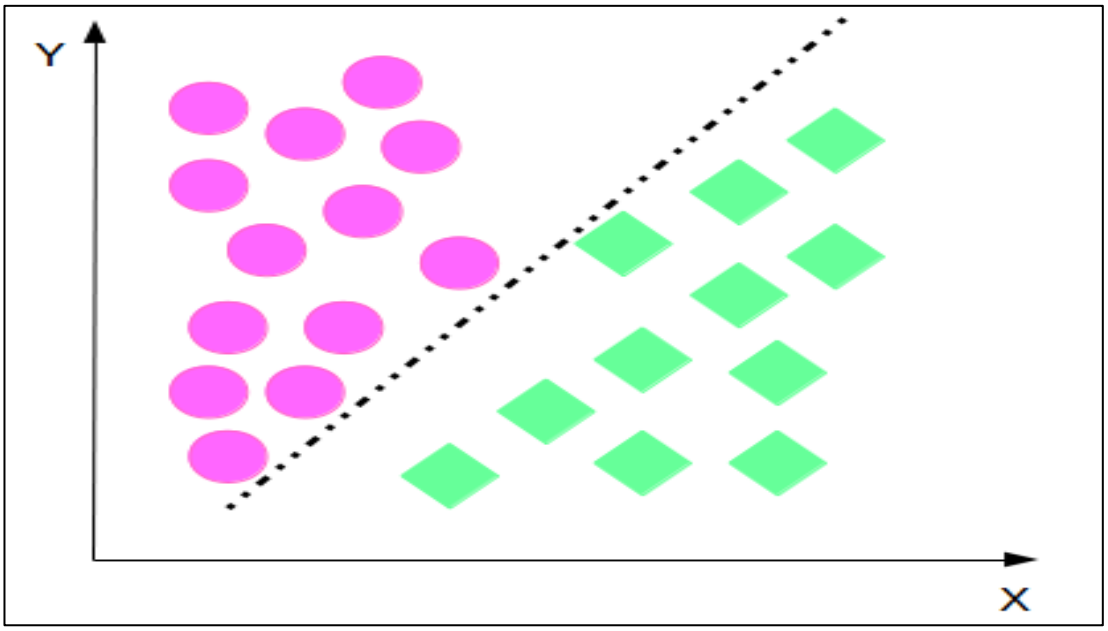
2.1.1. Denetimli Makine Öğrenmesi

Denetimli öğrenme, makine öğrenimi sistemine, sistemi eğitmek için örnek etiketli veriler sağlandığı ve sistemin bu etiketli verilere göre çıktıyı tahmin etmeye odaklandığı yöntemdir [3]. Sistem veri kümelerini anlamlandırmak için, etiketlenmiş verileri kullanarak bir model oluşturur ve ardından eğitim ve işleme aşamaları gerçekleştirilir (Şekil 2.2). Daha sonra kesin çıktı tahmin kontrolü için sisteme örnek veri seti sağlanır ve bu veri seti kullanılarak sistem test edilir. Denetimli öğrenmenin amacı girdi verilerini çıktı verileri ile eşlemektir. Denetimli öğrenme iki algoritma kategorisinde gruplandırılabilir: regresyon ve sınıflandırma.



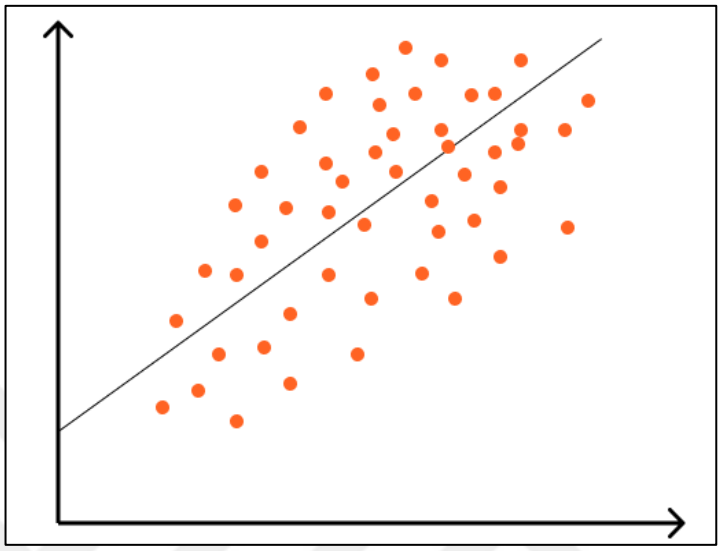
Şekil 2.2. Denetimli Öğrenme [4]

Sınıflandırma: Bir sınıflandırma probleminde, sonuçları ayrı ayrı çıktılarda tahmin etmeye odaklanılmaktadır (Şekil 2.3). Örnek verilecek olursa; tümörlü bir hastanın tümörünün iyi huylu veya kötü huylu olup olmadığını öngörmek, gelen bir mailin spam veya spam olmayan şeklinde kategorize etmektir.



Şekil 2.3. Sınıflandırma

Regresyon: Bir regresyon probleminde, sonuçları sürekli bir çıktı içinde tahmin etmeye odaklanılmaktadır; yani, girdi değişkenlerini bazı sürekli fonksiyonlara eşlemeye çalışmaktadır (Şekil 2.4). Örnek verilecek olursa Bir firmanın reklam harcamalarının satışlarını nasıl etkilediğini tahmin etmek.

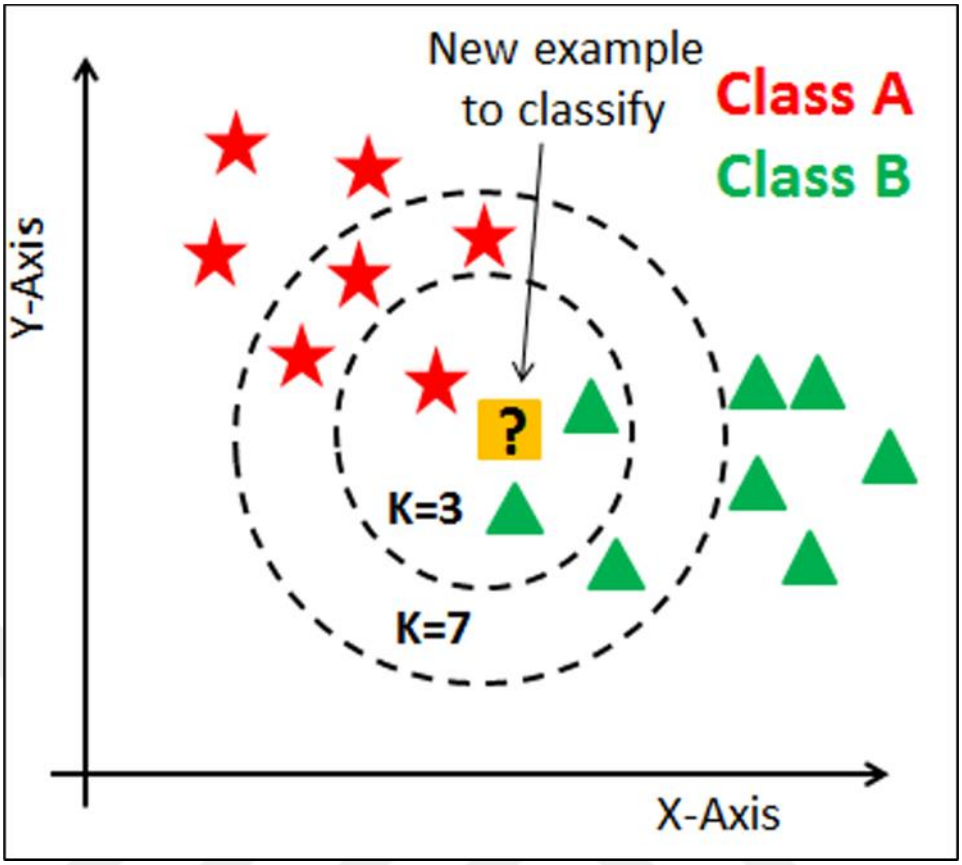


Şekil 2.4. Regresyon

2.1.1.1. K-NN

K-NN sınıflandırma ve regresyon problemlerini çözmek için kullanılabilecek istatistiksel bir tekniktir [2]. Basit anlamıyla KNN, tahmin edilecek değerin bağımsız değişkenlerinin oluşturduğu vektörün sınıfını, en yakın komşuların hangi sınıfta yoğun olduğu bilgisinden yola çıkarak tahmin etmeye dayanır (Şekil 2.5). KNN (K-En Yakın Komşular) Algoritması iki ana değer üzerinden tahmin yapar;

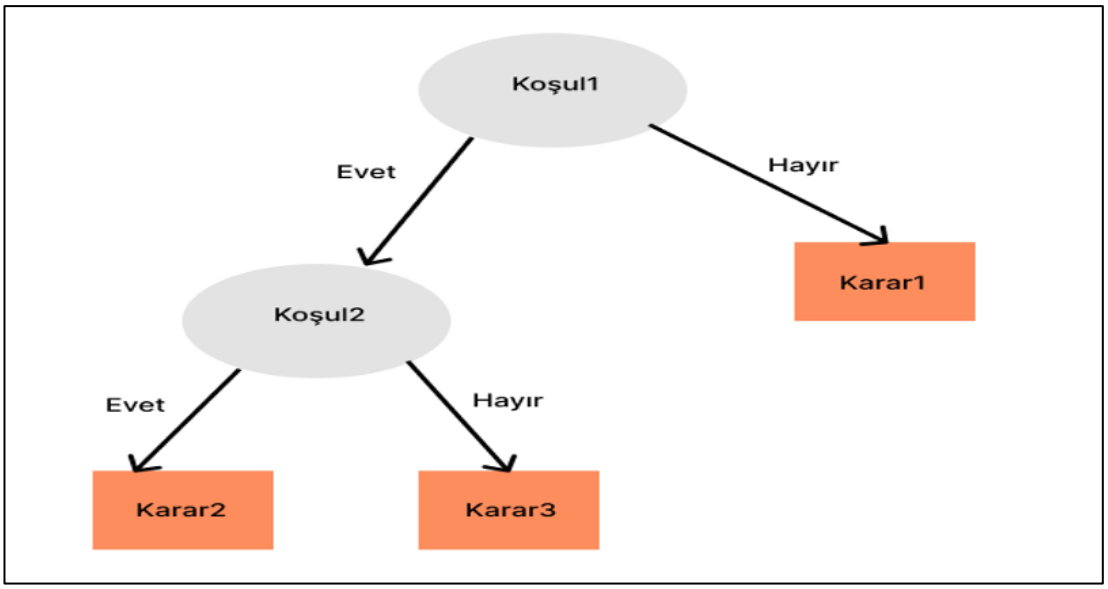
- Uzaklık: Tahmin için bir nokta belirlenir. Tahmini yapılacak noktanın diğer noktalardan uzaklığı hesaplanır. Uzaklık hesabı için Minkowski mesafe hesaplama fonksiyonu kullanılır [6].
- K (Komşuluk Sayısı): En yakın kaç komşu üzerinden tahmin yapılacağı belirlenir. Seçilen K değeri sonucu doğrudan etkileyecektir. K değeri çok küçük olursa overfit etme olasılığı çok yüksek olacaktır. K değeri çok büyük olursa çok genel sonuçlar verecektir ve yanlış tahmine yol açacaktır. K değerinin seçimi büyük önem taşımaktadır. K değeri için optimum değer seçilmelidir [6].



Şekil 2.5. KNN [5]

2.1.1.2. Karar Ağaçları

Karar Ağaçları hem sınıflandırma hem de regresyon problemlerinde kullanılan en popüler makine öğrenme algoritmalarından biridir. Amacı, veri özelliklerinden basit kurallar çıkararak ve bu kuralları öğrenerek bir değişkenin değerini tahmin eden bir model oluşturmaktır. Diğer yöntemlerle kıyaslandığında karar ağaçlarının yapılandırılması, anlaşılması ve yorumlanması daha kolaydır. İki tür eleman kullanılarak inşa edilirler: düğümler ve dallar. Her düğümde, eğitim sürecindeki gözlemleri bölmek veya tahmin yaparken belirli bir veri noktasının belirli bir yolu izlemesini sağlamak için verilerimizin özelliklerinden biri değerlendirilir. Karar ağaçları, örnekleri ağaçta kök düğümden bazı yaprak düğümlere doğru sıralayarak sınıflandırır, yaprak düğüm örneğe sınıflandırma sağlar (Şekil 2.6)



Şekil 2.6. Karar Ağaçları

Sınıflandırma: Karar Ağaçları sınıflandırma için kullanılırken, karar ağacının her adımında veya düğümünde, veri kümesinde yer alan tüm etiketleri veya sınıfları ayırmak amacıyla özellikler üzerinde bir koşul oluşturmaya odaklanılır.

Karar ağaçları yönteminde sınıflandırma yapmak için elimizdeki verilerden ağaç oluşturulur ve veri setindeki kayıtlar bu ağaca uygulanır, çıkan sonuca göre kayıtların sınıflandırma işlemi gerçekleşir.

Karar ağaçları ile sınıflandırma süreci kök düğüm ile başlar. Niteliklerin değerine göre düğümler alt dallara ayrılır, bu süreç yaprak elde edilinceye kadar devam eder. Burada düğümler, test işlemine tabi tutulan nitelikleri gösterir ancak bilinmelidir ki tüm değişkenlerin ya da niteliklerin karar ağacında yer alması beklenmez. Dallar, teste tabi tutulan düğümlerin sonuçlarını gösterir. Elde edilen dallar ile tahmin edilecek sınıfa giden yol belirlenir. Dalın sonucunda sınıflama tamamlanmıyorsa tekrar düğüm oluşturulur. Yapraklar ise veri setindeki bir karar sınıfını temsil etmektedir [7,8].

Regresyon: Karar ağaçları hem kategorik hem de sayısal verileri işleyebilir. J. R. Quinlan'ın ID3 adlı karar ağaçları oluşturmak için çekirdek algoritması, olası dallar alanında yukarıdan aşağıya, geri izleme olmaksızın çalışır. ID3 algoritması, Bilgi Kazanımını Standart Sapma Azaltma ile değiştirerek regresyon için bir karar ağacı oluşturmak için kullanılabilir. Bir karar ağacı, bir kök düğümden yukarıdan aşağıya

oluşturulur ve verilerin benzer değerlere sahip (homojen) örnekler içeren alt kümelere bölünmesini içerir. Sayısal bir örneğin homojenliğini hesaplamak için standart sapma kullanılır.

$$S = \sqrt{\frac{(x_i - \dot{x})^2}{n-1}} \quad (1)$$

Standart Sapma (S) ağaç yapımı (dallanma) içindir. Dallanmanın ne zaman durdurulacağına karar vermek için Sapma Katsayısı (CV) kullanılır. Girdi sayısı (n) de kullanılabilir. Ortalama, yaprak düğümlerindeki değerdir. Standart sapmada iki özellik vardır hedef (target) ve tahmin edici (predictor).

Karar Ağaçları ile regresyon analizi yapılırken, tahmin sonucu elde edildikten sonra Ortalama Kare Hatası (MSE) hesaplanır.

$$MSE = \frac{1}{n}\sum_{i=n}^{n}(Y_i - \hat{Y}_i)^2 \quad (2)$$

Formülde kullanılan, n toplam veri sayısını, Y_i gözlemlenen değeri ve Y^i tahmin edilen değeri ifade etmektedir. Algoritmanın arkasındaki temel fikir, veri setini 2 parçaya bölmek için bağımsız değişkendeki noktayı bulmaktır, böylece ortalama kare hatası bu noktada en aza indirgenir. Algoritma bunu tekrarlayan bir şekilde yapar ve ağaç benzeri bir yapı oluşturur.

2.1.1.3. Naive Bayes

Naive Bayes, büyük hacimli veriler için kullanılan bir makine öğrenme modelidir, milyonlarca veri kaydına sahip verilerle çalışıyor olsanız bile önerilen yaklaşım Naive Bayes'tir. Duygusal analiz gibi NLP görevleri söz konusu olduğunda çok iyi sonuçlar verir. Hızlı ve karmaşık olmayan bir sınıflandırma algoritmasıdır. Naive Bayes algoritmasını anlamak için Bayes teoremini anlamak gerekir.

Bayes Teoremi: Koşullu olasılık üzerinde çalışan bir teoremdir. Koşullu olasılık bize bir olayın ön bilgisini kullanarak olasılığını verebilmektedir.

$$P(A\backslash B) = \frac{P(B\backslash A) * P(A)}{P(B)} \quad (3)$$

P(A): H hipotezinin doğru olma olasılığı. Bu, önceki olasılık olarak bilinir.

P(B): Kanıtın olasılığı.

P(A|B): Verilen hipotezin doğru olma olasılığı.

P(B|A): Kanıtın doğru olduğu verilen hipotezin olasılığı.

Naive Bayes Sınıflandırıcısı: Sınıflandırıcı, değişkenlerin belirli özelliklerine göre farklı nesneleri ayıran bir makine öğrenme modelidir. Bayes teoremi üzerinde çalışan bir tür sınıflandırıcıdır. Belirli bir sınıfla ilişkili veri noktalarının olasılığı gibi her sınıf için üyelik olasılıklarının tahmini yapılır. Yüksek olasılığa sahip sınıf, en uygun sınıf olarak değerlendirilir. Bu aynı zamanda Maksimum A Posteriori (MAP) olarak da adlandırılır [9].

Bir hipotez için MAP:

$MAP\ (H)$ = maksimum $P((H|E))$

$MAP\ (H)$ = maksimum $P((H|E) * (P(H))\ /P(E))$

$MAP\ (H)$ = maksimum$(P(E|H) * P(H))$

$P\ (E)$ kanıt olasılığıdır ve sonucu normalleştirmek için kullanılır. $P(E)$ kaldırıldığında sonuç etkilenmeyecektir.

