**T.C.**
**ONDOKUZ MAYIS ÜNİVERSİTESİ**
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**
**BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI**

# TÜRKÇE İSTENMEYEN E-POSTALARIN DERİN ÖĞRENME İLE TESPİT EDİLMESİ

Yüksek Lisans Tezi

**Ersin Enes ERYILMAZ**

Danışman

**Prof. Dr. Erdal KILIÇ**

SAMSUN
2021

## TEZ KABUL VE ONAYI

**Ersin Enes ERYILMAZ** tarafından, **Prof. Dr. Erdal KILIÇ** danışmanlığında hazırlanan **"Türkçe İstenmeyen E-postaların Derin Öğrenme ile Tespit Edilmesi"** başlıklı bu çalışma, jürimiz tarafından 27.1.2021 tarihinde yapılan sınav sonucunda oy birliği ile başarılı bulunarak Yüksek Lisans Tezi olarak kabul edilmiştir.

| | **Unvanı Adı Soyadı**<br>**Üniversitesi**<br>**Ana Bilim/Ana Sanat Dalı** | **İmza** | **Sonuç** |
|---|---|---|---|
| **Başkan** | Prof. Dr. Mustafa ULUTAŞ<br>Karadeniz Teknik Üniversitesi<br>Bilgisayar Mühendisliği Anabilim Dalı | | ☒ Kabul<br>☐ Ret |
| **Üye**<br>(Danışman) | Prof. Dr. Erdal KILIÇ<br>Ondokuz Mayıs Üniversitesi<br>Bilgisayar Mühendisliği Anabilim Dalı | | ☒ Kabul<br>☐ Ret |
| **Üye** | Doç. Dr. Sedat AKLEYLEK<br>Ondokuz Mayıs Üniversitesi<br>Bilgisayar Mühendisliği Anabilim Dalı | | ☒ Kabul<br>☐ Ret |

Bu tez, Enstitü Yönetim Kurulunca belirlenen ve yukarıda adları yazılı jüri üyeleri tarafından uygun görülmüştür.

ONAY
... / ... / ...
Prof. Dr. Ali BOLAT
Enstitü Müdürü

## BİLİMSEL ETİĞE UYGUNLUK BEYANI

Hazırladığım yüksek lisans/doktora/sanatta yeterlik tezinin bütün aşamalarında bilimsel etiğe ve akademik kurallara riayet ettiğimi, çalışmada doğrudan veya dolaylı olarak kullandığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin Kaynaklar'da gösterilenlerden oluştuğunu, her unsurun enstitü yazım kılavuzuna uygun yazıldığını ve TÜBİTAK Araştırma ve Yayın Etiği Kurulu Yönetmeliği'nin 3. bölüm 9. maddesinde belirtilen durumlara aykırı davranılmadığını taahhüt ve beyan ederim.

27/01/2021
Ersin Enes ERYILMAZ

## TEZ ÇALIŞMASI ÖZGÜNLÜK RAPORU BEYANI

**Tez Başlığı:** Türkçe İstenmeyen E-postaların Derin Öğrenme ile Tespit Edilmesi

Yukarıda başlığı belirtilen tez çalışması için şahsım tarafından 25/12/2020 tarihinde intihal tespit programından alınmış olan özgünlük raporu sonucunda;

Benzerlik oranı : % 8

Tek kaynak oranı : % 2 çıkmıştır.

27/01/ 2021
Prof. Dr. Erdal KILIÇ

# ÖZET

TÜRKÇE İSTENMEYEN E-POSTALARIN DERİN ÖĞRENME İLE TESPİT EDİLMESİ

Ersin Enes ERYILMAZ
Ondokuz Mayıs Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Ana Bilim Dalı
Yüksek Lisans, Ocak/2021
Danışman: Prof. Dr. Erdal KILIÇ

E-postalar günümüzün en etkili iletişim araçlarından biridir. E-postaların içinde meşru e-postalar bulunduğu gibi istenmeyen e-postalar da bulunmaktadır. Yaramaz, önemsiz, gereksiz e-posta anlamına istenmeyen e-postalar internet kullanıcılarına maddi ve manevi ciddi zararlar vermekte olup internet trafiğini de meşgul etmektedir. İstenmeyen e-postaların tespiti için birçok yöntem bulunmakla birlikte mevcut çözümler çoğunlukla spam göndericilerin yenilikçiliğinin ve geliştirdiği tekniklerin gerisinde kalmaktadır. Bu tez çalışmasında literatürde bulunan istenmeyen e-postaların tespitinde kullanılan yöntemler incelenmiş olup Türkçe istenmeyen e-posta tespiti için 6 farklı model önerilmiştir. 4 farklı derin öğrenme modeli Python programlama dili Keras kütüphanesi kullanılarak Spyder geliştirme ortamı ile geliştirilmiştir. Önerilen derin öğrenme modelleri RNN, LSTM, GRU ve BLSTM modelleridir. 2 farklı derin öğrenme modeli ve hiperparametre ince ayarı ile en iyi hiperparametre seçimi internet tabanlı Google Colaboratory ile geliştirilmiştir. Google Colaboratory ile test edilen derin öğrenme modelleri BERT ve DistilBERT modelleridir. Google Colaboratory ile de Tensorflow tabanlı Keras kütüphanesi kullanılmaktadır. İstenmeyen e-posta tespitinde önerilen modeller geliştirilirken 400 adet istenmeyen, 400 adet meşru olmak üzere toplam 800 adet Türkçe e-posta veri kümesi kullanılmıştır. Bu modellerden 5 katlamalı çapraz doğrulama ile BLSTM 0.0373 ile en az test kaybına sahip olup LSTM ve BLSTM istenmeyen e-posta tespitinde %99.38 başarım oranına ulaşmıştır. İnce ayarlı BERT modeli ise %98.75 başarım oranına ulaşmıştır. RNN derin öğrenme modeli için hiperparametre ince ayarı Izgara Arama tahmin edici ile yapılmıştır. Hiperparametre ince ayarı yapılarak %97.66 başarım elde edilmiştir. Ayrıca tez çalışması kapsamında 350 adet e-posta içeren yeni bir Türkçe e-posta veri kümesi oluşturulmuştur. Daha sonraki çalışmalarda bu e-posta veri kümesinin boyutu artırılarak derin öğrenme modellerinde deneyler yapılması düşünülmektedir.

**Anahtar Sözcükler:** derin öğrenme, makine öğrenmesi, RNN, LSTM, GRU, BLSTM, BERT, DistilBERT, Keras, Google Colaboratory, istenmeyen e-posta tespiti, hiperparametre ince ayar

## ABSTRACT

DETECTION OF TURKISH SPAM EMAIL BY DEEP LEARNING
Ersin Enes ERYILMAZ
Ondokuz Mayıs University
Institute of Graduate Studies
Department of Computer Engineering
Master, January/2021
Supervisor: Prof. Dr. Erdal KILIÇ

E-mails are one of today's most effective communication tools. E-mails contain legitimate e-mails as well as spam e-mails. Spam e-mails, which mean naughty, junk, unnecessary e-mails, cause serious material and moral damage to internet users and also occupy internet traffic. Although there are many methods of detecting spam e-mails, current solutions often fall behind the innovation and techniques developed by spammers. In this thesis, the methods used in the detection of unsolicited e-mails in the literature were examined and 6 different models were proposed for the detection of spam e-mails in Turkish. 4 different deep learning models were developed with the Spyder development environment using the Python programming language Keras library. Recommended deep learning models are RNN, LSTM, GRU and BLSTM models. With 2 different deep learning models and hyperparameter fine-tuning, the best hyperparameter selection has been developed with the internet-based Google Colaboratory. Deep learning models tested with Google Colaboratory are BERT and DistilBERT models. Tensorflow-based Keras library is also used with Google Colaboratory. While developing the suggested models for spam detection, a total of 800 Turkish e-mail data sets, 400 of which are spam and 400 are legitimate, were used. Among these models, 5-fold cross validation has the least test loss with BLSTM 0.0373, and LSTM and BLSTM have achieved 99.38% success rate in spam detection. The fine tuned BERT model has achieved 98.75% performance rate. Hyperparameter fine-tuning for the RNN deep learning model was done with the Grid Search estimator. A performance of 97.66% was achieved by fine tuning the hyperparameter. Also, a new Turkish e-mail data set containing 350 e-mails was created within the scope of the thesis study. In future studies, it is planned to increase the size of this e-mail data set and experiment with deep learning models.



**Keywords:** deep learning, machine learning, RNN, LSTM, GRU, BLSTM, BERT, DistilBERT, Keras, Google Colaboratory, spam detection, hyperparameter fine tuning

# ÖNSÖZ VE TEŞEKKÜR

Akademik eğitim sürecimde hep yanımda olan, çalışmalarım boyunca yardım ve desteğini benden esirgemeyen yüksek lisans tez danışmanım değerli hocam Sayın Prof. Dr. Erdal Kılıç'a, yönlendirme ve yardımları için değerli Araştırma Görevlisi Durmuş Özkan Şahin'e teşekkürlerimi sunarım.

Tez çalışmalarım boyunca desteğini esirgemeyen sevgili annem, babam ve kardeşlerime,

Hayatımın her anında olduğu gibi tez yazım süresince desteğini esirgemeyen kıymetli eşim Öznur Eryılmaz'a, tez çalışmalarıma ağırlık verip kendileri ile ilgilenmem gereken zamanları azaltmak zorunda kaldığım oğlum Ömer Alp ve kızım Eylül'e çok teşekkür ederim.

Ersin Enes ERYILMAZ

# İÇİNDEKİLER

## SİMGELER VE KISALTMALAR

### SİMGELER

| | |
|---|---|
| $\theta$ | Aktivasyon fonksiyonu |
| $h_t$ | Hidden state - Saklı durum vektörü |
| $x_t$ | Girdi vektörü |
| **U** | Girdi ağırlığı |
| **W** | Ağırlık matrisi |
| $y_t$ | Çıktı vektörü |
| $i_t$ | Giriş / güncelleme kapısının aktivasyon vektörü |
| $o_t$ | Çıkış kapısının aktivasyon vektörü |
| $f_t$ | Unut kapısı vektörü |
| $\sigma$ | Sigmoid aktivasyon fonksiyonu |
| $r_t$ | Sıfırlama kapısı vektörü |
| $z_t$ | Güncelleme kapısı vektörü |
| $\hat{h}_t$ | Aday aktivasyon vektörü |
| $\hat{c}_t$ | Hücre girişi aktivasyon vektörü |
| $c_t$ | Hücre durum vektörü |

### KISALTMALAR

| | |
|---|---|
| **AIS** | Yapay Bağışıklık Sistemi |
| **ANN** | Artificial Neural Network (Yapay Sinir Ağları: YSA) |
| **ARPANET** | Gelişmiş Araştırma Projeleri Ajans Ağı |
| **ASCII** | American Standard Code for Information Interchange |
| **BERT** | Bidirectional Encoder Representations from Transformers (Transformatörlerden Çift Yönlü Kodlayıcı Temsili) |
| **BOW** | Bag of Words (Kelime Çantası) |
| **BCC** | Blind Carbon Copy (Gizli bilgi kopyası) |
| **BLSTM** | Çift Yönlü Uzun Kısa Süreli Bellek |
| **BTK** | Bilgi Teknolojileri Kurumu |
| **CAPTCHA** | Completely Automated Public Turing test to tell Computers and Humans Apart |
| **CNN** | Convolution Neural Network (Evrişimsel Sinir Ağı) |
| **CC** | Carbon Copy (Bilgi kopyası) |
| **CPU** | Central Processing Unit (Merkezi İşlemci Birimi) |
| **DCC** | Dağıtılmış Checksum Clearinghouse |

| | |
|---|---|
| **DistilBERT** | Distelled BERT (Damıtılmış-saf BERT) |
| **DKIM** | Alan Adı Anahtarlarıyla Tanımlanmış E-posta |
| **DMARC** | Alan Adı Esaslı İleti Kimlik Doğrulaması, Raporlama ve Uyumluluk |
| **DNS** | Domain Name System (Alan Adı Sistemi) |
| **DVM** | Destek Vektör Makinesi (Support Vector Machine:SVM) |
| **Email** | Elektronik mail |
| **E-posta** | Elektronik posta |
| **GA** | Genetik Algoritma |
| **GB** | Gigabyte |
| **GPU** | Graphic Processing Unit (Grafik İşleme Birimi) |
| **GRU** | Gated Recurrent Unit (Geçitli Tekrarlayan Birim) |
| **GridSearchCv** | Izgara Arama |
| **HTML** | Hyper Text Markup Language |
| **IMAP** | Internet Message Access Protocol |
| **IP** | İnternet Protokolü |
| **ISS** | İnternet Bilgi Servisi |
| **KB** | Kilobyte |
| **KNN** | K-Nearest Neighbor (k-En Yakın Komşu) |
| **LR** | Lojistik Regresyon |
| **LSTM** | Long-short Term Memory (Uzun-kısa Süreli Bellek) |
| **MAE** | Mean Absolute Error |
| **MAPE** | Mean Absolute Percentage Error |
| **ML** | Machine Learning (Makine Öğrenmesi) |
| **MLP** | Multi Layer Perceptron (Çok Katmanlı Algılayıcı: ÇKA) |
| **MNIST** | Modified National Institute of Standards and Technology |
| **NB** | Naive Bayes |
| **NLP** | Natural Language Processing (Doğal Dil İşleme) |
| **NSA** | Negatif Seçim Algoritması |
| **ONEIROS** | Açık Uçlu Nöro-Elektronik Akıllı Robot İşletim Sistemi |
| **PCA** | Temel Bileşen Analizi |
| **POP3** | Post Office Protokol 3 (Postane protokolü 3) |
| **PSO** | Parçacık Sürüsü Optimizasyonu |
| **RF** | Random Forest (Rastgele Orman) |
| **RNN** | Recurrent Neural Network (Tekrarlayan Sinir Ağları) |

| | |
|---|---|
| **RoBERTa** | A Robustly Optimized BERT Pretraining Approach (Sağlam Bir Şekilde Optimize Edilmiş BERT Ön Eğitim Yaklaşımı) |
| **RS** | Rough Sets (Kaba Kümeler) |
| **SOM** | Self-Organizing Maps (Özdüzenleyici Haritalar) |
| **SMO** | Sequential Minimal Optimization (Sıralı Minimum Optimizasyon) |
| **SMS** | Kısa Mesaj Hizmeti |
| **SMTP** | Simple Mail Transfer Protocol (Basit posta iletim protokolü) |
| **Spam** | İstenmeyen elektronik posta |
| **SPF** | Gönderen politikası çerçevesi |
| **URL** | Tekdüzen Kaynak Bulucu |
| **WSD** | Kelime Anlamında Belirsizlik Giderme |
| **WWW** | World Wide Web (Dünya çapında ağ) |

# ŞEKİLLER DİZİNİ

# TABLOLAR DİZİNİ

# 1. GİRİŞ

Elektronik postalar kullanımının kolay, maliyetlerinin ucuz olmasından dolayı, propaganda, reklam, oltalama yapmak isteyen kişi veya topluluklar tarafından hedefe konulmuştur. Amaçlarının gerçekleştirmek isteyen kişi veya topluluklar hiç tanımadıkları e-posta hesaplarına gereksiz ve istenmeyen e-posta göndermektedir. İstenmeyen e-posta yaramaz, önemsiz, gereksiz e-posta anlamına gelmektedir. Gereksiz e-postalar internet kullanıcılarına maddi ve manevi ciddi zararlar vermekle birlikte internet trafiğini de meşgul etmektedir.

