ÜRETKEN RAKİP AĞLAR İLE TÜRKÇE METİN ÜRETİMİ

Barış GÜCÜK

2021
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ

Tez Danışmanı
Dr. Öğr. Üyesi Rafet DURGUT

ÜRETKEN RAKİP AĞLAR İLE TÜRKÇE METİN ÜRETİMİ

Barış GÜCÜK

T.C.
Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalında
Yüksek Lisans Tezi
Olarak Hazırlanmıştır

Tez Danışmanı
Dr. Öğr. Üyesi Rafet DURGUT

KARABÜK
Ocak 2021

Barış GÜCÜK tarafından hazırlanan "ÜRETKEN RAKİP AĞLAR İLE TÜRKÇE METİN ÜRETİMİ" başlıklı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Dr. Öğr. Üyesi Rafet DURGUT .....
Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından Oy Birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 29/01/2021

Unvanı, Adı SOYADI (Kurumu) İmzası

Başkan : Prof. Dr. Oğuz FINDIK (KBÜ) .....

Üye : Dr. Öğr. Üyesi Rafet DURGUT (KBÜ) .....

Üye : Dr. Öğr. Üyesi İlker YILDIZ (BAİBÜ) .....

KBÜ Lisansüstü Eğitim Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Prof. Dr. Hasan SOLMAZ .....
Lisansüstü Eğitim Enstitüsü Müdürü

"Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim."

Barış GÜCÜK

ÖZET

Yüksek Lisans Tezi

ÜRETKEN RAKİP AĞLAR İLE TÜRKÇE METİN ÜRETİMİ

Barış GÜCÜK

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Dr. Öğr. Üyesi Rafet DURGUT
Ocak 2021, 49 sayfa

Makinelerin çeşitli algoritmalar aracılığı ile kendisine verilen örneklerden öğrenip, gelecek durumlar için tahminlerde bulunmasına makine öğrenmesi denir. Makine öğrenmesi yöntemlerinde eğitim aşamasının başarısı için kullanılan eğitim veri seti kümesi oldukça önemlidir. Doğal dil işlemede en çok karşılaşılan problemlerden birisi yeterli veri bulunamaması veya bulunan verilerin etiketsiz olmasıdır. Özellikle sınıflandırma problemlerinde belirli bir sınıftaki verinin azlığı sınıflandırmanın başarısını düşürmektedir. Bu problemin doğal dil işleme alanında çözümü için metin üretimi kullanılmaktadır. Metin üretimi, metnin ayrık doğası ve sözlükte bulunmayan farklı yüzey formlarına sahip olduğundan çözülmesi zor bir problemdir. Bu çalışmada veri kümesinde bulunan metinlerin arttırılması amacı ile üretken rakip ağlar yöntemi kullanılmıştır. Üretilen bu metinlerin konuşma diline yakın olması amaçlanmıştır. Çalışmada morfolojik açıdan zengin bir dil olan Türkçe üzerinde üretken rakip ağlar kullanılarak normal dağılımlı olmayan bir veri setindeki eksik sınıfa ait metinlerin

üretimi yapılmıştır. Çalışmada problem olarak haber metinlerinin olumlu veya olumsuz olarak sınıflandırılması ele alınmıştır.

Oluşturulan veri kümesinde toplam 3058 haber metni bulunmaktadır. Bu haber metinlerinin 2949 tanesi olumlu 109 tanesi olumsuz sınıfa aittir. Olumsuz sınıfa ait örneklerin az olması nedeniyle bu sınıfta başarının düşük olduğu gözlenmiştir. Ardından, üretken rakip ağ ile olumsuz sınıftaki veriler test aşamasında 50 örnekten başlayarak 2750 örneğe kadar çoğaltılmıştır. Elde edilen sonuçlar n-gram, destek vektör makinesi, TF-IDF ve lojistik regresyon gibi makine öğrenmesi teknikleriyle birlikte kullanılarak performansları değerlendirilmiştir. Elde edilen sonuçlara göre üretken rakip ağların Türkçe metin üretimi için kullanılması sınıflandırma başarısını yaklaşık % 47 oranında arttırmıştır. Sınıflara ait örnek sayılarında aşırı farklılık olduğu durumda başarı oldukça düşük çıkmakta, örnek sayısı yapay zekâ ile artırıldığında ise başarı % 90 üzerine çıkmaktadır. Ayrıca üretilen sonuçlar incelendiğinde çalışmada kurulan model ile konuşma diline yakın cümleler üretilebileceği gözlenmiştir.

Anahtar Sözcükler : Doğal dil işleme, üretken rakip ağlar, metin üretimi, sınıflandırma.

Bilim Kodu **:** 92432

ABSTRACT

M. Sc. Thesis

TURKISH TEXT GENERATION WITH GENERATIVE ADVERSARIAL NETWORKS

Barış GÜCÜK

Karabük University
Institute of Graduate Programs
Department of Computer Engineering

Thesis Advisor:
Assist. Prof. Dr. Rafet DURGUT
January 2021, 49 pages

Machine learning is when machines learn from the examples given to them through various algorithms and make predictions for future situations. The training data set used for the success of the training phase in machine learning methods is very important. One of the most common problems in natural language processing is the lack of sufficient data or the untagged data found. Especially in classification problems, the scarcity of data in a certain class reduces the success of the classification. Text generation is used to solve this problem in natural language processing. Text generation is a difficult problem to solve as it has the discrete nature of the text and different surface forms not found in the dictionary. In this study, generative adversarial network method was used to increase the texts in the data set. These texts are aimed to be close to the spoken language. In the study, texts belonging to the missing class in a non-normally distributed data set were produced by using generative adversarial network in Turkish, a morphologically rich language. The problem of the study is to

categorize news texts as positive or negative.

There is a total of 3058 news texts in the data set created. 2949 of these news texts belong to the positive and 109 of them belong to the negative category. It was observed that success was low in this class due to the small number of samples belonging to the negative class. Then, with the generative adversarial network, data in the negative class were replicated from 50 samples to 2750 samples in the test phase. The results obtained were evaluated together with machine learning techniques such as n-grams, support vector machine, TF-IDF and logistic regression. According to the results, the use of generative adversarial network for Turkish text generation increased the success of classification by approximately 47%. In cases where there is an excessive difference in the number of samples belonging to the classes, the success is low, and when the number of samples is increased with artificial intelligence, the success increases over 90%. In addition, when the results produced were examined, it was observed that sentences close to the spoken language could be produced with the model established in the study.

Key Word **:** Natural language processing, generative adversarial networks, text generation, classification.

Science Code **:** 92432

TEŞEKKÜR

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren, cesaretlendirme ve kılavuzluğu sayesinde bu yüksek lisans tez çalışmasını gerekli akademik disiplinlerle şekillendirmemi sağlayan sayın hocam Dr. Öğr. Üyesi Rafet DURGUT'a;

Değerli katkılarıyla çalışmama destek veren bilgi ve birikiminden yararlandığım sayın hocam Prof. Dr. Oğuz FINDIK'a sonsuz teşekkürlerimi sunarım.

Sevgili aileme maddi ve manevi hiçbir yardımı esirgemeden her an ve her koşulda yanımda oldukları için tüm kalbimle teşekkür ederim.

İÇİNDEKİLER

Sayfa

Sayfa

ŞEKİLLER DİZİNİ

Sayfa

Sayfa

ÇİZELGELER DİZİNİ

Sayfa

SİMGELER VE KISALTMALAR DİZİNİ

SİMGELER

Σ : Toplam

log : Logaritma

D_{loss} : Ayırıcı Kaybı

D(x) : Ayırıcı Fonksiyonu

G(x) : Üretici Fonksiyonu

G_{loss} : Üretici Kaybı

KISALTMALAR

AI : Artificial Intelligence (Yapay Zekâ)

BigGAN : Big Generative Adversarial Network (Büyük Üretken Rakip Ağ)

CIFAR-10 : Makine Öğrenimi ve Bilgisayar Görme Algoritmalarını Eğitmek İçin Yaygın Olarak Kullanılan Bir Görüntü Koleksiyonu

CoGAN : Coupled Generative Adversarial Networks (Çiftli Üretken Rakip Ağ)

DCGAN : Deep Convolutional Generative Adversarial Network (Derin Evrişimli Üretken Rakip Ağ)

FCGAN : Face Conditional Generative Adversarial Network (Yüz Koşullu Üretken Rakip Ağ)

GAN : Generative Adversarial Network (Üretken Rakip Ağ)

IDF : Inverse Document Frequency (Ters Doküman Sıklığı)

LeakGAN : Generative Adversarial Network with Leaked Information (Bilgi Sızdırmalı Üretken Rakip Ağ)

Log Reg : Logistic Regression (Lojistik Regresyon)

LSTM : Long Short-Term Memory (Uzun Kısa Süreli Bellek)

MaliGAN : Maximum-Likelihood Augmented Generative Adversarial Network (Maksimum Olabilirliği Arttırılmış Üretken Rakip Ağ)

MNIST : Mnist Database (Makine Öğrenimi ve Bilgisayar Görme Algoritmalarını Eğitmek İçin Yaygın Olarak Kullanılan Bir Veri Seti)

NLP : Natural Language Processing (Doğal Dil İşleme)

RankGAN : Adversarial Ranking Generative Adversarial Network (Karşılıklı Sıralamalı Üretken Rakip Ağ)

RL : Reinforcement Learning (Pekiştirmeli Öğrenme)

RNN : Recurrent Neural Networks (Tekrarlayan Sinir Ağı)

SeqGAN : Sequence Generative Adversarial Network (Sıralı Üretken Rakip Ağ)

SRGAN : Super-Resolution Generative Adversarial Network (Yüksek Çözünürlüklü Üretken Rakip Ağ)

SVM : Support Vector Machine (Destek Vektör Makinesi)

TF : Term Frequency (Terim Sıklığı)

BÖLÜM 1

GİRİŞ

Türkçesi Doğal Dil İşleme olarak bilinen Natural Language Processing (NLP) bir bilgisayar mühendisliği terimidir. Dil bilimi ile yapay zekâ teknolojisinin beraber kullanılmasıyla doğal dil işleme ortaya çıkmıştır. Birçok uygulama alanının yanında doğal dil işleme bilgisayarlarda geliştirilen programlar yardımı ile metinlerin dijital ortama geçirilmesini de konu almaktadır. Bunun yanında ses dalgalarının çözümlenerek dijital ortama aktarılması da doğal dil işlemenin çalışma alanlarından biridir. Doğal dil işleme çalışmaları ilk olarak 1954 yılında gerçekleşmiştir [1].

İnsanların ihtiyaçlarını karşılayabilmek için kullandığı ilk ve en önemli iletişim araçlarından biri doğal dildir. Dil öğrenimi uzun vakit alan bir süreçtir. İnsanlar doğdukları andan itibaren ana dillerini öğrenip, geliştirmeye başlar. Bir dilde aynı anlama gelen ve kullanıldığı zamana göre değişen birçok kelime bulunabilir. İnsanlar öğrendikleri dillerdeki bilgileri ve tecrübeleri sayesinde bu kelimelerin hangi anlamda kullanıldığını anlayabilir. Fakat bilgisayarın dijital ortamda bu kelimelerin hangi anlama geldiğini algılaması zordur. Bundan dolayı doğal dil işlemenin ilk yıllarından beri bu sorunun çözülmesi için uğraşılmıştır [1].

Yapay zekâ (Artificial Intelligence) ilk olarak 1956'da ortaya çıkmıştır [2]. Kısaca yapay zekâ makinelerin kendilerine verilen verilerden öğrenmesi ve daha sonra bu öğrenimini yeni problemlerin çözümünde kullanılması için tasarlanmasıdır. Bu amaçla eğitilen makineler yeni veri değerlerine göre aynı ya da farklı bir problemi çözebilme kapasitesine sahiptir. Buradaki amaç makinenin insan gibi hedeflenen problemi çözmesidir.

Her geçen yıl büyüyen veri boyutları, işlemcilerin hesaplayabilme kapasiteleri ve daha iyi algoritmaların ortaya çıkması ile yapay zekâ konusunun popülerliği artmaktadır. Popülerliğinin artmasıyla uygulama alanları da artmaktadır. Telefonlardaki sanal asistanlar, anti virüs çözümleri, bilgisayar oyunları ve arama motorları gibi alanlarda yapay zekâ kullanımı yaygındır. İlerleyen yıllarda yapay zekâ teknolojisindeki gelişmeler ile daha fazla uygulama alanındaki problemlerin çözümünde de kullanılabilecektir [2].

Makine Öğrenmesi (Machine Learning) yapay zekânın bir alt koludur. Yapay zekâda olduğu gibi kendisine verilen örnekler üzerinden öğrenim gerçekleştirir. Algoritma ile verilen ögelerin niteliklerinden çıkarımlar yaparak tahminlerde bulunur. Makine öğrenmesi ilk olarak 1959 yılında model sınıflandırma probleminin çözümünde kullanılmıştır [3]. Gözetimsiz ve gözetimli öğrenme olarak iki sınıfta incelenmektedir.

Gözetimli öğrenmede eğitim yapılan verinin girdi değerleri ve bunlara karşılık gelen sonuç değerleri bilinir. Burada gözetimli öğrenme yapılarak farklı bir girdi değerine karşılık gelecek sonuç değeri tahmin edilmesi sağlanır.

Her durumda bir girdi değerine karşılık sonuç değeri olmayabilir. Bu durumda gözetimsiz öğrenme yöntemi kullanılır. Burada sonuç değerlerinin varlığından etkilenmeden girilen ögeler arasındaki bağlantı ve ilişkileri inceler. Burada kullanıcı girişine gerek kalmadan algoritmanın kendi çıkarımlar yapması beklenir.

Gözetimli öğrenmede verilen bir ögenin hangi sınıfa ait olduğunu tahmin etme işlemine sınıflandırma denir. Buradaki sınıfların özellikleri eğitim sırasında ögeler arasındaki ilişkilerden çıkarılmış olup bu bilgiler sonrasında yeni gelen verilerin sınıflandırma tahmininde kullanılır.

Gözetimsiz öğrenmede girdilerin sonuçları bilinmediğinden dolayı sınıflandırma yerine verilerin kümelenmesi yöntemi kullanılır. Burada bir veri kümesini oluşturan ögeler diğer veri kümelerindeki ögelere göre birbirleriyle daha benzer özellikler içerir ve bu şekilde gruplanmış olurlar.

Bir makine öğrenmesi modelinin eğitiminde etiketli veri kullanmak modelin çıktı başarısını doğrudan etkilemektedir. Etiketli verilerden öğrenim makine öğrenmesinin temel taşlarındandır. Verilerin etiketlenmesi konuyla ilgili hâkim bir kişi tarafından yapılmalıdır. Etiketli verinin sayısı da modelin performansını etkilemektedir. Fakat etiketli veri bulunması zordur. Çünkü verilerin etiketlenmesi uzun süre alan maliyetli bir işlemdir. Etiketleme işlemini yapan kişinin uzmanlığı gereklidir. Etiketleme sırasındaki insan hataları modelin başarısını etkilemektedir. Bazı verilerde gizlilik gerekebilir bu gibi durumlarda etiketleme yapılamayabilir.

Bu gibi durumlarda etiketsiz verilerden gözetimsiz öğrenme ya da yarı gözetimli öğrenme yöntemi seçilebilir. Elde etiketli veri var ise yarı gözetimli olarak etiketli verilerden bir ön eğitim işlemi gerçekleştirilip modelin eğitim işlemini başarıyla tamamlaması sağlanabilir. Yok ise gözetimsiz öğrenme tercih edilerek modelin veriler arasındaki bağlantıları ve özellikleri kendisinin çıkarması beklenebilir.

Üretken Rakip Ağlar (Generative Adversarial Network) iki ağın birlikte çalışmasıyla meydana gelmektedir. Bir ağ üretim ile sorumlu iken diğer ağ ayırt edici olarak çalışmaktadır. Bir dengede çalışan bu iki ağ eğitilmesi sonrasında eğitim setinden farklı özgün yeni resimler, sesler üretilebilir. Ne kadar dengeli bir sistem kurulursa gerçeğe o kadar yakın sonuçlar elde edilir.