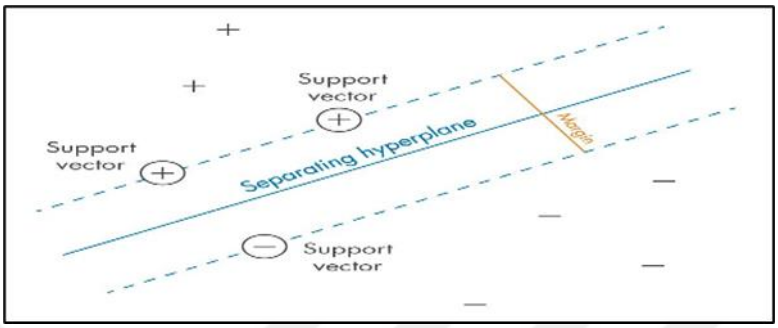
Naive Bayes algoritmaları, duygu analizi, istenmeyen e-posta filtreleme, öneri sistemleri vb. için yaygın olarak kullanılmaktadır. Hızlı ve kullanımı daha kolaydır, ancak en büyük dezavantajı "tahmin edicilerin bağımsız olması gerekliliğidir".

2.1.1.4. Destek Vektör Makineleri

Birçok uygulamada uygulanan önemli öğrenme algoritmalarından biridir ve şu anda birçok uygulama alanında en çok kullanılan yöntemlerden biri olarak temsil edilmektedir. SVM, tıbbi teşhis, metin veya görüntü sınıflandırması, istenmeyen posta tespiti, bir nesnenin tespiti ve tanınması, yüz tespiti, doğrulama ve tanıma, sinyal işleme, tahmin, bilgi gibi birçok farklı alanda birçok alanda uygulanmıştır.

Destek Vektör Makineleri, herhangi birçok değişkenli işlevi farklı doğrulukta tahmin etme yeteneğine sahiptir ve bu nedenle, Destek Vektör Makineleri, karmaşık sistemleri ve doğrusal olmayan sistemleri modellemek için çok uygundur.

Destek Vektör Makineleri, iki sınıf arasında sınıflandırma yapmak ve bir karar yüzeyi vermek için kullanılır. Bu karar yüzeyi, sınıflandırma sınıfındaki en yakın noktaları ayırmaya bağlı olabilecek maksimum mesafe ile temsil edilir (Şekil 2.7). En yakın noktalar, destek vektörleri verileri olarak bilinir [10].

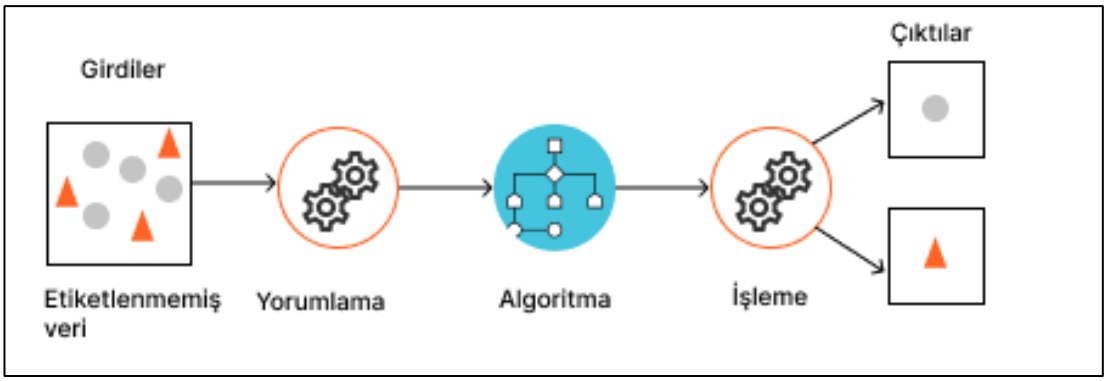


Şekil 2.7. Destek Vektör Makineleri [6]

Destek Vektör Makinelerinin, girdi verilerinin doğrusal olarak ayrılabilir olarak desteklenebilmesi koşulunu kolaylaştırmak için maksimum marj sınıflandırıcıları olduğu düşünülebilir. Bu varsayım altında, herhangi iki sınıf, aralarındaki lineer hiper düzlem elde edilebilir.

2.1.2. Denetimsiz Makine Öğrenmesi

Denetimsiz öğrenme, bir makinenin herhangi bir denetim olmadan öğrendiği bir öğrenme yöntemidir. Eğitim, etiketlenmemiş, sınıflandırılmamış veya kategorize edilmemiş veri seti ile makineye verilir ve algoritmanın herhangi bir denetim olmaksızın bu veriler üzerinde hareket etmesi gerekir. Denetimsiz öğrenmenin amacı, girdi verilerini yeni özelliklere veya benzer kalıplara sahip bir grup nesneye yeniden yapılandırmaktır. Denetimsiz öğrenmede önceden belirlenmiş bir sonuç yoktur. [12] Makine, büyük miktarda veriden faydalı bilgiler bulmaya çalışma üzerine odaklanmıştır (Şekil 2.8).



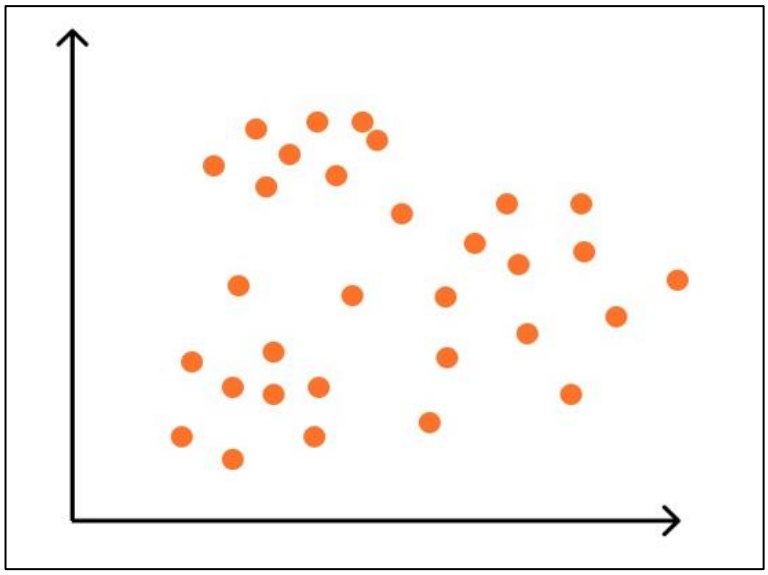
Şekil 2.8. Derin Öğrenme Yapısı [13]

2.1.2.1. K Ortalamalar Kümesi

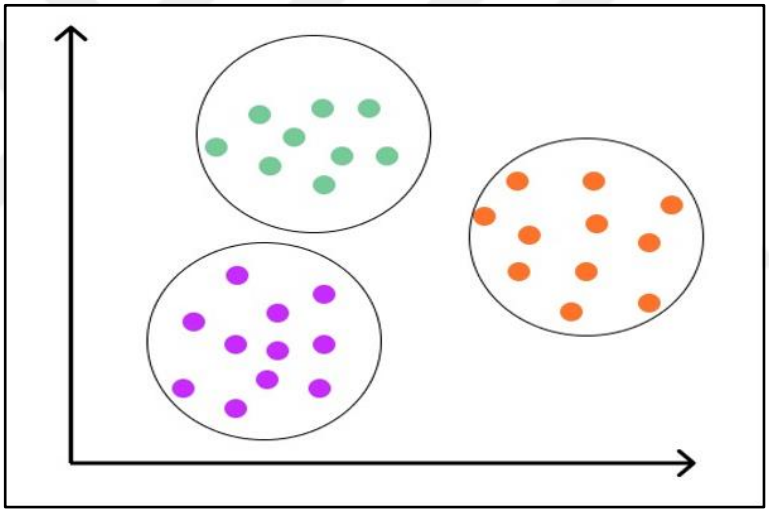
K-Ortalamalar Kümeleme, etiketlenmemiş veri kümesini farklı kümeler halinde gruplayan denetimsiz makine öğrenmesi algoritmasıdır (Şekil 2.9). K, oluşturulması gereken önceden belirlenmiş küme sayısını ifade eder. K- Ortalamalar Kümesi etiketlenmemiş verilerin herhangi bir eğitime ihtiyaç duymaksızın farklı gruplara ayrılmasına odaklanır (Şekil 2.10). Bu algoritmanın temel amacı, veri noktası ve bunlara karşılık gelen kümeler arasındaki mesafelerin toplamını en aza indirmektir. Algoritma, etiketlenmemiş veri kümesini girdi olarak alır, veri kümesini K sayıda kümeye böler ve en iyi kümeleri bulana kadar işlemi tekrarlar. Bu algoritmada K değeri önceden belirlenmektedir.

K-Ortalamalar kümeleme algoritması temel olarak iki görevi yerine getirir:

- Yinelemeli bir işlemle K merkez noktaları veya ağırlık merkezleri için en iyi değeri belirler.
- Her veri noktasını en yakın komşuluk merkezine atar. Belirli komşuluk merkezine yakın olan bu veri noktaları bir küme oluşturur. Dolayısıyla, her kümenin bazı ortak noktaları olan veri noktaları vardır ve diğer kümelerden uzaktır.



Şekil 2.9. K Ortalamalar Kümeleme Öncesi

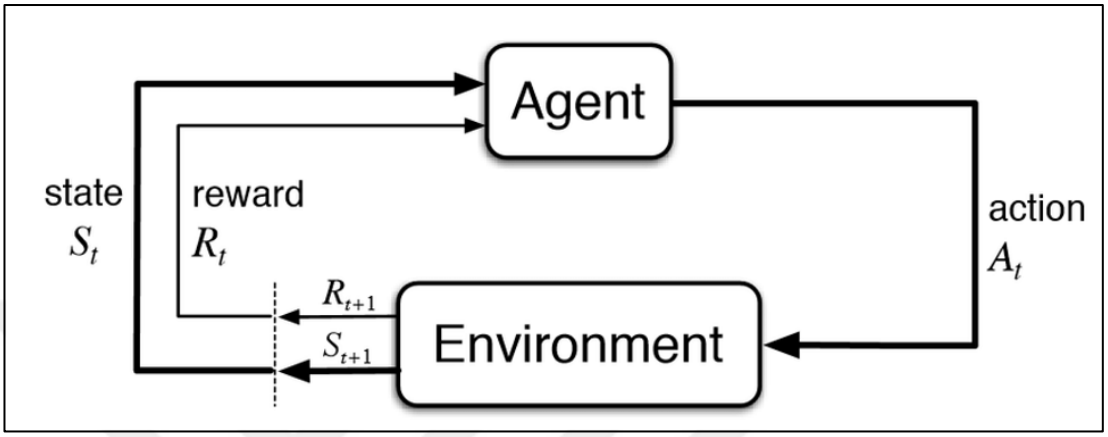


Şekil 2.10. K Ortalamalar Kümeleme Sonrası

2.1.3. Pekiştirmeli Makine Öğrenmesi

Pekiştirmeli Öğrenme, sistemin eylemleri gerçekleştirerek ve gerçekleştirdiği eylemin sonuçlarına göre davranmayı öğrendiği geri bildirime dayalı bir makine öğrenmesi tekniğidir. Sistem, iyi her eylem için olumlu dönüt alır, her kötü eylem için olumsuz dönüt veya ceza alır. Sistem, herhangi bir etiketli veri olmadan alınan dönütleri kullanarak otomatik olarak öğrenir.

Pekiştirmeli Öğrenme, denetimli öğrenmeden, denetimli öğrenmede eğitim verilerinin cevap anahtarına sahip olması ve böylece modelin doğru cevabın kendisi ile eğitilmesi, takviyeli öğrenmede cevap olmaması, ancak takviye ajanının ne yapacağına karar vermesi bakımından farklıdır. Bir eğitim veri kümesinin yokluğunda, deneyimlerinden öğrenmesi zorunludur (Şekil 2.11).



Şekil 2.11. Pekiştirmeli Öğrenme [14]

Pekiştirmeli öğrenmede önemli olan anahtar kavramlar aşağıdaki gibi sırlanabilir. Giriş: Giriş, modelin başlayacağı bir başlangıç durumu olmalıdır. Çıktı: Belirli bir soruna çeşitli çözümler olduğu için birçok olası çıktı vardır.

Eğitim: Eğitim girdiye dayanır, model bir durum döndürür ve kullanıcı modeli çıktısına göre ödüllendirmeye veya cezalandırmaya karar verir.

Pekiştirmeli Öğrenmede, model öğrenmeye devam etmekteyken en iyi çözüm maksimum ödüle göre kararlaştırılır. Takviyeli öğrenme, sırayla kararlar vermekle ilgilidir. Basit bir deyişle, çıktının mevcut girdinin durumuna bağlı olduğunu ve bir sonraki girdinin önceki girdinin çıktısına bağlıdır.

2.1.3.1. Pekiştirmeli Öğrenme Türleri

Temel olarak iki tür pekiştirmeli öğrenme vardır; Olumlu Pekiştirmeli Öğrenme ve Olumsuz Pekiştirmeli Öğrenme [15].

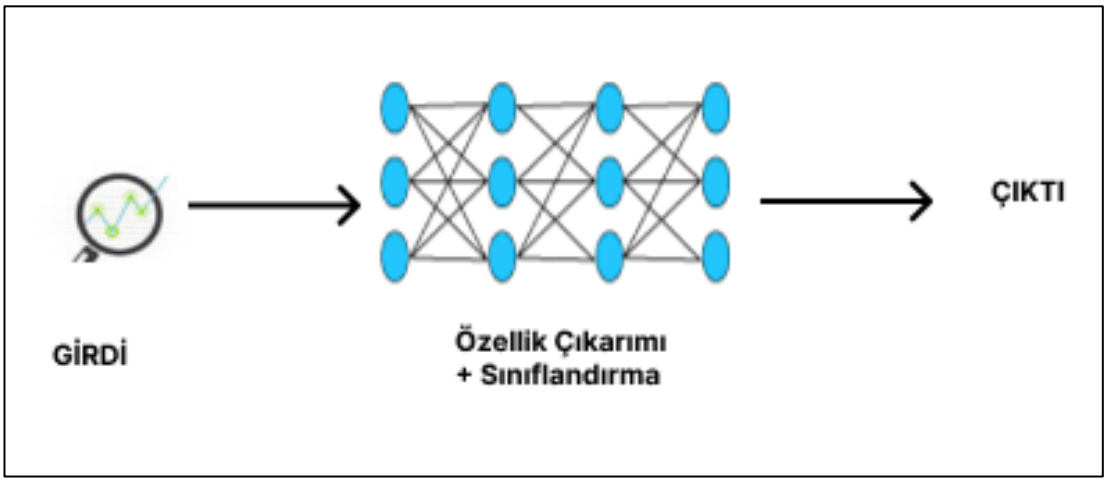
Olumlu Pekiştirme, belirli bir davranış nedeniyle bir olayın meydana gelmesi, davranışın gücünü ve sıklığını artırması olarak tanımlanır. Başka bir deyişle, davranış üzerinde olumlu bir etkisi vardır.

Takviyeli öğrenmenin avantajları şunlardır: Performansı En Üst Düzeye Çıkarır Değişimi uzun süre sürdürün Çok fazla Güçlendirme, sonuçları azaltabilecek durumların aşırı yüklenmesine neden olabilir.

Olumsuz Pekiştirme, olumsuz bir durum durdurulduğu veya kaçınıldığı için davranışın güçlendirilmesi olarak tanımlanır. Takviyeli öğrenmenin dezavantajları: Asgari performans standardına meydan okuma sağlamak yalnızca minimum davranışı karşılamaya yetecek kadar sağlar.

2.2. Derin Öğrenme

Derin Öğrenme, yapay sinir ağlarına (ANN), daha özel olarak evrişimsel sinir ağlarına (CNN) dayalı bir modeldir. Derin öğrenmede, birden çok işleme katmanından oluşan hesaplama modelleri bulunmaktadır [16]. Bu modellerin, birden çok soyutlama düzeyiyle verilerin temsillerini öğrenmesi amaçlanmaktadır. Derin öğrenme, bir makinenin önceki katmandaki temsilden her katmandaki temsili hesaplamak için kullanılan dahili parametrelerini nasıl değiştirmesi gerektiğini belirtmek için geri yayılım algoritmasını kullanarak büyük veri kümelerinde karmaşık yapıyı keşfeder (Şekil 2.12).



Şekil 2.12. Derin Öğrenme Yapısı

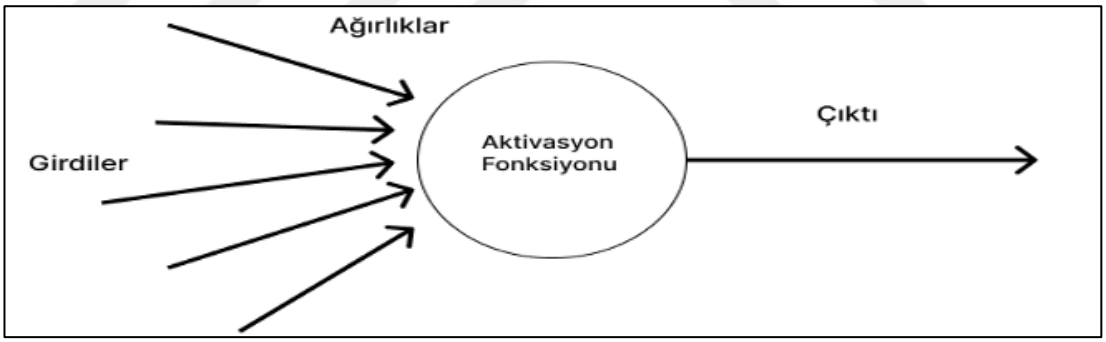
2.2.1.Yapay Sinir Ağları

Yapay sinir ağı, sinir yapısını oluşturan katsayılar (ağırlıklar) ile bağlantılı yüzlerce tek birimden, yapay nöronlardan oluşan biyolojik olarak esinlenilmiş bir hesaplama modelidir. Bilgiyi işledikleri için işleme elemanları (PE) olarak da bilinirler [17].

Her PE'nin ağırlıklı girdileri, transfer işlevi ve bir çıktısı vardır. PE esasen girdileri ve çıktıları dengeleyen bir denklemdir. Bağlantı ağırlıkları sistemin hafızasını temsil ettiği için yapay sinir ağlarına bağlantı tabanlı modeller de denir.