Akıllı telefonların piyasaya girmesiyle ortaya çıkan mesajlaşma uygulamaları her ne kadar fazla olsa da e-postaların kullanımına uzunca bir süre daha devam edileceği aşikârdır. E-postaların kolay, ucuz, internete bağlı her cihazdan erişilebilir olması reklam amaçlı veya kötü niyetli internet kullanıcıları için bir kaynak olmaya devam edeceğini göstermektedir. Günümüzün en etkili iletişim araçlarından olan e-postaların arasında hala bu nedenle istenmeyen e-postalar da olabilmektedir. Bu istenmeyen e-postaların ayrıştırılması için birçok yöntem bulunmakla birlikte mevcut çözümler çoğunlukla spam göndericilerin sürekli olarak getirdiği yenilikçiliğin ve geliştirdiği tekniklerin gerisinde kalmaktadır. Bu nedenle istenmeyen e-posta tespiti ve filtrelenmesi güncel bir problem olarak kalmakta dolayısıyla istenmeyen e-posta tespiti için sürekli yeni yöntemler geliştirilmektedir. Bu tez çalışmasında istenmeyen e-posta tespiti için literatürde bulunan yöntemler yapay zekâ tabanlı olmayan istenmeyen e-posta tespit yöntemleri ve yapay zekâ tabanlı olan istenmeyen e-posta tespit yöntemleri şeklinde iki ana grupta incelenmiştir. Ayrıca istenmeyen e-posta tespiti için kullanılan yapay zekâ, makine öğrenmesi ve derin öğrenme kavramları da açıklanmıştır.

İstenmeyen e-postaları tespit etme çalışmaları çoğunlukla İngilizce veri kümeleri üzerinde yapılmıştır. Bu tez çalışmasında ise yapay zekâ tabanlı olan derin öğrenme yöntemleri kullanılarak Türkçe istenmeyen e-postaların filtrelenmesi hedeflenmektedir.

Tez beş bölümden oluşmaktadır. Tezin girişi olan birinci bölümde tezin konusu, gerekçesi, amacı, araştırma probleminin niteliği, kuramsal çerçevesi, yöntemi sunulmuş, temel kaynaklara ilişkin genel bir değerlendirme yapılmıştır. İkinci bölümde kuramsal temeller üzerinde durulmuş olup literatürde bulunan kaynaklar

özetlenmiştir. Üçüncü bölümde kullanılacak materyal ve yöntem tanımları yapılmıştır. Dördüncü bölümde bulgular ve tez çalışması ile elde edilen bulguların literatürdeki çalışmalar ile karşılaştırılması tartışılmıştır. Beşinci bölümde sonuç ve önerilere yer verilmiştir.

### 1.1. Tezin Amacı

Bu çalışmanın amacı istenmeyen e-postaları tespit edebilmek için literatürde bulunan araştırmaları irdelemek ve insan beyninin davranışlarından ilham alan makine öğrenmesi yöntemlerinden derin öğrenme modelleri kullanılarak Türkçe istenmeyen e-postaları başarılı bir şekilde tespit etmektir.

### 1.2. Tezin Kapsamı

Bu tez kapsamında, istenmeyen e-postaları tespit etmek için kullanılan yöntemler incelenmiştir. İstenmeyen e-postayı tespit edebilmek için genel olarak yapay zekâ tabanlı ve yapay zekâ tabanlı olmayan sistemler ana başlıkları içinde irdelenmiştir. Yapay zekâ, makine öğrenmesi, derin öğrenme kavramları ve bunların istenmeyen e-posta tespitinde kullanım yöntemleri, Python, Keras ve Google Colaboratory araçları ile gerçeklenmiştir. Literatürde bulunan çalışmalar ve araştırma sonuçları hakkında bilgiler verilmiştir. Ayrıca Derin öğrenme kullanarak istenmeyen e-postaları tespit etmek için modeller geliştirilmiş, tespit sonuçları açıklanmış ve sonraki çalışmalarla ilgili hedeflenenlere yer verilmiştir. Çalışma ile Türkçe istenmeyen e-posta tespiti yapılmış aynı zamanda yeni bir Türkçe e-posta veri kümesi de oluşturulmuştur.

# 2. KURAMSAL TEMELLER VE KAYNAK ÖZETLERİ

## 2.1. Kuramsal Temeller

Bu kısımda e-postanın yapısı ve tarihçesi ile istenmeyen e-posta tanımı yapılmıştır.

### 2.1.1. Elektronik Posta

Günümüzde internete erişim sağlayan hemen hemen herkes e-posta ile bilgi alışverişinde bulunmaktadır. Herhangi bir kullanıcı kaydı için dahi bir e-posta adresine ihtiyaç bulunmaktadır. E-postalar ile kişilerin gerçek kişi olup olmadığı belirlenebilmekte, saniyeler içerisinde dünyanın herhangi bir yerinde bulunan bir veya birden çok alıcıya yazı, resim, video gibi birçok içerik gönderilebilmektedir.

İlk e-posta örneği Massachusetts Institute of Technology bilgisayarlarında MAILBOX adı ile anılan ve 1965 yılına uzanan programlardır. Bu bilgisayarların kullanıcıları bu programla mesajlarını bilgisayarda bir sonraki oturum açıldığında mesajları görebilecek diğer kullanıcılar için üniversitedeki bilgisayarda bırakabilir. Yalnızca birbiriyle iletişim kurmak isteyen insanlar aynı bilgisayarı düzenli olarak kullanınca sistem oldukça etkiliydi. 1969 yılında ABD Savunma Bakanlığı kurum içi iletişim için çok sayıda bilgisayarı birbirine bağlayan bir ağ olan ARPANET'i (Gelişmiş Araştırma Projeleri Ajans Ağı) uygulamaya başladı ve ilk mesaj 29 Ekim 1969 yılında bilgisayardan bilgisayara gönderildi. 1972 yılında ARPANET'in ağa bağlı e-posta sistemini oluşturarak e-posta icat edildi. Kimsenin adında olmayan @ sembolü kullanıldı. kullanıcıadı@bilgisayar adı şeklinde e-postalar tanımlandı. 1974 yılında yüzlerce askeri e-posta kullanıcısı vardı. ARPANET e-postayı teşvik etti. Larry Roberts patronu için postalarını sıralayabilmek adına bazı e-posta klasörleri icat etti. John Vital e-postayı sıralamak için bir yazılım geliştirdi. Birkaç yıl içinde ARPANET trafiğinin %75'i e-posta ile gönderildi. E-posta bizi ARPANET'ten internete götürmüştür (Peter, 2004).

İlk önemli e-posta standardı SMTP veya basit mesaj aktarma protokolü olarak adlandırıldı. SMTP çok basit ve hala kullanılmakta olan oldukça naif bir protokoldür ve mesaj göndermeyi iddia eden kişinin iddia ettiği kişi olup olmadığını anlamaya çalışmaz. Böylelikle sahtecilik yaygınlaşmıştır. Protokoldeki bu basitlik virüs, solucan ve güvenlik sahtekârları spammer (spam yayınlayıcı) tarafından sömürülmektedir. Daha sonraları POP3 protokolü ile sunucular standart olarak görünmeye başlamıştır

İlk önemli e-posta standardı SMTP veya basit mesaj aktarma protokolü olarak adlandırıldı. SMTP çok basit ve hala kullanılmakta olan oldukça naif bir protokoldür ve mesaj göndermeyi iddia eden kişinin iddia ettiği kişi olup olmadığını anlamaya çalışmaz. Böylelikle sahtecilik yaygınlaşmıştır. Protokoldeki bu basitlik virüs, solucan ve güvenlik sahtekârları spammer (spam yayınlayıcı) tarafından sömürülmektedir. Daha sonraları POP3 protokolü ile sunucular standart olarak görünmeye başlamıştır (Peter, 2004).

IMAP ile de bilgisayara ya da diğer cihazlara gelen e postalarınızın indirilmesine yarayan bir iletişim kaynağıdır. Kullanılan bu kaynak ya da protokol sayesinde internete bağlı olunmasa bile daha önce indirilmiş olan birçok farklı postaya bakılabilmektedir.

Günümüzde www ile Yahoo, Outlook, Gmail, Yandesk ile oldukça kullanışlı e-posta ara yüzleri geliştirilmiş olup ücretsiz olan e-posta adresleri milyonlarca insan tarafından kullanılmaktadır.

### 2.1.2. İstenmeyen Elektronik Posta

İstenmeyen e-posta, spam e-posta, yaramaz, önemsiz, gereksiz e-posta anlamına gelir. Genellikle, talep etmeyen çok sayıda alıcıya bir reklam veya alakasız içeriği olan bir mesajın gönderildiği anlamına gelir. Spam aynı zamanda özellikle kamuya açık veya özel e-posta adreslerine mesaj gönderme izniyle ilgili olarak tüm dünya mahkemelerinde tartışmalı bir konu olarak da bilinmektedir.

Birçok durumda içerik probleminden ziyade bir rıza meselesi haline gelmiştir. E-posta iletilerinin yanı sıra, spam gönderenler, anında mesajlaşma (spim), webloglar, SMS veya sahte arama motoru optimizasyon hizmetleri (spamdexing) kullanarak yeni saldırı yöntemleri geliştirmektedir. Spam göndericilerin amacı, bilgisayar kullanıcılarını yasal veya yasaklayıcı nitelikte ürünler ve hizmetler almaya teşvik etmektir. Geçmişte bir haber grubunu ya da e-posta listesini alakasız ya da uygunsuz mesajlarla doldurmak olan istenmeyen e-posta, günümüzde ticari amaçlı ve para odaklı hale gelmiştir (avira.com, 2019).

İstenmeyen e-postalar çok farklı şekilde karşımıza çıkabilir. Spam e-posta tanınmaması için kelimeler arasına özel karakterler, HTML etiketi, veya ASCII kodları uygulanabilir.

- b-e-d-a-v-a → özel karakter

- beda<!---->va →HTML etiketleri
- <a href= “ 8ud3eu339ej?,2%12”>bedava</a>→karakter varlık kodlaması
- b#0101edava →ASCII kodlarla

Özenle hazırlanmış e-postaların kullanıldığı bir tür istatistiksel saldırı olan Bayes zehirleme atakları ile Bayesian filtresinin kalbine saldırılır ve böylece filtrenin etkinliğini düşürür. Bir e-posta gönderildiğinde, e-postanın teslim edilememesi veya teslimin bir nedenden dolayı ertelenmesi durumunda normal olarak gönderen bilgilendirilir. Posta sunucuları yanlış yönlendirilen bilgilendirme e-postası gönderdiğinde geri saçılım denilen ek spam e-posta oluşabilir. Resim spam, bot spam, sosyal mühendislik denilen oltalama e-postaları bir diğer istenmeyen e-posta şekilleridir (Bhowmick ve Hazarika, 2016).

### 2.2. İstenmeyen E-posta Tespit Yöntemleri ve Kaynak Özetleri

Spam ile mücadelede kullanıcılar bu tür e-posta aldığında cevap vermeyerek veya spam olarak işaretleyebilir. İstenmeyen e-posta ile mücadele için yasal ve kişisel önlemler vardır. Spam ile başa çıkmak için etkili teknik tedbirler bulunmaktadır.

Makine öğrenmesi tekniklerinden önce kullanılan kurallara dayalı spam filtreleme, beyaz listeler (whitelists), kara listeler (blacklists), gri listeler (graylists) CAPTCHA kullanılan meydan okuma yanıtlama sistemleri (Challenge-Response),bal kapları (honey pots), Optik karakter tanıma filtreleri itibar tabanlı filtreler vardır (Bhowmick ve Hazarika, 2016).

Spam tespit sistemleri yapay zekâ tabanlı olmayan ve yapay zekâ tabanlı olan şeklinde iki ana grupta incelenmiştir. Mevcut çözümler çoğunlukla spam göndericilerin sürekli olarak getirdiği yenilikçiliğin ve geliştirdiği tekniklerin arkasında kaldığından dolayı sürekli olarak spam tespiti ile ilgili çalışmalar gelişmektedir (Eryılmaz ve Kılıç, 2020).

#### 2.2.1. Yapay Zekâ Tabanlı Olmayan İstenmeyen E-posta Tespit Sistemleri

Bu sistemlerin çoğu bağımsız yazılım programları veya çevrimiçi tabanlı çözümler gibi farklı platformlarda yaygın bir şekilde kullanılabilen istenmeyen e-posta önleme çerçeveleridir. Bunlar, sunucu yetkilendirme kimlik doğrulama sistemleri, işbirlikçi yöntemler, sezgisel filtreleme teknikleri ve içeriğe dayalı yaklaşımlar olarak sıralanabilir (Karim vd, 2019).

#### 2.2.1.1. Sunucu Yetkilendirme ve Kimlik Doğrulama Sistemleri

SPF, DKIM ve DMARC posta sunucusunun kimliğini doğrulamanın ve ISS'lere, posta hizmetlerine ve gönderenlerin e-posta gönderme konusunda gerçekten yetkili olduğu diğer alıcı posta sunucularına kanıtlamanın yoludur. Uygun şekilde kurulduğunda, her üçü de gönderenin meşru olduğunu, kimliklerinden ödün verilmediğini ve başka biri adına e-posta göndermediklerini kanıtlar. DKIM tüm giden iletilerin üstbilgisine şifreli bir imza ekler. İmzalanmış iletileri alan e-posta sunucuları, ileti üstbilgisinin şifresini çözmek ve gönderildikten sonra iletinin değiştirilmediğini doğrulamak için kullanır. Gönderen politikası çerçevesi anlamına gelen SPF, sunucuya ileti gönderebilen alanları belirtmektedir. SPF temel olarak dolandırıcıların e-postaları başka birinin adına dağıtmasını yetkilendirilmiş bir IP adresinden geldiğini doğrulayıp engeller (Karim vd, 2019). Alan adı esaslı ileti kimlik doğrulaması, raporlama ve uyumluluk anlamına gelen DMARC, alanınızın şüpheli e-postaları nasıl ele alacağını belirler (Hameed vd, 2013).