Daha yeni olmasına karşın üretken rakip ağlar üzerindeki çalışmalarda gerçeğe çok yakın görüntüler üretilmiştir. Buradaki başarıdan yola çıkarak, üretken rakip ağlar ile kurabilecek bir modelin metin üretimi gibi bir doğal dil işleme uygulamasında kullanılabileceği akla gelmektedir. Türkçe metin üretimlerinde daha çok Yinelenen Sinir Ağı (Recurrent Neural Network) ve Uzun Kısa Süreli Bellek (Long Short Term Memory) kullanılmaktadır. Bu yöntem ile üretilen metinlerin başarısı yüksek değildir.

Bu çalışmada üretken rakip ağlar kullanılarak normal dağılımlı olmayan bir veri seti üzerinde Türkçe metin üretimi işlemi yapılmıştır.

BÖLÜM 2

ÜRETKEN RAKİP AĞLAR

Üretken rakip ağlar bir makine öğrenmesi yöntemidir. İlk olarak 2014 yılında Ian Goodfellow vd. tarafından tanıtılmıştır [4]. Yapısı iki sinir ağının birbirine karşı çalışacak şekilde oluşturulmasından meydana gelmektedir. Bu iki ağ arasında sıfır toplamlı oyun (zero-sum game) Nash Dengesi vardır. Bunu bir ağın kazanması için diğerinin kaybetmesi gerekliliği gibi düşünülebilir. Her iki ağ aynı anda kazanamaz. Bu durumda toplam kazanç ve toplam kayıp toplamı sıfır çıkmalıdır. Zero-sum game problemleri min-max teoremi ile çözülmektedir.

Modelin çalışma mantığından bahsetmek gerekirse, bir sinir ağı üretici (generator) diğer sinir ağı ise ayırıcı (discriminator) olarak adlandırılmaktadır. Bir ön eğitimden sonra üretici eğitim setinden öğrenmeye başlar ve bu öğrendiklerinden tahminlerde bulunur. Ayırıcı ise bu çıktıları eleyerek üreticiyi istenilen gerçeğe yakın çıktılara yönlendirmeye çalışır. Bunu sıcak soğuk oyunu gibi düşünürsek istenilen çıktıdan uzaklaşıldığında ayırıcı üreticiye soğuk şeklinde geri bildirimde (feedback) bulunur. Bu durumda üretici geri yayılım (back propagation) kullanarak hatasını minimalize etmeye çalışır ve ağırlıklarını günceller. Diğer durumda ayırıcı üreticiden gelen çıktıların gerçeğe yaklaştığı fark eder ve bu durumda sıcak şeklinde geri bildirimde bulunur. Üretici ise bu durumda kazancını maksimize etmeye çalışarak en iyi sonuçları üretmek için geri yayılım kullanarak ağırlıklarını günceller.

Üretken rakip ağlar denetimsiz eğitimin yanında, denetimli eğitim ve yarı-denetimli eğitim olarak da uygulanabilir. Üreticinin ön eğitimi kabul edilebilir düzeye gelene kadar devam ettirilir.

Üretici gerçeğe yakın sonuçlar üretmeye başladığında ayırıcı üretilenler ve gerçek olanlar arasında ayırım yapmakta zorlanır. Bu durumda ayırıcı da kendi ağırlıklarını geri yayılım algoritması kullanarak günceller. Böylece üretilenler ve gerçek olanları ayırt etmede daha yetenekli olur.

Sadece üretici ağı ters evrişimli sinir ağı (deconvolutional neural network), ayırıcı ağı ise evrişimli sinir ağı (convolutional neural network) gibi düşünebiliriz. Bütün olarak da üretken rakip ağları bir oto kodlayıcı (autoencoder) gibi görebiliriz [5]. Oto kodlayıcılar gibi veri setine yakın yeni sonuçlar üretir. Fakat bu gibi bir sisteme kelime üretimi gibi bir doğal dil işleme uygulaması uygulamaya kalktığımızda modelin biraz değişmesi gerekmektedir.

Basitçe bir üretken rakip ağın algoritması aşağıdaki gibidir;

1. Eğitimdeki iterasyon kadar alttaki adımları yap
2. k adım kadar alttaki adımları yap
3. Gürültü vektörü pg(z)'den m adet örnek {z(i), ..., z(m)} mini grubu al
4. Üretilmiş pdata(x) verilerinden dağıtılmak üzere m adet örnek {x(1), ..., x(m)} mini grubu al
5. Eşitlik 2.1 kullanılarak stokastik eğim düşüm yöntemi ile ayırıcı ağırlıkları güncelle

$$Dloss_{real} = \log\left(D(x)\right)$$
$$Dloss_{fake} = \log\left(1 - D\left(G(z)\right)\right)$$
$$Dloss = Dloss_{real} + Dloss_{fake}$$
$$\log\left(D(x)\right) + \log\left(1 - D\left(G(z)\right)\right)$$
$$\frac{1}{m}\sum_{i=1}^{m} \log\left(D\left(x^i\right)\right) + \log\left(1 - D\left(G\left(z^i\right)\right)\right) \quad (2.1)$$

6. Döngüyü tamamla
7. Gürültü vektörü pg(z)'den m adet örnek {z(i), ..., z(m)} mini grubu tekrar al

8. Eşitlik 2.2 kullanılarak stokastik eğim düşüm yöntemi ile üretici ağırlıkları güncelle

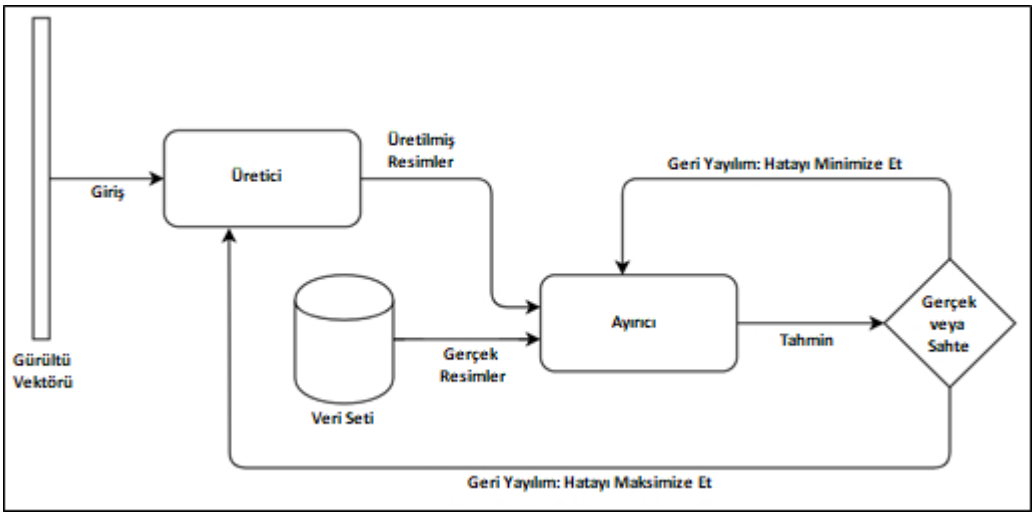
$$Gloss = \log\left(1 - D\left(G(z)\right)\right) \quad veya \quad -\log\left(D\left(G(z)\right)\right)$$

$$\frac{1}{m}\sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^i\right)\right)\right)$$

veya

$$\frac{1}{m}\sum_{i=1}^{m} -\log\left(D\left(G\left(z^i\right)\right)\right) \tag{2.2}$$

9. Döngüyü tamamla



Şekil 2.1. Üretken rakip ağlar için akış diyagramı.

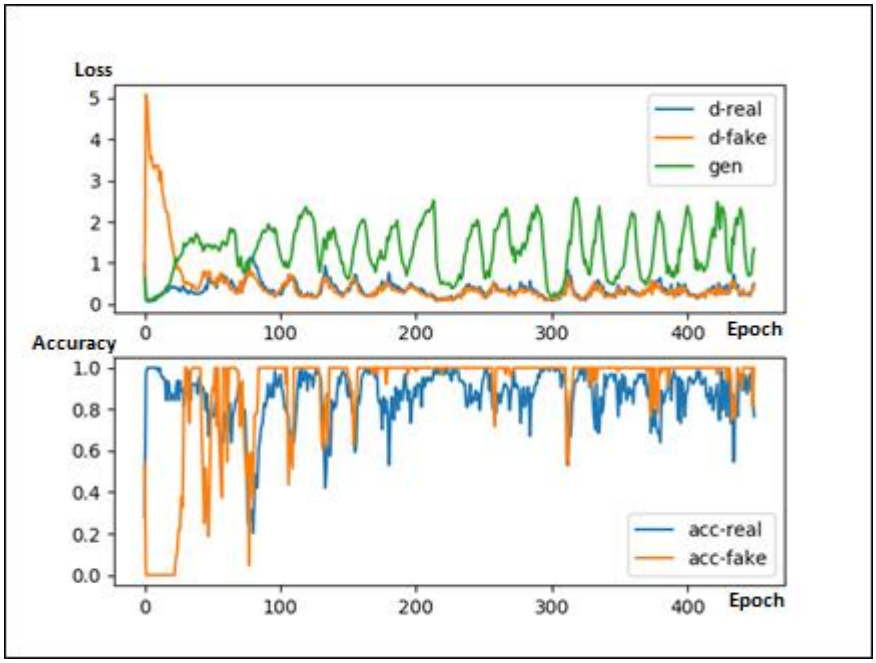
Üretken rakip ağların yapısı algoritma ve akış diyagramından anlaşılabileceği gibi bir gürültü vektörü (noise vector) ile başlar. Gürültü ile başlandığında sistemin Nash dengesine gelmesi uzun sürebildiğinden veri setinin bir kısmından ön eğitim yapılabilir. Bu hem öğrenme süresini hızlandırır hem de sistemin kararlılığının arttırır. Bu işlemden sonra küçük gruplar (minibatch) halinde çıktılar alınır ve ayırıcıya gönderilir. İki sinir ağının ağırlıkları geri yayılım ile güncellenir. Daha sonrasında sonuç olarak sistemin Nash dengesine ulaşması ve en gerçeğe yakın çıktılar üretmesi hedeflenir. Akış diyagramı Şekil 2.1'deki gibidir.

Geri yayılımdan bahsedersek ayırıcı x'in gerçek bir çıktı olma olasılığını gösteren bir D(x) oluşturur. Buradaki amaç çıktıların gerçeğe en yakın olma olasılığını maksimize, sahte olanları da minimize etmektir. Buradaki hatayı hesaplamak için çapraz düzensizlik (cross-entropy) p log(q) kullanılır. Burada geçen p gerçek çıktılar için 1 sahteler için ise 0'dır. Burada hedeflenen üretken rakip ağ modelinin, en yüksek D(x) ile başarılı sonuçlar üretmesidir.

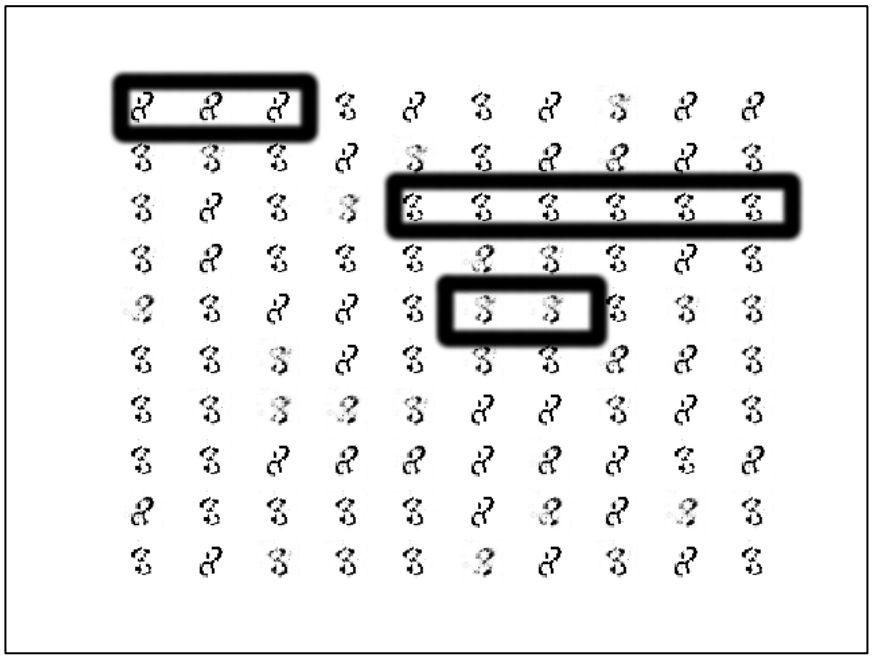
Üretken rakip ağların yapısı gereği bazı sorunları da beraberinde getirmektedir. Örneğin eğim kaybolması (vanishing gradients) ayırıcının üreticiye yeterli dönüş yapmamasından kaynaklanır. Veri eksikliğinden kaynaklanan bu durumda üretici gerçek veriyi çözmekte sıkıntı yaşar ve çıktılar gerçeğe uzak olur. Aktivasyon fonksiyonunun ürettiği sonuçlar eğitim süresince sıfıra yakınlaşır. Bu sorunun çözümü için Wasserstein loss [6] ya da Modified minimax loss [7] yöntemleri kullanılır.

Bir başka problem mode collapse denilen problemdir. Bu problem eğitim sırasında üreticinin başarılı bir sonuç bulması sonrasında devam eden çıktıların o sonuca yakınsamasıdır. Üretkenlik ve özgünlük azalmış olduğundan model başarılı sonuçlar üretmeyi bırakır. Yerel minimum noktasından çıkamaz ise üretici aynı çıkışları verir ve ayırıcı bunları reddetmeye başlar. Bu sorundan kaçınmak için Wasserstein loss [6] veya Unrolled GAN [8] yöntemleri kullanılır.

Mode collapse yaşanan bir çalışmada alınan sonuçlar aşağıda verilmiştir [9]. Şekil 2.2. deki ilk grafikte üreticinin kayıp değişimleri yeşil ile gösterilmiştir. Eğitim sırasında büyük dalgalanmaların yaşandığı görülebilir. Buna karşılık ayırıcıda ise küçük salınımlar görünmektedir. İkinci grafikte ise görülebileceği gibi üretici başarısız ve tekrarlı çıktılar vermesi sonucunda ayırıcı çıktıların gerçek ya da sahte olduğu konusunda test boyunca yüksek başarı göstermiştir. Bunun sonucunda çıktılardaki benzerlik sorunu Şekil 2.3'te gösterilmiştir.

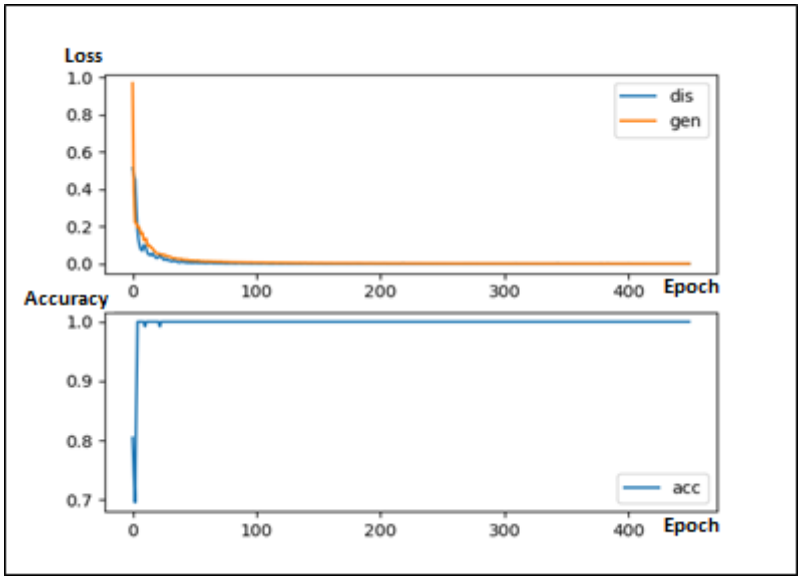


Şekil 2.2. GAN'da görülen mode collapse yaşanması sırasındaki doğruluk ve kayıp çizgi grafiği [9].