2.2.1.1. Nöronlar

Yapay nöron, biyolojik nöronun işlevini göstermek etmek için tasarlanmış yapay sinir ağının yapı bileşenidir. Girdi olarak adlandırılan ve bağlantı ağırlıkları ile çarpılan (ayarlanmış) gelen sinyaller önce toplanır (birleştirilir) ve daha sonra o nöron için çıktı üretmek üzere bir transfer fonksiyonundan geçirilir (Şekil 2.13). Aktivasyon fonksiyonu, nöronun girdilerinin ağırlıklı toplamıdır [17].



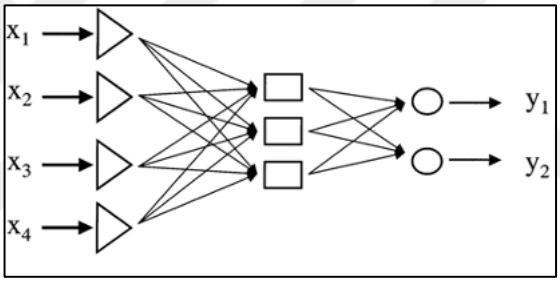
Şekil 2.13. Yapay Sinir Ağları

2.2.1.2. Bağlantı Formülü

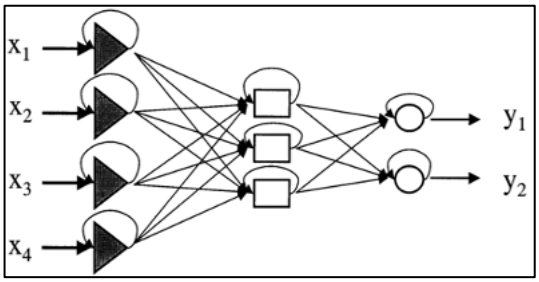
Nöronların birbirine bağlanma şekli, yapay sinir ağının işleyişi üzerinde önemli bir etkiye sahiptir. Tıpkı 'gerçek' nöronlar gibi, yapay nöronlar da uyarıcı veya engelleyici girdiler alabilir. Uyarıcı girdiler, bir sonraki nöronun toplama mekanizmasının eklenmesine neden olurken, engelleyici girdiler onun çıkarılmasına neden olur. Bir nöron aynı katmandaki diğer nöronları da inhibe edebilir. Buna lateral

inhibisyon denir. Ağ, en yüksek olasılığı 'seçmek' ve diğerlerini engellemek istemektedir. Bu kavramın adı da rekabettir.

Geri besleme, bir katmanın çıktısının bir önceki katmanın girişine veya aynı katmana geri yönlendirildiği başka bir bağlantı türüdür. Bir ağda geri besleme bağlantısının olmamasına veya varlığına göre iki tür mimari tanımlanabilir. İleri beslemeli mimarinin çıktıdan girdi nöronlarına geri bağlantısı yoktur ve bu nedenle önceki çıktı değerlerinin kaydını tutmaz (Şekil 2.14). Geri besleme mimarisi, çıktıdan girdi nöronlarına bağlantılara sahiptir. Her nöron, eğitim hatasını en aza indirmeye çalışırken ek bir serbestlik derecesine izin verecek bir girdi olarak ek bir ağırlığa sahiptir (Şekil 2.15). Böyle bir ağ, önceki durumun bir hafızasını tutar, böylece bir sonraki durum sadece giriş sinyallerine değil, aynı zamanda ağın önceki durumlarına da bağlıdır.



Şekil 2.14. İleri Besleme [17]



Şekil 2.15. Geri Besleme [17]

2.2.1.3. Öğrenme Kuralı

Pek çok farklı öğrenme kuralı vardır, ancak en sık kullanılanı Delta kuralı veya Geriye yayılma kuralıdır [17]. Bir sinir ağı, ağırlıkların yinelemeli ayarlanmasıyla bir dizi girdi verisini eşlemek için eğitilir. Ağırlıklı bağlantıların kullanımı, yapay sinir

ağının tanıma yetenekleri için esastır. Girdilerden gelen bilgiler, nöronlar arasındaki ağırlıkları optimize etmek için ağ üzerinden ileriye doğru beslenir. Ağırlıkların optimizasyonu, eğitim veya öğrenme aşamasında hatanın geriye doğru yayılmasıyla yapılır. Yapay Sinir Ağları, eğitim veri setindeki giriş ve çıkış değerlerini okur ve tahmin edilen ve hedef değerler arasındaki farkı azaltmak için ağırlıklı bağlantıların değerini değiştirir.

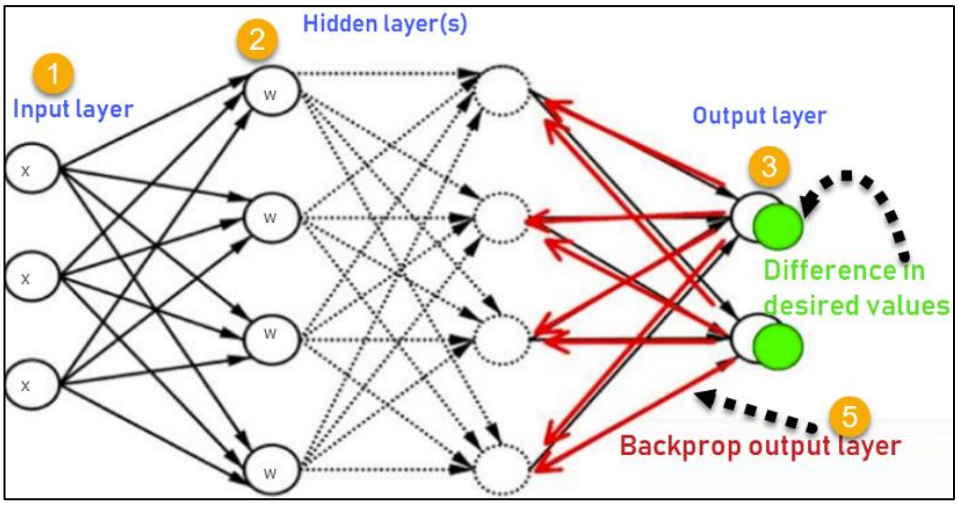
2.2.2. Geri Yayılım Algoritması

Geri yayılım, sinir ağı eğitiminin özüdür. Bir önceki aşamada elde edilen hata oranına dayalı olarak bir sinir ağının ağırlıklarına ince ayar yapma yöntemidir. Ağırlıkların uygun şekilde ayarlanması, hata oranlarını azaltmanıza ve genellemeyi artırarak modeli güvenilir hale getirmenize olanak tanır.

Sinir ağlarında geriye yayılım, "hataların geriye doğru yayılması" için kısa bir formdur. Yapay sinir ağlarını eğitmek için standart bir yöntemdir. Bu yöntem, ağdaki tüm ağırlıklara göre bir kayıp fonksiyonunun gradyanını hesaplamaya yardımcı olur.

Sinir ağındaki geri yayılım algoritması, zincir kuralına göre tek bir ağırlık için kayıp fonksiyonunun gradyanını hesaplar. Yerel bir doğrudan hesaplamanın aksine, her seferinde bir katmanı verimli bir şekilde hesaplar. Geri yayılım hızlı, basit ve programlanması kolaydır [18,19]. Giriş sayılarından başka ayarlanacak parametresi yoktur.

Ağ hakkında önceden bilgi gerektirmediğinden esnek bir yöntemdir. Genellikle iyi sonuç veren standart bir yöntemdir. Öğrenilecek işlevin özelliklerinden özel olarak söz edilmesine gerek yoktur.



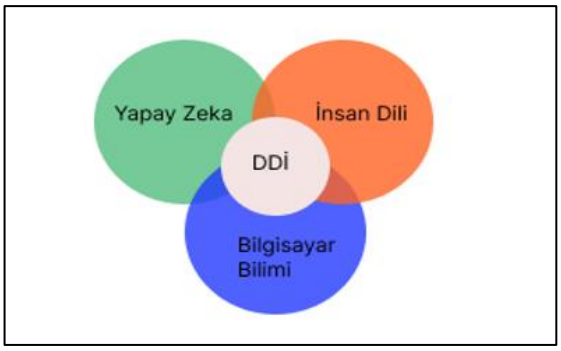
Şekil 2.16. Geri Yayılım Algoritması [19]

X girişleri, önceden bağlanmış yoldan gelir. Girdi, gerçek ağırlıklar W kullanılarak modellenir. Ağırlıklar genellikle rastgele seçilir. Girdi katmanından gizli katmanlara ve çıktı katmanına kadar her nöron için çıktı hesaplanır. Çıkışlardaki hatayı hesaplanır. Ağırlıkları hata azaltılacak şekilde ayarlamak için çıktı katmanından gizli katmana geri dönülür (Şekil 2.16).

2.3. Doğal Dil İşleme

Dil insanların duygu ve düşüncelerini ifade etmek, aktarmak için kullandıkları bir iletişim aracıdır. Her dilin kendine özgü bir yapısı ve kuralları vardır. Dil insanlık tarihi boyunca var olmuş ve insanlık tarihi var oldukça yaşayacak bir yapıdır. Ayrıca dil sürekli gelişmekte ve değişmekte olan canlı bir varlıktır.

Sözcük ve cümle birimleri aracılığıyla, düşünceyi konuşmayla ilişkilendiren çok seviyeli bir sistemdir. (N. Chomsky). Doğal Dil İşleme, Yapay Zeka ve Dil Biliminin bir alt kategorisidir. Doğal Dil İşleme, makineler aracılığıyla doğal dillerin kurallı yapısının çözümlenerek anlaşılır hale getirilmesi veya yeniden üretilmesi amacı sağlar (Şekil 2.17).



Şekil 2.17. Doğal Dil İşleme

2.3.1. Doğal Dil İşlemenin Tarihçesi

Doğal Dil İşleme alanında ilk çalışmalar 2. Dünya Savaşından sonra 1940'larda başlamıştır. İlerleyen zamanlarda, insanlar bir dilin diğer dile çevirinin önemini fark ettiler ve bu tür çevirileri otomatik olarak yapabilen bir makine geliştirmeyi planladılar.

1954 yılında yapılan Geogretown-IBM deneyi bu alandaki atılan ilk önemli atılım olmuştur. Bu deneyde, 60'tan fazla Rusça cümlenin bilgisayarlar tarafından otomatik olarak tercüme edilmesi amaçlanmıştır. 1960'larda bilgisayarlar anlamayı öğrenmiştir. Yapay Zekâ teorisyeni Roger Schank, 1969'da doğal dili anlamak için kavramsal bağımlılık teorisi modelini geliştirmiştir. Geliştirilen bu teorinin amacı, bilgisayarların, yazılan sözcükten bağımsız olarak anlam okuması yapmasını sağlamaktı. Bu yaklaşım, bilgisayarlara girdilerin nasıl girildiğine bakılmaksızın anlamları aynı olduğu sürece yazılma şekillerinin önemli olmadığını öğretti. Bu teori ile bilgisayarlar "Dün eve geldim." ile "Eve dün geldim." cümlelerinin aynı anlama geldiğini kavrayacak bir biçimde eğitilmeye başlandı. Doğal Dil İşleme araştırmacısı olan William A. Woods, 1970 yılında arttırılmış geçiş ağını (ATN) tanıttı. ATN doğal dil girdisini temsil etmekteydi ve bilgisayarlara cümle yapısını analiz etme kabiliyeti kazandırdı. 1970'li yıllar programcıların kavramsal ontolojiler geliştirmesiyle geçti [20].

Doğal Dil İşleme 1980'lerin büyük bir bölümünde kural tabanlı modellere dayandı. 1980'lerin sonunda geliştirilen ilk Makine Öğrenimi algoritmaları şartlı kural mekaniklerine dayanan karar ağaçları olarak şekillendi ve istatiksel modellere odaklandı. Günümüzde Doğal Dil İşleme altın çağını yaşamaktadır. Değişen ve gelişen

teknoloji ile Doğal Dil işleme her zamankinden daha ön plana çıkmaktadır. Aynı zamanda Yapay Zekâ teknolojisinin gelişimiyle doğru orantılı olarak sürekli daha tutarlı ve doğru sonuçlar vererek işlevselliğini arttırmaktadır [20].

2.3.2. Doğal Dil İşlemenin Bileşenleri

2.3.2.1. Doğal Dil Anlama

Bir bilgisayarın yapılandırılmamış metni makine tarafından okunabilir bir formatta işleyebilmesi için önce makinelerin insan dilinin özelliklerini anlaması gerekir.

Doğal Dil Anlama (Natural Language Understanding (NLU)), kavramlar, anahtar kelimeler, ilişkiler ve anlamsal roller gibi içerikten temel verileri çıkararak makinenin insan dilini anlamasına ve analiz etmesine yardımcı olur. Metin veya konuşma biçiminde yapılan girdilerin bilgisayar tarafından anlaması için bilgisayar yazılımını kullanan yapay zeka dalıdır. NLU doğrudan insan-bilgisayar etkileşimini sağlar. Doğal dil anlayışı bir makinenin insan dilini anlama yeteneğine odaklanır. NLU, makinelerin onu "anlaması" ve analiz etmesi için yapılandırılmamış verilerin nasıl yeniden düzenlendiğini ifade eder [20,21].

2.3.2.2. Doğal Dil Üretimi

NLG, hesaplanan verileri doğal dil temsiline dönüştüren bir çevirmen görevi görür. Doğal dil üretimi (NLG), yapay zekâ kullanarak verileri doğal dile dönüştürme sürecidir. Doğal Dil Üretimi için yapay zekâ yazılımları kullanılmaktadır. Yazılımlar bunu, sayıları insanların anlayabileceği doğal dil metnine veya konuşmaya dönüştürmek için makine öğrenimi ve derin öğrenmeyle desteklenen yapayzeka modellerini kullanarak yapar. Chatbot'lar, sesli asistanlar ve AI blog yazarları (birkaç isim) doğal dil oluşturmayı kullanır. NLG sistemleri, önceden ayarlanmış şablonlara dayalı olarak sayıları anlatılara dönüştürebilir. Daha sonra hangi kelimelerin oluşturulması gerektiğini tahmin edebilirler (örneğin, aktif olarak yazdığınız bir e-postada). Veya en karmaşık sistemler, özetlerin, makalelerin veya yanıtların tamamını formüle edebilir [20,21].

2.3.3. Doğal Dillerde Analiz

2.3.3.1. Morfolojik Analiz

Morfolojik veya Sözcüksel Analiz, metinle bireysel kelime düzeyinde ilgilenir. Bir kelimenin en küçük birimi olan morfemleri arar. Örneğin; irrasyonel, ir (ön ek), rasyonel (kök) ve -ly (sonek) olarak ayrılabilir. Sözcük Analizi, bu biçimbirimler arasındaki ilişkiyi bulur ve sözcüğü kök biçimine dönüştürür. Sözcük analizcisi ayrıca olası konuşma bölümünü kelimeye atar. Dilin sözlüğünü dikkate alır. Örneğin, "karakter" kelimesi isim veya fiil olarak kullanılabilir.

2.3.3.2. Sözdizimsel Analiz

Sözdizimi Analizi, belirli bir metin parçasının doğru yapıda olmasını sağlar. Cümle düzeyinde doğru dilbilgisini kontrol etmek için cümleyi ayrıştırmaya çalışır. Önceki adımda oluşturulan olası konuşma bölümü göz önüne alındığında, bir sözdizimi çözümleyicisi, cümle yapısına dayalı olarak konuşma bölümü etiketlerini atar.

2.3.3.3. Anlamsal Analiz

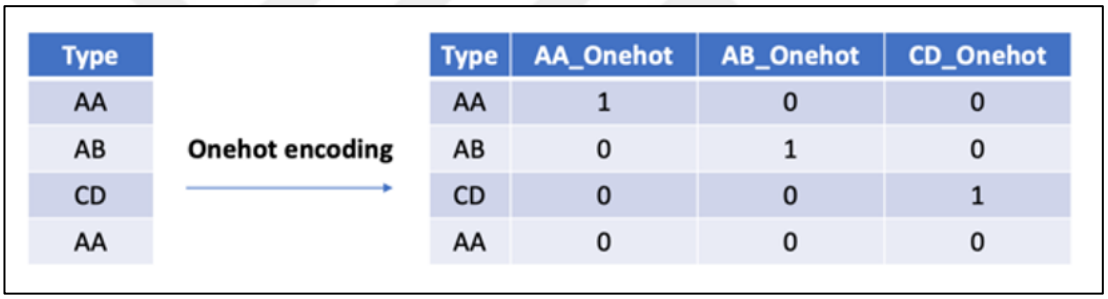
Semantik Analiz, anlamlar atayan sözdizimsel çözümleyici tarafından oluşturulan bir yapıdır. Bu bileşen, doğrusal sözcük dizilerini yapılara aktarır. Semantik yalnızca kelimelerin, deyimlerin ve cümlelerin gerçek anlamlarına odaklanır. Bu, yalnızca sözlük anlamını veya verilen bağlamdan gerçek anlamı soyutlar. Sözdizimsel çözümleyici tarafından atanan yapılara her zaman bir anlam atanmıştır [22].

2.3.4. Kelime Gömme Teknikleri

Metinlerin makineler tarafından anlaşılabilir, yorumlanabilir ve işlenebilir olması için, kelimelerin matematiksel olarak karşılığının oluşturulması gerekmektedir. Metin temsil yöntemleri, kelimeleri matematiksel olarak temsil etmek için kullanılır. One Hot Encoding, TF-IDF, Word2Vec, FastText, sık kullanılan Word Gömme yöntemleridir. Bu tekniklerden biri (bazı durumlarda birkaçı) verinin işlenme durumuna, boyutuna ve amacına göre tercih edilir ve kullanılır.

2.3.4.1. One Hot Encoding

Verileri sayısal olarak temsil etmek için kullanılan tekniklerden biri One Hot Encoding tekniğidir. One Hot Encoding, sağlanacak kategorik veri değişkenlerini makine ve derin öğrenme algoritmalarına dönüştürmek için gerekli süreç olarak tanımlanabilir, bu da bir modelin sınıflandırma doğruluğunun yanı sıra tahminleri de geliştirir. One Hot Encoding, makine öğrenimi modelleri için kategorik özellikleri önceden işlemenin yaygın bir yoludur. Verilen metin parçasının her kelimesi - semboller de dahil- sadece 1 ve 0'dan oluşan vektörler halinde yazılır (Şekil 2.18). Her kelime için benzersiz bir vektör yazılır veya kodlanır. Bu yöntemde toplam benzersiz kelime sayısı büyüklüğünde bir vektör oluşturulur. Bir kelime bir vektör olarak temsil edildiğinden, cümledeki kelimeleri listesi bir vektör dizisi veya bir matris olarak ifade edilebilir [23].