SPF ve DKIM daha geniş bir şekilde benimsenirken, DMARC potansiyel kimlik avı e-postalarını yakalamak için ciddi bir yol olsa da, yaygın olarak benimsenen bir politika değildir. Kriptografik bir hata veya zayıflık bu yöntemler için sorun teşkil etmektedir. Aynı zamanda şifreleme işlemleri bazen e-posta sunucusunun yavaşlamasına neden olmaktadır.

Sunucu yetkilendirme ve kimlik doğrulama sistemlerinin yanında ortak çalışma yaklaşımına dayalı işbirlikçi modeller bulunmaktadır.

#### 2.2.1.2. İşbirlikçi Modeller

Ortak çalışmaya dayalı spam filtreleme modelleme stratejilerinde bir mesaj başka bir kullanıcı tarafından alınır ve değerlendirilir. İşbirliğine dayalı modeller, bu kararların erken yakalanması, kaydedilmesi ve sorgulanması sürecini sergiler. Kriptografik hash, bulanık hash, DCC, gri liste, DNS kara liste-beyaz liste ve sosyal güven temelli çözümler işbirlikçi modeller arasındadır (Karim vd, 2019).

Ayrıca literatürde Bayes filtreleme, imza ağaçları ve veri sıkıştırma tabanlı benzerlik kombinasyonuna dayanan spam algılama yöntemleri de vardır. Bu yöntemleri kullanarak spam algılama hassasiyetinde %99'a varan bir iyileşme olduğu ileri sürülmektedir (Prilepok vd, 2013).

İmza tabanlı teknikler, bilinen her spam ileti için benzersiz bir özet imza değeri

oluşturur. İmza oluşturma teknikleri, yasal bir e-posta iletisinin, spam iletisiyle aynı karma değere sahip olmasını istatistiksel olarak imkânsız hale getirir. Bu imza filtreler yanlış pozitif değerlerini (FP veya YP) çok düşük seviyeye indirmeyi sağlar (Geerthik ve Anish, 2013). Ayrıca bulanık özet fonksiyonu ile iki e-posta arasındaki benzerliği tespit etmek için de kullanılır. Chen ve arkadaşlarının önerdiği istenmeyen e-postayı ortak hedeflerle aynı spam kampanyasında kümelemek için kullanılan bulanık özetleme işleminde 540 bin istenmeyen e-postadan oluşan üç yıllık bir veri kümesi işlenmiş, spam kampanyalarının ve ilgili botnet'lerin (Chen vd, 2014). tipik davranışları incelenmiştir.

DCC fikri, e-posta alıcılarının aldıkları postaları karşılaştırabilmeleri durumunda, istenmeyen toplu postaları tanıyabilmek için ortaya atılmıştır. DCC bir iletinin spam olup olmadığına karar vermez. Sadece bir iletinin kaç kopyasının alındığını bildirir (Gansterer vd, 2005). Literatürde bu fikri kullanan çeşitli çalışmalarda çeşitli antispam filtreleme sistemleri deneysel olarak geliştirmiştir. Bu çalışmalarda ön işleme tabi tutulan Enron2 veri kümesinden 2400 normal e-posta ve 800 spam e-posta kullanılmıştır. İlgili yöntemlerin yüksek hassasiyet ve doğruluk oranları elde ettiği görülmektedir (Wang vd, 2009).

Gri liste yaklaşımı ise tanınmayan bir göndericiden gelen herhangi bir e-postayı geçici olarak reddeden bir spam önleme yöntemidir. Ancak, spam e-posta yeniden gönderilerek bu yaklaşım etkisiz kılınabilir (Bajaj vd, 2011).

Bir diğer yaklaşım olan DNS (Alan Adı Sunucusu) kara listesi merkezi bir veri tabanında spam oluşturucu olarak tanımlanan posta sunucusu IP'lerinin tutar (Caruana ve Li, 2008; Geerthik ve Anish, 2013; Liu ve Moh, 2016). İstenmeyen e-postayı genellikle alan adları veya web sitelerine göre kara listeler oluşturarak tespit etmektedir (Chiba vd, 2018). Ancak, bu tür yaklaşımda, kara liste veritabanı güncelleme işlemleri yeterince hızlı yapılamadığından, saldırının ardından kötü amaçlı kimlik avı URL'leri erken saptanamamaktadır (Ramachandran vd, 2007).

Beyaz liste, yalnızca onaylanmış yasal yöneticiler tarafından yönetilen posta sunucularının bir listesini tutma veya iyi niyetli kullanıcılardan gelen içeriği kabul etme uygulamasıdır. Farklı organizasyonlar, kullanıcıları daha kolay tanıyabilmek için kendi beyaz listelerine sahiptir (Karim vd, 2019). Ancak zamanla beyaz listeden olan bir sunucu spam gönderici durumuna dönüşebilir. Kara liste ve beyaz liste

yaklaşımlarında temel problem listelerin veri tabanlarında yeterince hızlı güncellenememesidir.

Güvene duyarlı bir işbirliğine dayalı spam azaltma sisteminde e-posta sınıflandırma işlevselliği olmayan düğümlerin, bir ana bilgisayarın spam göndericisi olup olmadığını sorgulamasını sağlar (Lin vd, 2013; Sirivianos vd, 2011). Güvene dayalı yaklaşımlarda kullanılan sunucuların güncellenme ve saldırılara açık hale gelebilmesi nedeniyle zamanla güvenilen alan dışında kalabilmesi mümkündür.

İşbirlikçi modellerde etkili sonuçlar vermesine rağmen kriptografik özet ve bulanık özet fonksiyonlarının zayıf noktaları olabileceği, kara liste - beyaz liste veri tabanları güncelleme sorunu, güven belirleyen sunucuların bir şekilde pasifize olması ihtimali bu modellerin başarımlarına olan güveni sarsabilmektedir. Kural tabanlı sezgisel filtreleme yaklaşımları ile düzenli ifade oluşturulması önerilmektedir.

#### 2.2.1.3. Sezgisel Filtreleme Modelleri

Kural tabanlı olan statik spam e-posta filtrelemede düzenli ifade (regex) tabanlı filtre sistemleri sezgisel filtreleme modelleri olarak bilinirler. Bu yapıdaki kurallar çoğunlukla düzenli ifadeler kullanılarak geliştirilirler. Eşleşen kuralların her biri için puanlar hesaplanır. Hesap sonucunda elde edilen toplam değerin, önceden belirlenmiş bir eşik değerin üstünde olup olmadığı kontrol edilir ve ilgili e-postanın gerçekten spam olup olmadığına karar verilir (Khanna vd, 2012; Revar vd, 2017). Sezgisel sistemler hızlı ve kolaydır, ancak dolandırıcıların kural setini ele geçirmeleri durumunda, filtreleme sisteminden kaçınmak için kolayca mesaj oluşturabilirler.

Anlatılan bu yöntemlerin yanında istenmeyen e-posta yakalamada daha etkili olan içeriğe bağlı yaklaşımlar da kullanılmaktadır.

#### 2.2.1.4. İçeriğe Dayalı Filtreleme Çözümleri

Bu sistemler öncelikle e-postanın gövdesinin veya içeriğinin incelenmesine dayanır. Bunlar; içerik filtreleme sistemleri, bağlama duyarlı öneriler ve bulanık mantık tabanlı sistemlerdir.

Bu sistemlerde, ana bilgisayar mesajında bulunan metinlerindeki kalıpları bulmak için kapsamlı bir analiz yapılır, bunlar önceden tanımlanmış ve onaylanmış spam kalıplarıyla eşleştirilir ve bir puan kaydedilir. Puanlar eşik değeri ile karşılaştırıldıktan sonra spam veya spam değil kararı verilir (Khanna vd, 2012). İçerik

tabanlı filtreleme sistemlerine tipik bir örnek Kural tabanlı uzman sistemlerdir. Bu tür bir sınıflandırma, söz konusu sınıflar statik olduğunda ve bileşenleri özellik bakımından ayırt edilebilirliği sağlayabildiğinde uygulanabilir (Kumar vd, 2012). Son derece etkili olmasına rağmen, sistem içerikte yazanları anlayamamaktadır. Yani gerçek amaçlanan mesaj ve tartışmanın arka planı dikkate alınmayabilir. Örneğin "virüs" kelimesi hakkında tartışma ve eğitsel mesajlar istenmeyen e-posta olarak tanımlanabilir.

İçeriğe dayalı filtreleme yaklaşımında bulunan bağlamsal sorunları ele almak için bağlam duyarlı çalışmalar yapılmıştır. Laorden vd., sözdizimsel ve iletilerdeki terimlerin temel anlamlarını açıklayabilmek için WSD adında bir ön işleme adımı ekleyerek spam filtrelemede anlambilimin kullanımını araştırmıştır (Laorden vd, 2012). Uzun mesajlarda performans iyileştirilmesine ihtiyaç duymaktadır.

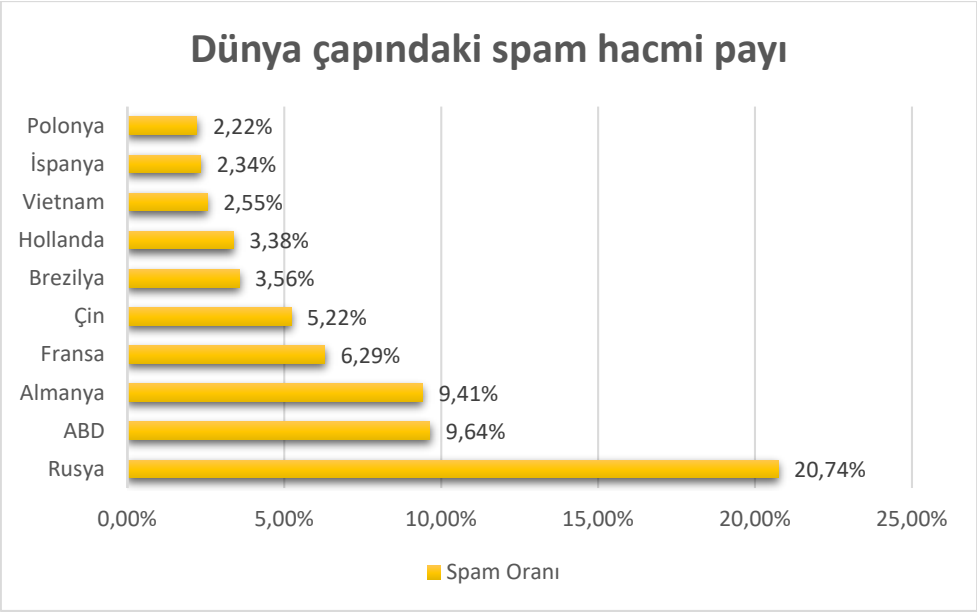
Kimlik avı e-postasıyla başa çıkmak için üç farklı kısımda çalışan algoritmanın ilk kısmında, benzer anlamdaki sözcükler birlikte gruplanır. İkinci kısımda kimlik avı e-postalarını sınıflandırmak için kullanılan kategori veritabanı oluşturulur. En yakın kategoriyi belirlemek için bir bulanık kontrol fonksiyonu kullanılır (Zadeh vd, 1996). Son olarak, öncelikle türetilen mantığa dayalı yeni gelen e-postaların sınıflandırılması konusunda kullanıcılara öneriler eklenir (Che vd, 2017).

İçeriğe dayalı istenmeyen e-posta tespiti yaklaşımlarında, üstbilgi veya alan adı bilgileri yerine e-postanın içeriğine en fazla önem verilir.

Yapay zekâ tabanlı olmayan yukarıdaki anlatılan yöntemler haricinde ülkeye göre filtreleme yapılan ve eşler arası altyapı iletişime dayanan ve istenmeyen e-posta tespit yöntemleri de mevcuttur.

#### 2.2.1.5. Yapay Zekâ Tabanlı Olmayan Diğer Çözümler

Ülke tabanlı filtreleme, eşler arası altyapı diğer spam tespiti yaklaşımlarıdır. Bazı e-posta sunucuları çoğu zaman belirli ülkelerden gelen e-posta akışlarını tamamen engeller, çünkü belirli coğrafi sınırlar genellikle büyük bir spam kaynağıdır (Hu vd, 2010). Dünya genelinde spam hacminin payı ile 2019'un 3. çeyreğinde istenmeyen spam e-postaları için önde gelen kaynak ülkelerin oranı Şekil 3'te verilmektedir. Statista'ya göre 2020 yılında spam oluşturma oranlarında ilk üç sırayı alan ülkeler Rusya, ABD ve Almanya olup sırasıyla %20.74, %9,64 ve %9,41 oranlarında spam oluşturmuşlardır (Statista, 2020).




Şekil 2.1. 2020 ilk çeyreğinde spam için ülkelerin kaynak spam hacmi oranı

Literatürde Bitcoin işlemlerinde kullanılan benzer iş kanıtı konseptine dayanan 'Bitmessaging' olarak bilinen farklı yaklaşımlarda vardır (Bradbury, 2014; Nakamoto, 2019). Bu yöntemler BitMessage eşler arası iletişim protokolüne dayanır ve tamamen merkezi olmayan ve şifreli bir ağ kullanırlar. Bu tür yaklaşımı kullanan yöntemler, mevcut e-posta altyapısıyla henüz tam olarak uyumlu olmadığından, bazı ölçeklenebilirlik sorunları vardır. Ayrıca Dark Mail olarak bilinen temelde farklı bir e-posta sistemi de literatürde mevcuttur (Bradbury, 2014).

Şimdiye kadar açıklanan yapay zekâ tabanlı olmayan istenmeyen elektronik posta tespit sistemleri bağımsız yazılım programları olarak veya çevrimiçi tabanlı çözümler gibi farklı platformlarda kullanılabilen yaygın istenmeyen e-posta önleme çerçeveleridir. Bu çözümlerin yapay zekâ tabanlı sistemlerle birlikte kullanımı ile toplam gönderilen ve alınan istenmeyen e-posta sayısında önemli bir düşüş yaşanmıştır.

Yapay zekâ tabanlı olmayan sistemler bazı sunucularda tek başına çalıştığı, büyük sunucularda ise hem yapay zekâ tabanlı olmayan hem de yapay zekâ tabanlı olan sistemlerin birlikte çalıştığı bilinmektedir. Örneğin en büyük e-posta sunucularına sahip Google ve Microsoft SPF, DKIM, DMARC gibi yapay zekâ tabanlı olmayan sistemlerle makine öğrenmesi yöntemlerini birlikte kullanmaktadır. Fakat küçük çaplı sunucularda beyaz liste, kara liste, gri liste, içerik filtreleme yaklaşımları kullanılmaktadır. Bu sistemlerin birlikte kullanılması yani melez yapıların kullanımı önerilmektedir. Yapay zekâ tabanlı olmayan sistemler tek başına istenmeyen e-

postalara engel olamadığından yapay zekâ tabanlı sistemler üzerinde çalışmalar günümüzde yaygın hale gelmiştir.