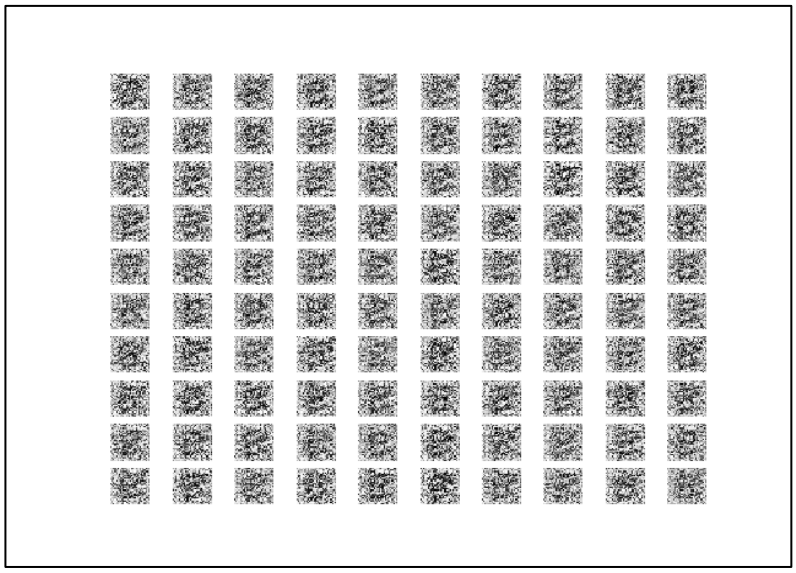


Şekil 2.3. Mode collapse sorunu görülen bir GAN çalışmasında el yazısı ile yazdırılmaya çalışılan 8 rakamının tekrar edilmesi [9].

En çok karşılaşılan diğer bir problem ise yakınsama başarısızlığıdır (failure to converge). Burada üretken rakip ağ modeli yakınsama yapamadığından üreticinin çıktıları gerçekten uzaktır [9]. Burada sorun için uygulanan çözüm yöntemlerden ikisi ayırıcının girdilerine noise ekleme veya ayırıcının ağırlıklarına ceza uygulanmasıdır. Şekil 2.4 ve Şekil 2.5'te yakınsama problemi olan modelin test sonuçları verilmiştir.



Şekil 2.4. Yakınsama başarısızlığı görülen bir GAN modeli için doğruluk ve kayıp grafiği [9].



Şekil 2.5. El yazısı ile 8 yazdırmak istenen bir GAN modelinin yakınsama başarısızlığı görülmesi sonucu çıktıları [9].

BÖLÜM 3

LİTERATÜR TARAMASI

Üretken rakip ağlar yapay sinir ağı mimarisine benzer bir üretici modeldir. Diğer üretici modellerde olduğu gibi üretken modelleme mevcutta olmayan yeni örnekler oluşturmayı hedefler.

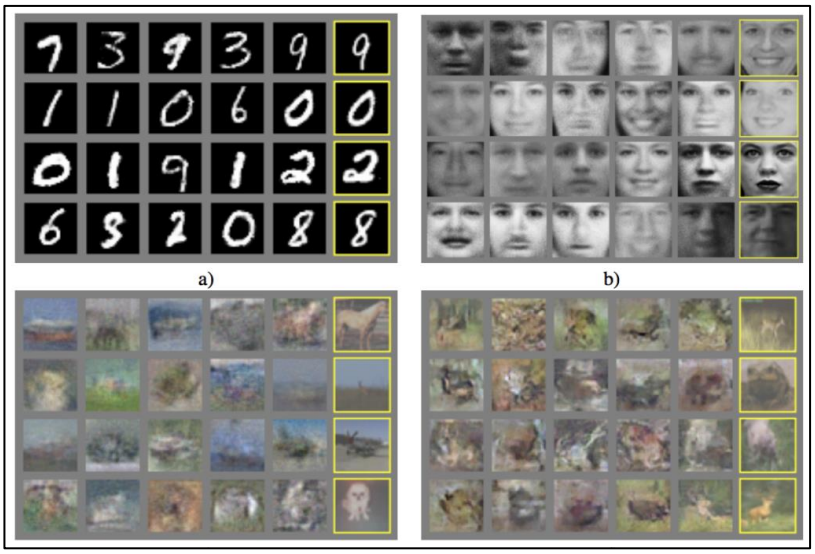
Bir üretken rakip ağ iki yapay sinir modelinden meydana gelmektedir. Birisi üretici ağdır ve hedefi yeni çıktılar üretmektir. Diğer ağ ise ayıcı olarak adlandırılır ve üretilenlerin gerçek mi sahte mi olduğunu anlamaya çalışır.

Bu iki ağ birbirleriyle çekişme içindedir ve aralarında oyun teorisindeki gibi bir denge durumu olması sağlanır. Eğitim sonrasında üretken modelden yeni çıktılar üretmesi istenebilir [10].

Üretken rakip ağları uygulama alanlarına göre birkaç bölgeye ayırmak istersek:

1. Resim veri setleri üzerinden üretilen çalışmalar [4, 11, 31]
2. Fotoğraflardan insan yüzleri üretme üzerine çalışmalar [11, 13, 14, 21, 22, 28]
3. Yeni çizgi kahramanlar üretme üzerine yapılan çalışmalar [15, 29]
4. Resimlerden resime geçiş çalışmaları [23, 24, 30]
5. Yüz açılarına göre tahmin çalışmaları [25, 26]
6. Yaşlandırma çalışmaları [27]
7. Çözünürlük üzerine yapılan çalışmalar [18, 19, 20]
8. 3 Boyutlu obje çalışmaları [12, 16, 17]
9. Metin üretimi çalışmaları [32, 33, 34, 35, 36, 37, 38]

Üretken rakip ağlar üzerindeki ilk çalışma Ian Goodfellow vd. tarafından 2014'teki yayınladığı makalede geçmektedir [4]. Bu çalışmada MNIST el yazısı rakam veri seti, CIFAR-10 obje resimleri veri seti ve Toronto yüz veri seti üzerinde üretken rakip ağlar kullanılarak alınan sonuçlar paylaşılmıştır. Alınan sonuçlar Şekil 3.1'dedir.

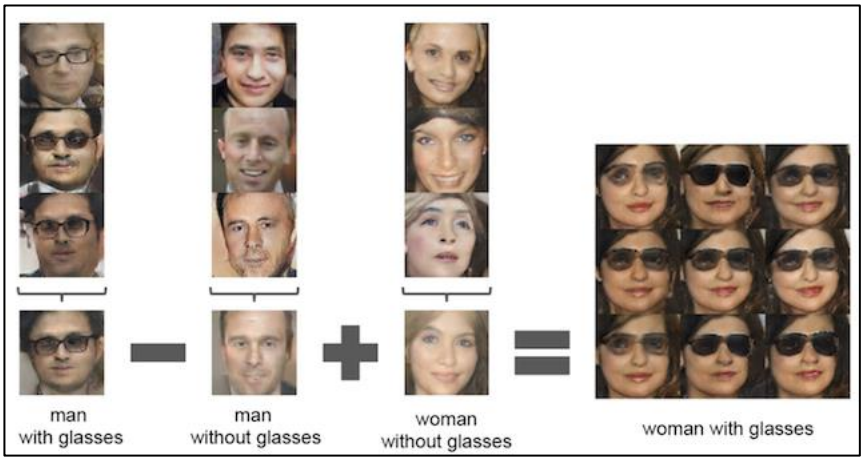


Şekil 3.1. 2014'teki üretken rakip ağlar modelinin çıktıları [4].

2015'te Alec Radford vd. tarafından oluşturulan DCGAN normal bir üretken rakip ağlar modeline derin evrimsel ağlar yapısı ekleyerek oluşturmuştur [11]. Bu çalışmasında büyük veri setlerinde de başarılı bir eğitim gerçekleştirebileceğini ortaya koymuştur. Alttaki resimde DCGAN ile üretilmiş yeni yatak odası resimleri bulunmaktadır. Üretilen yeni çıktılar Şekil 3.2'dedir.

Şekil 3.2. DCGAN kullanılarak yeni olarak üretilen yatak odası örnekleri [11].

Ayrıca bu makalesinde DCGAN'ın aritmetik vektör becerilerini kullanarak insan yüzleri üzerinde yaptığı çalışmayı göstermiştir. Şekil 3.3'de sonuçlar verilmiştir.

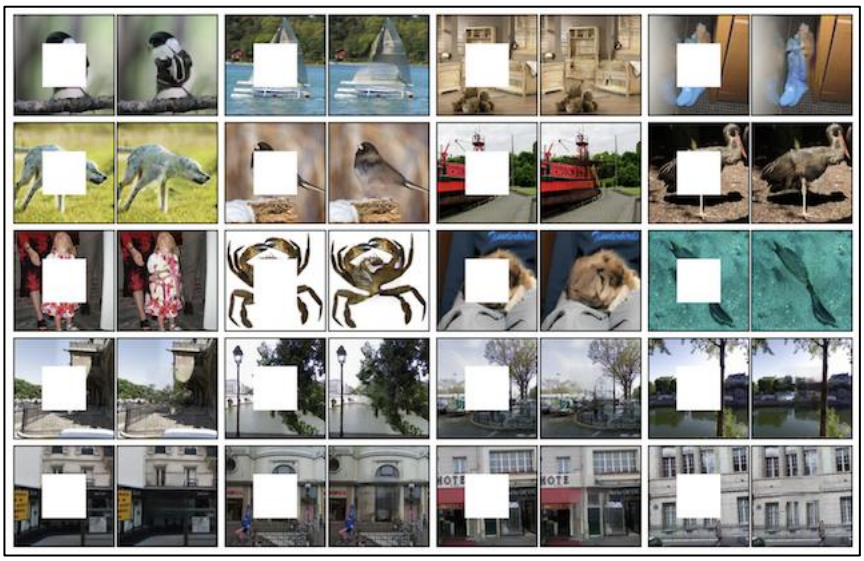


Şekil 3.3. Aritmetik vektör becerileri kullanılarak insan yüzü üzerinde gözlük ekleme çalışması [11].

Carl Vondrick vd. tarafından 2016'da yaptığı çalışmasında üretken rakip ağlar kullanarak videodaki sahnenin bir sonraki karesinin tahmin etmekte başarılı olduğunu ortaya koymuştur [12]. Buradaki başarı statik kısımlarda daha yüksek olduğu belirtilmiştir. Örnek video karesi tahminleri Şekil 3.4'tedir.

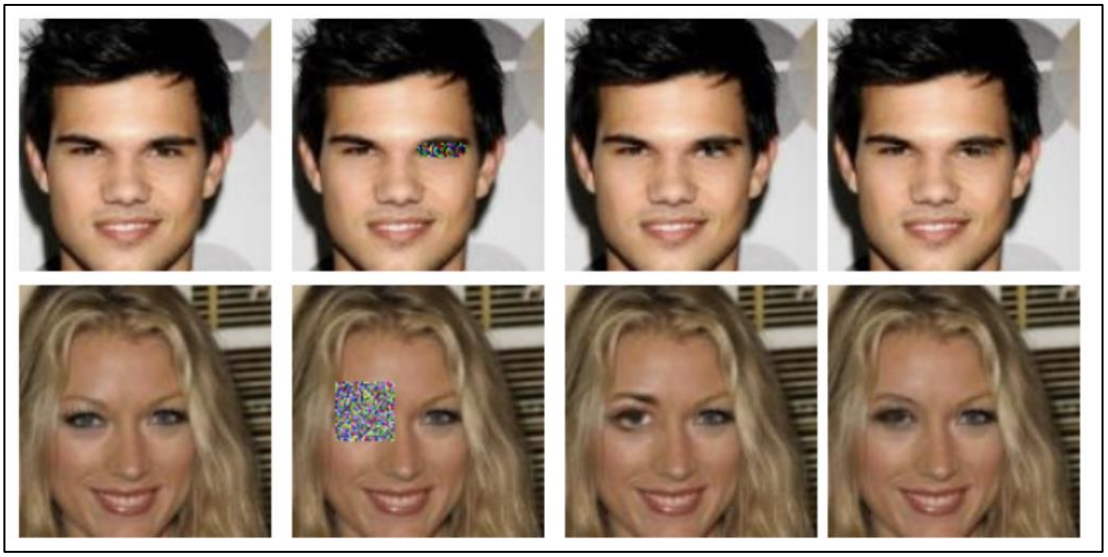
Şekil 3.4. Üretilen videonun değişik karelerine ait çıktıları [12].

2016 yılında yayınlanan bir başka çalışmada ise Deepak Pathak vd. fotoğraflardaki eksik ya da boş kısımların üretken rakip ağ kullanılarak tamamlanması ile alakalı çalışmasını yayınlamıştır [13]. Şekil 3.5'de çalışma sonuçları verilmiştir.



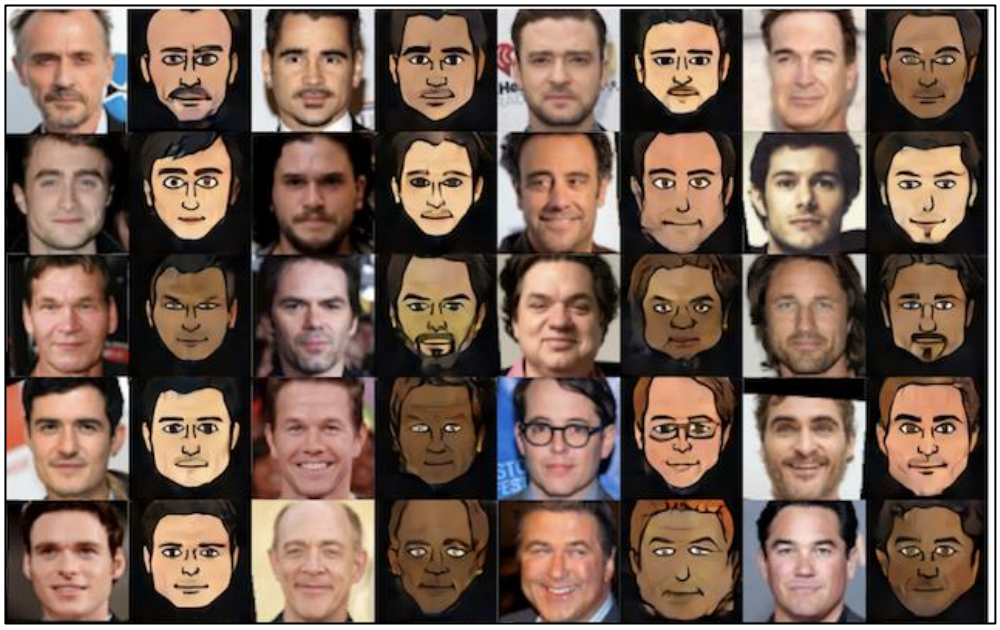
Şekil 3.5. Fotoğraflardaki eksik kısımların tamamlanması [13].

Bu çalışmadan esinlenerek Yijun Li vd. 2017'deki makalesinde üretken rakip ağ kullanarak hasarlı insan yüzü fotoğraflarının düzeltilmesi sağlamıştır [14]. Alınan çıktılar Şekil 3.6'dadır.



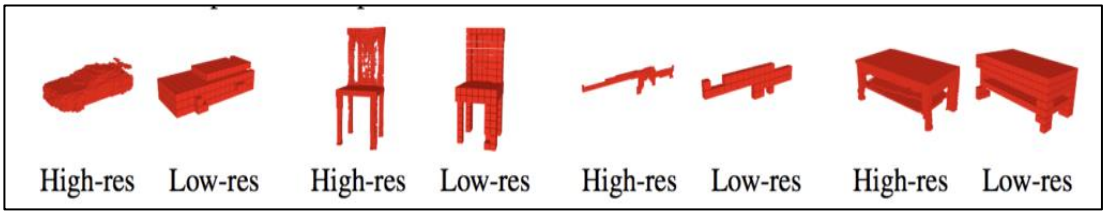
Şekil 3.6. Hasarlı fotoğrafların üretken rakip ağlar ile düzeltilmesi çalışması [14].

Yaniv Taigman vd. yayınladığı makalesinde üretken rakip ağlar kullanarak ünlülerin fotoğraflarından emojiler üretmeyi başarmıştır [15]. Üretilen emojiler Şekil 3.7'dedir.