Şekil 2.18. One Hot Encoding

2.3.4.2. TF-IDF

TF-IDF, belgelerdeki kelimelerin matematiksel önemini belirlemek için kullanılan istatistiksel bir ölçüdür. Vektör oluşturma işlemi, One Hot Encoding'e benzer. Alternatif olarak, kelimeye karşılık gelen değere 1 yerine TF-IDF değeri atanır. TF-IDF değeri, TF ve IDF değerlerinin çarpılmasıyla elde edilir [23].

Bir örnek ile gösterilecek olursa;

[O bir kedidir.]

[O bir kuştur.]

[O, panda veya köpek değildir.]

Yukarıdaki örnekte "O" 3 belgenin tamamında, "bir" 2 belgede ve "veya" yalnızca bir belgede kullanılmıştır. Bunlara göre sırasıyla TF ve ardından IDF değerlerini bulalım.

Terim Sıklığı (TF): Belgedeki hedef terim sayısının belgedeki toplam terim sayısına oranıdır. TF değeri yukarıdaki örneğe göre hesaplanacak olursa;

[0.33, 0.33, 0.33],
[0.33, 0.33, 0.33],
[0.20, 0.20, 0.20, 0.20, 0.20]

Ters Belge Sıklığı (IDF): IDF değeri, toplam belge sayısının hedef terimin geçtiği belge sayısına oranının logaritmasıdır. Bu aşamada, terimin belgede kaç kez geçtiği önemli değildir. Geçip geçmediğini belirlemek yeterlidir. Bu örnekte alınacak logaritmanın taban değeri 10 olarak belirlenmiştir. Ancak farklı değerlerin kullanılmasında bir sakınca yoktur.

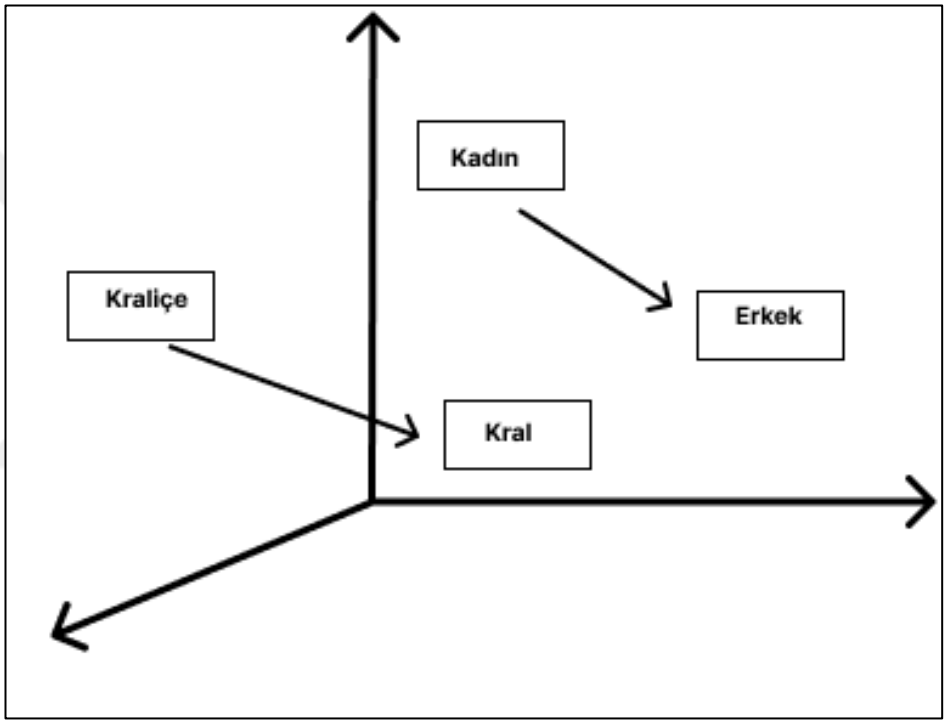
"O": Log (3/3) = 0,
"Bir": Log (3/2):0.1761,
"Veya": Log (3/1): 0.4771

Böylece hem TF hem de IDF değerleri elde edilmiştir. Bu değerlerle vektörleştirme oluşturulursa, ilk olarak her belge için tüm belgelerdeki benzersiz kelime sayısı kadar öğelerden oluşan bir vektör oluşturulur (bu örnekte 8 terim vardır). Bu aşamada çözülmesi gereken bir problem vardır. He teriminde görüldüğü gibi IDF değeri 0 olduğu için TF-IDF değeri de sıfır olacaktır. Ancak vektörleştirme işlemi sırasında belgede yer almayan kelimelere (örneğin "panda" ibaresi 1. cümlede yer almamaktadır) 0 değeri atanacaktır. Burada karışıklığı önlemek için TF-IDF değerleri vektörleştirme için yumuşatılır. En yaygın yöntem, elde edilen değerlere 1 eklemektir. Amaca göre daha sonra bu değerlere normalizasyon uygulanabilir [23].

[1., 1.1761, 1.4771, 0., 0., 0., 0., 0.],
[1., 1.1761, 0., 1.4771, 0., 0., 0., 0.],
[1., 0., 0., 0., 1.4771, 1.4771, 1.4771, 1.4771],

2.3.4.3. Word2Vec

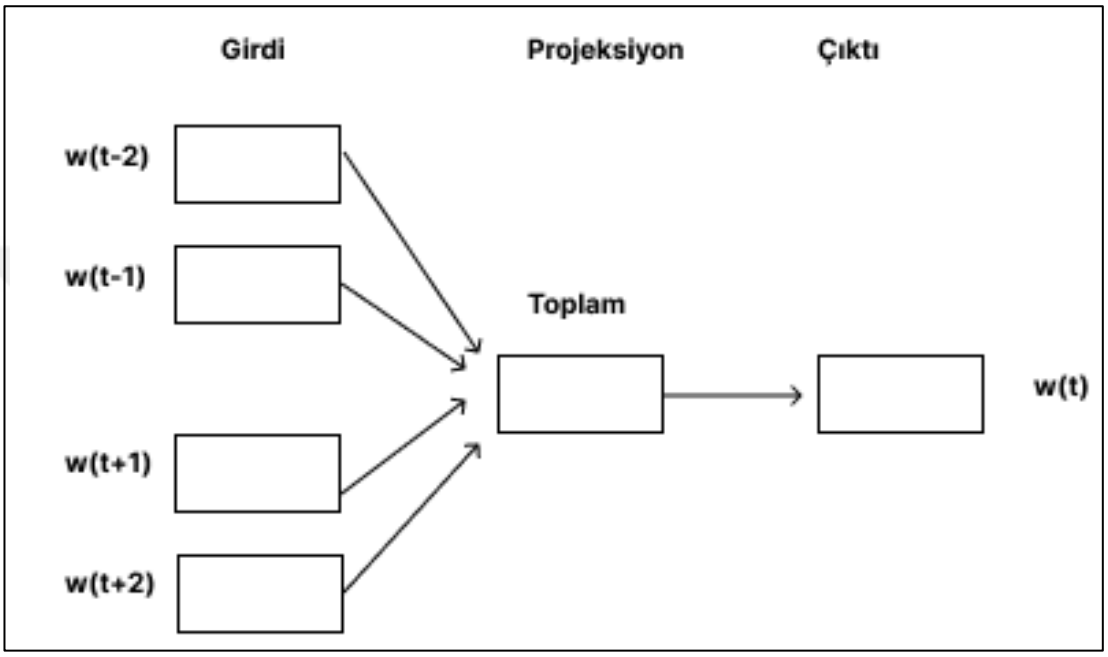
Word2vec, sık kullanılan kelime gömme tekniklerinden bir diğeridir. Tüm derlem taranır ve hedef kelimenin hangi kelimelerle daha sık geçtiği belirlenerek vektör oluşturma işlemi yapılır. Bu sayede kelimelerin birbirine anlamsal yakınlığı da ortaya çıkar. Örneğin; .x y A z w.. , .. x y B z k.. ve ..x l C d m… dizilerindeki her harf bir kelimeyi temsil etsin. Bu durumda kelime_A, kelime_B'ye kelime_C'den daha yakın olacaktır. Vektör oluşumunda bu durum dikkate alındığında kelimeler arasındaki anlamsal yakınlık matematiksel olarak ifade edilmektedir [23].



Şekil 2.19. Anlamsal Yakınlık [25]

Şekil 2.19, Word2Vec'te en sık kullanılan görüntülerden birini göstermektedir. Bu kelimeler arasındaki anlamsal yakınlık, vektör değerlerinin birbirine matematiksel yakınlığıdır. Sıkça verilen örneklerden biri de “kral-adam + kadın = kraliçe” denklemidir. Burada olan vektörlerin birbirinden çıkartılıp eklenmesi sonucu elde edilen vektör değerinin “kraliçe” ifadesine karşılık gelen vektöre eşit olmasıdır. “Kral” ve “kraliçe” kelimelerinin birbirine çok benzediği, ancak vektörel farklılıkların sadece cinsiyetlerinden dolayı ortaya çıktığı anlaşılabilir.

Word2Vec, kelimelerin vektörel temsillerini elde edip kelimeler arasındaki mesafeyi hesaplayarak aralarındaki anlamsal benzerliği tespit etmek için geliştirilmiş bir yöntemdir [26]. Temelinde yapay sinir ağları barındırmaktadır. CBOW ve Skip-Gram olmak üzere iki öğrenme modeli içermektedir. CBOW, hedef kelimenin çevresindeki komşu kelimeleri girdi olarak almakta ve bu kelimelerden hedef kelimeyi tahmin etmeye çalışmaktadır (Şekil 2.26).



Şekil 2.20.CBOW modeli [27]

2.3.4.4. FastText

FastText algoritmasının çalışma mantığı Word2Vec'e benzer ancak en büyük farkı eğitim sırasında N-gram'lık kelimeleri de kullanmasıdır. Bu, modelin boyutunu ve işlem süresini artırırken, modele farklı kelime varyasyonlarını tahmin etme yeteneği de verir. Örneğin, "Kelimeler" kelimesinin eğitim veri setinde olduğunu ve eğitim bittikten sonra "Klimeler" kelimesinin vektörünü almak istediğimizi varsayalım. Bu işlemler için Word2Vec modeli kullanılırsa "Klimeler" kelimesinin sözlükte olmadığı hatası verecek ve herhangi bir vektör döndürmeyecektir. Ancak bu işlem için FastText modeli kullanılırsa hem vektör dönecek hem de "Kelimeler" kelimesi en yakın kelimeler arasında olacaktır. Yukarıda açıklandığı gibi eğitime sadece kelimenin kendisi değil N-gram varyasyonları da dahildir ("Kelimeler" kelimesi için örnek 3 gramlık ifadeler-> Kel, eli, lim, ime). FastText modeli günümüzde birçok farklı alanda

kullanılsa da özellikle kelime gömme tekniklerine ihtiyaç duyulduğunda sıklıkla tercih edilmektedir. FastText, doğrudan kendi sözlüğünde olmayan kelimelerin bile vektörlerini elde etmede büyük avantaj sağlar. Bu nedenle kelime hatalarının oluşabileceği problemlerde diğer alternatiflere göre bir adım öndedir.

2.4. Doğal Dil İşleme Dil Modelleri

2.4.1. Çift Yönlü Dil Modelleme

2.4.1.1. Transfer Öğrenme

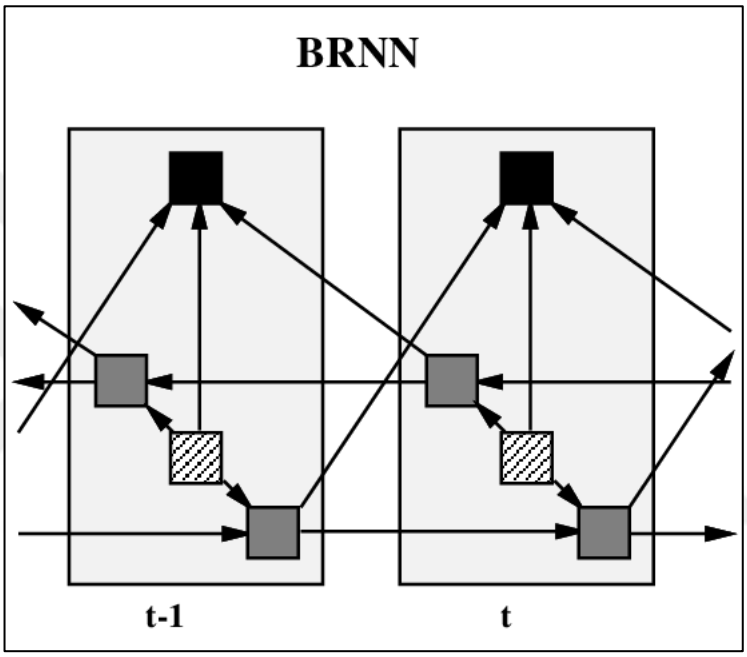
Transfer öğrenimi, büyük metin yapılarında model eğitimini ve elde edilen bilginin sonraki görevlerde kullanılmasını kapsamaktadır. Transfer öğrenimi için transformatör mimarisinin ortaya çıkmasından önce, tek yönlü dil modelleri yaygın olarak kullanılıyordu, ancak bu modeller tek yönlü tekrarlayan sinir ağı (RNN) mimarisine güvenme ve sınırlı bağlam vektör boyutu gibi birçok sınırlamayla karşı karşıya kaldı. Bu boşlukların üstesinden gelmek için, (BERT) iki yönlü kodlayıcı gösterimi gibi iki yönlü dil modelleri tanıtıldı. Çift yönlü dil modelleri, doğal dil çıkarımı, cümle düzeyinde açıklama, soru cevaplama (QA) sistemleri ve simge düzeyinde varlık tanıma gibi çok çeşitli görevlerde uygulanabilir. Başlangıçta, çift yönlü dil modellerinin ön eğitimi, denetimli öğrenme yoluyla yapıldı, ancak insan etiketli veri kümeleri sınırlıdır. Bu sorunu çözmek için büyük bir derlem tabanlı denetimsiz öğrenmenin kullanımı arttı.

2.4.1.2. Çift Yönlü RNN

Temel amaç, normal bir Yinelenen Sinir Ağının durum nöronlarını, pozitif zaman yönünden (ileri durumlar) ve negatif zaman yönünden (geri durumlar) sorumlu olacak şekilde parçalamaktır. İleri durumların çıktıları, geri durumların girişlerine bağlı değildir ve bunun tersi de geçerlidir. Aynı ağda her iki zaman yönünün de halledilmesiyle, şu anda değerlendirilen zaman çerçevesinin geçmişteki ve geleceğindeki girdi bilgileri, normal tek yönlü için olduğu gibi, gelecekteki bilgileri dahil etmek için gecikmelere gerek kalmadan amaç fonksiyonunu en aza indirmek için doğrudan kullanılabilir. BRNN, iki tip durum nöronu arasında herhangi bir etkileşim olmadığından ve bu nedenle genel bir ileri beslemeli ağa açılabileceğinden, prensip olarak normal bir tek yönlü RNN ile aynı algoritmalarla eğitilebilir. Ancak, örneğin,

herhangi bir biçimde zamanda geri yayılım kullanıldığında, durum ve çıktı nöronlarının güncellenmesi artık birer birer yapılamadığı için ileri ve geri geçiş prosedürü biraz daha karmaşıktır. Eğitim verilerinin yalnızca başında ve sonunda bazı özel işlemler gereklidir.

Çift yönlü bir RNN (Şekil 2 21), ileri ve geri bir tekrarlayan sinir ağından oluşur ve görüntüde görülebileceği gibi, herhangi bir t zamanında her iki ağın sonuçları birleştirilerek nihai tahmin yapılır.



Şekil 2.21.BRNN [28]

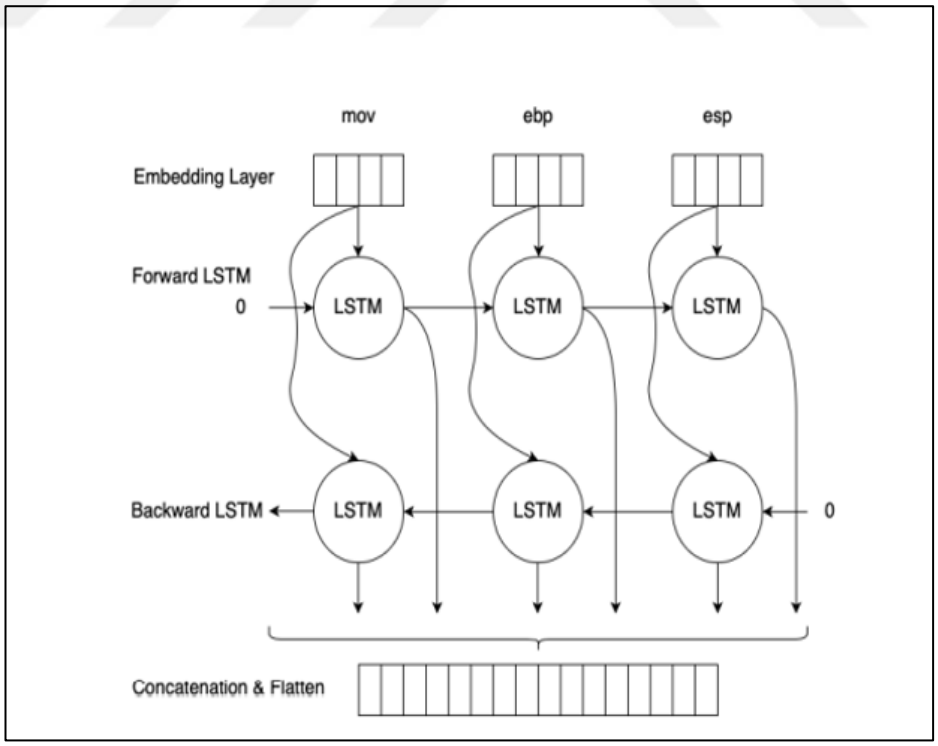
2.4.1.3. Çift Yönlü LSTM

Yinelemeli Sinir Ağları (RNN), değişken uzunlukta ve sıralı verileri işleyebilen bir grup sinir ağıdır. Bilgi ağdan geçerken hafızasında süresiz olarak kalabilir. Bu, sıralı bağımlı şehirleri yakalama sürecini kolaylaştırır. RNN, giriş dizisinin her bir öğesini işledikten sonra bir tahminde bulunur. Bu nedenle, çıkış dizisi, giriş dizisiyle aynı uzunlukta olabilir. RNN mimarisi, önerilen NER görevi için doğal bir model olarak kendini gösterir.