### 2.2.2. Yapay Zekâ Tabanlı İstenmeyen E-posta Tespit Sistemleri

İstenmeyen e-postaların tespitinde kullanılan klasik yöntemlerin büyük veriyi işlemedeki başarı oranlarının tıkanması sonucu yapay zekâ tabanlı yeni bir istenmeyen e-posta tespit alanı doğmuştur. Literatürde bu tür sistemler biyolojik esinli zekâ temelli, makine öğrenmesi ve makine öğrenmesinin bir çeşidi olan derin öğrenme tabanlı sistemler olarak karşımıza çıkmaktadır.

#### 2.2.2.1. Biyolojik Esinli Zekâya Dayalı Spam Tespit Sistemleri

Bunlar, doğal davranışlar ve çeşitli doğal canlılarda sıklıkla gözlenen mekanizmalar tarafından motive edilen hesaplama algoritmalarıdır (Darwish, 2018). GA tabanlı sistemler, NSO ve PSO tabanlı sistemler biyolojik esinli zekâya dayalı sistemlerdir.

Spam için genetik algoritma ile düzenli ifade (reggoogle-gooex) filtreleri geliştiren ve spam ve spam olmayanlar arasında ayrım yapan bazı testleri %94'ün üzerinde doğrulukla bulmasına rağmen bu tür yöntemler Fitness fonksiyonu her çalıştırıldığında, her mesajı incelemek zorunda kaldıkları için bu oldukça yavaştır (Greenstadt ve Kaminsky, 2002). Literatürde genetik algoritma kullanan farklı spam tespit algoritmaları vardır. Düzenli ifadelerle GenRegex adlı spam tespit sistemi yeni bir otomatik spam kalıp bulma aracı olan DiscoverRegex ile oluşturulan kalıplar ile YP hatalarından kaçınarak daha iyi performans göstermiştir (Ruano-Ordás vd, 2018). Örneğin NSA ile Gerçek Pozitif ve Gerçek Negatif tespit oranının % 6 oranında artıran, %98.5 doğrulukla spam tespiti yapan bir yöntemler bulunmaktadır (Saleh vd, 2019). Ancak bu tür yöntemler veri sözlüğündeki kelimeler GA ile belirtilmeyen bir performans metriği ile test edilmiş olup yöntemin performansı genellikle standart diğer çalışmalarla karşılaştırılmamıştır (Choudhary ve Dhaka, 2013; Shrivastava ve Bindu, 2013).

Literatürde NSA modeli, farklı melez modellerle yapılan çalışmalarla karşılaştırıldığımızda çok düşük performans gösterdiği görülmektedir (Idris ve Selamat, 2014). Ayrıca NSA ile PSO algoritmasının birleştirildiği farklı tip melez e-posta spam algılama çalışmaları da mevcuttur (Idris vd, 2015).

Bu çalışmalardaki temel problemler kullanılan veri kümelerinin genelde küçük

olmasıdır. Dolayısıyla bu yöntemlerin performanslarını büyük veri temelli derin öğrenme kullanan yöntemler ile karşılaştırmak olası değildir. Ayrıca makine öğrenmesi algoritmalarına kıyasla biyolojik temelli yapay zekâya dayalı algoritmalarında istenmeyen e-posta tespitinde kullanılmasının performans açısından çok fazla kazanç sağlamadığı aksine düşük performans gösterdikleri görülmektedir.

Günümüzde veri boyutunun çok yüksek boyutlara erişmesi nedeniyle bu verileri kullanarak istenmeyen e-posta tespitinde genellikle makine öğrenmesi tabanlı yöntemler ön plana çıkmaktadır. Bu sebeple çalışmanın bundan sonraki kısmında bu tür çalışmalara değinilecektir.

#### 2.2.2.2. Makine Öğrenmesi Tabanlı Spam Tespit Sistemleri

Sabit bir bilgiye dayanan sistemlerin karşılaştığı zorluklar sonucunda, yapay zekâ sistemlerinin ham bir veriden örüntüler çıkarıp kendi bilgilerini elde etme kabiliyeti ile birlikte makine öğrenmesi ortaya çıkmıştır. Makine öğrenmesi, öznel görünen kararlar vermeyi ve gerçek dünyadaki bilgileri içeren problemlerle baş etmeyi sağlamıştır (Bhowmick ve Hazarika, 2016; Goodfellow vd, 2016).

Makine öğreniminin en yaygın şekli denetimli öğrenmedir. Bir e-postanın spam veya spam olup olmadığını sınıflandırabilmek için önce, her biri kategorisiyle etiketlenmiş spam ve spam olmayan geniş bir veri kümesinin toplanması gerekir. Eğitim sırasında, makineye bir e-posta gösterilir ve her kategori için bir tane olmak üzere bir skor vektörü şeklinde bir çıktı üretilir. İstenilen kategorinin tüm kategorilerde en yüksek puana sahip olması istenir, ancak bunun eğitimden önce gerçekleşmesi olası değildir. Burada çıktı puanları ile istenen puan deseni arasındaki hatayı (veya mesafeyi) ölçen nesnel bir fonksiyon kullanılmaktadır. Makine daha sonra bu hatayı azaltmak için dâhili ayarlanabilir parametrelerini değiştirir. Genellikle ağırlık olarak adlandırılan bu ayarlanabilir parametreler, makinenin giriş-çıkış fonksiyonunu tanımlayan gerçek ayar düğmeleri gibidir. Ağırlık vektörünü uygun şekilde ayarlamak için, öğrenme algoritması, her ağırlık için, ağırlık küçük bir miktar arttırılınca hatanın ne kadar artacağını veya azalacağını gösteren bir gradyan vektörü (Bousquet vd, 2007) hesaplar. Ağırlık vektörü daha sonra gradyan vektörüne zıt yönde ayarlanır (LeCun vd, 2015).

Yapay sinir ağları, naïve bayes, karar ağaçları, rastgele orman, lojistik regresyon, destek vektör makinesi, adaboost, k-en yakın komşu istenmeyen e-posta tespitinde en

çok kullanılan denetimli makine öğrenmesi algoritmalarıdır. Spam algılamada denetimsiz ve yarı denetimli öğrenme içeren yöntemler de kullanılmaktadır. Denetimli öğrenmeye kıyasla, denetimsiz öğrenme, etiket kümesini değil, yalnızca özellik kümesini alır. Denetimsiz öğrenme için verilen eğitim kümesi, etiketlenmemiş veri kümesidir. Denetimsiz öğrenme, kümeleme, olasılıksal tahmini, özellikler arasında ilişki bulma ve boyut azaltmayı amaçlamaktadır (Chao, 2011). K-ortalama kümeleme, SOM tabanlı öneriler (Ra vd, 2018), PCA tabanlı çerçeveler, birliktelik tabanlı öneriler denetimsiz öğrenmeye örnek verilebilir. Makine öğrenmesi tekniği kullanan çok sayıda spam tespit yöntemi, bunların kullandığı veri kümeleri ve spam tespit algoritmalarının performansları Tablo 2.1.'de verilmiştir.

Tablo 2.1. Makine öğrenmesi tabanlı spam tespit çalışmaları

| **Çalışma adı** | **Veri kümesi** | **Kullanılan yöntemler / En başarılı yöntemler** | **Kullanılan performans ölçütleri ve başarım oranları (%)** | **Yöntemin Türü (Bilinen/ Yeni/ Melez)** |
|---|---|---|---|---|
| (Zhao ve Zhang, 2005) | 1518 adet e-posta içeren TE 943 | Kaba Küme, Naïve Bayes/ Kaba Küme | Doğruluk: 97.37<br>Kesinlik: 86.58<br>Hassasiyet: 96.99 | Yeni |
| (Altunyaprak, 2006) | 767'si spam olan 2387 adet Türkçe e-posta | Bayes | Kesinlik: 84<br>Hassasiyet: 93.2 | Yeni |
| (Norte Sosa, 2010) | 2200 e-posta | YSA | Doğruluk: 96.1 | Yeni |
| (Yumak, 2011) | 100 adet e-posta | Bulanık Mantık, NB/ NB | Doğruluk: 81.8 | Bilinen |
| (Awad ve ELseuofi, 2011) | SpamAssassin | Bayes, kNN, YSA, DVM, AIS ve RS / NB | Doğruluk: 99.46<br>Kesinlik: 99.66<br>Hassasiyet: 98.46 | Bilinen |
| (Ergin vd, 2012) | 800 adet Türkçe e-posta içeren "TurkishEmail" veri kümesi | Olasılıklı ve İkili (Binary) Bayes/ İkili Bayes | Doğruluk: 93 | Yeni |
| (Idris ve Abdulhamid, 2014) | Spambase | AIS | FP: 1.2 | Bilinen |
| (Bhagyashri vd, 2013) | SpamAssassin | Bayes | Doğruluk: 90<br>Kesinlik: 82.35<br>Hassasiyet: 93.33 | Bilinen |
| (Ateş, 2014) | (Ergin vd, 2012) tarafından oluşturulan 800 Türkçe e-posta veri kümesi ve İngilizce Lingspam_public veri kümesi | DVM, Gauss Karışım Modeli (GKM), NB. / Türkçe veri kümesinde NB<br>/İngilizce veri kümesinde doğrusal DVM | Doğruluk: 99<br>Doğruluk: 98.6 | Bilinen |

Tablo 2.1. (devam)

| | | | | |
|---|---|---|---|---|
| (Sharma vd, 2014) | TREC07 | MLP, NB / MLP | Doğruluk: 93<br>Hassasiyet: 93.2<br>Kesinlik: 93 | Bilinen |
| (Karthika ve Visalakshi, 2015) | Spambase | kNN, NB, DVM ve Hibrid ACO-DVM / ACO-DVM | Doğruluk: 81.25<br>Kesinlik: 87.02<br>Hassasiyet: 75.1 | Yeni ve Melez |
| (Renuka vd, 2015) | Spambase | GA-Naive Bayes, ACO-Naive Bayes / ACO-Naive Bayes | Doğruluk: 84<br>Kesinlik: 89<br>Hassasiyet: 78<br>F-ölçütü: 87 | Yeni ve Melez |
| (Tuteja ve Bogiri, 2016) | 100 spam olmak üzere 200 adet e-posta | K-ortalamalı Geri yayılımlı Sinir Ağı (BPNN) | Kesinlik: 98.42<br>Hassasiyet: 93.5 | Melez |
| (Palanisamy vd, 2016) | Lingspam | Negatif Seçim Algoritması (NSA) kullanan PSO, DVM, NB, DFS-DVM / Negatif Seçim Algoritması (NSA) kullanan PSO | Doğruluk: 93.2 | Melez |
| (Zavvar vd, 2016) | Spambase | PSO, SOM, k-ortalama, DVM / DVM | AUC: 93.07 | Bilinen |
| (Foqaha, 2016) | Spambase | RBF, MLP ve YSA ve Melez HC-RBFPSO / MLP | Doğruluk: 93.28 | Melez |
| (Sharma ve Suryawanshi , 2016) | Spambase | Bayes, KNN, DVM / KNN | Doğruluk: 97.54<br>Kesinlik: 97.72<br>Hassasiyet: 93.52<br>F-ölçütü: 95.6 | Bilinen |
| (Alkaht ve Al-Khatib, 2016) | CSDMC 2010, SpamAssassin, Tarassul | Kendi kendini organize eden Küresel Sıralama Haritası ve İleri besleme algoritmalarının birleşimi ile Several Stage Neural Network (SNN) | Doğruluk: 95.40<br>Kesinlik:99.45<br>Hassasiyet: 91.28<br>F-ölçütü: 95.19 | Yeni |
| (Rajamohana vd, 2017) | (Ott vd, 2011) tarafından oluşturulan veri kümesi | Naive Bayes Uyarlanabilir İkili Çiçek tozlaşma algoritması (ABFPA) | Doğruluk: 91.42 | Yeni |
| (Akinyelu ve Adewumi, 2014) | 2000 kimlik avı ve normal e-posta | Rastgele Orman | Doğruluk: 99.7<br>Kesinlik: 99.47<br>Hassasiyet: 97.5<br>F-ölçütü: 98.45 | Bilinen |
| (Yıldız, 2017) | 310 adet Türkçe e-posta | NB, DVM, YSA, Adaboost, J48, JRIP / Çok Terimli NB | Doğruluk: 96.31<br>Kesinlik: 91<br>Hassasiyet: 100<br>Kappa: 94 | Bilinen |
| (Şahin, 2018) | 55888 e-posta | 12 klasik makine öğrenmesi algoritması / Naive Bayes Kernel ve Doğrusal SVM | Doğruluk: 99.89<br>F-ölçütü: 99.81 | Bilinen |
| (Kale, 2018) | Louis Dorard'ın 2013 yılında kendine ait 4.709 adet e-posta | Karar Ağaçları, Derin öğrenme, Gradient Boosted Tree (GBT), kNN, NB, RF ve LR/ Çok terimli NB | Doğruluk: 95.5<br>Kesinlik: 100<br>Hassasiyet: 91<br>F-ölçütü: 95.8 | Bilinen |
| (Nazlı, 2018) | Enron (300 e-posta) | DVM (Poly) | Doğruluk: 98.33 | Bilinen |