Şekil 3.7. Üretken rakip ağlar kullanılarak üretilen ünlülere ait karikatür suratları [15].

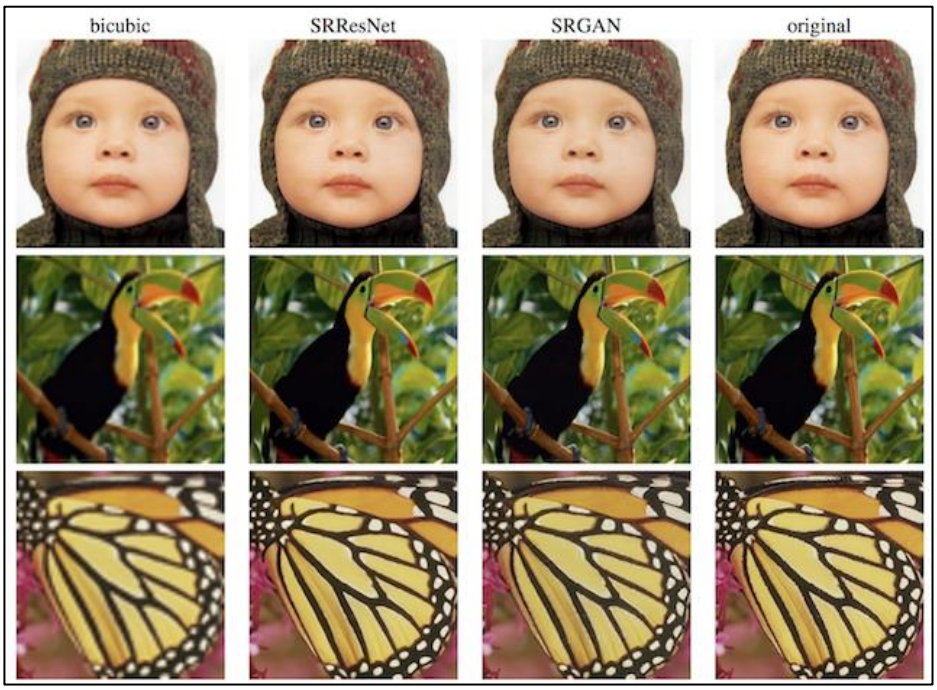
Jiajun Wu vd. 2016'daki makalesinde üretken rakip ağlar ile üç boyutlu objeler üretmeyi başarmıştır [16]. Çalışma sonuçları Şekil 3.8'dedir.



Şekil 3.8. Üç boyutlu üretilen yüksek ve düşük poligonlu objeler [16].

2016'daki bir başka makalede Matheus Gadelha vd. iki boyutlu resimlerden üç boyutlu objeler üretmeyi başarmıştır [17].

Christian Ledig vd. tarafından yayınlanan 2016'daki makalesinde özellikle kullandığı SRGAN modeli ile çözünürlüğü çok daha yüksek çıktılar üretebileceğini ortaya koymuştur [18]. Şekil 3.9'da çalışmayla alakalı örnek verilmiştir.

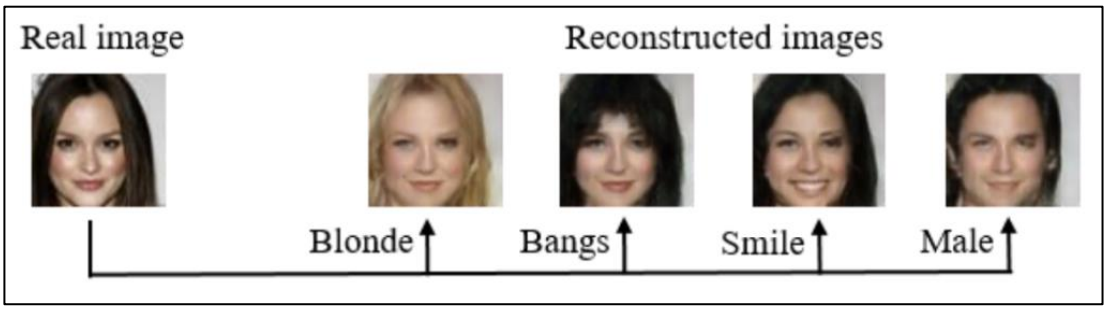


Şekil 3.9. Çözünürlük yükseltme çalışmasından birkaç örnek [18].

Huang Bin vd. çözünürlük yükseltme çalışmasını insan yüzleri üzerinde özelleştirilmiş versiyonunu 2017'de yayınlamıştır [19]. Yüz koşullu üretken rakip ağ (FCGAN) herhangi bir yüz bilgisi almadan ve düşük çözünürlüklü bir fotoğraftan bile yüksek çözünürlüklü bir yüz oluşturabilmektedir.

Subeesh Vasu vd. ise 2018'de yaptığı çalışmasında manzara fotoğraflarının çözünürlüğünü yükseltmeyi başarmıştır [20]. Bu çalışmadaki model gelişmiş algısal bir ağ kullanarak özellikle yakınlaştırıldığındaki bozuklukları giderme konusunda başarı sağlamıştır.

2016 yılında yayınladığı makalede Guim Perarnau vd. insan suratlarını belirli özellikler çerçevesinde üretmeyi başarmıştır [21]. Çalışmanın prensibi Şekil 3.10'da gösterilmiştir.

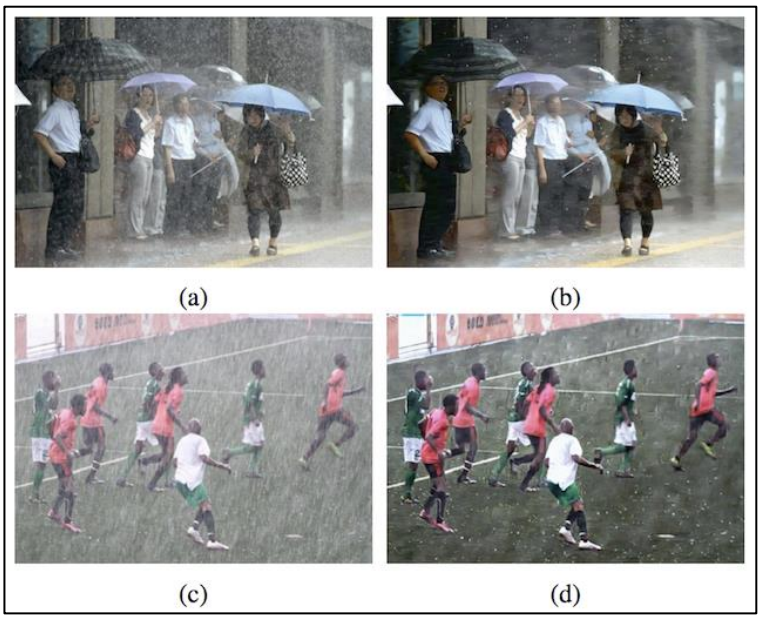


Şekil 3.10. Örnek bir fotoğraf üzerinde özelliklere göre çıktının değişmesi [21].

Ming-Yu Liu vd. 2016'daki yayınladığı makalesinde CoGAN ile renk ve derinlik gibi özellikleri öğrenebildiğini ortaya koymuştur [22]. Sonuçlar Şekil 3.11'de verilmiştir.

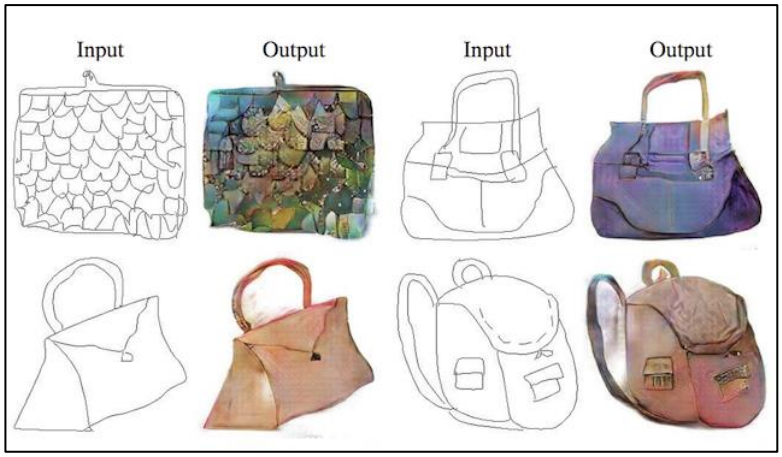
Şekil 3.11. CoGAN ile üretilen insan yüzlerinden birkaç örnek [22].

He Zhang vd. 2017'deki yaptığı çalışmasında fotoğraflardan yağmur ve kar silinmesi için üretken rakip ağ kullanımının bir örneğini ortaya koymuştur [23]. Çalışmanın çıktıları Şekil 3.12'de gösterilmiştir.



Şekil 3.12. Örnek iki fotoğraf üzerinde yağmur ve karın silinmesi [23].

Phillip Isola vd. tarafından yayınlanan makalesinde özellikle görüntüden görüntüye geçişte üretken rakip ağlar kullanmıştır [24]. Bu çalışmasında gündüzden geceye, siyahtan beyaza ve kara kalem çizimlerinden renklendirmeye geçişte başarılı olmuştur. Çalışma ile alakalı birkaç örnek Şekil 3.13'de verilmiştir.

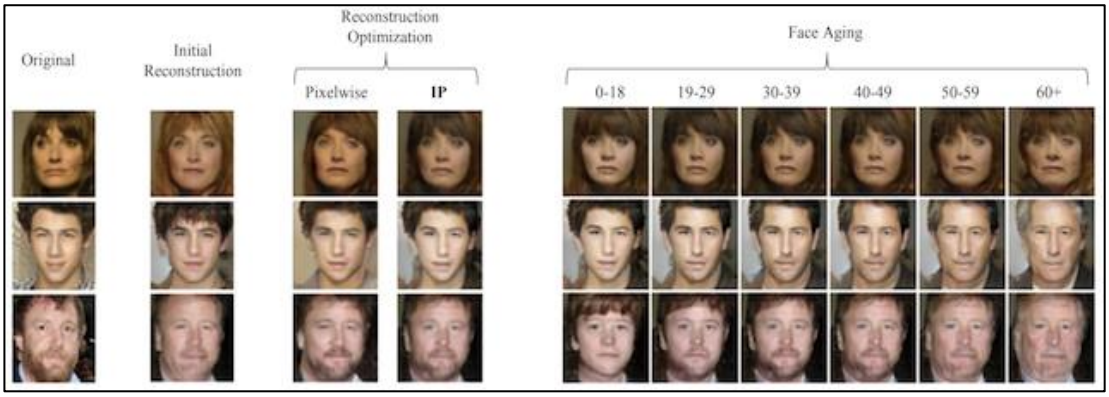


Şekil 3.13. Çizimlerin renklendirilmesi sırasında üretken rakip ağlar kullanılması [24].

Rui Huang vd. 2017'de yayınladığı makalede farklı açılardaki insan fotoğraflarından yüzün karşıdan görünüşünü üretken rakip ağ ile üretilmesini sağlamıştır [25]. Burada üretilen resimlerin yüz tanımlama ve doğrulama sistemlerinde kullanılması hedeflenmiştir. Yüz açılarıyla ilgili olan bu çalışmanın sonuçları Şekil 3.14'teki gibidir.

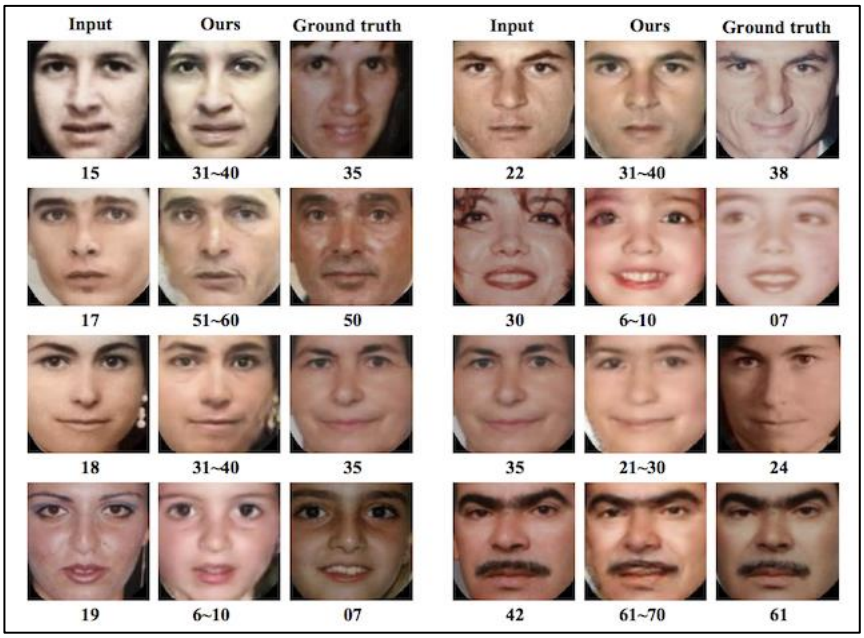
Şekil 3.14. Yandan çekilen fotoğraflardan yüzün karşıdan görünüşünün tahmini [25].

2017 yılında yayınlanan bir başka makalede Grigory Antipov vd. üretken rakip ağlar ile verilen bir insan fotoğrafının farklı yaşlardaki halinin üretimini sağlamıştır [26]. Yayınlanan çalışmanın çıktıları Şekil 3.15'te verilmiştir.



Şekil 3.15. Yandan çekilen fotoğraflardan yüzün karşıdan görünüşünün tahmini [26].

Zhifei Zhang vd. makalesinde ileri ve geri yönlü yaşlardaki yüz tahmini yapmak için üretken rakip ağlar kullanmıştır [27]. Bu tahminler Şekil 3.16'da gösterilmiştir.

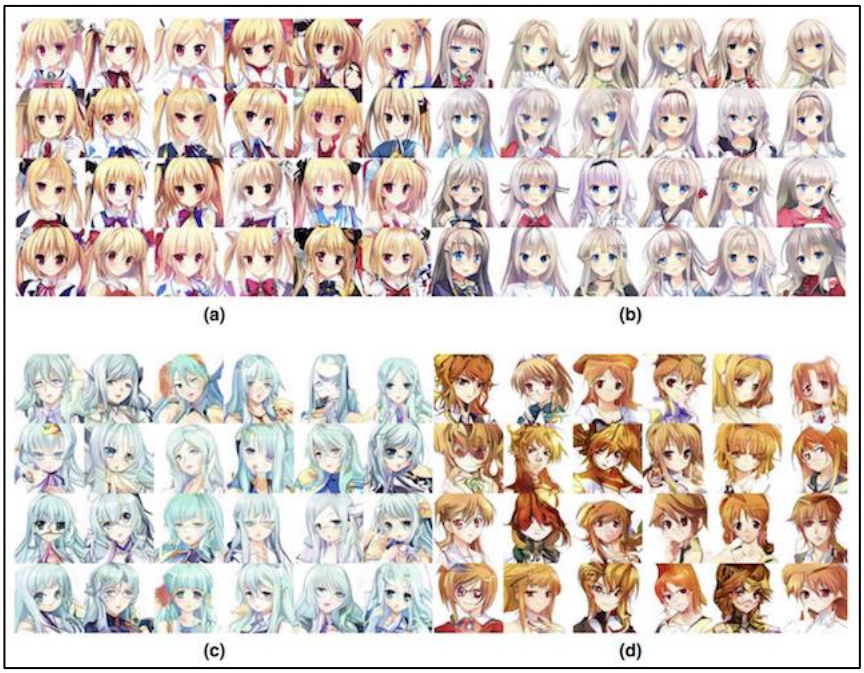


Şekil 3.16. İki fotoğraf arasındaki bir yaşa ait insan yüzü üretimi [27].

Tero Karras vd. yayınladığı makalede gerçeğe yakın insan yüzleri üretebileceğini göstermiştir. Bu çalışmada elde edilen sonuçların başarısı birçok kişinin dikkatini çekmiştir [28]. Şekil 3.17'de bu 2017'deki modelin sonuçları görülebilir.

Şekil 3.17. Ünlülere ait fotoğrafların eğitimde kullanıldığı üretken ağların çıktıları [28].