RNN, zaman algoritması yoluyla hata geri yayılımı (Werbos, 1990) ve gradyan iniş algoritmasının bir varyantı kullanılarak eğitilir. Bununla birlikte, bu modellerin eğitimi, özellikle uzun girdi dizileri ile eğitildiğinde, patlayan ve yok olan gradyanlar

sorunu nedeniyle herkesin bildiği gibi zordur (Bengio ve diğerleri, 1994). Patlayan gradyan problemi için gradyanlar kırpılarak sayısal kararlılık sağlanabilir (Graves, 2013). Kaybolan gradyanlar sorunu, standart RNN hücresinin giriş dizisi boyunca sabit bir hata akışına izin veren uzun kısa süreli bellek (LSTM) hücresiyle değiştirilmesiyle ele alınabilir (Hochre iter ve Schmidhuber, 1997). Daha sabit bir hata aynı zamanda ağın giriş dizisi üzerinde daha iyi uzun vadeli bağımlılıkları öğrenebileceği anlamına gelir. Bilgileri zıt yönlerde ileten iki RNN'nin çıktılarını birleştirerek, dizinin her iki ucundan da bağlamı yakalamak mümkündür. Ortaya çıkan mimari, Çift Yönlü LSTM olarak bilinir (Şekil 2 22).

Çift yönlü Uzun Kısa Vadeli Bellek ağları, giriş verilerine iki LSTM'nin uygulandığı açıklanan Uzun Kısa Vadeli Bellek modellerinin bir uzantısıdır. İlk aşamada, ileri katman, giriş dizisi Uzun Kısa Vadeli Bellek modeline beslenir. İkinci turda, geri katmanda, giriş dizisinin ters formuna bir Uzun Kısa Vadeli Bellek modeli uygulanır. Bu yaklaşım, uzun vadeli bağımlılıkların öğrenilmesini geliştirir, modelin bir cümlede hangi kelimelerin hemen ardından ve bir kelimeden önce geldiğini bilmesini sağlar ve böylece modelin doğruluğunu artırır [29].

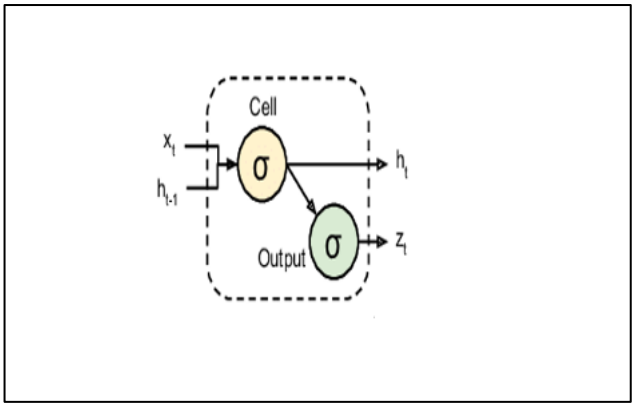


Şekil 2.22. Çift Yönlü LSTM [29]

2.4.2. Yinelemeli Sinir Ağları (RNN)

Yinelemeli Sinir Ağları (RNN), değişken uzunlukta ve sıralı verileri işleyebilen bir grup sinir ağıdır. Başka bir deyişle, Tekrarlayan Sinir Ağları, değişken uzunlukta dizi girişleri ile çalışmak için geleneksel ileri beslemeli sinir ağlarının genişletilmiş bir versiyonudur (Şekil 2.23). Yinelemeli Sinir Ağları her zaman adımında çıktı üretebilmektedir.

Doğal Dil İşleme görevleri hem yazılı metin hem de konuşma olabilen dil verileriyle ilgilenir ancak her ikisi de sıralıdır. Kelime düzeyindeki dil modellerinde, zaman adımı her kelimenin konumudur. Sonuç olarak, bu avantaj nedeniyle Yinelemeli Sinir Ağları dil modellemede diğer algoritmalara göre daha iyidir. Yinelemeli Sinir Ağları, uzun seriler için önceki sıralı bilgilerden yararlanmaya odaklansa da bellek sınırlamaları nedeniyle, sıralı bilgilerin boyutu birkaç aşama geriye indirgenir. Bu problem, "yok olan gradyanlar" olarak bilinir. Bu sorun, RNN'lerin uzun vadeli bağımlılıkları yakalamasını zorlaştırır ve bu nedenle, Yinelemeli Sinir Ağlarının eğitimi oldukça zor olacaktır. RNN'leri eğitmeyi zorlaştıran bir diğer sorun da "patlayan gradyanlar" olarak bilinir [30].

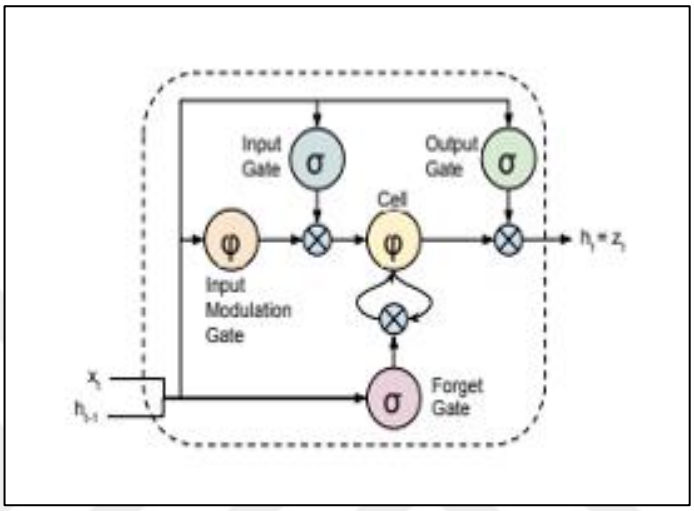


Şekil 2.23. RNN [30]

2.4.3. Uzun Kısa Vadeli Bellek Ağları (LSTM)

LSTM Hochreiter & Schmidhuber tarafından kaybolan/patlayan gradyan probleminin üstesinden gelmek için 1997 yılında tanıtılmıştır. Uzun Kısa Vadeli Bellek Ağları modeli, kaybolan gradyan problemini ele almak için girdilerin uzun vadeli bağımlılıklarını depolamak ve öğrenmek için RNN'lerin hafızasını artırarak

modeli genişletir [31] ve Yinelemeli Sinir Ağı Modellerine bilgiyi geri çağırma ve bilgiyi depolamaya veya atmaya karar verme yeteneği verir. Eğitim sürecinde bilgilere atanan ağırlık değerleri bu kararı etkiler. Bu nedenle LSTM modelleri, verileri RNN Modellerinden daha uzun süre saklar. Ağırlıklar, bir LSTM modelinin hangi bilgilerin korunması gerektiğini veya hangi bilgilerin kaldırılması gerektiğini öğrenmesini sağlar [32].



Şekil 2.24. LSTM [33]

Uzun Kısa Vadeli Bellek ağlarının, standart RNN'lerden daha uzun vadeli bağımlılıkları oluşturma ve öğrenmede daha iyi olduğu yapılan deneylerle gösterilmiştir. Uzun Kısa Vadeli Bellek ağları üç kapıdan yararlanır: giriş kapısı, unutma kapısı ve çıkış kapısı. Bu kapıları kullanarak, Uzun Kısa Vadeli Bellek ağları artık ihtiyaç duyulmayan bilgileri kaldırabilir ve ayrıca bağlam için önemli olan bilgileri ekleyebilir (Şekil 2.24).

2.5. Doğal Dil İşleme Adımları (Pipeline)

Cümle Segmentasyonu: Yapılacak ilk adım, metni ayrı cümlelere ayırmaktır. Tek bir cümleyi anlamak için bir program yazmak, bütün bir paragrafı anlamaktan çok daha kolay olacaktır.

Kelime Tokenizasyonu: Metin cümlelere bölündükten sonra birer birer işlenebilir. Boru hattımızdaki bir sonraki adım, bu cümleyi ayrı kelimelere veya belirteçlere ayırmaktır. Buna tokenizasyon denir. Kelimeler aralarında boşluğa göre

ayrılabilir. Ayrıca noktalama işaretlerinin de anlamı olduğu için noktalama işaretlerini ayrı belirteçler olarak ele alınmalıdır.

Her Token için Konuşma Bölümlerini Tahmin Etme: Daha sonra, her bir simgeye bakılır ve simgenin bir isim mi, bir fiil mi, bir sıfat mı olduğunu tahmin etmeye çalışılır. Cümledeki her kelimenin rolünü bilmek, cümlenin neden bahsettiğini anlamaya yardımcı olacaktır. Bu, her bir kelimeyi (ve bağlam için etrafındaki bazı ekstra kelimeleri) önceden eğitilmiş bir konuşma parçası sınıflandırma modeline besleyerek yapılabilir.

Metin Çözümleme: Bir bilgisayarda metinle çalışırken, her iki cümlenin de aynı kavramdan bahsettiğini bilmeniz için her kelimenin temel biçimini bilmek yardımcı olur. NLP'de, bu sürece kök çözümleme- cümledeki her kelimenin en temel biçimini bulma olarak adlandırılır. Aynı şey fiiller için de geçerlidir. Ayrıca fiilleri köklerini, çekimsiz formlarını bularak çözümlenir.

Durdurulan Sözcükleri Tanımlama: Daha sonra, cümledeki her bir kelimenin önemi ele alınır. İngilizce'de "and", "the" ve "a" gibi çok sık kullanılan birçok dolgu kelimesi vardır. Metin üzerinde istatistik yaparken, bu kelimeler diğer kelimelerden çok daha sık göründükleri için çok fazla gürültü çıkarır. Bazı NLP ardışık düzenleri onları durdurma sözcükleri, yani herhangi bir istatistiksel analiz yapmadan önce filtreleme ihtiyacı olan sözcükler olarak işaretler.

Bağımlılık Ayrıştırma: Bir sonraki adım, cümlemizdeki tüm kelimelerin birbirleriyle nasıl ilişkili olduğunu bulmaktır. Buna bağımlılık ayrıştırma denir.

Amaç, cümledeki her kelimeye tek bir ana kelime atayan bir ağaç oluşturmaktır. Ağacın kökü cümledeki ana fiil olacaktır.

Adlandırılmış Varlık Tanıma (NER): Kişi, yer, organizasyon gibi önceden tanımlanmış kategorilerin metin dokümanları üzerinden çıkarılma işlemidir. Bilgi çıkarımının bir alt dalı olup makine çevirilerinden tutun da duygu analizine kadar birçok Doğal Dil İşleme probleminde kullanılmaktadır.

2.6. Algoritma Performans Değerlendirme Parametreleri

Herhangi bir makine öğrenimi modeli için, modele "iyi bir uyum" sağlamak son derece önemlidir. Bu, önyargı ve varyans arasındaki bir dengeyi sağlamamakla mümkün olmaktadır.

Kesinlik (Presicion), Gerçek Pozitifler ile tüm Pozitifler arasındaki orandır (Şekil 2.25).

$$Precision = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Positive(FP)}$$

Şekil 2.25. Kesinlik

Geri çağırma (Recall) modelin Gerçek Pozitifleri doğru bir şekilde tanımlamasının ölçüsüdür (Şekil 2.26).

$$Recall = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Negative(FN)}$$

Şekil 2.26. Geri Çağırma

Doğruluk (Accurarcy), toplam doğru tahmin sayısı ile toplam tahmin sayısının oranıdır. Yorumlaması kolaydır. Verilerin nasıl dağıtıldığını hesaba katmaz (Şekil 2.27).

$$\text{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{(True Positive + False Positive + True Negative + False Negative)}}$$

Şekil 2.27. Doğruluk

F1 puanı, kesinlik ve geri çağırmanın harmonik ortalamasıdır. Verilerin nasıl dağıtıldığı önemlidir. Veriler dengesiz dağılmış ise, F1 puanı model performansının

daha iyi bir değerlendirmesini sağlar. F1 puanı, modelin kesinliği ve geri çağrılmasının bir karışımıdır ve bu da yorumlamayı biraz daha zorlaştırır (Şekil 2.28).

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Şekil 2.28.F-1 puanı

2.7. Siber Zorbalık Tespiti

2.7.1. Siber Zorbalık Nedir

Siber zorbalık, internet teknolojisinin gelişmesi ve sosyal ağlara erişim kolaylığı ile büyük bir sorun haline gelmiştir. Siber zorbalık, dijital teknolojilerin kullanımıyla zorbalıktır. Bir kişi veya grup tarafından siber zorbalık, başkalarını taciz etmek amacıyla bilgi ve iletişim teknolojilerinin kötü yönde kullanılması anlamına gelir. Sosyal medya platformları, insanların birbirleriyle iletişim kurmaları, güncel olaylardan haberdar olmaları ve sosyallik kazanmaları için faydalı araçlardır. Bununla birlikte bazen kötü niyetli kullanıcıların elinde tehlikeli bir araca dönüşebilir. Siber Zorbalık, sosyal medyada, mesajlaşma platformlarında, oyun platformlarında ve cep telefonlarında yer alabilir. Siber zorbalığa örnek olarak alay etme, tehdit etme ve taciz etme gibi tekrarlayan ve korkutmayı, kızdırmayı, utandırmayı amaçlayan siber zorbalık davranışları gösterilebilir.

2.7.2. Siber Zorbalık Türleri

Karalama (Aşağılama): Siber zorbalığın en yaygın türlerinden biri olarak belirtilebilir. Genellikle ergenlik çağında sorunlu bireylerin iletişim tarzlarını kullanmaları sonucu ortaya çıkar. Bir kişi veya grup hakkında yalan haber paylaşmak veya elektronik mesaj göndermek olarak tanımlanmaktadır. Özellikle son dönemde sıkça karşılaşılan bir yöntemdir. Bu durumun temel nedeni, gençler arasında artan sosyal medya kullanımı olarak belirtilmektedir.

Kimliğe Bürünme: Kimliğe bürünme, birinin başka bir kişinin adına sahte bir profil oluşturması veya başka bir kişinin hesabını hacklemesidir. Siber zorba,

çevrimiçi ortamda kurbanı gibi davranır ve kurbanın itibarını zedeler. Daha çok Facebook, Twitter ve Instagram gibi sosyal medya platformlarında görülmektedir. Siber zorba, zarar vereceği kişinin sahte hesabını oluşturarak siber suç mağduru gibi davranarak paylaşımlarda bulunabilir. Bu sayede siber zorba, mağduru zor durumda bırakabilir ve diğer kişilere uygunsuz mesajlar gönderebilir [34,35].

İfşalama: Siber zorba, kurbanı incitmek veya aşağılamak amacıyla özel bilgileri izinsiz olarak dağıtmak ve yaymak için teknolojik araçları kullandığında gerçekleşir.

Hile: İfşalama ile bazı ortak özelliklere sahiptir. Ancak farklı olarak, güven kazanma ve bu güveni manipüle etme durumu vardır. Kişi internette tanıştığı birine güvenir ve ortaya çıktığında utanabileceği bilgi veya görüntüleri paylaşır. Güvendiği kişi, mağdurun güvenini kötüye kullanarak onun gizli bilgilerini paylaşır. Siber zorba bilgiyi elde ettikten sonra, "kurbanı utandırmak amacıyla" alenen başkalarına yayarak mağdura karşı kullanacaktır [34,35].

Taciz: Taciz, küfretme, müstehcen video görüntüsü gönderme veya hakaret içeren metin mesajları olarak tanımlanır. Taciz uzun süreli devam edebilir. Taciz, zorba, elektronik iletişim biçimleri aracılığıyla hedefine saldırgan ve tehdit edici mesajlar gönderdiğinde ortaya çıkar. Hatta birden fazla kişi, kurbana aynı anda binlerce mesaj göndermek için bir araya gelebilir [34,35].

Siber Takip: Hakaret içeren materyaller veya ses/görüntü kayıtları, görüntüler içeren elektronik mesajlarla mağduru küçük düşürerek mağduru korkutan siber zorbalık türüdür. Siber zorba adres bilgilerini isteyerek kurbanı yaralayacağını, öldüreceğini veya döveceğini söyleyerek korkutur. Mağdurlar genellikle siber zorbalardan tehditkâr ve göz korkutucu elektronik mesajlar alırlar. Mağdurlar genellikle "gözdağı verenin çevrimdışı olabileceğine ve onlara fiziksel olarak zarar verebileceğine" inanmaya başlayabilir, bu da onların çevrelerinden de aşırı derecede şüphelenmelerine neden olabilir [34,35].

2.7.3. Siber Zorbalığın Olumsuz Etkileri

Düşük Benlik Kaygısı: Siber zorbalık, kurbanların özgüvenini etkiler. Çevrimiçi olarak istismara uğrayan bireyler, içedönük hale gelebilir ve düşük öz

saygıya sahip olmaya başlayabilir. Bu, özellikle bu tür tacizlerin fiziksel görünümlerine ve görünümlerine yönelik olduğu durumlarda geçerlidir. Görünüşleri nedeniyle alay edilen gençler ve genç yetişkinler vücutlarından hoşlanmamaya başlayabilir. Bu, kendilerini başkalarına daha az çekici olarak gören bu bireylere dönüşecektir. Toplum içinde görünmekten veya başkalarıyla sosyal olarak etkileşimde bulunmaktan utanmaya başlayabilirler.

Başarıda Düşüş: Siber zorbalığın etkileri genellikle bireyin yaşamının her alanında kendini gösterir. Siber zorbalığa maruz kalan bireyler sosyal hayatlarında ve iş hayatlarında başarısız olma eğilimindedirler. Öğrenciler için bu, notlarında açıkça görülebilir.