Tablo 2.1. (devam)

| | | | | |
|---|---|---|---|---|
| (Al-Azzawi, 2018) | Spambase | Kaotik ateş böceği algoritmasına dayanan sarmal öznitelik seçimli NB | Doğruluk: 95.14 | Yeni |
| (Karamollaoglu vd, 2018) | 800 adet Türkçe e-posta içeren "TurkishEmail" veri kümesi | NB, Vektör Uzay Modeli/NB | Doğruluk: 95.5 | Bilinen |
| (Salihi, 2019) | 355 spam gönderici olan 1183 Twitter'dan elde edilen veri kümesi | NB, J48, IBK, RF / RF | Doğruluk: 92.95<br>Kesinlik: 92<br>Hassasiyet: 88<br>F-ölçütü: 89 | Yeni |
| (Deniz vd, 2019) | 800 adet Türkçe e-posta içeren "TurkishEmail" veri kümesi | Distributed Bag of Word (DBoW) ve Distributed Memory (DM) öznitelik seçim yöntemleri ile DVM | Doğruluk: 78.75 | Yeni |
| (Ablel-Rheem vd, 2020) | Spambase | NB, Karar Ağaçları ve Kolektif (Ensemble) öğrenme algoritmaları 10 katlı çapraz doğrulama ve Bilgi Kazancı (IG) öznitelik seçimi/ Melez Kolektif | Kesinlik: 94.4<br>Hassasiyet: 94.4<br>F-ölçütü: 94.4 | Yeni |
| (Zamir vd, 2020) | Kullanıcı ve istenmeyen e-posta sözlüğü özelliklerine dayalı özellik merkezli bir spam e-posta algılama modeli | IG, kazanç oranı ve Relief-F gibi özellik seçme teknikleri ile Derin Sinir Ağı | Doğruluk: 97.2 | Yeni |
| (Mohammad, 2020) | Enron-Spam | "Ayarlanabilir Veri Kümesi Bölümleme Kullanan Topluluk Tabanlı Yaşam Boyu Sınıflandırma" Kolektif (Ensemble) modeli | Doğruluk: 95.80<br>Kesinlik: 94.40<br>Hassasiyet: 95.80<br>F-ölçütü: 95.10 | Yeni |
| (Kumar ve Sonowal, 2020) | Kaggle web sitesinden "spam.csv" isimli veri kümesi | 7 farklı makine öğrenmesi algoritması içinden en başarılı NB | Doğruluk: 98 | Bilinen |
| (Eryilmaz vd, 2020) | 800 adet Türkçe e-posta içeren "TurkishEmail" veri kümesi | CHI öznitelik seçme yöntemi ile DVM tabanlı SMO algoritması | F-ölçütü: 98.5 | Yeni |

Tablo 2.1. (devam)

| | | | | |
|---|---|---|---|---|
| (Eryılmaz vd, 2020b) | 800 adet Türkçe e-posta içeren "TurkishEmail" veri kümesi | CHI, IG, ACC, OR, DF öznitelik seçme yöntemi kullanılmış, ayrıca öznitelik seçimi yapılmadan sonuçlar verilmiştir. "TRHamSpamEmailv1.0" adında yeni bir e-posta veri kümesi oluşturulmuştur. | "TurkishEmail" veri kümesi CHI ile SMO: 98.5 F-ölçütü, "TRHamSpamEmailv1.0" veri kümesi CHI ile RF ve NB:74.8 F-ölçütü. Öznitelik seçimi yapılmadan "TurkishEmail" RF: 51.4 F-ölçütü, "TRHamSpamEmailv1.0" RF:53.5 F-ölçütü | Yeni |

Tablo 2.1. incelendiğinde veri kümesinin az olduğu spam tespit çalışmalarında başarım oranlarının yüksek çıktığı, Spambase açık veri kümesi üzerinde makine öğrenmesi tekniklerinin yoğunlaştığı, doğruluk, kesinlik, hassasiyet performans metriklerinin yanında, F-ölçütünün de kullanıldığı, MLP, YSA'ya dayanan spam tespit algoritmalarında model eğitimlerinin zaman aldığı görülmektedir. Bayes, Naive Bayes, DVM, YSA ve Melez yaklaşımların başarım oranlarının ise yüksek olduğu tespit edilmiştir. Açık kaynak kodlu Weka (Hall vd, 2009) kütüphanesinin birçok çalışmada makine öğrenmesi algoritmalarını test ederken kullanıldığı görülmüştür. . K-ortalama kümeleme algoritmasının kullanımının kesinlik oranının artırdığı ifade edilmiştir (Tuteja ve Bogiri, 2016).

Yukarıdaki çalışmalardan daha anlamlı sonuçlar çıkarmak için daha büyük e-posta (istenmeyen e-posta / normal e-posta) içeren veri kümelerine ihtiyaç duyulduğu, farklı algoritmaların kullanımı sonucunda ortaya çıkan başarım oranlarının karşılaştırılması gerektiği görülmektedir. Ayrıca farklı performans metriklerinin algoritma performanslarını değerlendirirken bir arada kullanılmasının gerekli olduğu düşünülmektedir. Sinir ağları temelli istenmeyen e-posta tespit algoritmaların eğitiminde, merkezi işlemcilerin yanında grafik işlemcilerinin de kapasitelerinin artmasıyla, derin öğrenme tekniklerinin kullanımının daha da yaygınlaşacağı öngörülmektedir.

Son yıllarda merkezi işlemci biriminin yanında grafik işlemci birimlerinin hesaplama güçlerinin artmasıyla derin öğrenme tekniklerine özel algoritmaların çalışma zamanları ve performansları iyileşmiştir.

#### 2.2.2.3. Derin Öğrenme Temelli Spam Tespit Sistemleri

Derin öğrenmede bilgisayarların gereksinim duyduğu biçimsel kuralların insan

eliyle girilmesine ihtiyaç kalmaması için kavramlar hiyerarşisi oluşturulur. Kavramlar hiyerarşisi bilgisayarın karmaşık kavramları daha basit kavramlardan öğrenmesine olanak sağlar. Bu kavramların birbiri üzerine nasıl inşa edildiğini gösteren bir çizge olduğunu düşündüğümüzde, bu çizge çok katmanlı olan birçok düğümün birbiriyle ilişkide olduğu derin bir çizge oluşturur. Bu karmaşık işlemlerin yapıldığı yaklaşıma derin öğrenme denmektedir (Goodfellow vd, 2016). Derin öğrenme bir makine öğrenmesi yöntemi olup denetimli veya denetimsiz olabilir. Önceleri tamamen denetimli öğrenmenin başarısının gölgesinde kalan denetimsiz öğrenme, derin öğrenmeye duyulan ilgiyi de canlandırmıştır (Kingma vd, 2014).

Derin öğrenmede öznitelik ve özellik seçimini bir yöntem belirlemeden gizli sinir ağlarında yapılmaktadır. Normal ve istenmeyen etiketli veri kümesinde bulunan veriler, veri vektörlerine dönüştürülerek elde edilen bu veri kümesi eğitim ve test bölümlerine ayrılır. Kullanılacak derin öğrenme katmanları ile model eğitilir. Eğitimden sonra, sistemin performansı test kümesi adı verilen farklı bir dizi örnek üzerinde ölçülür.

Derin öğrenme klasik yapay öğrenme yani makine öğrenmesi algoritmalarının yetersiz kaldığı bazı durumlarda insan performansına yakın çıktıların elde edilmesini sağlayabilmektedir. Geleneksel makine öğrenme teknikleri, doğal verileri ham formlarında işleme yetenekleriyle sınırlıdır. Onlarca yıl boyunca, bir desen tanıma veya makine öğrenme sistemi oluşturmak, ham verileri (örn: bir görüntünün piksel değerleri) uygun bir iç gösterime veya özellik vektörüne dönüştüren bir özellik çıkarıcı tasarlamak için dikkatli bir mühendislik ve önemli bir alan uzmanlığı gerektiriyordu. Derin öğrenmenin en önemli özelliği, özellik katmanlarının insanlar tarafından tasarlanmamasıdır.

Derin öğrenme, çoklu soyutlama seviyelerine sahip verilerin gösterimini öğrenmek için çoklu işleme katmanlarından oluşan hesaplama modellerine izin verir. Tipik bir derin öğrenme sisteminde, makineyi eğitmek için yüz milyonlarca ayarlanabilir ağırlık ve yüz milyonlarca etiketli örnek olabilir. Tekrarlayan sinir ağları (RNN), Uzun-kısa süreli bellek (LSTM) ağları ve Evrişimsel sinir ağları (CNN) derin öğrenmenin en çok kullanılan yöntemlerindendir (Hochreiter ve Schmidhuber, 1997; Mikolov vd, 2013; Russakovsky vd, 2015; Szegedy vd, 2015).

Derin evrişimsel ağlar, görüntü, video, konuşma ve ses işlemede çığır açarken,

tekrarlayan ağlar metin ve konuşma gibi sıralı veriler üzerinde başarılı biçimde kullanılmaktadır (LeCun vd, 2015). Literatürde derin öğrenme ile istenmeyen e-posta tespiti üzerine çalışmalar da görülmektedir. Bu, çalışmalarda kullanılan veri kümeleri, en başarılı yöntemler, kullanılan performans metrikleri, yöntemin türü, tespit algoritmalarının performansları, Tablo 2.2.'de verilmiştir.

Tablo 2.2. Derin öğrenme tabanlı spam tespit çalışmaları

| **Çalışma** | **Veri Kümesi** | **Kullanılan Yöntem /Başarısı En Yüksek Yöntem** | **Kullanılan performans metrikleri ve en yüksek değerleri (%)** | **Yöntemin Türü (Bilinen, Yeni, Melez)** |
|---|---|---|---|---|
| (Tyagi, 2016) | PU1, PU2, PU3, PUA ve Enron-Spam | Yoğun MLP, Stacked Denoising Autoencoder, (SDAE), Derin İnanç Ağı (DBN) / DVM, SDAE | Doğruluk: 96.21<br>Kesinlik: 96.78<br>Hassasiyet: 95.57<br>F-ölçütü: 96.17 | Bilinen |
| (Shang ve Zhang, 2016) | 52934 görüntü içeren yeni bir spam veri kümesi | Tahmin Katmanında DVM kullanan CNN | Doğruluk: 82 | Yeni ve Melez |
| (Roy vd, 2016) | Spambase | Derin DVM, YSA, DVM / Derin DVM | Doğruluk: 92.8<br>Kesinlik: 91.4<br>Hassasiyet: 89.9<br>F-ölçütü: 90.7<br>AUC: 97.3 | Yeni |
| (Kaynar vd, 2016) | 800 adet Türkçe e-posta içeren "TurkishEmail" veri kümesi | Derin öğrenme tekniklerinden oto kodlayıcı ile ince ayar öncesi %98 ince ayar sonrası %97 başarım | Doğruluk: 98 | Yeni |
| (Seth ve Biswas, 2017) | Toplanan 1521'den fazla spam resim ve Enron metin veri kümesi | Görüntü CNN, Metin CNN, Çoklu Öğrenme Modeli / Çoklu Öğrenme Modeli | Doğruluk: 98.11<br>F-ölçütü: 98 | Yeni ve Melez |
| (Yawen vd, 2018) | Spambase | Derin Sinir Ağları (DNN), Naive Bayes / DNN | Doğruluk: 90 | Bilinen |
| (Ra vd, 2018) | IWSPA-AP 2018 | CNN, RNN, LSTM, MLP / Kelime Yerleştirme (Word Embedding) + LSTM | Doğruluk 99.1 | Melez |
| (Bagui vd, 2019) | 3416'sı kimlik avı e-postası olan 18366 etiketli e-posta veri kümesi | Naive Bayes, DVM, Karar Ağacı, LSTM, CNN ve Kelime Yerleştirme / Kelime Yerleştirme | Doğruluk: 98.89 | Bilinen |
| (Yang vd, 2019) | Enron, Personal Image, Spam Archive | LSTM ve CNN | Doğruluk: 98.48<br>Hassasiyet: 98.52<br>Kesinlik: 98.52<br>F-ölçütü: 98.45 | Yeni ve Melez |
| (Jain vd, 2019) | SMS Spam ve Twitter spam veri kümesi | DVM, Naive Bayes, ANN, k-NN, RF, LSTM / LSTM | Doğruluk: 99.01<br>Hassasiyet: 99.35<br>Kesinlik: 98.74<br>F-ölçütü: 99.24 | Yeni |

Tablo 2.2. (devam)

| | | | | |
|---|---|---|---|---|
| (Nagisetty ve Gupta, 2019) | UNSW-NB15 ve NSL-KDD99 | MLP, CNN, DNN, Oto Kodlayıcı / DNN ve MLP | Doğruluk(DNN): 99.24<br>F-ölçütü(MLP): 99.28<br>RMSE(DNN): 0.4 | Bilinen |
| (Roy vd, 2020) | 747 spam ve 4.827 normal SMS veri kümesi | NB, RF, GB, LR, LSTM, SGD / CNN | Doğruluk: 99.44<br>Hassasiyet: 99.8<br>Kesinlik: 99.6<br>F-ölçütü: 99.8<br>AUC: 97.7 | Bilinen |
| (Eryılmaz vd, 2020a) | 800 adet Türkçe e-posta içeren "TurkishEmail" veri kümesi | Derin öğrenme kütüphanesi Keras ile farklı hiperparametre seçimleri ve LSTM modeli | Doğruluk: 100 | Yeni |

Tablo 2.2. dikkatli bir şekilde incelendiğinde, Spam tespiti için derin öğrenme ile açık veri kümelerinde test yapılmasının yan ısıra, araştırmacılarının kendisinin organize edip bir araya getirdiği veya oluşturduğu veri kümeleri de kullanılmıştır.

Önerilen yöntemlerde genel olarak LSTM ve CNN algoritmaları kullanıldığı, performans ölçütü olarak da genellikle doğruluk, hassasiyet, kesinlik ve F-ölçütü metrikleri kullanılmakla birlikte AUC, RMSE metriklerinin de çeşitli çalışmalarda kullanıldığı görülmektedir.

Sadece metin içeren veri kümeleri olduğu gibi spam resim içeren veri kümeleri ile de çalışmalar yapıldığı görülmüştür. Genel olarak metin içeren veri kümelerinde LSTM algoritması ile daha yüksek başarım elde edilirken resim içeren veri kümelerinde CNN algoritmasının başarımı daha yüksektir. Dengesiz veri kümesi kullanılan bir çalışmada CNN, LSTM'den daha iyi başarım sonucu vermektedir (Roy vd, 2020). Ayrıca melez spam tespit yöntem yaklaşımlarının ve spam algılama sistemlerinin başarımı artırdığı görülmektedir (Ra vd, 2018; Yang vd, 2019). %99 civarı başarım oranı veren çalışmalarda LSTM, CNN, Kelime Yerleştirme, DNN ve MLP teknikleri kullanılmıştır.

Metin içeren veri kümelerinde LSTM, görüntü içeren veri kümelerinde CNN algoritmasının daha etkili olduğu, melez yaklaşımların yüksek başarımlara ulaşabildiği görülmüştür.

Bazı veri kümelerinde DNN, MLP (Nagisetty ve Gupta, 2019; Yawen vd, 2018) ve DVM (Roy vd, 2016; Tyagi, 2016) algoritmalarının daha iyi başarım sonuçları verdiği saptanmıştır. Hiperparametre ince ayarı ile en iyi hiperparametre seçimi de yapan çalışmaların olduğu görülmüştür (Kaynar vd, 2016).

Klasik makine öğrenme algoritmaları ile derin öğrenme algoritmalarının başarımını karşılaştıran çalışmalar dikkate alındığında, derin öğrenme algoritmalarının daha yüksek başarım oranı ile spam tespit ettiği görünmektedir.