Çizgi kahramanların üretken rakip ağlar ile üretilebileceğini 2017 yılında Yanghua vd. tarafından yayınlanan makalede gösterilmiştir [29]. Bu çalışmadan esinlenerek DCGAN ile yeni Pokemon karakterleri üretimi ile alakalı denemeler yapılmıştır. Bu karakterlere ait sonuçlar Şekil 3.18'dadır.



Şekil 3.18. Yanghua vd. yaptığı çalışma sonucunda üretilen dört karaktere ait fotoğraflar [29].

Ting-Chun Wang vd. tarafından 2017 tarihinde yayınlanan makalesinde belirli bölgelere ayrılmış bir girdi için üretken rakip ağlar kullanarak gerçeğe yakın bir fotoğraf çıktıları almayı başarmıştır [30]. Bunla ilgili örnek Şekil 3.19'da verilmiştir.

Şekil 3.19. Bir anlamsal görüntünün üretken rakip ağ kullanılarak çıktısının oluşturulması [30].

2018 yılında Andrew Brock vd. tarafından yayınlanan çalışmada BigGAN teknikleri kullanılarak gerçekten ayırt edilemeyecek düzeyde çıktılar elde edilmiştir [31]. Bu çıktılar Şekil 3.20'de verilmiştir.

Şekil 3.20. BigGAN tekniği ile üretilen gerçeğe yakın çıktılar [31].

Bunların dışında üretken rakip ağlar ile metin üretimi üzerine de çalışmalar yapılmaktadır. Metnin ayrık ve soyut doğası gereği üretken rakip ağlar ile metin üretimi aşamasında bazı problemler ile karşılaşılmaktadır. Burada akla gelen ilk çözümlerden biri pekiştirmeli öğrenme (reinforcement learning) uygulamaktır.

Tong Che vd. tarafından 2017 yılında yayınlanan makalesinde üretken rakip ağlardaki kararsızlığı yok etmek için maksimum olabilirliği arttırılmış üretken rakip ağları (MaliGAN) önermiştir [32]. Bu yöntemde hedefi optimize etmektense ayırıcı sonuçları kullanılarak yeni bir hedef üretilir.

Jiaxian Guo vd. yine 2017 yılında LeakGAN ile sızan bilgilerle uzun metin üretimi modelini yayınlamıştır [33]. Bu modelde ayırıcı üreticiye daha fazla bilgi yönlendirmesine izin verilerek daha başarılı ve anlamını kaybetmeyen uzun metinler üretilebileceği ortaya koyulmuştur.

SeqGAN eğim tedbirli bir sıra oluşturucudur. Lantao Yu vd. tarafından 2017 yılında yayınlanan makalesinde üretken rakip ağın eğitiminde pekiştirmeli öğrenme uygulamıştır [34]. Üreticinin sınıflandırma problemlerini atlayıp direk olarak eğim tedbirli bir şekilde eğitimine devam eder.

Kevin Lin vd. 2017 yılında yaptığı çalışmada RankGAN ile ayırıcıyı sınıflandırmak için eğitmek yerine sıralama ve bir referans grubu oluşturacak şekilde değiştirmiştir [35]. Daha sonrasında puanlandırma sistemi ile daha iyi değerlendirme yaptığını göstermiştir.

William Fedus vd. 2018 yılında kelimelerin bir önceki kelimeye göre koşullandırılması yerine maksimum olasılık yöntemi ile eğitilir [36]. MaskGAN metodu ile üretici daha kaliteli sonuçlar üretilebileceği gösterilmiştir.

Üstteki yöntemler performans olarak başarılı olsalar dahi optimizasyon aşamalarında zorluk yaşamaktadır.

Zhang vd. 2017 yılında yaptığı çalışmada TextGAN diğer modellerin aksine pekiştirmeli öğrenme içermeden metin üretimi yapmıştır [37].

Matt Kusner vd. 2016 yılında yayınladığı çalışmasında üretken rakip ağ üzerinde Gumbel-softmax dağılımı kullanarak parametrelerin farklılaşması engellenmiştir [38]. Böylelikle eğitimin hızı ve istikrarı artmıştır.

Üretken rakip ağlar ile metinler üzerinde veri çoğaltma çalışmaları incelendiğinde 2017 yılında Antreas Antoniou vd. yayınladığı çalışmasında veri artırmak için üretken rakip ağ kullanmıştır. Çalışmada kullandığı DAGAN ile sınıflandırmada başarı artışı gözlenmiştir [39].

Georgios Douzas vd. 2018 yılında normal dağılımlı olmayan veri seti üzerinde sınıflandırma problemini çözmek için üretken rakip ağ kullanmıştır. Çalışmada kullandığı cGAN ile sınıflandırma başarısını artırmıştır [40].

BÖLÜM 4

ÜRETKEN RAKİP AĞLAR İLE HABER METİNİ ÜRETİMİ

Bu çalışmada üretken rakip ağlar kullanılarak kendisine verilen haber metinlerinden yeni metinler üretmek amaçlanmıştır. Kullanılan veri seti internet üzerindeki Türkçe haber sitelerinden toplanmıştır. Bu veri seti 3058 haber ve 832 bin 302 kelimeden oluşmaktadır. Veri setinden bir kesit Çizelge 4.1'de verilmiştir. Bu haberler iki kategoriye ayırılmıştır. Bu kategoriler anlamına göre olumlu haberler ve olumsuz haberler şeklindedir. Bu kategorileme sonucunda 2 bin 949 olumlu habere karşılık 109 olumsuz haber bulunmaktadır. Tahmin edileceği üzerine veri seti üzerindeki bu eşit olmayan dağılım sınıflandırmayı negatif olarak etkilemektedir.

Çizelge 4.1. Haber metinlerinin ve sınıflandırma sonuçlarının bulunduğu veri setinden bir kesit.

No	Sınıf	Veri
61	Olumlu	Başkan Atabay'dan yeni yıl ziyaretleri Didim Belediye Başkanı Deniz Atabay ilçedeki bankaların yöneticilerine yeni yıl ziyaretinde bulundu. Başkan Atabay yeni yıl ziyaretleri kapsamında İlçede faaliyet gösteren Işbank Müdürü Ömer Aksoy, Garanti Bankası Müdürü Pınar Sönmez Metin, Vakıfbank Müdürü Hüseyin Turpçu ve TEB Bankası Müdürü Nurçin Yılmaz'ı ziyaret ederek 201 8 yılını başarı dostluk ve mutluluk içerisinde geçmesini diledi.
62	Olumsuz	Tünelden geçmedik çünkü... TRAFİK KİLİT AMA AVRASYA YİNE BOŞ M Avrasya Tiincli'nden geçişlerin beklenenin altında kalması İstanbul dünya trafik yoğunluğunda ilk 10'da yer almasına rağmen sürücüler neden tüneli tercih etmedi?' sorusunu gündeme getirdi. Gözler, tek yönde 16 TL'lik geçiş ücretine çevrildi. Tüneldeki ikinci krizi vatandaş şikayetleri ortaya koydu: M OGS'de para olmasına rağmen ceza yazıldı. M 'Plakaya ait ceza yoktur' yazısından üç gün sonra iki geçiş için 10 kat ceza kesildi. S/5
63	Olumlu	En çok dolar konuşuldu ama zirve borsanın Dünya ekonomisindeki toparlanma, 2018'de şirket karlarına, dolayısıyla borsa ve emtialara yarayacak Piyasalar, oynaklığın yüksek, sürprizlerin bol olduğu bir yılı daha geride bıraktı. Dolar çok konuşuldu ama 115 bin 840 puanla tüm zamanların en yüksek seviyesine çıkan Borsa Endeksi, yıllık yüzde47.6 ile 2017'nin getiri şampiyonu oldu. Eurodaki yükseliş yüzde 22.1 olurken dolardaki artış yüzde 7.5'te kaldı. Cumhuriyet altınındaki prim yüzde 20.9 olarak gerçekleşti. Yılın başında 1.000 TL'si olan için mevduatın getirisi 106 TL, tahvilin getirisi ise 123 TL olarak hesaplandı. Küresel ekonominin

4.1. KULLANILAN YÖNTEMLER

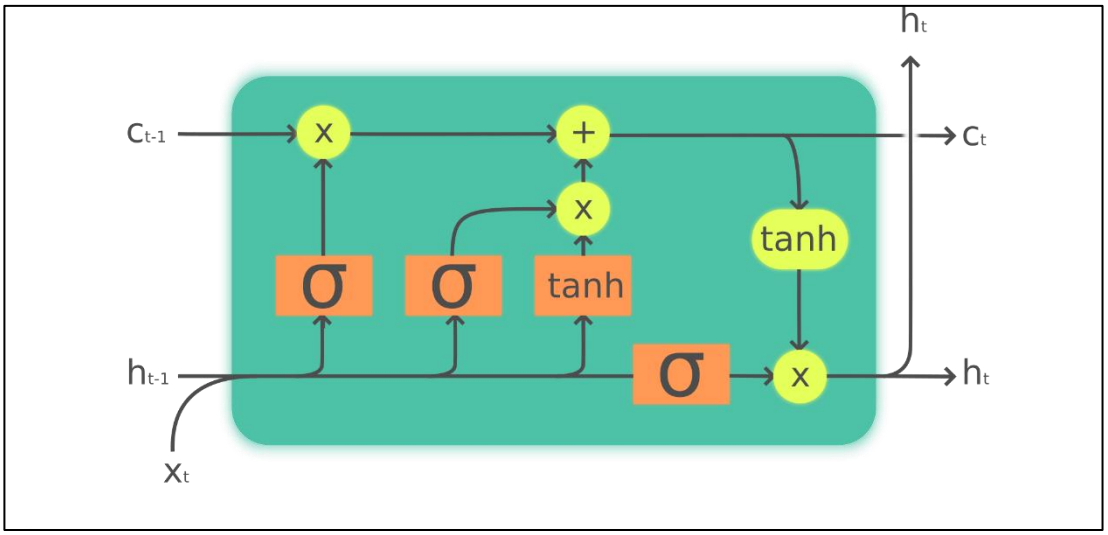
Üretken rakip ağlar ile metin üretimi yapmak için birçok farklı metot kullanılabilir. Bu yöntemlerin uygulanan problem ve istenilen sonuca göre başarı oranları değişmektedir. Bu çalışmada sınıflandırma başarısını arttırmak için gerekli olan eksik kategoriye ait metinlerin üretiminde aşağıda metotlardan yararlanılmıştır.

4.1.1. Tensorflow

Tensorflow Google Brain takımı tarafından kurulan açık kaynak kodlu bir yazılım kütüphanesidir [41]. 2017 yılında Google tarafından yayınlanmıştır. Tensorflow kurulan yapay sinir ağının anlaşılırlığını arttırarak tur (epoch) sonrasında sonuçları incelemeye yardımcı olan basit bir matematik kütüphanesidir. Bu çalışmada üretken rakip ağların kurulmasında Keras API kitaplıkları kullanılmıştır. Keras kütüphaneleri yüksek seviyeli bir dildir. Keras ara yüzü alt yapı olarak Tensorflow'u kullanmaktadır. Geri arama fonksiyonu (callback function) ile epoch sonralarında kayıtlar alınarak sonradan inceleme imkânı sağlar. Bu kayıtları incelemek için TensorBoard kullanılmaktadır. Ayrıca büyük veri setlerinin eğitimi uzun süre almaktadır. Keras GPU destekli çalışmasından dolayı eğitim aşamasını kısaltmaktadır.

4.1.2. LSTM

Uzun kısa süreli bellek (LSTM) özel bir RNN türüdür. Kavram ilk olarak 1995 yılında Sepp Hocreiter ve Jürgen Schmidhuber tarafından önerilmiştir. Uzun kısa süreli bellek bilgileri uzun süre hatırlamak için kullanılır. LSTM hücreleri, lojik kapılar ile neyin saklanacağına neyin unutulacağına karar verir. Şekil 4.1'de örnek bir LSTM hücresi görülmektedir. Yapısından dolayı zaman serili verilerde sınıflandırma ve tahmin konularına uygundur [42].

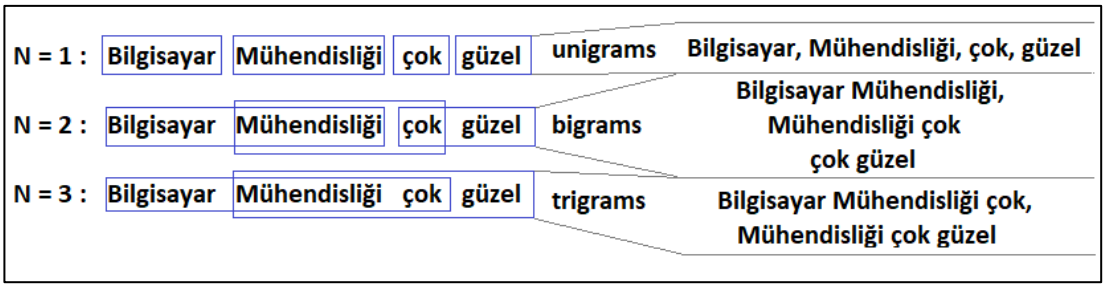


Şekil 4.1. Bir uzun kısa süreli belleğin iç yapısı [42].

Ağ şekli tekrarlayan sinir ağı (RNN) biçimindedir. Her adımda bir cümledeki sonraki kelimeyi tahmin ederek eğitilirler. Bu çalışmada 3 gizli katmanlı 128 düğümlü bir LSTM kullanılmıştır. Parti boyutu (batch size) 128 olup toplamda epoch sayısı 90'dır. Eğitim sırasında her epoch sonrası ağırlıklar incelenerek en iyi ağırlık değerleri seçilerek devam ettirilmiştir. Bu şekilde ağın sürekli kendini geliştirmesi amaçlanmıştır.

4.1.3. N-gram

N-gramlar temelde üçe ayrılır. Bunlar Unigram, Bigrams ve Trigrams olup cümleleri sırasıyla bir, iki ve üç kelimelik bölümlere ayırır [43]. Üçten büyükler ise n-gram olarak adlandırılır. Buradaki n sayısı büyüklüğü saklayabildiği bilgi ile doğru orantılıdır. N-gramlar ile tekrar oranı bulunabilir. Çalışma mantığı istatistiğe dayanmaktadır. Doğal dil işleme çalışmalarında n-gram modelleri yaygın olarak kullanılmaktadır. Şekil 4.2'de n-gram çalışma prensibi verilmiştir.



Şekil 4.2. N-gram'ın çalışma metodu.

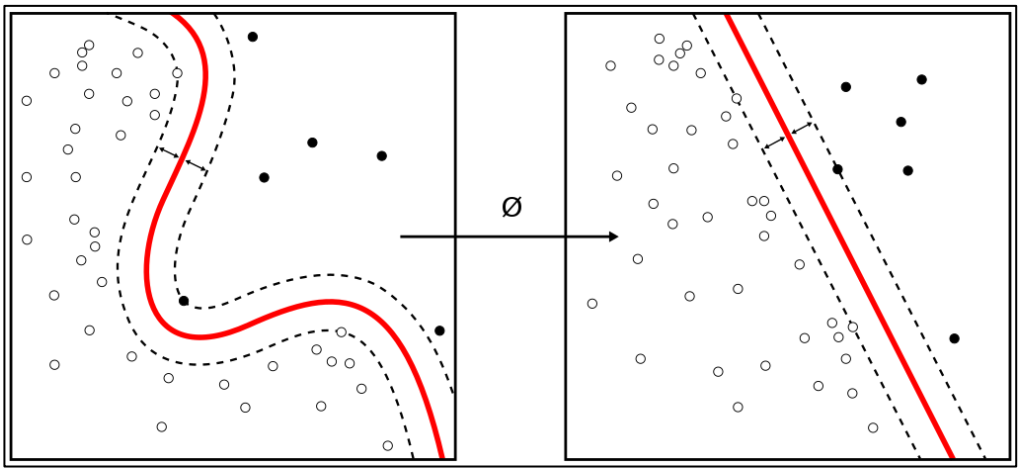
4.1.4. TF-IDF

Terim frekansı (term frequency) veri seti içerisinde bulunan kelimelerin sıklık oranlarını incelemek için kullanılır [44]. Ters belge frekansı (inverse document frequency) yönteminde ise bu sıklık oranlarına göre bağlaçlar tespit edilip çıkartılarak daha başarılı eğitim amaçlanmıştır.