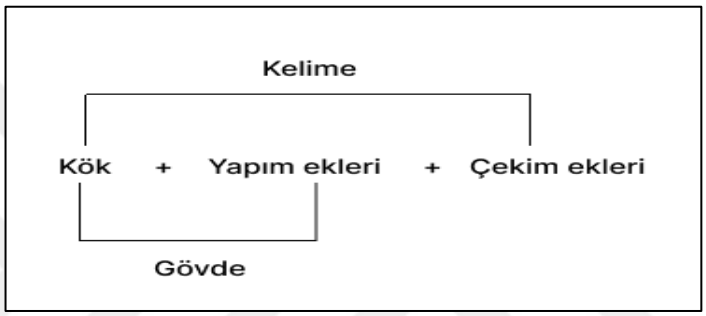
Depresyon: Siber zorbalık kurbanları genellikle stresle ilgili çeşitli koşullara maruz kalır. Kaygı bozukluğu ve depresyon bu koşullardan sadece birkaçıdır. Kurbanlar kendilerine olan güvenlerini kaybetmeye başladıklarında, bu onların ruh sağlıklarına zarar verir. Ek olarak, çevrimiçi zorbalığın yükünü taşımak onları mutsuz eder. Birey üzüntü ve hoşnutsuzlukla karşılaştığında, zamanla depresyona girebilir.[36]

İntihar Eğilimleri: Araştırmalar, siber zorbalığın bazı kurbanlarda intihar eğilimlerini teşvik edebileceğini gösteriyor. Umutsuzluk duygusuna kapılan bireyler, internette sürekli taciz edildiğinde, bireyler ölümü tek kaçış yolu olarak görmeye başlayabilir.

Sık Sık Hastalanma: Siber zorbalığa uğramanın getirdiği stres, bir kişinin hastalanmasına neden olabilir. Uykusuzluk, baş ağrıları, göğüs ağrısı ve diğer fiziksel komplikasyonlar doğrudan çevrim içi tacizden kaynaklanabilir. Bu stres, aynı zamanda, bireyin benlik saygısını daha da aşındıran cilt komplikasyonlarına ve kopmalara yol açabilir. Siber zorbalığa uğrayan kişiler, bir başa çıkma mekanizması olarak yeme bozuklukları geliştirebilir. Çok zayıf göründükleri söylendiğinde kilo verebilir veya anormal biçimde yemek yiyebilirler. Bu eylemlerin sağlık üzerinde olumsuz etkileri vardır [36].

2.8. Türk Dili Modellemenin Zorlukları

Morfolojik olarak zengin diller, özellikle Türkçe, hesaplamalı bir NLP modeli oluştururken kendi dil zorluklarına sahiptir [37,38]. Analitik bir dil olan Türkçe'ye benzer morfolojik açıdan zengin diller İngilizce'nin aksine sondan eklemeli karmaşıklığa sahiptir. Dilin en basit üyesi olan kelime, temelde bir kök ve bağlamı oluşturan çekim eklerinden oluşur. İngilizce benzeri analitik diller, bu bağlamı oluşturmak için ayrı edatlardan yararlanır ve doğal olarak, Türkçe gibi morfolojik dillere kıyasla analizleri daha kolaydır. Türkçenin morfolojik yapısı Şekil 2.32'de gösterilmiştir (Şekil 2.29).



Şekil 2.29. Türk dilinin morfolojik yapısı [39]

Türkçe 'de kelimeleri, kökler, ön ekler ve ekler oluşturulmaktadır ve bundan dolayı tek bir sözcük birçok karşılık gelen çekim biçimine sahip olacaktır [41]. Kökler, türetme ekleri ile gövdeye dönüştürülür. Türetim ekleri kökün/gövdenin anlamını değiştirirken, çekim ekleri sözcüğün yerini, zamanını ya da özünü verir. Türkçede çekim ekleri genellikle yapım eklerinden sonra gelir. Oflazer, ayrıntılı çalışmasında Türkçenin bu muazzam kelime üretme yeteneğini ve "dil işlemedeki zorluklarını" göstermiştir [41,42]. Tek bir kelimeye eklenebilecek eklerin sayısı ve bunların kombinasyonları, olası türetmelerden asıl kök elde etmek için ciddi bir dil analizi sorunu oluşturmaktadır.

Türkçe NLP çalışmalarının hedef probleme bir çözüm modellemeden önce dil işleme görevleriyle ilgilenmesi gerektiği söylenebilir. Türkçenin sondan eklemeli bir dil olması ve fiillerin çok sayıda eke sahip olması Doğal Dil İşleme problemlerini arttırmaktadır. Bu problemler, örneğin İngilizce için geliştirilmiş en son modellerin ve algoritmaların uygulanmasını kısıtlar. Türkçe 'deki veri azlığının üstesinden gelmek

için, NLP ardışık düzenlerinden önce, köklendirme veya kök çözümleme gibi ön işleme görevleri başlatılmalıdır [43,44]. Hem köklendirme hem de kök çözümleme, kelimelerin çekim veya türetme biçimlerini ortak bir temel biçime indirgeme amacına sahiptir. Köklendirme, temel bir biçim elde etmek için türetilen kelimelerin uçlarını kesen buluşsal bir süreç iken, kök çözümleme, kelimenin sözlük biçimini elde etmek için bir kelime hazinesi ve morfolojik analizden yararlanır [45]. Ancak, iki yöntem birbirinin yerine geçemez ve karşılık gelen dil sorunu için hangisinin daha iyi olduğu dikkatle incelenmelidir.

Köklendirme ve kök çözümleme dışında geleneksel ön işleme görevleri aynı zamanda duran kelime kaldırma (stop-word) kaldırma, küçük harf dönüştürme ve simgeleştirme görevlerini de içerebilir. Edatlar, zamirler ve bağlaçlar bitiş sözcükleridir ve bu (o), orda (orada) bazı örnek Türkçe sözcüklerdir. Sözcüksel olarak, belirteçleştirme, girdi metin dizesini sözcükler veya deyimler gibi anlamlı parçalara (belirteçler) bölme işlemi olarak tanımlanır. Pek çok Türk NLP görevi, gelişmiş bir ML modeli elde etmek için çeşitli özellik işleme stratejilerinden de yararlanır. Pratik olarak, morfolojik olarak zengin diller için geliştirilen geleneksel ML algoritmaları, ön işleme adımlarından elde edilen belirteç tabanlı dil modellerine ihtiyaç duyar. Geleneksel ML modelleri için ön işleme görevleri gereklidir. Belirteç tabanlı modeller ön işleme adımlarından sonra oluşturulsa bile, yukarıda belirtilen veri azlığı sorunu başka bir sorundur. Başka bir deyişle, morfolojik işleme görevleri, eğitim derleminde çok nadir veya muhtemelen var olmayan kelime formları ürettiğinden, tahmin aşamasında ML algoritmalarının performansı düşecektir [45]. Özet olarak, Türk NLP problemlerinin, veri azlığı ve ilgili problemleri işe alırken verimli ML modelleri elde etmek için birkaç ön işleme adımına ihtiyaç duyduğu gözlemlenebilir. Ayrıca, morfolojik olarak zengin diğer diller de verimli ML modelleri üretirken Türkçe'ye benzer morfolojik işleme görevleri gerektirecektir. Genel problemler için çözüm vaka çalışmamız için muhtemelen çift yönlü sinirsel dil modellerinin veya BERT'nin değerlendirilmesidir.

3. LİTERATÜR TARAMASI

Siber zorbalığı otomatik olarak tespit etmek için yapılan önceki çalışmaların çoğu İngilizce' diline aittir. Türkçe için yapılmış çalışmalar sınırlıdır. Siber zorbalığın otomatik tespiti ile ilgili ilk çalışma 2009 yılında yapılmıştır [46]. Ağustos 2008 itibariyle Perverted Justice (PJ) web sitesinde bulunan 288 sohbet günlüğünü indirerek ilk veri setini geliştirmiştir. Bu veri setini internet avcılarını kategorize etmek için kullanmışlardır. Bu çalışmada kullanılan veri seti İngilizce diline aittir. Bu veri setini kullanarak iki set metin madenciliği deneyi gerçekleştirdiler. İlk deney, iletişim stratejilerini kategorize etmeye çalışır ve yırtıcı ile kurbanı veya yırtıcı ile normal sohbeti ayırt etmeye çalışır. İkinci deneyde, çocukları cezbetmek için farklı iletişim stratejilerinin kullanılıp kullanılmadığını belirlemek için kümeleme kullanılır. Kontostatis ve arkadaşları (2010) ikinci çalışmalarında, derlemlerini Formspring.me'den bir gönderi koleksiyonu olarak alıyorlar. Siber zorbalığı tespit etmek için bu veri setini kullandılar. Sorgular zorbalık terimleriyle genişletilir. Her gönderi Amazon Mechanical Turk tarafından etiketlenir. Kontostatis ve ark. (2010), zorbalık terimlerini bulmak için Gizli Semantik İndeksleme ve Tekil Değer Ayrıştırma'yı kullandı. Siber zorbalık tespiti için %91,25'lik bir başarı oranı elde ettiler.

2011 yılında, metinsel siber zorbalığı tespit etmek için bir dizi ikili ve çok sınıflı sınıflandırıcı uygulayarak 4500 YouTube yorumundan oluşan bir külliyat kullandı. YouTube yorumlarını manuel olarak etiketlediler ve belgeleri sınıflandırmak için Naive Bayes, Kural tabanlı JRip, Ağaç tabanlı J48 ve SVM algoritmalarını kullandılar. JRip doğruluk açısından en iyi performansı verirken, kappa istatistiği ile ölçülen SVM en güvenilir sınıflandırıcıdır ve siber zorbalığın tespitinde %66,7 doğruluk elde edilmiştir. Çalışmaları, ikili sınıflandırıcılar oluşturmanın, bu tür hassas mesajları tespit etmede çok sınıflı sınıflandırıcılardan daha etkili olduğunu göstermektedir [47].

Naïve Bayes sınıflandırıcısı ile siber zorbalığı tespit etmek için Twitter yorumlarını kullanılmıştır. İngilizce dili için twitter veri setinde cinsiyete özel bir zorbalık tespiti gerçekleştirilir. Naïve Bayes sınıflandırıcı ile %67,3 doğruluk değerleri elde edilmiştir.

Dadvar ve arkadaşları (2013) uzman bilgisi aracılığıyla YouTube kullanıcılarının davranışlarını ve özelliklerini daha iyi anlamak için çok kriterli bir değerlendirme sistemi kullanmıştır. Tüm kullanıcılara, sistem tarafından önceki faaliyetlerine göre verilen puanlar atanır. Bu puanlar onların siber zorbalık seviyelerini gösterir. Puanların, bir kullanıcının zorbalık yapıp yapmadığına karar vermede yardımcı olduğu bulunmuştur. Puanlar, zorbalık geçmişi olan kullanıcılar ile incitici eylemlerde bulunmamış kullanıcılar arasında ayrım yapmak için kullanılabilir ve bir kullanıcının zorbalık yapıp yapmadığına karar vermede yardımcı olabilir [48].

4. DENEYSEL ÇALIŞMALAR

4.1. Problem Tanımı

Türkçe gibi morfolojik açıdan zengin olan dillerde Doğal Dil İşleme çalışmaları Makine Öğrenmesi tabanlı algoritmalarla yapılmaktadır. Bu algoritmalar ile Doğal Dil İşleme çalışmaları yapılırken, ilk olarak dilin makinenin anlayabileceği bir biçimde temsil edilmesi gerekmektedir. Dolaysıyla, dil çeşitli ön işleme teknikleri kullanılarak analiz edilir. Bu adımların kombinasyonları veya varyasyonları ve Türkçe veri azlığı, tahmin verimliliği açısından elde edilen sonuçları değiştirir. Doğal Dil İşleme için Google tarafından BERT algoritması alternatif olarak tanıtılmıştır. BERT algoritması ön eğitimli bir algoritma olmasında dolayı, böyle bir ön işleme adımı gerekli değildir ve çoğu görevin analizi en son sonuçlarla yapılır.

4.2. BERT Algoritması

BERT, Doğal Dil İşleme için bir makine öğrenimi algoritmasıdır. BERT, bağlam oluşturmak için çevreleyen metni kullanarak bilgisayarların metindeki belirsiz dilin anlamını anlamasına yardımcı olmak için tasarlanmıştır. BERT algoritması, ön eğitimli bir algoritmadır ve soru-cevap veri kümeleriyle ince ayar yapılabilmektedir. Her çıkış ögesinin her giriş ögesine bağlı olduğu ve aralarındaki ağırlıkların bağlantılarına göre dinamik olarak hesaplandığı bir derin öğrenme modeli olan dönüştürücüleri (transformers) temel alır [49].

Tarihsel olarak incelenecek olursa, dil modelleri metin girişini yalnızca sırayla- soldan sağa veya sağdan sola- okuyabilmekteydi, fakat ikisini aynı anda yapamazdı. BERT, aynı anda her iki yönde de okumak üzere tasarlandığından diğer algoritmalardan ayrılmaktadır. Bert, çift yönlü bir algoritma olarak da bilinmektedir. Bert Algoritması, iki farklı ancak birbirleri ile bağlantılı Maskeli Dil Modelleme (MLM) ve Sonraki Cümle Tahmini (NSP) olan Doğal Dil İşleme görevleri üzerine ön eğitimli bir algoritma olarak geliştirilen bir algoritmadır. Maskeli Dil Modeli (MLM) eğitiminin amacı, bir cümlede bir kelimeyi gizlemek ve ardından programın, gizli kelimenin bağlamına dayalı olarak hangi kelimenin gizlendiğini (maskelendiğini) tahmin etmesini sağlamaktır. Sonraki Cümle Tahmini eğitiminin amacı, programın

verilen iki cümlenin mantıksal, sıralı bir bağlantısı olup olmadığını veya ilişkilerinin basitçe rastgele olup olmadığını tahmin etmesini sağlamaktır [49,50].

4.2.1. Arka Plan

Transformatörler ilk olarak Google tarafından 2017'de tanıtılmıştır. O yıllarda, dil modellerinin Doğal Dil İşleme görevlerini yerine getirmesi için Tekrarlayan Sinir Ağları (RNN) ve Evrişimli Sinir Ağları (CNN) kullanılmaktaydı. Tekrarlayan Sinir Ağlarının ve Evrişimli Sinir Ağlarının, eğitim yaparken herhangi bir sabit sırada işlenecek veri dizilerine ihtiyaç duymamasından dolayı eğitilen veri miktarı transformatörlere göre sınırlıydı. Transformatörler, verileri herhangi bir sırayla işleyebildiğinden, daha önce hiç olmadığı kadar büyük miktarda veri üzerinde eğitime olanak tanımaktadır. Bu da piyasaya sürülmeden önce büyük miktarda dil verisi üzerinde eğitilmiş olan BERT gibi ön eğitimli modellerin oluşturulmasını kolaylaştırmıştır.

2018'de Google, BERT'i tanıttı. Algoritma, araştırma aşamalarında, duygu analizi, anlamsal rol etiketleme, cümle sınıflandırması ve çok anlamlı kelimelerin veya çok anlamlı kelimelerin belirsizliğini giderme dahil olmak üzere 11 doğal dil anlama görevinde kayda değer sonuçlar elde etti. Bu sonuçlara bağlı olarak BERT algoritması, bağlam ve çok anlamlı sözcükleri yorumlarken sınırlı olan word2vec ve GloVe gibi önceki dil modellerinden ayrıldı. Ekim 2019'da Google, Amerika Birleşik Devleti merkezli arama algoritmalarına BERT algoritmasını uygulamaya başlayacaklarını duyurdu. Aralık 2019'da BERT, 70'ten fazla farklı dile uygulandı [50,51].

4.2.2. Bert Algoritması Nasıl Çalışır

Herhangi bir Doğal Dil İşleme tekniğinin amacı, doğal olarak konuşulan insan dilini anlamaktır. Bunu yapmak için, modellerin genellikle geniş bir özel, etiketli eğitim verisi deposu kullanarak eğitmesi gerekir. Bu, veri etiketlemesini gerektirir. Ancak BERT, yalnızca etiketlenmemiş, düz metin bir bütünce (Vikipedi'nin tamamı ve Brown Corpus'un tamamı) kullanılarak ön eğitimli hale getirilmiştir. Pratik uygulamalarda kullanılsa bile etiketlenmemiş metinden denetimsiz olarak öğrenmeye ve gelişmeye devam etmektedir. BERT algoritmasının ön eğitimi, inşa edilecek bir "bilgi" temel katmanı olarak hizmet eder. BERT sürekli büyüyen aranabilir içerik ve

sorgulara uyum sağlayabilir ve bir kullanıcının özelliklerine göre ince ayar yapabilir. Bu süreç transfer öğrenme olarak bilinir. BERT, Google'ın transformatörler üzerine yaptığı araştırma sayesinde mümkün olmuştur. Transformatör, BERT'e dilde bağlamı ve belirsizliği anlama kapasitesini artıran modelin parçasıdır. Transformatör bunu, herhangi bir kelimeyi birer birer işlemek yerine, bir cümledeki diğer tüm kelimelere göre işleyerek yapar. Transformatör, çevreleyen tüm kelimelere bakarak, BERT modelinin kelimenin tam bağlamını anlamasını ve arama amacını daha iyi anlamasını sağlar. Metin girişini sırayla (soldan sağa veya sağdan sola) okuyan yönlü modellerin aksine, Transformatör kodlayıcı tüm kelime dizisini bir kerede okur. Bu özellik, modelin bir kelimenin bağlamını tüm çevresi (kelimenin solu ve sağı) temelinde öğrenmesini sağlar. Girdi, önce vektörlere gömülü olan ve daha sonra sinir ağında işlenen bir dizi belirteçtir. Çıktı, her vektörün aynı indekse sahip bir giriş belirtecine karşılık geldiği H boyutunda bir vektör dizisidir. BERT iki eğitim stratejisi kullanır:

Maskeli Dil Modeli: Kelime dizilerini BERT'e beslemeden önce, her dizideki kelimelerin %15'i bir [MASK] belirteci ile değiştirilir. Ardından model, dizideki diğer, maskelenmemiş kelimeler tarafından sağlanan bağlama dayalı olarak, maskelenmiş kelimelerin orijinal değerini tahmin etmeye çalışır. Teknik terimlerle, çıktı kelimelerinin tahmini şunları gerektirir:

- Kodlayıcı çıktısının üstüne bir sınıflandırma katmanı ekleme.
- Çıktı vektörlerini gömme matrisi ile çarpmak, onları kelime boyutuna dönüştürmek.
- Softmax ile kelime dağarcığındaki her kelimenin olasılığını hesaplama.