Derin öğrenme tabanlı sistemlerle, makine öğrenmesindeki çalışmalar öznitelik mühendisliğinden ziyade model ve mimari mühendisliğine dönüşmüştür. Elimizde çok küçük veri kümesi varsa problem derin öğrenme ile çözümü çok uygun olmayabilir. Derin öğrenme modelinde bulunan hiperparametrelerin çok iyi ayarlanması veya sistemin en iyi sonucu verecek şekilde çalışılan modele güncelleyebilecek yapılar oluşturulması gereklidir.

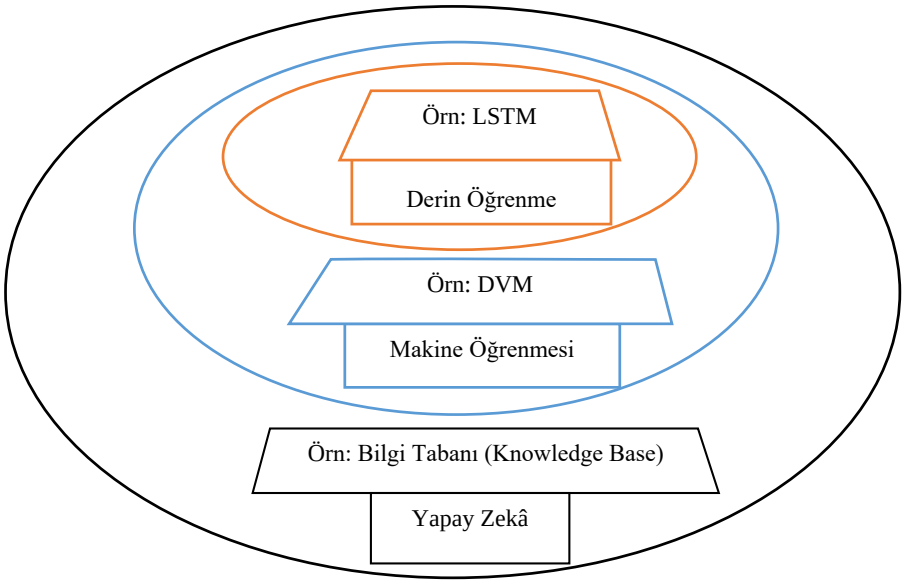
# 3. MATERYAL VE YÖNTEM

## 3.1. Materyal

İstenmeyen elektronik posta tespiti için günümüzde çoğunlukla yapay zekâ teknikleri kullanılmıştır. Bu nedenle yapay zekâ, makine öğrenmesi ve derin öğrenme kavramlarını çok iyi anlamak gerekmektedir. Bu kısımda bu kavramlar açıklanmıştır.

### 3.1.1. Yapay Zekâ

Yapay zekâ (AI veya YZ), görüntü tanıma, akıllı hoparlörler ve otomatik pilot otomobiller gibi, bir sistemin "harici verileri doğru bir şekilde yorumlayabilmesi, bu tür verilerden öğrenebilmesi ve bu bilgileri kullanabilmesi" olarak tanımlanmaktadır. 1950'lerde akademik bir disiplin olarak kurulan yapay zekâ, yarım asırdan fazla bir süre için göreceli bilimsel belirsizlik ve sınırlı pratik ilgi alanı olarak kaldı. Bugün, Büyük Veri'nin (Big Data) yükselişi ve bilgisayar gücündeki gelişmeler nedeniyle, iş ortamına ve kamuya açık konuşmalara girmiştir. AI, analitik, insandan ilham alan bilişsel, duygusal ve sosyal zekâ gibi zekâ türlerine bağlı ve insanlaştırılmış AI olarak sınıflandırılabilir (Haenlein ve Kaplan, 2019). Şekil 3.1'de yapay zekânın, derin öğrenme ile makine öğrenmesi arasındaki ilişkisi gösterilmiştir. Makine öğrenmesi yapay zekânın bir alt disiplini iken, derin öğrenme de makine öğrenmesinin bir alt disiplini sayılmaktadır.




Şekil 3.1. Derin öğrenme, makine öğrenmesi ve yapay zekâ ilişkisi

Canlıların düşünme şeklini ve davranışlarını model alan yapay zekâ çalışmaları; varlıkların akıllı davranışlarını öğrenip yapay olarak üretmeyi amaçlamaktadır. Yapay

zekâ teknolojileri arasında uzman sistemler, genetik algoritmalar, bulanık mantık, yapay sinir ağları, makine öğrenmesi gibi teknikler sıralanabilir. Doğa ve canlıların izlenmesiyle karınca kolonisi, parçacık sürüsü ve yapay arı gibi algoritmalar oluşturulmuştur. İnsan gibi düşünüp karar veren yapılar bu akılcı sistemlerle modellenerek makinelere aktarılmaktadır. Yapay zekâ daha çok insan olmak üzere canlıların zekice kabul edilen davranışlarına sahip bilgisayar sistemleridir ve makine öğrenmesi bu anlamda yapay zekânın son evresi olarak kabul edilmektedir (Atalay ve Çelik, 2017).

### 3.1.2. Makine Öğrenmesi

Makine öğrenmesi veya yapay öğrenme, öğrenebilen ve veriler üzerinden tahminlerde bulunabilen algoritmaların çalışmasını sağlayan bir sistemdir. Bir makine öğrenmesi algoritmasıyla bazı özelliklere göre insanların göğüs kanseri olup olmadığı belirlendiği gibi, kişinin dini inanışı, eğitime harcadığı para miktarı, yaşadığı bölgeye göre siyasi görüşü ile ilgili sınıflandırma yapılabilmektedir. Müşterilerin daha önce yaptığı alışverişleri analiz edip neleri beğeneceğini ve satın alabileceğini öngören önerici sistemler de makine öğrenmesine örnek teşkil etmektedir. Telefonumuzla internette arama yapmak için veya sözlü komutlarla yönlendirdiğimiz teknolojiler de makine öğrenmesi ile olmaktadır. Sanal kişisel asistanlar Siri, Cortana, Google Asistan ses tanıma teknolojisi ile verilen komutlara göre insana konuşmasını makine öğrenmesi algoritmaları ile işleyip buna göre her geçen gün daha doğal tepki vermektedir. Bankacılık sistemlerinde, hastalık teşhislerinde, görüntü işleme vb. birçok alanda makine öğrenmesi kullanılmakta olup her geçen gün hayatımızı daha fazla etkilemektedir.

Makine öğreniminin en yaygın şekli, derin olsun olmasın, denetimli öğrenmedir. Bir e-postanın spam olup olmadığını sınıflandırabilecek bir sistem kurmak istediğimizde, öncelikle, her biri kategorisiyle etiketlenmiş spam ve spam olmayan geniş bir veri kümesi toplanır. Eğitim sırasında, makineye bir e-posta gösterilir ve her kategori için bir tane olmak üzere bir skor vektörü şeklinde bir çıktı üretilir. İstediğiniz kategorinin tüm kategorilerde en yüksek puana sahip olmasını istiyoruz, ancak bu eğitimden önce gerçekleşmesi olası değildir. Çıktı puanları ile istenen puan deseni arasındaki hatayı (veya mesafeyi) ölçen nesnel bir fonksiyon hesaplanır. Makine daha sonra bu hatayı azaltmak için dâhili ayarlanabilir parametrelerini değiştirir. Genellikle ağırlık olarak adlandırılan bu ayarlanabilir parametreler, makinenin giriş-çıkış

fonksiyonunu tanımlayan gerçek ayar düğmeleri gibidir. Ağırlık vektörünü uygun şekilde ayarlamak için, öğrenme algoritması, her ağırlık için, ağırlık küçük bir miktar arttırılınca hatanın ne kadar artacağını veya azalacağını gösteren bir gradyan vektörü hesaplanmaktadır. Ağırlık vektörü daha sonra gradyan vektörüne zıt yönde ayarlanmaktadır (LeCun vd, 2015).

Uygulamada genellikle stokastik gradyan inişi (SGD) adı verilen bir prosedür kullanılır. Bu optimize edici fonksiyon, giriş vektörünü birkaç örnek için göstermek, çıktıları ve hataları hesaplamak, bu örnekler için ortalama gradyanı hesaplamak ve ağırlıkları buna göre ayarlamak için kullanılır. Eğitim seti ile birçok küçük örnek grubu için işlem tekrarlanır. Buna stokastik denir, çünkü her küçük örnek kümesi, tüm örnekler üzerindeki ortalama gradyanın gürültülü bir tahminini verir (Bousquet vd, 2007). Eğitimden sonra, sistemin performansı test seti adı verilen farklı bir dizi örnek üzerinde ölçülür. Bu, makinenin eğitim sırasında hiç görmediği yeni girdiler hakkında hassas cevaplar üretme yeteneğini ve genelleştirme yeteneğini test eder. Tipik bir derin öğrenme sisteminde, makineyi eğitmek için yüz milyonlarca ayarlanabilir ağırlık ve etiketli örnek olabilmektedir.

### 3.1.3. Derin Öğrenme ve Modelleri

Derin öğrenme bir veya daha fazla gizli katman içeren yapay sinir ağları (Zavvar vd, 2016) ve benzeri makine öğrenme algoritmalarını kapsayan çalışma alanıdır.

Derin öğrenme, çoklu soyutlama seviyelerine sahip verilerin gösterimini öğrenmek için çoklu işleme katmanlarından oluşan hesaplama modellerine izin verir. Bu yöntemler konuşma tanıma, görsel nesne tanıma, nesne algılama ve ilaç keşfi ve genomik gibi diğer birçok alanda son teknolojiyi önemli ölçüde geliştirmiştir. Derin öğrenme, makinenin her katmandaki gösterimi önceki katmandaki gösterimden hesaplamak için kullanılan dâhili parametrelerini nasıl değiştirmesi gerektiğini göstermek için geri yayılma algoritmasını kullanarak karmaşık veri kümelerinde karmaşık yapıyı keşfeder. Derin evrimsel ağlar, görüntü, video, konuşma ve ses işlemede çığır açarken, tekrarlayan ağlar metin ve konuşma gibi sıralı veriler üzerinde ışık tutmaktadır (LeCun vd, 2015).

İnsanın öğrenmesine en yakın olan derin öğrenme klasik makine öğrenmesi algoritmalarının yetersiz kaldığı bazı durumlarda daha da öne çıkmaktadır. Google derin öğrenme kütüphanesi Tensorflow 2015 yılında Google Brain Team tarafından

yayınlandıktan sonra, son yıllarda derin öğrenme konusuna ilgi oldukça artmıştır. Bu ilginin nedeni derin öğrenme ile “natural language processing” (doğal dil işleme - NLP), “computer vision” (bilgisayarlı görü), “speech recognition” (konuşma tanıma), “image processing” (görüntü işleme) gibi birçok alanda başarılı sonuçlar vermektedir. Derin öğrenme genellikle öğrenim alt sisteminde bir sınıflandırıcı modeli tanımlayabilir ve sınıflandırır. Temsil öğrenme, bir makinenin ham verilerle beslenmesini ve tespit veya sınıflandırma için gerekli gösterimleri otomatik olarak keşfetmesini sağlayan bir dizi yöntemdir Derin öğrenmenin en önemli özelliği, bu özellik katmanlarının insanlar tarafından tasarlanmadığıdır. Verilerden öğrenilen genel amaçlı bir öğrenme izleği kullanılır.

Tezin bu bölümünde derin öğrenme modellerinden RNN, LSTM, GRU, BLSTM, BERT, DistilBERT açıklanmıştır.

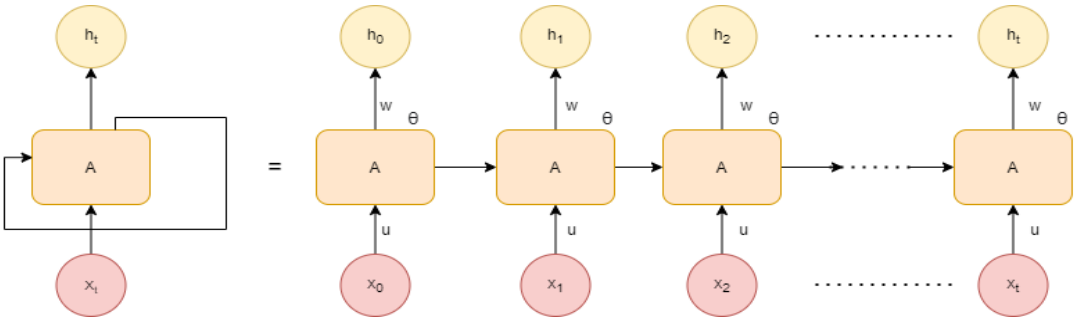
#### 3.1.3.1. Tekrarlayan Sinir Ağları RNN

Tekrarlayan sinir ağları, 1990'lı yıllarda araştırma ve geliştirmenin önemli bir odağı olmuştur. Sıralı veya zamanla değişen kalıpları öğrenmek için tasarlanmıştır. Birbirini takip eden sıralı uygulamalar, hareket algılama ve müzik sentezi ve finansal tahminler gibi konularda çalışan yapay sinir ağları modelidir (Medsker ve Jain, 1999).

Günümüzde özellikle görüntü ve dil işleme alanındaki birçok problemin çözümünde sinir ağları kullanılmaktadır. Tekrarlayan sinir ağları (RNN) bir derin öğrenme mimarisi türü olup sinir ağlarının yinelenmesi ile oluşmaktadır. RNN, zaman serileri veya doğal dil işleme gibi dizi verilerini modellemek için güçlü olan bir sinir ağları sınıfıdır. RNN haricinde olan yapay sinir ağlarında girdilerin ağa geliş sırasının bir önemi yoktur ve girdiler birbirlerinden bağımsız olarak işlenir. Bundan sonra girdinin ağdaki işlemi tamamlanıp çıktı üretildikten sonra sinir ağı bu girdi bilgisini unutur. Cümlede anlam olabilmesi için bir kelimeden sonra gelebilecek kelimeyi tahmin etmeye çalışırken önceki kelimelerin bilgisine ihtiyaç bulunur. Bu tür problemlerde çözüm olarak hafızalı olan sinir ağları RNN ortaya çıkmıştır. RNN modeli derin öğrenme ağında eğitim gerçekleştirirken bir önceki veriye ait bilgileri hafızasında tutar.