Yapılan çalışmada üretken rakip ağlar ile üretilen metinlere bir dizi işlem uygulanmıştır. Örneğin TF-IDF kullanarak cümlenin anlamını değiştiren olumlu ve olumsuz sözcükler aranmıştır. Bu sayede en olumlu kelimeler ve en olumsuz kelimeler gibi bir sıralama yapılabilir. Bağlaçlar ve noktalama işaretleri gibi istenmeyen ögeler bulunabilir ve filtrelenebilir.

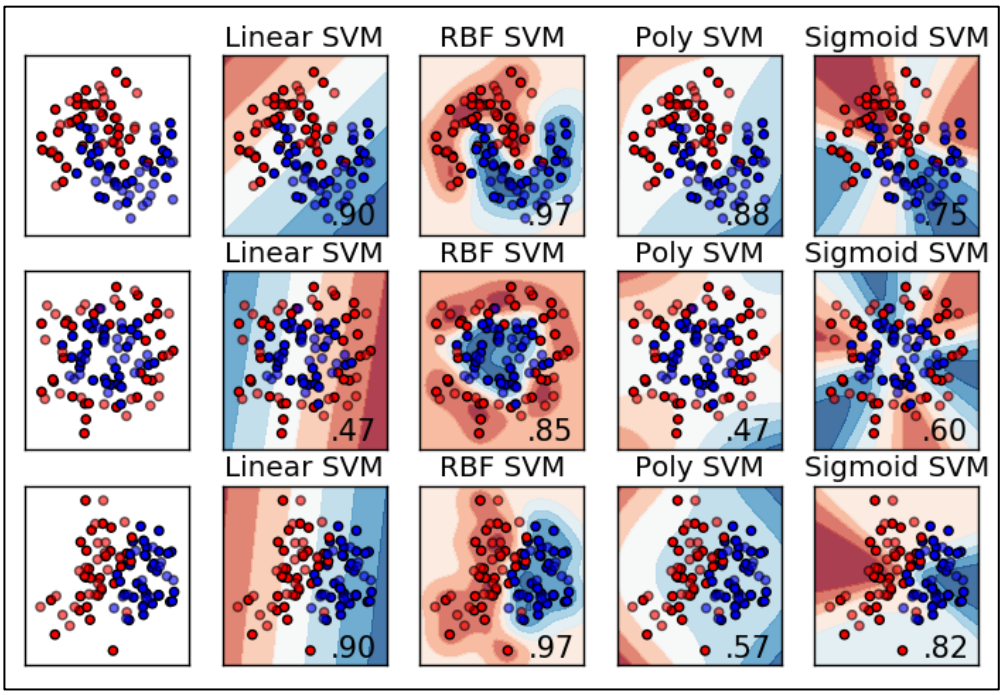
4.1.5. SVM

Destek vektör makinesi (support vector machine) sınıflar arasına bir karar vektörü oluşturur. Destek vektör makinelerinde çekirdek fonksiyonları (kernel function) sayesinde birçok duruma uygun karar çizgileri çizilebilir. Burada veri seti olumlu ve olumsuz olmak üzere iki kategoriden oluşmaktadır. Olumlu ve olumsuz gruplar SVM ile Şekil 4.3'deki gibi bir optimal karar çizgisi ile ayrılabilir [45].



Şekil 4.3. Destek vektör makinesinin doğrusal olmayan ve doğrusal olan karar çizgileri [45].

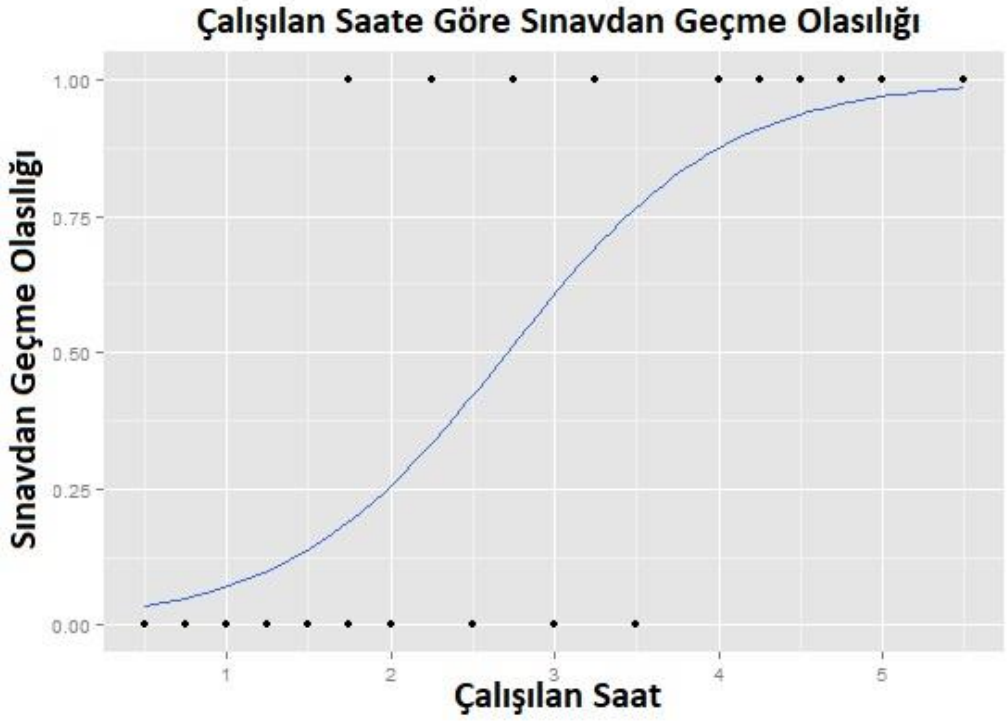
Diğer kernel fonksiyonları ile verilen veri seti özelinde istenilen sonuca göre farklı çekirdek fonksiyonları kullanılması daha başarılı karar çizgilerinin çizilmesinde etkilidir. Çekirdek fonksiyonlarının karşılaştırmalarının yapıldığı bir çalışmada alınan sonuçlar görselleştirilmesi Şekil 4.4'de verilmiştir [46].



Şekil 4.4. Destek vektör makinesinin farklı çekirdek fonksiyonlarına göre karar çizgileri [46].

4.1.6. Lojistik Regresyon

Belirli bir sınıfın ya da durumun olasılığını modellemek için lojistik regresyon kullanılabilir. Lojistik regresyonda her ögeye 0 ile 1 arasında toplamı bir olacak şekilde bir olasılık atanır [47]. Bu çalışmada alınan sonuçların değerlendirilmesi aşamasında diğer yöntemlerin yanında lojistik regresyonda kullanılmıştır. Şekil 4.5'de bu yöntemin çalışma prensibi gösterilmiştir.



Şekil 4.5. Sınavdan geçme ihtimali için lojistik regresyon grafiği [47].

4.1.7. Zemberek

Zemberek bir doğal dil işleme aracıdır. Türkçe üzerine özelleşmiştir [48]. Bu çalışmada üretilen metinlerin normalizasyonun aşamasında Zemberek kullanılmıştır.

4.1.8. Vektörizasyon

Makinelerin kelimeleri anlayabilmesi, işleyebilmesi, diğer kelimelerle ilişkiler kurabilmesi ve üzerinde matematiksel işlemler yapılabilmesi için kelimelerin vektörlere çevrilmesi gerekmektedir. Bu işleme kelime vektörizasyonu denilmektedir. Bu vektörizasyon işlemi sonrasında elde edilen değerler ile kelimelerin hangi sınıfa ait olduğu yada hangi kelimelerin daha çok birlikte kullanıldığı anlaşılabilir.

One-Hot Encoding yönteminde kelimelerin index değerleri ve metin içindeki konumları NxN tipindeki bir matris içinde saklanır. Her kelime kendi satır ve sütununda 1 değerini almaktadır. Matrisin boyutu veri setindeki kelime sayısının karesiyle doğru orantılı olarak artması sebebi ile büyük veri setlerinde bellek boyutunda yetersizlik görülebilir [49].

Word2Vec 2013 yılında Google tarafından bulunmuştur. Bir gözetimsiz öğrenme türüdür. Temel çalışma mantığı seçilen bir kelimenin sağ ve solundaki kelimeler ile ilişkilerini inceleyerek vektör değerlerini belirler ve sonrasında farklı kelimeler için sağ ve solundan yakınsamalar yaparak anlamını tahmin etmeye çalışır [50].

Word2Vec yöntemi iki farklı algoritmadan oluşmaktadır. Skip-gram algoritması ortadan alınan bir center word ile bunun iki yanındaki kelimelerin vektör değerlerinin tahmin edilmesi yöntemidir. Cbow algoritması ise skip-gram algoritmasının tam tersi mantığında çalışarak kenarlardan alınan kelimelere göre ortadaki kelimenin tahmin edilmesi yöntemidir [51].

Bu çalışmada kullanılan Keras API kendi içerisinde kelime vektörizasyon modülü bulundurmaktadır. Bu vektörizasyon modülü çalışma aşamaları aşağıdaki gibidir [52];

1. Her verinin standartlaştırılması (küçük harfler ve noktalama kaldırma)
2. Kelimelerin alt dizelere bölünmesi
3. Alt dizelerin tokenlere çevrilmesi (N-gram)
4. Tokenlere değer atanması (her token için benzersiz bir değer)
5. Değerlere göre vektörlerin oluşturulması

BÖLÜM 5

DENEYSEL ÇALIŞMALAR

Bu çalışmada kurulan üretken rakip ağ Python üzerinde hazırlanmıştır. İlk olarak veri seti hazırlık işlemleri yapılmıştır. Veri çiftleri ve bilgisi eksik hücreler var mı diye incelenmiştir. Haber verileri çekilirken hücrelere istenmeyen ögeler eklenip eklenmediğine bakılmıştır.

Öncelikle veri seti Sklearn kütüphanesi yardımı ile eğitim ve test veri seti olarak iki gruba ayrılmıştır. Test için % 30'luk kısım eğitim için % 70'lik kısım kullanılmıştır. Sınıflandırma sonuçlarının herhangi bir işlem yapılmadan önceki karışıklık matrisi (confusion matrix) Çizelge 5.1'deki gibi dağılmıştır. Burada ana köşegende doğru sınıflandırılmış sonuçlar bulunmaktadır.

Çizelge 5.1. Karışıklık matrisi dağılımı.

N = 918	Tahmin: 0	Tahmin: 1
Gerçek: 0	15	16
Gerçek: 1	4	883

Görüldüğü gibi test veri seti 918 haber metninin olumlu haberlerin doğru sınıflandırma oranı % 99,54904 iken olumsuz haberlerin doğru sınıflandırma oranı % 48,3871'de kalmaktadır.

Buradaki olumlu haberlerin doğru sınıflandırılmasının olumsuz haberlerin doğru sınıflandırılmasına göre başarı oranındaki büyük farkın sebebi veri setinin normal dağılım göstermemesidir. Olumlu haberlerin sayısının olumsuz haberlere göre çok yüksek olması sınıflandırmayı doğrudan etkilemektedir.

Üretken rakip ağ modelinde üretici kısmında LSTM kullanılarak 128 düğüm içeren 3 katman oluşturulmuştur. Ayırıcı bölümünde ise sınıflandırma için SVM ve TF-IDF'ten yararlanılmıştır.

Eğitimde epochlar sonrası en başarılı ağırlıklar not edilip bu ağırlıklardan daha iyiye yönlendirmeye çalışılmıştır. Üreticinin ürettiği bu metinler arasından veri setinin başarısızlığının ana sebebi olan olumsuz haberlerin sayıca eksikliği giderilmesi sağlanmıştır. Yani üreticinin ürettiği metinler ayırıcıya yönlendirilip olumlu ya da olumsuz olduğu incelenmiştir. Olumsuz metinler ayrılarak bir kenarda saklanmıştır. Bu yeni üretilen olumsuz metinler Zemberek aracı ile normalizasyon çalışması geçirmiştir.

İlk olarak yeni üretilmiş 50 olumsuz haber orijinal veri setine eklenmiştir. Eklenmesi ile toplam haber sayısı 3108'e ulaşmıştır. Karar destek makinesi çekirdek fonksiyonu olarak 'linear' seçilip c katsayısı bir olarak tutulmuştur. Bunun üzerine alınan sınıflandırma başarı sonuçları not edilerek oluşturulan tablo Çizelge 5.2'dedir.

Çizelge 5.2. Elli olumsuz haberin eklenmesi sonucu karışıklık matrisi.

N = 933	Tahmin: 0	Tahmin: 1
Gerçek: 0	26	19
Gerçek: 1	3	885
Olumsuz Tahmin Yüzde	57,77778	
Olumlu Tahmin Yüzde	99,66216	

Yukarıdan da görüldüğü gibi 50 adet olumsuz haberin eklenmesi olumsuz tahmin başarı yüzdesinde yaklaşık % 9'luk bir artışa sebep olmuştur. Olumlu haber tahmin başarı yüzdesi ise % 0,11312'lik düşüş göstermiştir.

Eğitime devam edip 50 adet daha yeni üretilen olumsuz haberin veri setine eklenmesinden sonra sınıflandırma başarısı Çizelge 5.3'deki gibi değişmektedir.

Çizelge 5.3. Yüz olumsuz haberin eklenmesi sonucu karışıklık matrisi.

N = 949	Tahmin: 0	Tahmin: 1
Gerçek: 0	53	20
Gerçek: 1	1	875
Olumsuz Tahmin Yüzde	72,60274	
Olumlu Tahmin Yüzde	99,88584	

Bu seferde ise sonuçlarda olumsuz haber tahmin başarı yüzdesi yaklaşık % 14'lük bir artış görürken olumlu haberlerin tahmin yüzdesi de % 0,22368'lik artış görmüştür.

Eğitime devam edip toplamda 150 adet yeni olumsuz haberin orijinal veri setine eklenmesiyle alınan sonuçlar Çizelge 5.4'de karşılaştırılmıştır.

Çizelge 5.4. Yüz elli olumsuz haberin eklenmesi sonucu karışıklık matrisi.

N = 965	Tahmin: 0	Tahmin: 1
Gerçek: 0	56	29
Gerçek: 1	1	879
Olumsuz Tahmin Yüzde	65,88235	
Olumlu Tahmin Yüzde	99,88636	

Yukarıdaki tabloda görüldüğü gibi 50 yeni olumsuz haberin veri setine eklenmesi olumlu haberlerin başarılı tahmin yüzdesinde küçük bir artışa sebep olmuştur. Buna karşılık olumsuz haberlerin başarılı tahmin yüzdesinde yaklaşık % 7'lik azalış görülmüştür.

Yine kullanılan metot ile eğitime devam edilip yeni üretilen olumsuz haberler orijinal veri setine eklenmeye devam edilmiştir. Yapılan eklemelerden sonra alınan sonuçlar kaydedilmiştir. Toplamda 200 yeni olumsuz haberin eklenmesiyle oluşan karışıklık matrisi Çizelge 5.5'deki gibidir.

Çizelge 5.5. İki yüz olumsuz haberin eklenmesi sonucu karışıklık matrisi.

N = 981	Tahmin: 0	Tahmin: 1
Gerçek: 0	81	25
Gerçek: 1	0	875
Olumsuz Tahmin Yüzde	76,41509	
Olumlu Tahmin Yüzde	100	

200 yeni ögenin eklenmesi ile veri setindeki dengesiz dağılımın daha da azalması sonucu başarının arttığı gözlenmiştir. Bu yeni sonuçlarda olumsuz haberlerin başarılı tahmin yüzdesinde % 10,53274'lük artış görülerek % 76 seviyelerine kadar çıkmıştır. Bunun yanında olumlu haberlerin başarılı tahmininde ise seçilen test setinde % 100'lük başarı gözlenmiştir. 50 yeni haberin eklenmesinden sonra oluşan karışıklık matrisi şekil 5.6'da verilmiştir.

Çizelge 5.6. İki yüz elli olumsuz haberin eklenmesi sonucu karışıklık matrisi.