BERT kayıp fonksiyonu, yalnızca maskelenmiş değerlerin tahminini dikkate alır ve maskelenmemiş sözcüklerin tahminini yok sayar. Sonuç olarak, model yönlü modellere göre daha yavaş yakınsar; bu, artan bağlam farkındalığıyla dengelenen bir özelliktir [50,51].

Sonraki Cümle Tahmini: BERT eğitim sürecinde, model girdi olarak cümle çiftlerini alır ve çiftteki ikinci cümlenin orijinal belgedeki sonraki cümle olup olmadığını tahmin etmeyi öğrenir. Eğitim sırasında, girdilerin %50'si orijinal belgede ikinci cümlenin sonraki cümle olduğu bir çifttir, diğer %50'sinde ise derlemden rastgele bir cümle ikinci cümle olarak seçilir. Varsayım, rastgele cümlenin ilk

cümleden ayrılacağıdır. Modelin eğitimdeki iki cümleyi ayırt etmesine yardımcı olmak için girdi, modele girmeden önce aşağıdaki şekilde işlenir:

- İlk cümlenin başına bir [CLS] belirteci ve her cümlenin sonuna bir [SEP] belirteci eklenir.
- Her simgeye Cümle A veya Cümle B'yi belirten bir cümle gömme eklenir. Cümle yerleştirmeleri, kavram olarak 2 kelime dağarcığına sahip simge yerleştirmelerine benzer [50,51].
- Sıradaki konumunu belirtmek için her simgeye konumsal bir yerleştirme eklenir. Konumsal gömme kavramı ve uygulaması transformatör belgesinde sunulmaktadır.

İkinci cümlenin gerçekten birinciye bağlı olup olmadığını tahmin etmek için aşağıdaki adımlar gerçekleştirilir:

- Tüm giriş dizisi transformatör modelinden geçer.
- [CLS] belirtecinin çıktısı, basit bir sınıflandırma katmanı (öğrenilmiş ağırlık ve önyargı matrisleri) kullanılarak 2x1 şekilli bir vektöre dönüştürülür.
- Softmax ile Bir Sonraki Cümle “Mi” olasılığının hesaplanması.

BERT modeli eğitilirken, Maskeli Dil Modeli ve Sonraki Cümle Tahmini, iki stratejinin birleşik kayıp fonksiyonunu en aza indirmek amacıyla birlikte eğitilir.

4.2.3. BERT (İnce Ayar) Nasıl Kullanılır

BERT, temel modele yalnızca küçük bir katman eklerken çok çeşitli dil görevleri için kullanılabilir:

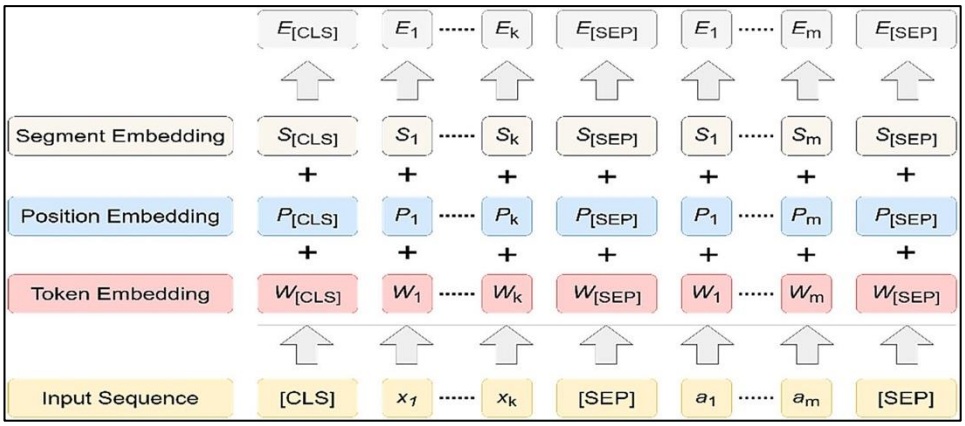
1. Duyarlılık analizi gibi sınıflandırma görevleri, [CLS] belirteci için Transformer çıktısının üstüne bir sınıflandırma katmanı eklenerek Sonraki Cümle sınıflandırmasına benzer şekilde yapılır.
2. Soru Yanıtlama görevlerinde, yazılım bir metin dizisiyle ilgili bir soru alır ve yanıtı sırayla işaretlemesi gerekir. BERT kullanarak, bir Soru-Cevap modeli, cevabın başlangıcını ve sonunu işaretleyen iki ekstra vektör öğrenilerek eğitilebilir.
3. Adlandırılmış Varlık Tanıma'da (NER), yazılım bir metin dizisi alır ve metinde görünen çeşitli varlık türlerini (Kişi, Kuruluş, Tarih, vb.) işaretlemesi gerekir.

BERT kullanılarak, bir NER modeli, her belirtecin çıktı vektörünü NER etiketini öngören bir sınıflandırma katmanına besleyerek eğitilebilir [51,52,53].

4.2.4. BERT Denetimsiz Ön Eğitim Görevleri

Sinirsel dil modelleme yaklaşımlarının çoğu, öğrenilen bilgiyi diğer görevlerden öğrenme kavramını kullanır. NLP görevlerinin akışını sağlamak için önceden eğitilmiş dil modellerini kullanırken, iki yaklaşım vardır; özellik tabanlı ve ince ayar. ELMo, ilgili NLP problemi için ek özellikler olarak önceden eğitilmiş temsilleri içeren ilk yaklaşımı kullanır. İki yaklaşımın dezavantajı, dil temsillerini elde etmek için tek yönlü bir dil modelini kullanmalarıdır. Başka bir deyişle, tek yönlü dil modellemesi, aşağı akış NLP görevlerini yerine getirirken transformatörlerin verimliliğini azaltabilir. Bunun nedeni, yönlü modellerin, tüm çevreye bağlı olarak kelimelerin bağlamsal anlamlarını kavramayı sınırlayan sol veya sağ yönde sırayla metni okuma ile ilerlemesidir. Bununla birlikte, BERT'nin dönüştürücü-kodlayıcı stratejisi, tüm komşularla ilişkili olarak elde etmesine yardımcı olan eksiksiz bir diziyi bir kerede özetler. Bu noktada çift yönlü dil gösterimine sahip olan BERT, ince ayar şeması ile NLP görevlerinin verimliliğini artıran devreye giriyor. BERT iki denetimsiz görevi kullanarak önceden eğitilir; Maskeli Dil Modeli ve Sonraki Cümle Tahmini.

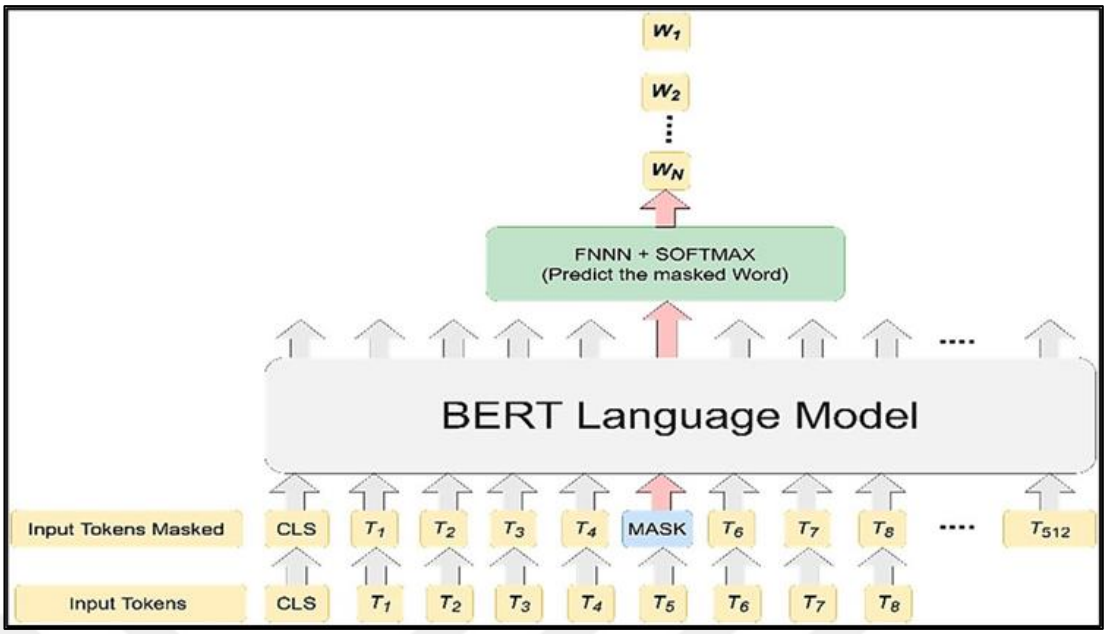
Maskeli Dil Modeli: BERT, maskeli dil modeli (MLM) ile tek yönlü sınırlamayı çözmüştür. Derin bir çift yönlü temsili eğitmek için, giriş belirteçlerinin bir yüzdesini rastgele olarak maskeleriz ve ardından bu maskelenmiş belirteçleri tahmin ederiz. Literatürde genellikle Close görevi olarak anılsa da bu prosedüre “Maskeli Dil Modeli” (MLM) adı verilmektedir. Bu durumda, maske belirteçlerine karşılık gelen son gizli vektörler, standart bir dil modellemede olduğu gibi sözlük üzerinden bir çıktı softmax'ına beslenir. BERT, ayrı belirteç dizileri ve bir dizi özel belirteç olarak temsil edilen sözcük dağarcığı üzerinde çalışır: SEP, CLS ve MASK. Bir giriş belirteci için, giriş gömme, belirteç-spesifik öğrenilmiş gömmelerin ve konum ve segment için karşılık gelen kodlamaların toplamı ile elde edilir. BERT giriş gösterimi Şekil 4.1'de [54] gösterilmiştir.



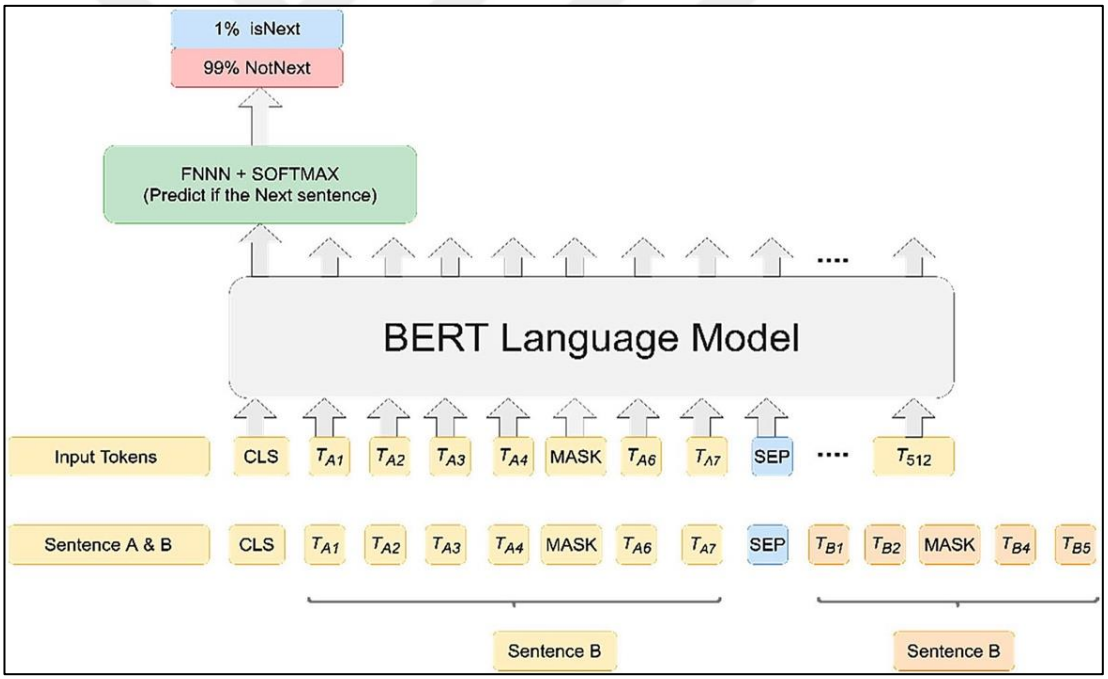
Şekil 4.1. BERT girdi gösterimi [54]

MLM görevi: girdi belirteçlerini ayrık kümelere bölerek gerçekleştirilir. Belirteçlerin yaklaşık %15'i aşağıdaki şema ile maskelenir (değiştirilir): (i) Maskelenmiş belirteçler zamanın %80'inde MASKE belirteci ile değiştirilir, (ii) %10 oranında rastgele bir sözcükle değiştirilir ve (iii) ve bunlar kısmın %10'u için değişmeden kaldı. Maskeleme sürecinin ardından, BERT modeli, gözlemlenen kümeye dayalı olarak maskelenmiş belirteçleri yeniden yapılandırmak için eğitilir [53].

Sonraki Cümle Tahmini**:** Soru Yanıtlama (QA) ve Doğal Dil Çıkarımı (NLI) gibi birçok önemli alt görev, doğrudan dil modellemesi tarafından yakalanmayan iki cümle arasındaki ilişkiyi anlamaya dayanır. Cümle ilişkilerini anlayan bir modeli eğitmek için, herhangi bir tek dilli bütünceden önemsiz bir şekilde oluşturulabilen ikili bir sonraki cümle tahmin görevi için ön eğitim yapıyoruz. Spesifik olarak, her bir ön eğitim örneği için A ve B cümlelerini seçerken, B zamanının %50'si A'dan sonra gelen asıl sonraki cümledir (IsNext olarak etiketlenir) ve zamanın %50'si derlemden rastgele bir cümledir (etiketlenmiş) NotNext olarak). İki görev sırasıyla Şekil 4.2 ve 4.3'te verilmiştir.



Şekil 4.2. BERT denetimsiz MLM görevi



Şekil 4.3. BERT denetimsiz NSP görevi

4.2.5. Siber Zorbalık Tespiti için BERT Algoritması İnce Ayarı

BERT tasarımcıları makalelerinde [56] küme büyüklüğü (batch-size), öğrenme oranı (learning rate) ve dönem boyutu (epoch-size) dışındaki tüm parametreleri sabit tutmayı önerir. Parametreler için önerilen değerler aşağıdaki gibidir:

- Batch-size: 16,32
- Learning Rate: 5e-5, 3e-5, 2e-5
- Epoch-size: 2,3,4

Tüm deneylerde sırasıyla küme büyüklüğü, öğrenme oranı ve dönem büyüklüğü 16, 2e-5 ve 4'ü seçtik ve bunları sabit tuttuk. Görevlerimiz için, Transformer blok sayısı 12, gizli katman boyutu 768 ve öz dikkat başlığı 12 olan önceden eğitilmiş BERT tabanlı çoklu dil seçtik. BERT, Özellikle BERT'nin çok dilli versiyonu için, belirtilen ince ayar değerleri dışında Türkçe için özel bir parametre ayarlama stratejisi uygulanmamıştır. Türkçe dahil 100 dili destekleyen çoklu dil modeline sahiptir. Çok dilli uygulamalar için BERT ardışık düzeni belirli bir ayar görevi gerektirmez.

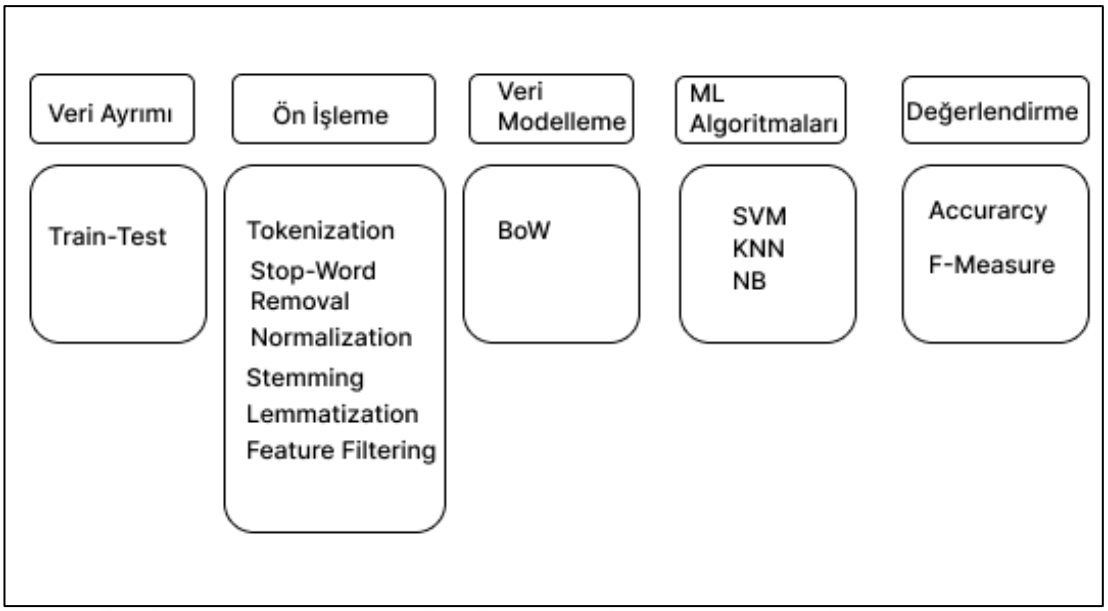
Deneyleri yaparken 128 GB RAM'e sahip Kişisel Bilgisayar (PC), Intel Core i9 9900X CPU ve 11 GB belleğe sahip GeForce RTX 2080 Ti GPU'dan faydalandık.

4.3. Veri Seti

Siber Zorbalık tespiti için Siber Zorbalık veri seti kullanılmıştır. Veri setinde iki sınıf bulunmaktadır. Her iki kategori için 1500 örnek bulunmaktadır. Maksimum kelime uzunluğu 25’tir.