RNN bir vektör girdisini çıktı olarak diziye; bir resmi yazı ile betimleme, bir dizi girdisini çıktı olarak vektöre; olumlu, olumsuz, nötr gibi duygu analizleri, bir dizi girdisini başka bir çıktı dizisine; bir dilden başka bir dile çeviri dönüştürebilir. Tipik

bir RNN ağı Şekil 3.2.'de olduğu gibidir. Xt girdi, ht saklı durum denilen tekrarlayan katmanlar boyunca bir önceki katmana ait olan bilgiyi - hidden state gösterir. X0 girişi hidden state ile birleşip, bir çıktı üretir ve hidden state'i güncellenir. X1 girdisi ve güncel hidden state'i alıp, diğer katmana girdi olarak verirken ağda yapılan işlemler bittikten sonra hidden state tekrardan güncellenir. Böylelikle model her yeni girdiyi aldığında girdiyle birlikte önceki girdi ile ilgili hidden state bilgisi verilir. Bu şekilde önceki girdiler ile ilgili bilgiler de bu RNN ağı sayesinde hafızada tutulur.




Şekil 3.2. RNN Modeli

RNN modelinin doğruluk açısından birçok rekabetçi dil modelleme tekniğinden önemli ölçüde daha iyi performans gösterdiği bilinmekle beraber hesaplama karmaşıklığı vardır. Hem eğitim hem de test aşamaları için 15 kattan fazla hızlanmaya yol açan yaklaşımlar, zaman algoritması aracılığıyla geri yayılım kullanmanın önemi, ileri beslemeli ağlarla deneysel bir karşılaştırma ve modeldeki parametre miktarının nasıl azaltılacağı (Mikolov vd, 2011) çalışmasında gösterilmiştir. RNN'lerin pratik uygulamaları genellikle çok küçük modelleri kullanır, çünkü büyük RNN'ler aşırı uyum-overfit eğilimindedir. Dropout (Srivastava, 2013) gibi hiperparametrelerin regulasyonu ile ezberlemenin azaltılarak dil modelleme, konuşma tanıma, makine çevirisi, resim yazısı oluşturma gibi çok çeşitli uygulamalarda performansı artırabileceğini belirtilmiştir (Zaremba vd, 2014).

Denklem 3.1 ile RNN iç ağ modelinin matematiksel açıklaması verilmiştir. W ve u girdi ile katman arasındaki ağırlıklar, θ ise aktivasyon fonksiyonudur. Burada ağırlık matrisi önceki ve şu anki verinin hangisinin sonuca etkisi daha çok veya daha az ise ona göre değerler almaktadır.

$$h_t = \theta(u_{hh}h_{t-1} + w_{xh}x_t) \quad \textbf{(3.1)}$$

Her ikisi de vektör olarak ifade edildiği için $x_t$ ve $h_{t-1}$ birleştirilebilir. Çıkan sonuç sigmoid, tanh vb. aktivasyon fonksiyonundan geçirilip fonksiyon değeri 0 ile 1 arasına

sıkıştırılıp bir değer üretir. İleri besleme işlemleri bittiği için yeni hidden state üretilir. Eğitimi gerçekleştirebilmek için geriye giderek geri yayılım işlemi ile optimizasyon yapılır. RNN için kullanılan yöntem ise Backpropagation Trough Time-BPTT diye bilinen zamana bağlı sıralı bir dizi hesaplamanın tümü için geri yayılım uygulamasıdır. Optimizasyon ile ağdaki ağırlıkların güncellenmesi için diğer derin öğrenme modellerinde de kullanılan gradient descent-derece alçalması-gradyan inişi tekniği kullanılır. Epoklar (devir-adım-devre-dönem-epoch) boyunca tüm bu işlemler tamamlandığında model eğitilir. Birbirine bağlı uzun ağlarda hatanın etkisi oldukça düşerek gradyan kaybolmaya başlayabilir. Bu da doğru sonucu bulmayı olanaksızlaştırır. Bütün katmanlar ve zamana bağlı adımlar birbirine çarpımla bağlı olduğundan, türevleri yokolma-vanishing gradient veya patlama-exploding gradient yani aşırı yükselme tehlikesindedir. Bu problemleri çözmek için threshold-eşik değeri koymak, ağırlık-W için uygun başlangıç değerleri seçmek, farklı aktivasyon fonksiyonları veya LSTM kullanmak çözüm yöntemlerindendir.

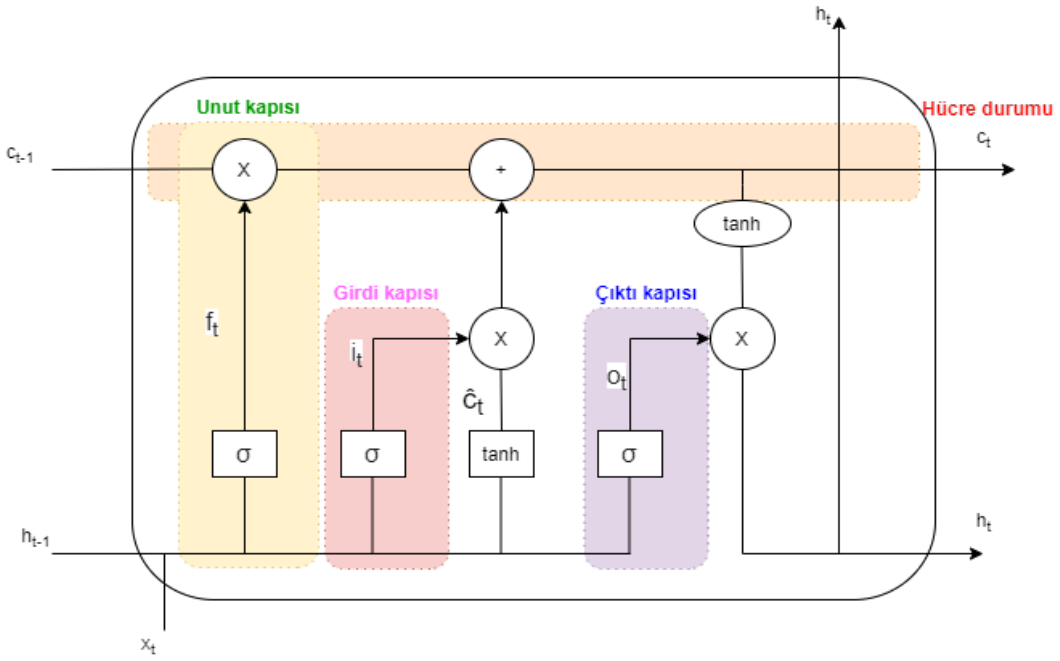
RNN konuşma ve dil gibi sıralı girdilerin eğitimi için daha iyidir. RNN'ler çok güçlü dinamik sistemlerdir ancak bu ağları eğitmek sorunludur. Çünkü geri yayılan gradyanlar her adımda büyür veya büzülür. Temel amacı uzun vadeli bağımlılıkları öğrenmek olsa da bilgiyi uzun süre saklamak zordur. Bunu düzeltmek için ağ açık bir bellekle güçlendirilmelidir. Doğal davranışları uzun süre girdileri hatırlayan ve özel gizli birimleri kullanan Uzun-kısa süreli bellek (LSTM) ağları bu sorunu düzeltmek için ortaya konulmuştur (Hochreiter ve Schmidhuber, 1997). LSTM ağları özellikle her bir adım adım için birkaç katmana sahip olduklarında, geleneksel RNN'lerden daha etkili olduklarını kanıtlamışlardır.

#### 3.1.3.2. Uzun Kısa Süreli Bellek LSTM

Uzun Kısa Süreli Bellek anlamına gelen LSTM, ilk olarak (Hochreiter ve Schmidhuber, 1997) tarafından önerilmiştir. Gerçek zamanlı tekrarlayan öğrenme, zamanda geri yayılma, tekrarlayan kademeli korelasyon, Elman ağları ve nöral sekans yığınlaması ile karşılaştırmalarda, LSTM çok daha başarılı çalışmalara yol açar ve çok daha hızlı öğrenir. LSTM ayrıca, önceki tekrarlayan ağ algoritmaları tarafından asla çözülmemiş karmaşık, yapay uzun zaman gecikmeli görevleri de çözmektedir. LSTM, özel birimlerdeki sabit hata karuselleri aracılığıyla sabit hata akışını zorlayarak 1000 farklı zaman adımını aşan minimum zaman gecikmelerini köprülemeyi öğrenebilir. Çarpımlı geçit birimleri, sabit hata akışına erişimi açıp kapatmayı öğrenir. LSTM, uzay

ve zamanda yereldir; zaman adımı ve ağırlığı başına hesaplama karmaşıklığı $O1$ olarak belirtilmiştir (Hochreiter ve Schmidhuber, 1997).

RNN'den LSTM'ye geçtiğimizde, eğitilmiş ağırlıklara göre girişlerin akışını ve karışımını kontrol eden daha fazla kontrol düğmesi sunulduğu söylenebilir. Dolayısıyla, LSTM bize en fazla kontrol yeteneğini ve dolayısıyla daha iyi sonuçları verir. Ama aynı zamanda daha fazla karmaşıklık ve işletme maliyetiyle birlikte gelmektedir. Tipik bir LSTM hücresi Şekil 3.3.'te olduğu gibidir. Derin öğrenme LSTM ağları bu gibi hücrelerin birleşiminden oluşur. Sıradan bir LSTM hücresi, bir hücre durumu, bir girdi kapısı, bir çıktı kapısı ve bir unut kapısından oluşur. Hücre, keyfi zaman aralıklarındaki değerleri hatırlar. Bu üç kapı, hücreye giren ve çıkan bilgi akışını düzenlemektedir.




Şekil 3.3. LSTM hücresi

Bir LSTM hücresinin parametrelerinin hesabı denklem (3.2)-(3.7) arasında verilmiştir. Denklem 3.2 ile giriş / güncelleme kapısının aktivasyon vektörü, denklem 3.3 ile unutma kapısı vektörü, denklem 3.4 ile çıkış kapısının aktivasyon vektörü, denklem 3.5 ile hücre girişi aktivasyon vektörü, denklem 3.6 ile hücre durum vektörü hesaplanmaktadır. Denklem 3.7 ile ise LSTM biriminin çıkış vektörü olarak da bilinen gizli durum vektörünü göstermektedir.

$$i_t = \sigma(x_t U^i + h_{t-1} W^i) \quad \textbf{(3.2)}$$

$$f_t = \sigma(x_t U^f + h_{t-1} W^f) \quad \textbf{(3.3)}$$

$$o_t = \sigma(x_t U^o + h_{t-1} W^o) \quad \textbf{(3.4)}$$

$$\hat{c}_t = tanh(x_t U^g + h_{t-1} W^g) \quad \textbf{(3.5)}$$

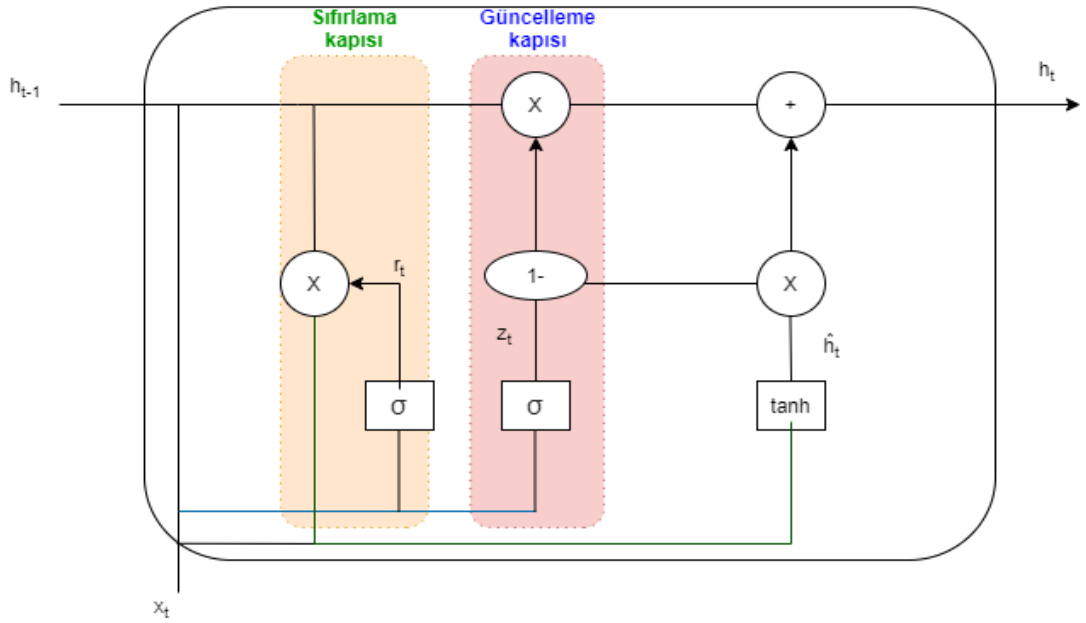
$$c_t = \sigma(f_t * c_{t-1} + i_t * \hat{c}_t) \quad \textbf{(3.6)}$$

$$h_t = tanh(c_t) * o_t) \quad \textbf{(3.7)}$$

#### 3.1.3.3. Geçitli Tekrarlayan Birim GRU

Geçitli Tekrarlayan Birim anlamına gelen GRU ilk olarak (Cho vd, 2014) tarafından önerilmiştir. GRU, iki tekrarlayan sinir ağından oluşan RNN Encoder-Decoder adlı yeni bir sinir ağı modelidir. Bir RNN, bir sembol dizisini sabit uzunlukta bir vektör temsiline kodlar ve diğeri, gösterimi başka bir sembol dizisine dönüştürür. Önerilen modelin kodlayıcı ve kod çözücüsü, bir kaynak dizisi verilen bir hedef dizinin koşullu olasılığını maksimize etmek için birlikte eğitilir. Bir istatistiksel makine çeviri sisteminin performansının, mevcut log-lineer modelde ek bir özellik olarak RNN Kodlayıcı-Kod Çözücü tarafından hesaplanan ifade çiftlerinin koşullu olasılıklarını kullanarak geliştirdiği deneysel olarak bulunmuştur. Niteliksel olarak, önerilen modelin dilbilimsel ifadelerin anlamsal ve sözdizimsel olarak anlamlı bir temsilini öğrendiğini göstermektedir (Cho vd, 2014).

GRU'nun geçidi sıfırlama (reset) ve güncelleme (update) olmak üzere iki kapısı vardır. Güncelleme kapısı, bir LSTM'nin unutma ve giriş kapısına benzer şekilde davranır. Hangi bilgilerin atılacağına ve hangi yeni bilgilerin ekleneceğine karar verir. Sıfırlama kapısı, ne kadar geçmiş bilginin unutulacağına karar vermek için kullanılan başka bir kapıdır. GRU daha az eğitim parametresi kullanır ve bu nedenle daha az bellek kullanır ve daha hızlı çalışır. GRU daha hızlı eğitim gereken, LSTM ise daha büyük veri kümelerinde kullanımı doğrudur. Şekil 3.4. ile GRU modelinin bir hücresi verilmiştir. GRU ağları bu hücrelerin bir araya gelmesi ile oluşmaktadır.