N = 996	Tahmin: 0	Tahmin: 1
Gerçek: 0	104	22
Gerçek: 1	1	869
Olumsuz Tahmin Yüzde	82,53968	
Olumlu Tahmin Yüzde	99,88506	

Eğitime devam edilip 50 yeni olumsuz haberin eklenmesinden sonra olumsuz haber tahmininde başarı oranı %82 seviyelerine ulaşmıştır. Olumlu haberlerde ise başarılı tahmin oranları devam etmektedir.

Bu aşamadan sonra her 250 yeni üretilen olumsuz haberin eklenmesiyle testlere devam edilmiştir ve kayıtlar tutulmuştur. Yeni eklenen olumsuz haber sayısı 2750'e kadar çıkartılıp olumlu ve olumsuz haber sayılarının denkliği sağlanmıştır. Eğitimin her adımında alınan sonuçlarla alakalı karışıklık matrisleri Çizelge 5.7, Çizelge 5.8, Çizelge 5.9, Çizelge 5.10 ve Çizelge 5.11'de verilmiştir.

Çizelge 5.7. Beş yüz olumsuz haberin eklenmesi sonucu karışıklık matrisi.

N = 1077	Tahmin: 0	Tahmin: 1
Gerçek: 0	170	34
Gerçek: 1	2	871
Olumsuz Tahmin Yüzde	83,33333	
Olumlu Tahmin Yüzde	99,7709	

Çizelge 5.8. Yedi yüz elli olumsuz haberin eklenmesi sonucu karışıklık matrisi.

N = 1158	Tahmin: 0	Tahmin: 1
Gerçek: 0	260	33
Gerçek: 1	1	864
Olumsuz Tahmin Yüzde	88,7372	
Olumlu Tahmin Yüzde	99,88439	

Çizelge 5.9. Bin iki yüz elli olumsuz haberin eklenmesi sonucu karışıklık matrisi.

N = 1321	Tahmin: 0	Tahmin: 1
Gerçek: 0	405	32
Gerçek: 1	1	883
Olumsuz Tahmin Yüzde	92,67735	
Olumlu Tahmin Yüzde	99,88688	

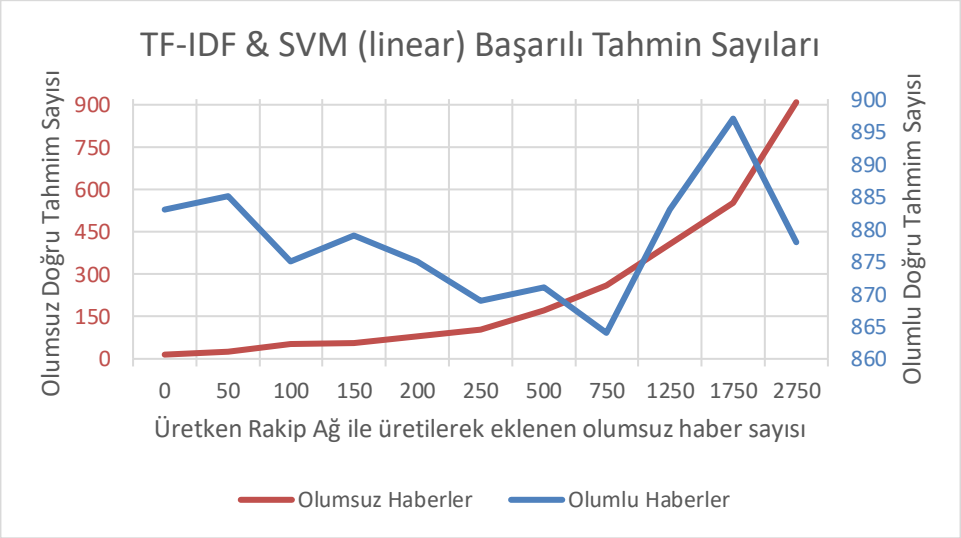
Çizelge 5.10. Bin yedi yüz elli olumsuz haberin eklenmesi sonucunda karışıklık matrisi.

N = 1491	Tahmin: 0	Tahmin: 1
Gerçek: 0	553	40
Gerçek: 1	1	897
Olumsuz Tahmin Yüzde	93,25464	
Olumlu Tahmin Yüzde	99,88864	

Çizelge 5.11. İki bin yedi yüz elli olumsuz haberin eklenmesi sonucu karışıklık matrisi.

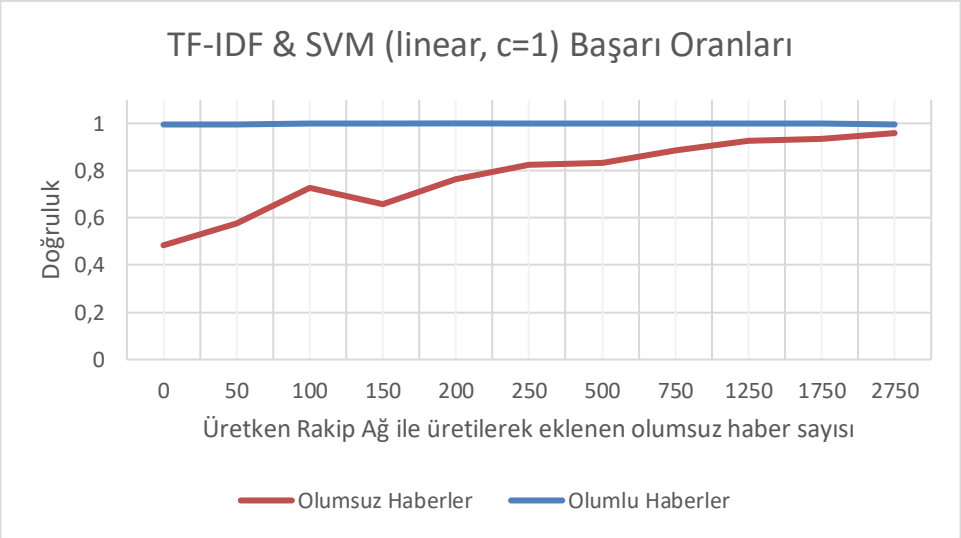
N = 1830	Tahmin: 0	Tahmin: 1
Gerçek: 0	909	39
Gerçek: 1	4	878
Olumsuz Tahmin Yüzde	95,88608	
Olumlu Tahmin Yüzde	99,54649	

Şekil 5.1'de verilen grafikte x ekseni üretken rakip ağ ile üretilen olumsuz haberlerin sayısını gösterirken, y ekseni ise rengine göre doğru sınıflandırılan haber sayılarını gösterir. Grafikten de görülebileceği gibi yeni olumsuz haber eklenmesi sonucunda tahmin başarısı hızlıca artmıştır.



Şekil 5.1. TF-IDF & SVM metodu ile eklenen olumsuz haber sayısına göre doğru sınıflandırılan haberlerin karşılaştırılması.

Başarı yüzdeleri ise Şekil 5.2.'te verilmiştir. Görüldüğü gibi eğitim sonrasında olumsuz haberlerin doğru tahmin etme oranı yaklaşık % 47 artar iken olumlu haberlerin doğru tahmin etme oranında gözle görülür bir azalış olmamıştır. Üretken rakip ağların metin üretimindeki başarısı sonuçta olumlu bir etkiye sahip olmuştur.



Şekil 5.2. TF-IDF & SVM metodu ile eklenen olumsuz haber sayısına göre doğru sınıflandırma oranları.

Aşağıda üretken rakip ağ ile üretilen metinlerden birkaç örnek verilmiştir:

“buna hakkınız da yok yaşlı adam parayı cumhuriyet altın-da bir kapat kardı konusu bir yapılan bu karşılığı konusu bir yönetim ve istanbul bankasından alacakları ile”

“reddeddi istanbul 4 asliye hukuk mahkemesinde bulunan şubesi ve iş bankası da krediyi yakın izlemeye aldı garanti bankasının geri alındı”

“konu devlet meselesi değil memleket merkezi ve markaralara bir gidin başkanı belirtileri dolan yer yaptığını bir türk heyeti alacakları ile”

“büyük darbeyi borsa vurdu rekor üstüne de kuruluyor bankalar bankası bir yıldan uzun yüzde 1,5 artış ile”

Eğitimin ardından en olumlu ve olumsuz kelimeleri belirlemek için puanlama işlemi yapılmıştır. Örnek olarak en olumlu beş kelime ve en olumsuz beş kelime aşağıda verilmiştir.

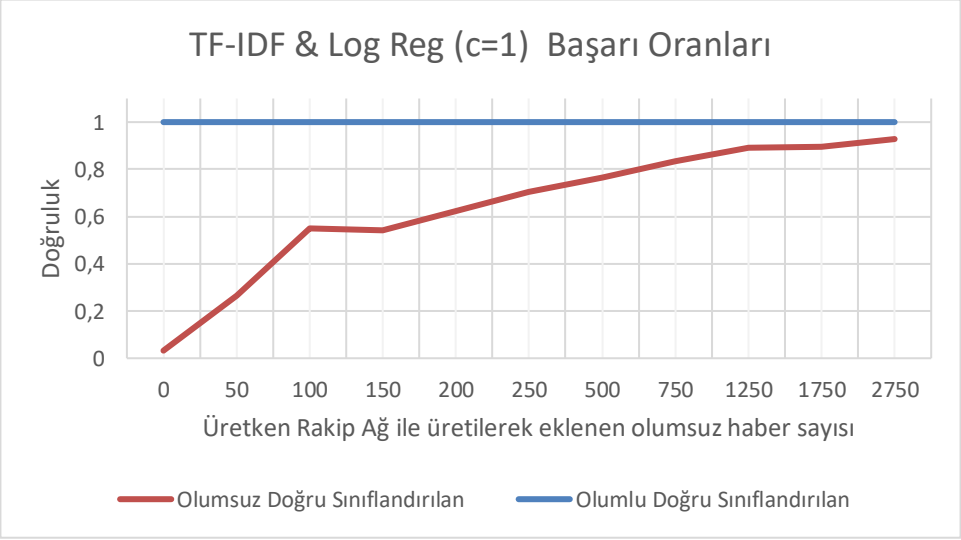
En olumlu beş kelime;

“finansal, türkiye, dijital, yeni, en”

En olumsuz beş kelime;

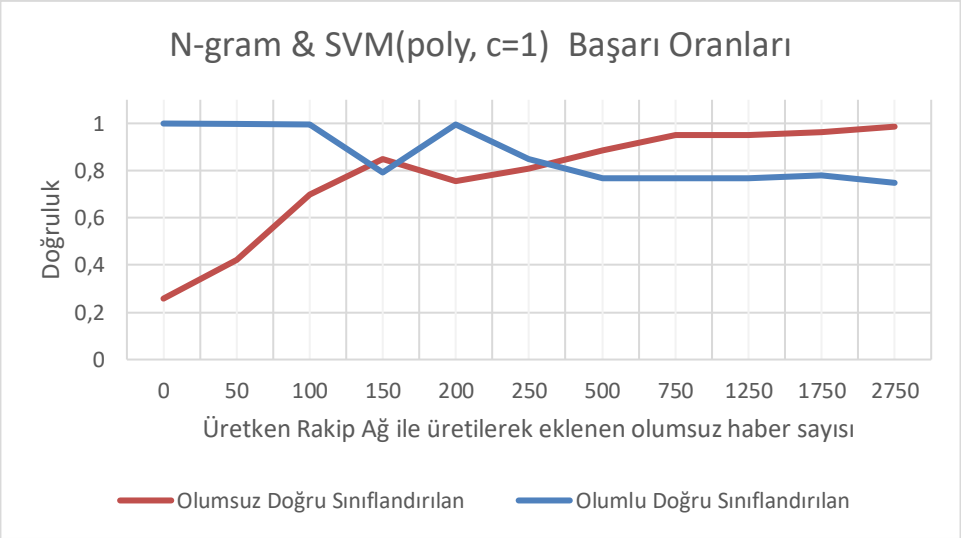
“telekom, karar, yakın, müdürü, bankası”

Burada yapılmış olan testleri ve eğitim işlemlerini n-gram, diğer karar destek makinesi çekirdek fonksiyonları ve lojistik regresyon metotları ile de yapılmıştır. Tüm sonuçlar kaydedilip ilgili tablolar 6’ıncı bölümde verilmiştir. Birkaç metodun başarı oranları aşağıdaki grafiklerde verilip sonuçları yorumlanmıştır.



Şekil 5.3. TF-IDF & Log Reg metodu ile eklenen olumsuz haber sayısına göre doğru sınıflandırma oranları.

TF-IDF & Log Reg metodunun TF-IDF & SVM metodu kadar başarılı sonuçlar elde edilemediği Şekil 5.3’de gösterilmiştir. Yeni üretilen iki bin yedi yüz elli olumsuz haberin eklenmesiyle yapılan test sonucu olumsuz haberlerin tahminindeki başarı oranı % 92,827’de kaldığı gözlenmiştir.



Şekil 5.4. N-gram & SVM metodu ile eklenen olumsuz haber sayısına göre doğru sınıflandırma oranları.

N-gram & SVM (poly) metodunu N-gram & SVM (linear) metodu ile karşılaştıracak olursak olumlu haber tahmin başarısında büyük düşüşler görülmüştür. Test sonucu olumlu haberler için başarılı tahmin oranı % 74,82993'e gerilediği gözlenmiştir. Buna karşılık olumsuz haberler için ise başarı oranı yaklaşık % 98 seviyelerindedir. Başarı oranları ise Şekil 5.4.'te verilmiştir.

BÖLÜM 6

SONUÇLAR

Sonuç olarak bu çalışmada üretken rakip ağlar ile Türkçe metin üretimi süreci yapılmıştır. Üretken rakip ağlar ile oluşturulan modelin Türkçe metinler üretilebileceği görülebilmektedir. Burada üretilen metinler normal dağılım göstermeyen bir veri seti üzerinde uygulanarak sınıflandırma başarısını arttırılmaya çalışılmıştır.

Hiçbir ekleme yapılmadan ki durumdan ve üretken rakip ağlar ile eksik sınıfa ait verilerin üretilip eklenmesiyle dengelenme durumunda sınıflandırma başarısı yaklaşık % 47 oranında arttırılmıştır. Bu da üretilen metinlerin başarılı olduğunu göstermektedir.

Kullanılan yöntemler ve parametrelerin değiştirilmesiyle sonuçlar değişmektedir. Daha da başarılı sonuçlar elde edilebilir. Metinlerdeki anlam bütünlüğünün artması sonuçları olumlu etkilemektedir. Üretken rakip ağlar ile üretilen metinlerin çıktılarının bir normalizasyon işlemi görmesi sonrasında gerçek konuşma diline yakın anlamlı cümleler oluşabildiği görülmüştür.

Üretken rakip ağlar ile eksik sınıfa ait haber üretimi olumlu ve olumsuz haberlerin sayıları eşitlendikten sonra durdurulmuştur. Bu yöntem ile daha fazla yeni üretilen metinlerin eklenmesiyle büyüyen veri seti ile daha başarılı sonuçlar elde edilebilir.

Yapılan çalışmada seçilen metot ve bunun dışındaki 9 farklı metot ile de testler gerçekleştirilmiştir. İlk durum ile veri setinin dengesizliğinin giderilmesiyle elde edilen sonuçların karşılaştırılması Çizelge 6.1, Çizelge 6.2, Çizelge 6.3, Çizelge 6.4, Çizelge 6.5, Çizelge 6.6, Çizelge 6.7, Çizelge 6.8, Çizelge 6.9 ve Çizelge 6.10'da verilmiştir.

Çizelge 6.1. TF-IDF & SVM (rbf, c=1) test sonuçları.

Eklenen	0	2750
Olumsuz Tahmin Yüzde	19,35484	94,51477
Olumlu Tahmin Yüzde	100	99,88662

Çizelge 6.2. N-gram & SVM (rbf, c=1)test sonuçları.

Eklenen	0	2750
Olumsuz Tahmin Yüzde	0	45,04219
Olumlu Tahmin Yüzde	100	99,88662

Çizelge 6.3. TF-IDF & SVM (poly, c=1) test sonuçları.

Eklenen	0	2750
Olumsuz Tahmin Yüzde	25,80645	94,93671
Olumlu Tahmin Yüzde	99,88726	99,88662

Çizelge 6.4. N-gram & SVM (poly, c=1) test sonuçları.