4.4. Deneyler

NLP problemlerini çözmek için Türkçe'nin hesaplamalı modellemesi genellikle uygun bir vektör temsili ile belirteçleştirme, kelime çıkarma, kök çıkarma gibi ön işleme görevlerini gerektirir. Metinsel bilgilerin gösterimi daha sonra hedef tanımlamayı elde etmek için ML algoritmalarına beslenir. Bu sık NLP analizi yaklaşımı Şekil 4.4'te gösterilmektedir.



Şekil 4.4.NLP problemleri için Türkçe dil modelleme

Önceden işlenmiş ham metinler için, BoW gibi bir matematiksel temsilin oluşturulması gerekir. Özellikle, BoW bir metni kelime sıklığı gibi bir kritere dayalı olarak basitleştirilmiş bir temsile indirger. Bu modelde ham metin bir kelime torbası olarak temsil edilirken kelimeler arasındaki anlamsal ilişkiler göz ardı edilmiştir. BoW her kelimeyi tek-sıcak kodlama şemasında kodladığından, model hızlı bir şekilde seyrek bir vektöre yakınsayabilir. Terim-frekans, BoW'da öznitelikleri elde etmek için en basit tekniktir. Derlemde sık kullanılan sözcüklerin etkisini azaltmak için metinden öznitelikler elde edilirken ağırlıklandırma şeması yani tf-idf sıklıkla kullanılır. Matematiksel olarak, tf-idf aşağıdaki şekilde temsil edilir:

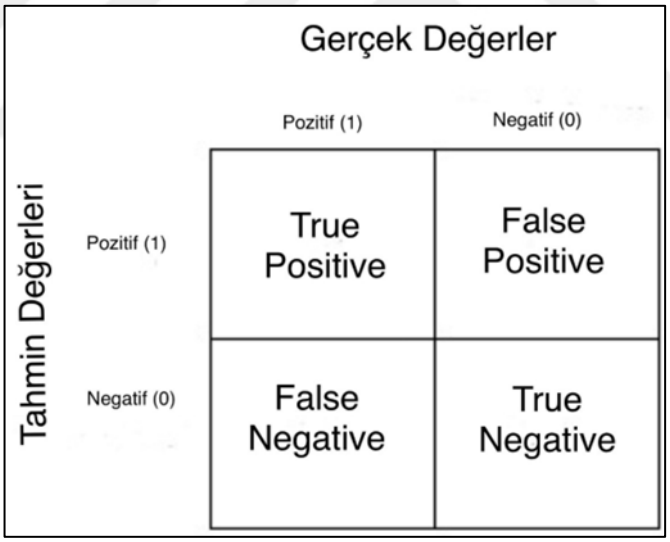
$$W(d,t) = TF(d,t) * \log(\frac{N}{df(t)}) \quad (4)$$

N, derlemdeki belge sayısını belirtirken, df(t), t terimini içeren belge sayısını belirtir [57]. BoW modelleri sadece sırasız kelime dizisi olduğundan, çıkarılacak bağlamsal bilgiyi zenginleştirebilecek anlamsal ilişkilerin farkında değillerdir. Bu probleme bir çözüm, N − 1 önceki kelimenin oluşumuna dayalı olarak kelimeler arasındaki ilişkiyi koruyan N-gram modelinin kullanılmasıdır [58].

Bu problemle başa çıkmak için, özellik mühendisliği (muhtemelen bir filtreleme) olarak bilinen bir diğer önemli adım, muhtemelen ortaya çıkan modele uygulanmalıdır. Filtreler, sarmalayıcılar ve gömülü öznitelik seçim metodolojileri

olmasına rağmen, en çok yönlü öznitelik seçim grubu öznitelik filtreleridir. NLP literatüründe Bilgi Kazanımı (IG), Chi-Square (CHI) ve Gini İndeksi (GI) sıklıkla kullanılan öznitelik filtreleme algoritmalarından bazılarıdır.

Metin gösterimi ardışık düzeni tamamlandıktan sonra, dil sorunu, eğitim/doğrulama/test şeması açısından bir ML yaklaşımına ihtiyaç duyar. NLP problemlerinde sıklıkla kullanılan birçok ML algoritması vardır. Açıkça, Naive Bayes (NB), Destek Vektör Makineleri (SVM), k-En Yakın Komşular (k-NN) Rastgele Ormanlar (RF) Boosting ve Bagging gibi yaklaşımları bir araya getirir. Son zamanlarda, DL algoritmaları, yani Evrişimli Sinir Ağları ve Tekrarlayan Sinir Ağları da NLP alanı için ortaya çıkan ML algoritmaları olarak büyük popülerlik kazanmıştır. Burada hatırlatılması gereken önemli bir nokta, algoritmaların performansının Accuracy, Recall, Precision ve F1-ölçümü gibi yaygın olarak kullanılan bazı metriklere dayalı olarak karşılaştırılabileceğidir [59]. Değerlendirme ölçütleri Şekil 4.5'te verilen karışıklık matrisinden türetilmiştir.



Şekil 4.5. Karmaşıklık Matrisi

Doğruluk, Geri Çağırma, Kesinlik, Karışıklık matrisinden F1 ölçümü aşağıdaki gibi tanımlanır:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$R = \frac{TP}{TP+FN} \quad (6)$$

$$P = \frac{TP}{TP+FP} \tag{7}$$

$$F1 - measure = \frac{2*P*R}{(P+R)} \tag{8}$$

Burada TP, FP, FN ve TN sırasıyla Doğru Pozitif, Yanlış Pozitif, Yanlış Negatif ve Gerçek Negatif'i gösterir. Ayrıca, ML algoritmalarının performansını ve istatistiksel doğrulamasını değerlendirmek için kullanılan iki ek anlamlı ölçümün altını çiziyoruz: Matthews Korelasyon Katsayısı (MCC) ve Kappa (κ).

MCC, sırasıyla mükemmel anlaşma ve anlaşmazlığı bildirmek için değeri 1 ile -1 arasında değişen hedef ve tahmin arasındaki bir korelasyon katsayısıdır. MCC, bir ML algoritmasının gücünü, dört karışıklık matrisi sonucunun tümü ile tahmin ettiğinden, kapsamlı bir ölçümdür. Bu nedenle, dengesiz veri uygulamalarında bile dengeli bir performans değerlendirmesi sağlar ve aşağıdaki gibi verilir.

$$K = \frac{(Po-Pe)}{(1-Pe)} \tag{9}$$

Kapsayıcılığa dayalı olarak, BERT deneylerinin performansını değerlendirmek için MM metriğini de kullandık. Kappa, yani κ, kategorik maddeler için değerlendiriciler arası güvenilirliği ölçen bir istatistik metriktir. Değerlendiriciler arası güvenirlik, puanlayıcılar arasındaki uyuşma derecesi olarak tanımlanmaktadır. Bu istatistiksel puan, tahmin edicilerin kararlarına dayalı olarak fikir birliği derecesini ölçer. Başka bir deyişle, her biri N öğeyi özel kategoriler halinde sınıflandıran iki öngörücü arasındaki uyuşmayı ölçer ve κ aşağıdaki gibi tanımlanır:

Burada Po aşağıda verilmiştir ve aynı zamanda doğrulukla aynıdır.

$$Po = \frac{TP+TN}{TP+FP+TN+FN} \tag{10}$$

Pe, bir puanlayıcının k kategorisini N gözlemiyle tahmin ettiği n_{ki} çarpı sayısı olarak tanımlanır ve şu şekilde verilir:

$$Pe = \frac{1}{N^2}\sum_k n_{k1}n_{k2} \tag{11}$$

Karışıklık matrisi terimleri cinsinden de aşağıdaki bağıntı ile hesaplanır.

$$Pe = \frac{(TP+FP)}{(TP+FP+TN+FN)} * \frac{(TP+FN)}{(TP+FP+TN+FN)} * \frac{(FN+TN)}{(TP+FP+TN+FN)} * \frac{(FP+TN)}{(TP+FP+TN+FN)} \tag{12}$$

K toplam puanı -1 ile 1 arasında değişir. Elde edilen puan, elde edilen sonuçların tesadüfen oluşmaktan ne kadar uzak olduğunun istatistiksel bir ölçüsüdür. 0.40'tan küçük değerler orta düzeyde uyum gösterirken, 0.40 ile 0.60 arasındaki değerler orta düzeyde uyum göstermektedir. Basitçe, güvenilir bir sınıflandırma performansı değerlendirmesi için, iyi uyum için 0.60–0.80 ve mükemmel uyum için 0.80'den yüksek elde edilmelidir [59]. Bu nedenle, elde edilen sonuçların istatistiksel güvenini değerlendirmek için deneylerde κ parametresini hesapladık.

5. SONUÇLARIN DEĞERLENDİRİLMESİ

Sonuçları kıyaslamak için ilgili makalede desteklenen Acc veya F-1 puanını kullanacağız Siber Zorbalık veri seti için, Makine Öğrenmesi tabanlı Yapay Sinir Ağları ile Acc veya F-1 performansını 91.00 olarak elde ettik. Aynı veri setini kullanarak BERT Acc'yi 98,27 olarak elde ettik. BERT algoritmasını farklı performans değerlendirme metriklerinde inceleyecek olursak; Acc = 0.9827, F-1 =0.9826,Presicion = 0.9819, Recall =0.9834 ,MCC =0.9653 ve K =0.9650 olarak elde ettik.

Morfolojik olarak zengin diller, elde edilen ML 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 sinir modeli, çeşitli dil görevlerini çözmek için nispeten basit bir uygulamaya sahiptir. BERT, çoklu dil versiyonu ile bile NLP problemlerinde dikkat çekici sonuçlar üretebilir. Türkçe Siber Zorbalık veri setini analiz etmek için BERT kullandık ve elde edilen sonuçları literatürdeki temel ML algoritmalarının en iyi tahminleriyle karşılaştırdık. Ayrıca Performans sayısal olarak +7.27 yükselmiştir.

KAYNAKLAR

[1] Yazğılı, E., Siber Zorbalık Tespit Yöntemleri Potansiyel Uygulama Alanları ve Zorluklar, Dicle Üniversitesi Mühendislik Dergisi, 2021,12:1,23-35
[2] Tutorial Sprint Machine Leaning Tutorial
[3] Liu, Q., Wu, Y., Supervised Learning, Springer, Boston, MA, 2012, 2-4
[4] https://www.javatpoint.com/supervised-machine-learning
[5] https://erkancetinyamac.medium.com/knn-algoritmas%C4%B1-ve-knn-s%C4%B1n%C4%B1fland%C4%B1rma-modeli-kullan%C4%B1larak-kredi-onay-de%C4%9Ferlendirilmesi-da5b06c85dda
[6] Saeed, Sherko Baper, Face Recognition System Based on Pca-Wavelet and Support Cector Machines, Hasan Kalyoncu Üniversitesi, Uygulamalı Bilimler Enstitüsü, Elektronik ve Bilgisayar Mühendisliği, Gazi Antep, 2016,79.(Yüksek Lisans Tezi).
[7] https://www.geeksforgeeks.org/decision-tree/
[8] Mitchell, T.M., Machine Learning, McGraw-Hill Science/Engineering/Math, 1997, 435
[9] https://www.geeksforgeeks.org/naive-bayes-classifiers/
[10] https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
[11] Toosi, Khaje Nasir, What is Unsupervised Learning, Strategic Intelligence Research Lab (SIRLAB), August 2018, IranAffiliation
[12] https://medium.com/@dilekamadushan/introduction-to-k-means-clustering-7c0ebc997e00
[13] https://www.softwaretestinghelp.com/data-mining-vs-machine-learning-vs-ai/8
[14] https://www.muhendisbeyinler.net/pekistirmeli-ogrenme-reinforcement-learning-nedir/
[15] https://www.javatpoint.com/reinforcement-learning
[16] Ravì, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., & Yang, G. Z. Deep learning for health informatics, IEEE journal of Biomedical and Health Informatics 21(1),2017, 4-21.
[17] S. Agatonovic-Kustrin *, R. Beresford, Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research, Journal of Pharmaceutical and Biomedical Analysis, 2000,22, 717–727
[18] Gershenson, C., Artificial Neural Networks for Beginners,2003,2-9.
[19] https://tr.on2vhf.be/back-propagation-neural-network
[20] https://www.guru99.com/nlp-tutorial.html
[21] https://www.wikitechy.com/tutorial/nlp/components-of-nlp
[22]https://www.hasanbaskin.com/dogal-dil-isleme-natural-language-processing/
[23] https://towardsdatascience.com/word-embedding-techniques-word2vec-and-tf-idf-explained-c5d02e34d08
[24] https://medium.com/@muhammedbuyukkinaci/word2vec-nedir-t%C3%BCrk%C3%A7e-f0cfab20d3ae
[25] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
[26] Erdinç, H. Y., & Güran, A., Semi-supervised Turkish text categorization with word2vec, doc2vec and Fasttext algorithms, 2019, 27 Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.4

[27] Hikari-dai, Seika-cho, Soraku-gun,BI-DIRECTIONAL RECURRENT NEURAL NETWORKS FOR SPEECH RECOGNITION ! Mike Schuster ATR, Interpreting Telecommunications Research Lab. 2-2, Kyoto 619-02,
[28] Bakewell, R., Withey S., Montana, G., Modelling Radiological Language with Bidirectional Long Short-Term Memory Networks, Savelie Cornegruta, Department of Biomedical Engineering, King's College London.
[29] Donahue, J., Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2014). Long-term recurrent convolutional networks for visual recognition and description. Arxiv, PP. https://doi.org/10.1109/TPAMI.2016.2599174
[30] Donahue, J., Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2014). Long-term recurrent convolutional networks for visual recognition and description. Arxiv, PP. https://doi.org/10.1109/TPAMI.2016.2599174
[31] Sherstinsky, A. (2018a). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. CoRR, abs/1808.03314.
[32] Demirci, Deniz, Middle East Technical University, Informatics Institute, Department of Cyber Security, Ankara,82. (Yüksek Lisans Tezi)
[33] Graves, A., Jaitly, N., Towards End-To-End Speech Recognition with Recurrent Neural Networks, *Proceedings of the 31st International Conference on Machine Learning*, 2014, United Kingdom
[34] Nergiz, Gözde, DERİN ÖĞRENME TEKNİKLERİ KULLANILARAK SOSYAL AĞLAR ÜZERİNDE SİBER ZORBALIK TESPİTİ,Mersin Üniversitesi, Fen Bilimleri Enstitüsü,Bilgisayar Mühendisliği,
[35] Yazğılı, E., Siber Zorbalık Tespit Yöntemleri Potansiyel Uygulama Alanları ve Zorluklar, Dicle Üniversitesi Mühendislik Dergisi, 2021,12:1,23-358
36] https://www.cyberwise.org/post/effects-of-cyberbullying-on-an-individual
[37] Sak H, Güngör T, Saraçlar M. Resources for Turkish morphological processing. Lang Resour Eval. 2011;45(2): 249–261.
[38] Vylomova E, Cohn T, He X, et al. Word representation models for morphologically rich languages in neural machine translation. Proceedings of the First Work- shop on Subword and Character Level Models in NLP. Copenhagen, Denmark: Association for Computational Linguistics. 2017:103–108.
[39] Kışla, T., Karaoğlan, B., A HYBRID STATISTICAL APPROACH TO STEMMING IN TURKISH: AN AGGLUTINATIVE LANGUAGE, Anadolu University Journal of Science and Technology,2016,1-12
[40] Hans K, Milton RS. Improving the performance of neural machine translation involving morphologically rich languages. ArXiv:1612.02482 [Cs]. January 8, 2017.
[41] Oflazer K. Turkish and Its challenges for language processing. Lang Resource Eval. 2014;48(4):639–653.
[42]Kutlu, Y., Tohma, K., Challenges Encountered in Turkish Natural Language Processing Studies, Natural and Engineering Sciences, 2020, DOI:10.28978/nesciences.833188
[43] Mertoğlu, Uğur, TÜRKÇE İÇİN SAHTE HABER TESPİT MODELİNİN OLUŞTURULMASI, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği, Ankara,2020,141.(Doktora Tezi)
[44] Nuzumlalı MY, Özgür A. Analyzing stemming approaches for Turkish multi-document summariza- tion. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguis- tics. 2014:702–706.

[45] Botha JA. Probabilistic modelling of morphologically rich languages. ArXiv:1508.04271[Cs]. August 18, 2015.
[46] Kontostathis, A., Edwards, L., and Leatherman, A., 2010, Text mining and cybercrime, Text Mining: Applications and Theory, M. W. Berry and J. Kogan Eds., John Wiley and Sons, New York, NY.
[47] Dinakar, K., Reichart, R., and Lieberman, H., 2011. Modelling the Detection of Textual Cyberbullying, Social Mobile Web Workshop at International Conference on Weblog and social media, Barcelona, Spain.
[48] Dadvar, Maral., Ordelman, Roeland., Trieschnigg, Dolf., Jong, Franciska. Conference: European Conference on Information Retrieval, 2013, Nederland.
[49] BERT tutorial, Jacob Devlin Google AI Language
[50] https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270
[51]https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model
[52] Chi,S., Xu, Y, Qiu,X., Huang, X. How to Fine-Tune BERT for Text Classification?,
arXiv:1905.05583 [cs.CL].14 May 2019.
[53] https://towardsdatascience.com/what-exactly-happens-when-we-fine-tune-bert-f5dc3288
[54] Wu X, Lv S, Zang L, et al. Conditional BERT contextual augmentation. December 17, 2018.
[55] Jahan,M. Khan,H.U. Akbar,S. Gul, S. Amjad,A. Bidirectional Language Modeling: A Systematic Literature Review. 2021.DOI: 10.1155/2021/6641832
[56] Devlin J, Chang M-W, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language under- standing. ArXiv:1810.04805 [Cs]. May 24, 2019.
[57] Kowsari K, Meimandi KJ, Heidarysafa M, et al. Text classification algorithms: a survey. Information. 2019;10(4):150.
[58] Mogotsi I, Christopher C, Manning D, et al. Intro- duction to information retrieval. Inf Retr Boston. 2010;13(2):192–195.