Eklenen	0	2750
Olumsuz Tahmin Yüzde	25,80645	98,62869
Olumlu Tahmin Yüzde	99,77452	74,82993

Çizelge 6.5. TF-IDF & SVM (sigmoid, c=1) test sonuçları.

Eklenen	0	2750
Olumsuz Tahmin Yüzde	41,93548	95,25316
Olumlu Tahmin Yüzde	99,4363	99,20635

Çizelge 6.6. N-gram & SVM (sigmoid, c=1) test sonuçları.

Eklenen	0	2750
Olumsuz Tahmin Yüzde	0	51,37131
Olumlu Tahmin Yüzde	99,66178	46,37188

Çizelge 6.7. TF-IDF & SVM (linear, c=1) test sonuçları.

Eklenen	0	2750
Olumsuz Tahmin Yüzde	48,3871	95,88608
Olumlu Tahmin Yüzde	99,54904	99,54649

Çizelge 6.8. N-gram & SVM (linear, c=1) test sonuçları.

Eklenen	0	2750
Olumsuz Tahmin Yüzde	45,16129	98,31224
Olumlu Tahmin Yüzde	99,4363	96,14512

Çizelge 6.9. TF-IDF & Log Reg (c=1) test sonuçları.

Eklenen	0	2750
Olumsuz Tahmin Yüzde	3,225806	92,827
Olumlu Tahmin Yüzde	100	100

Çizelge 6.10. N-gram & Log Reg (c=1) test sonuçları.

Eklenen	0	2750
Olumsuz Tahmin Yüzde	38,70968	98,31224
Olumlu Tahmin Yüzde	99,54904	97,16553

Verilen çizelgeler incelendiğinde veri setine uygulanacak metot ve parametrelerin değişimine göre olumlu ve olumsuz tahminlerin değişkenlik gösterdiği gözlenmektedir. Test için seçilen TF-IDF & SVM (linear) metodunun diğer metotlar ile karşılaştırıldığında olumlu ve olumsuz haberler beraber düşünüldüğünde başarılı tahminler yaptığı gözlenmiştir. Eğer olumsuz haberlerin tahminine önem verilmek istenilirse N-gram & SVM (poly) seçilebilir. Fakat bu yöntemde diğer metotlara göre olumlu haber tahmininde büyük düşüş görülmüştür. Farklı metotlar kullanılarak yapılan bu testler göz önüne alınır ise üretilen metinlerin genel olarak başarılı olduğu görülmektedir.

KAYNAKLAR

1. İnternet: Wikipedia, "Natural language processing", **https://en.wikipedia.org/wiki/Natural_language_processing** (2020).

2. İnternet: Wikipedia, "Artificial intelligence", **https://en.wikipedia.org/wiki/Artificial_intelligence** (2020).

3. Samuel A., "Some Studies in Machine Learning Using the Game of Checkers", ***IBM Journal of Research and Development,*** (1959).

4. Goodfellow I. J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A. and Bengio Y., "Generative Adversarial Nets", ***Advances in Neural Information Processing System (NIPS)***, 2672-2680 (2014).

5. İnternet: Wikipedia, "Generative Adversarial Network", **https://en.wikipedia.org/wiki/Generative_adversarial_network** (2020).

6. Adler J. and Lunz S., "Banach Wasserstein GAN", ***Advances in Neural Information Processing System (NIPS),*** (2019).

7. Tzeng E., Hoffman J., Saenko K. and Darrell T., "Adversarial Discriminative Domain Adaptation", ***Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)***, 7167-7176 (2017).

8. Metz L., Poole B., Pfau D. and Sohl-Dickstein J., "Unrolled Generative Adversarial Networks", ***International Conference on Learning Representations (ICLR)***, (2017).

9. İnternet: Machine Learning Mastery, "How to Identify and Diagnose GAN Failure Modes", **https://machinelearningmastery.com/practical-guide-to-gan-failure-modes/** (2019).

10. İnternet: Machine Learning Mastery, "18 Impressive Applications of Generative Adversarial Networks (GANs)", **https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/** (2019).

11. Radford A., Metz L. and Chintala S., "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", ***International Conference on Learning Representations (ICLR)***, (2016).

12. Vondrick C., Pirsiavash H. and Torralba A., "Generating Videos with Scene Dynamics", ***Advances in Neural Information Processing System (NIPS)***, Barcelona, (2016).

13. Pathak D., Krahenbühl P., Donahue J., Darrell T. and Efros A. A., "Context Encoders: Feature Learning by Inpainting", ***Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)***, (2016).

14. Li Y., Liu S., Yang J. and Yang M, "Generative Face Completion", ***Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)***, (2017).

15. Taigman Y., Polyak A. and Wolf L, "Unsupervised Cross-Domain Image Generation", ***International Conference on Learning Representations (ICLR)***, (2017).

16. Wu J., Zhang C., Xue T., Freeman W. and Tenenbaum J, "Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling", ***Advances in Neural Information Processing System (NIPS)***, (2016).

17. Gadelha M., Maji S. and Wang R., "3D Shape Induction from 2D Views of Multiple Objects", ***Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)***, (2017).

18. Ledig C., Theis L., Huszar F., Caballero J., Cunningham A., Acosta A., Aitken A., Tejani A. Totz J., Wang Z. and Shi W., "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", ***Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)***, (2017).

19. Bin H., Weihai C., Xingming W. and Chun-Liang L., "High-Quality Face Image SR Using Conditional Generative Adversarial Networks", ***Institution of Engineering and Technology***, (2017).

20. Vasu S., Madam N. and Rajagopalan A., "Analyzing Perception-Distortion Tradeoff using Enhanced Perceptual Super-resolution Network", (2018).

21. Perarnau G., Weijer J., Raducanu B. and Alvarez J., "Invertible Conditional GANs for image editing", ***Advances in Neural Information Processing System (NIPS)***, (2016).

22. Liu M. and Tuzel O., "Coupled Generative Adversarial Networks", ***Advances in Neural Information Processing System (NIPS)***, (2016).

23. Zhang H., Sindagi V. and Patel V., "Image De-raining Using a Conditional Generative Adversarial Network", ***IEEE***, (2019).

24. Isola P., Zhu J., Zhou T. and Efros A., "Image-to-Image Translation with Conditional Adversarial Networks", ***Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)***, (2017).

25. Huang R., Zhang S., Li T. and He R., "Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis", ***Center for Research on Intelligent Perception and Computing (CASIA)***, (2017).

26. Antipov G., Baccouche M. and Dugelay J., "Face aging with conditional generative adversarial networks", ***IEEE International Conference on Image Processing (ICIP)***, (2017).

27. Zhang Z., Song Y. and Qi H., "Age Progression/Regression by Conditional Adversarial Autoencoder", ***Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)***, (2017).

28. Karras T., Aila T., Laine S. and Lehtinen J., "Progressive Growing of GANs for Improved Quality, Stability, and Variation", ***International Conference on Learning Representations (ICLR)***, (2018).

29. Jin Y., Zhang J., Li M., Tian Y., Zhu H. and Fang Z., "Towards the Automatic Anime Characters Creation with Generative Adversarial Networks", ***Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)***, (2017).

30. Wang T., Liu M., Zhu J., Tao A., Kautz J. and Catanzaro B., "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs", ***Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)***, (2017).

31. Brock A., Donahue J. and Simonyan K., "Large Scale GAN Training for High Fidelity Natural Image Synthesis", ***International Conference on Learning Representations (ICLR)***, (2019).

32. Che T., Li Y., Zhang R., Hjelm R., Li W., Song Y. and Bengio Y., "Maximum-Likelihood Augmented Discrete Generative Adversarial Networks", (2017).

33. Guo J., Lu S., Cai H., Zhang W., Yu Y. and Wang J., "Long Text Generation via Adversarial Training with Leaked Information", ***Association for the Advancement of Artificial Intelligence***, (2017).

34. Yu L., Zhang W., Wang J. and Yu Y., "Long Text Generation via Adversarial Training with Leaked Information", ***Association for the Advancement of Artificial Intelligence***, (2017).

35. Lin K., Li D., He X., Zhang Z. and Sun M., "Adversarial Ranking for Language Generation", ***Advances in Neural Information Processing System (NIPS)***, (2017).

36. Fedus W., Goodfellow I. and Dai A., "Maskgan: Better Text Generation Via Filling In The _____", ***International Conference on Learning Representations (ICLR)***, (2018).

37. Cao Y., Zhou Z., Zhang W. and Yu Y., "Unsupervised Diverse Colorization via Generative Adversarial Networks", (2017).

38. Kusner M. and Hernandez-Lobato J., "GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution", (2016).

39. Antreas A., Amos S. and Harrison E., "Data Augmentation Generative Adversarial Networks", ***Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)***, (2018).

40. Georgios D. and Fernando B., "Effective data generation for imbalanced learning using conditional generative adversarial networks", ***Expert Systems with Applications***, 464-471 (2018).

41. İnternet: TensorFlow, "Neden TensorFlow", **https://www.tensorflow.org/about** (2020).

42. İnternet: Wikipedia, "Long short-term memory", **https://en.wikipedia.org/wiki/Long_short-term_memory** (2020).

43. İnternet: Wikipedia, "n-gram", **https://en.wikipedia.org/wiki/N-gram** (2020).

44. İnternet: Medium, "TF-IDF/Term Frequency Technique", **https://medium.com/analytics-vidhya/tf-idf-term-frequency-technique-easiest-explanation-for-text-classification-in-nlp-with-code-8ca3912e58c3** (2020).

45. İnternet: Wikipedia, "Support vector machine", **https://en.wikipedia.org/wiki/Support_vector_machine** (2020).

46. İnternet: GitHub, "Visualization of SVM Kernels Linear, RBF, Poly and Sigmoid on Python", **https://gist.github.com/WittmannF/60680723ed8dd0cb993051a7448f7805** (2016).

47. İnternet: Wikipedia, "Logistic regression", **https://en.wikipedia.org/wiki/Logistic_regression** (2020).

48. İnternet: GitHub, "Zemberek-NLP", **https://github.com/ahmetaa/zemberek-nlp** (2020).

49. İnternet: Wikipedia, "One-Hot", **https://en.wikipedia.org/wiki/One-hot** (2020).

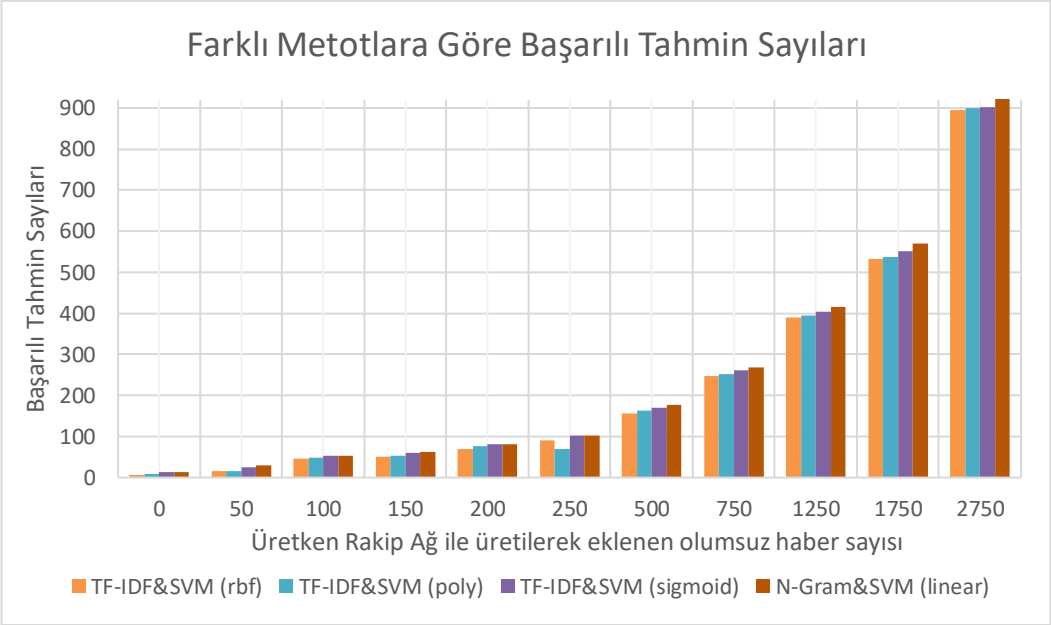
50. İnternet: Wikipedia, “Word2Vec”, **https://en.wikipedia.org/wiki/Word2vec** (2020).

51. İnternet: Medium, “Kelime Vektörleri”, **https://ugurozker.medium.com/kelime-vekt%C3%B6rleri-e6bfd75cfb92** (2020).

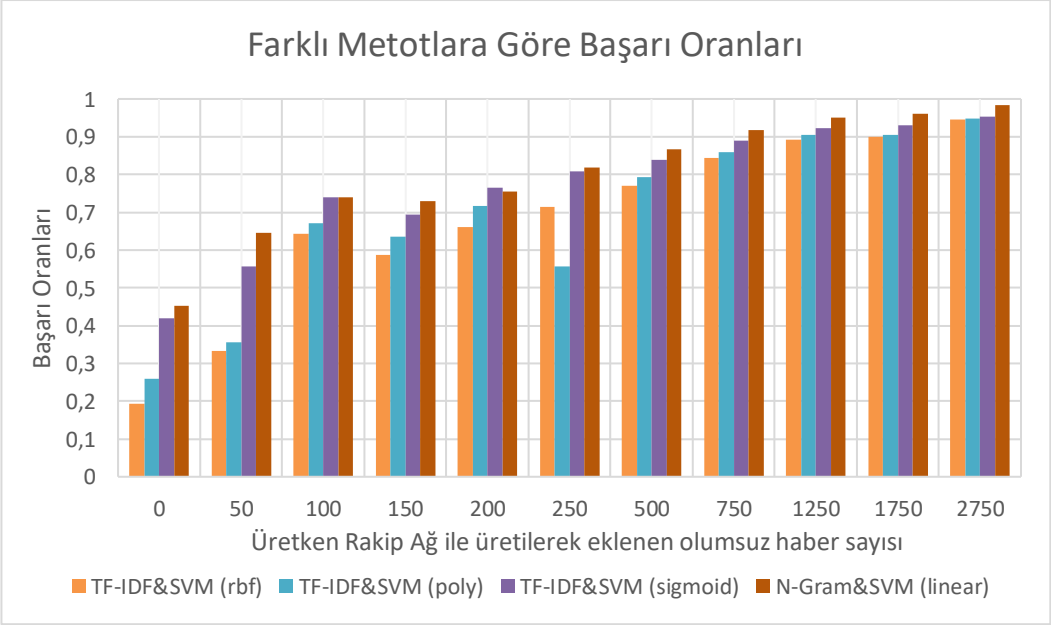
52. İnternet: TowardsDataScience, “You should try the new TensorFlow’s TextVectorization layer.”, **https://towardsdatascience.com/you-should-try-the-new-tensorflows-textvectorization-layer-a80b3c6b00ee** (2020).

EK AÇIKLAMALAR A.

DİĞER TEST SONUÇLARI



Şekil Ek A.1. Farklı metotlara göre olumsuz haberlerin başarılı tahmin sayılarının karşılaştırılması.



Şekil Ek A.2. Farklı metotlara göre olumsuz haberlerin başarılı tahmin oranlarının karşılaştırılması.

ÖZGEÇMİŞ

Barış GÜCÜK 1992 yılında Karabük'te doğdu. İlk ve orta öğrenimini aynı şehirde tamamladı. Kıymet ve Mustafa Yazıcı Anadolu Lisesi'nden 2010 yılında mezun oldu. 2010 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü'nde öğrenime başlayıp 2017 yılında iyi derece ile mezun oldu. 2018 yılında Karabük Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda yüksek lisans eğitimine başladı. 2020 yılında Karabük Üniversitesi Teknoloji Geliştirme Bölgesinde bir dönem çalıştı.

ADRES BİLGİLERİ

Adres : Karabük Üniversitesi
Demir-Çelik Kampüsü
Mühendislik Fakültesi
Balıklarkayası Mevkii / KARABÜK
Tel : (544) 574 8789
E-posta : barisgucuk@gmail.com