Şekil 3.4. GRU hücresi

Bir GRU hücresinin parametrelerinin hesabı denklem (3.8)-(3.11) arasında verilmiştir. Denklem 3.8 ile sıfırlama kapısının vektörü, denklem 3.9 ile güncelleme kapısı vektörü, denklem 3.10 ile aday aktivasyon vektörü, denklem 3.11 ile çıktı vektörü hesaplanmaktadır.

$$r_t = \sigma(x_t W^r + h_{t-1} W^r) \quad \textbf{(3.8)}$$

$$z_t = \sigma(x_t W^z + h_{t-1} W^z) \quad \textbf{(3.9)}$$

$$\hat{h}_t = tanh(W.[r_t * h_{t-1}, x_t] \quad \textbf{(3.10)}$$

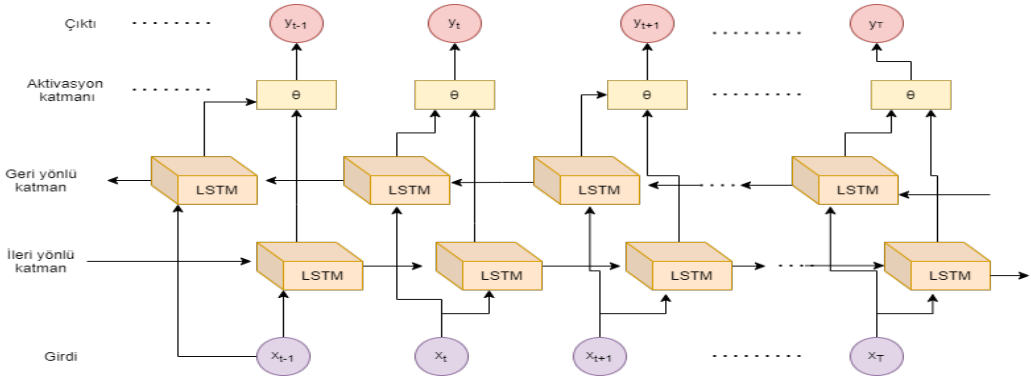
$$h_t = (1 - z_t) * h_{t-1} + z_t * \hat{h}_t] \quad \textbf{(3.11)}$$

GRU'ların daha az tensör işlemi vardır. Bu nedenle, LSTM'lere göre modeli eğitmek için biraz daha hızlıdırlar. Hangisinin daha iyi olduğu konusunda net bir kazanan bulunmamaktadır. Hangisinin daha iyi çalıştığı kullanım durumlarına göre belirlenmeye çalışılmaktadır.

### 3.1.3.4. Çift Yönlü Uzun Kısa Süreli Bellek BLSTM

Çift yönlü LSTM'ler (Schuster ve Paliwal, 1997), sınıflandırma problemlerinde model performansını artırabilen geleneksel LSTM'lerin bir uzantısıdır. Çift yönlü LSTM'lerin kullanımı, tüm dizi tahmin problemleri için anlamlı olmayabilir, ancak uygun olduğu alanlarda daha iyi sonuçlar açısından bazı faydalar sağlayabilir. Çift yönlü ağların tek yönlü ağlardan önemli ölçüde daha etkili olduğu gösterilmiştir (Graves ve Schmidhuber, 2005).

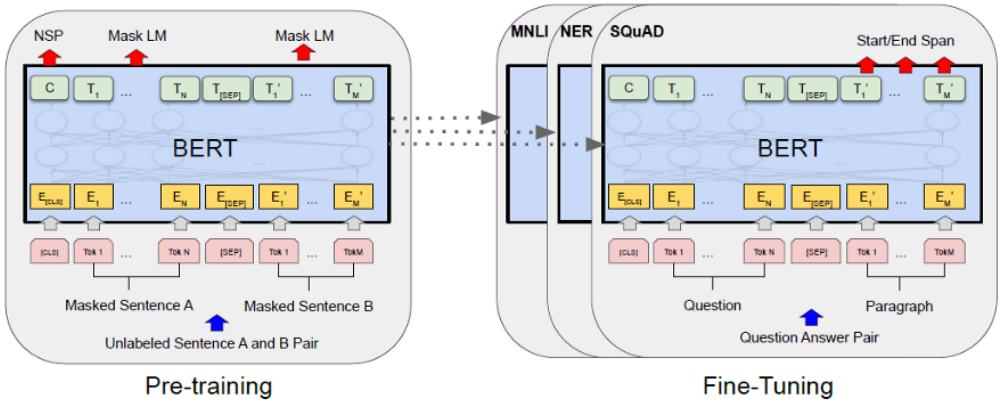
Giriş dizisinin tüm zaman adımlarının mevcut olduğu problemlerde, Çift Yönlü LSTM'ler giriş dizisinde bir LSTM yerine iki tane çalıştırır. Birincisi olduğu gibi giriş dizisi üzerinde ve ikincisi ise giriş dizisinin ters kopyası üzerindedir. Bu, ağa ek bir bağlam sağlayabilir ve sorunla ilgili daha hızlı ve hatta daha eksiksiz öğrenmeye sebep olabilir. Şekil 3.5 ile genel bir BLSTM modeli gösterilmiştir.




Şekil 3.5. BLSTM modeli

### 3.1.3.5. BERT ve DistilBERT

2018 yılı Transfer Öğrenmeyi tanıtarak Doğal Dil İşleme (NLP) alanında devrim niteliğinde bir değişiklik getirmiştir. Transformatörlerden Çift Yönlü Kodlayıcı Temsili (BERT), Google AI ekibi tarafından 2018 yılında sunulan ve çeşitli NLP görevlerinde son teknoloji sonuçlarıyla NLP topluluğunu sarsan klasik bir Transfer Öğrenimi örneğidir. Oldukça pragmatik yaklaşımı ve yüksek performansı sebebiyle BERT çeşitli NLP görevleri için kullanılır ve dil modellerinde son teknoloji ürünü sonuçlar elde eder. Son dil temsil modellerinden farklı olarak, BERT, tüm katmanlarda hem sol hem de sağ bağlamda birlikte koşullandırılarak etiketlenmemiş metinden derin çift yönlü gösterimleri önceden eğitmek için tasarlanmıştır. Sonuç olarak, önceden eğitilmiş BERT modeli, soru yanıtlama ve dil çıkarımı gibi çok çeşitli görevler için son teknoloji modeller oluşturmak için göreve özgü mimari değişiklikleri olmadan yalnızca bir ek çıktı katmanı ile ince ayar yapılabilir. Şekil 3.6. ile BERT modelinin önceden eğitilmiş ve ince ayarlanmış yapıdaki modeli verilmiştir (Devlin vd, 2018).




Şekil 3.6. BERT modeli

BERT modellerinin birçok türü bulunmaktadır. Bunlar arasında; BERT, DistilBERT (Sanh vd, 2019), RoBERTa (Liu vd, 2019), ALBERT, StructBERT, CamemBERT, MobileBERT, SpanBERT gibi birçok çeşidi vardır. Bunlar arasında eğitim süresi, eğitilen veri kümesi, parametre sayısı vb. gibi farklılıklar bulunmaktadır. Doğal dil işleme alanında ayrıca, GloVe, Elmo, ELECTRA, XLNET, T5, Turing NLG, GPT, GPT-2,GPT-3 gibi birçok ön eğitimli veya ince ayarlı model bulunmaktadır.

### 3.1.4. Derin Öğrenme Araçları

Bu tez çalışmasında modülerlik ve genişletilebilir yapısı (Keras.io) nedeniyle Spyder yazılım geliştirme ortamında Python derin öğrenme kütüphanesi Keras (Chollet, 2015) ve Google Python geliştirme ortamı (Google-Colaboratory) kullanılmıştır.

## 3.2. Yöntem

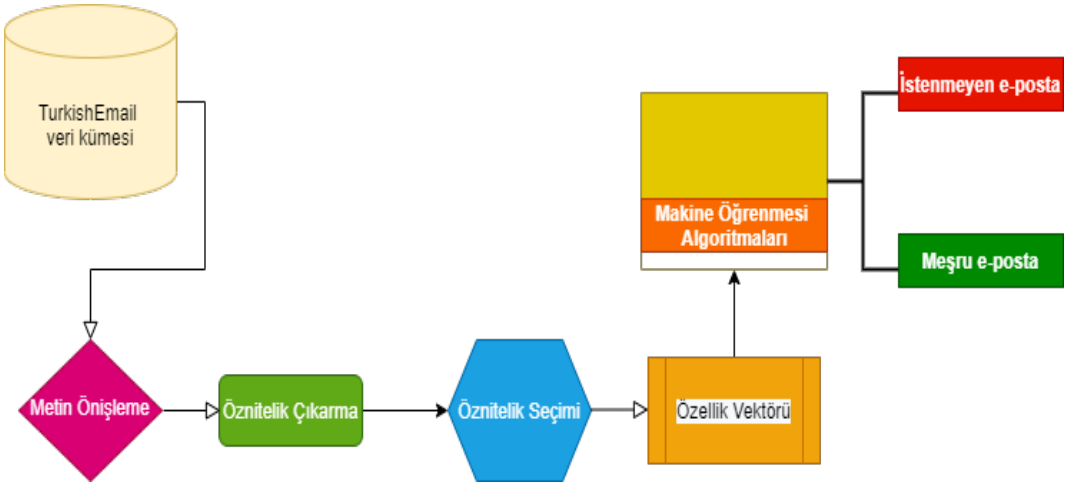
Makine öğrenmesi ve derin öğrenme ile istenmeyen e-posta tespitinin nasıl yapıldığı bu bölümde gösterilmiştir. Aynı zamanda veri kümesinin hazırlanması ve kullanılan performans ölçütleri açıklanmıştır.

### 3.2.1. Makine Öğrenmesi ile Spam Tespiti

Spam filtreleme için öncelikle verinin ön işleme tabi tutulması gerekir. Bunun için ilk olarak spam verisini sözcüksel analiz yani dizgeciklere ayırılır (spam veya ham e-posta temsili, başlıklar, ekler ve HTML etiketleri, e-posta gövdesi, konu satırı, IP adresleri, alan adları) ve bu verilerden bağlaç vb. gibi durak kelimeleri (stop-words) kaldırılır. Daha sonrasında ise kelimeleri köklerine ayırma ve temsil olarak kullanılan

e-posta mesajının, kullanılan makine öğrenmesi algoritmasının gerektirdiği şekilde özellikli veya yapılandırılmış bir formata dönüştürülür. Özellik çıkarımı, seçilen özelliğin seçimi, e-posta başlık analizi, içerik olmayan özelliklere dayalı filtreler (geçici özellik, SMTP yolu, davranış ve kullanıcıların sosyal ağ analizi) de bu ön işleme adımlarındandır (Bhowmick ve Hazarika, 2016).

Şekil 3.7.de makine öğrenmesi algoritmaları ile genel bir istenmeyen e-posta tespit modeli gösterilmiştir. Makine öğrenmesi ile veri kümesi ön işlem adımlarından geçtikten sonra öznitelik çıkarılıp seçilir daha sonrasında bilgisayarın anlayacağı şekilde özellikler sayısallaştırılıp vektöre çevrilir. Makine öğrenmesi algoritmaları bu sayısal değerler içinde hesaplamalar yapıp hedeflenen ve tahmin edilen çıktılara göre bir başarım oranı oluşturur.

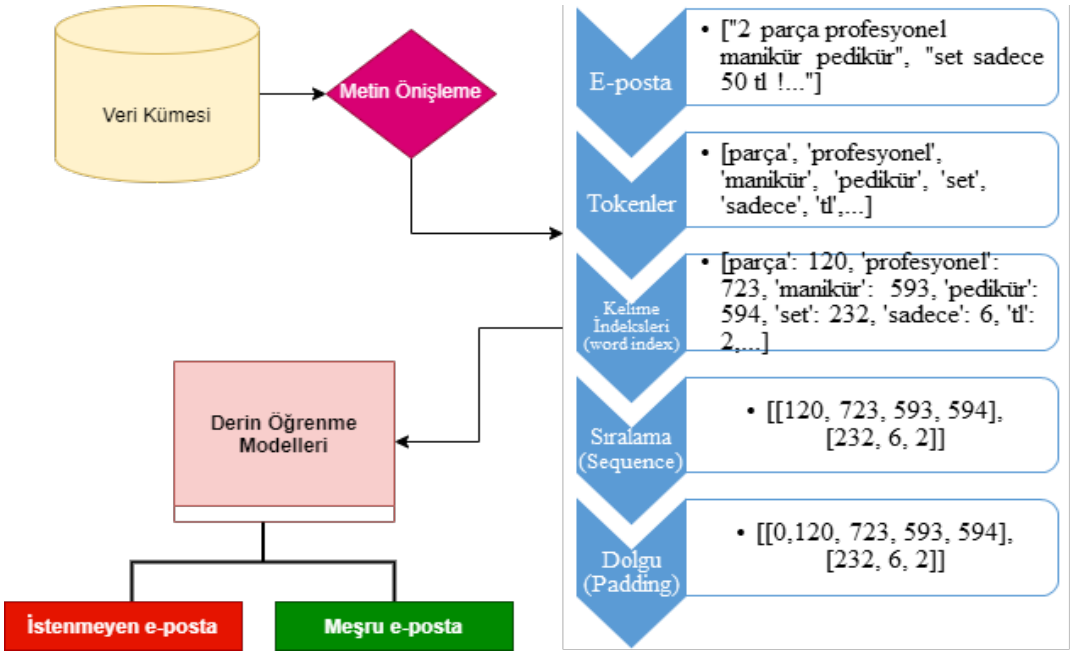



Şekil 3.7. Makine öğrenmesi ile istenmeyen e-posta tespiti

Metin sınıflandırma için öznitelik seçimi üzerine çok büyük boyutlu veri kümelerinde sınıflandırma algoritmalarının çalışması zaman alırken ezberleme problemide ortaya çıkar. Burada bütün kelimeleri kullanmak yerine ilgili en iyi alt kümenin seçilmesi işleri kolaylaştırabilir (Şahin ve Kılıç, 2019). Makine öğrenmesi teknikleri için öznitelik seçimi hayati öneme sahiptir. (Şahin vd, 2016)'nin çalışmasında, öznitelik seçiminde sıkça kullanılan CHI, IG ve OR metrikleri, (Eryılmaz vd, 2020b)'nin çalışmasında ise 5 farklı öznitelik seçme yöntemi (CHI, IG, ACC, OR, DF) kullanılmıştır. Ayrıca öznitelik seçimi yapılmadan da başarım sonuçları verilerek öznitelik seçim yöntemi ile başarımının ne kadar arttığı gösterilmiştir. Aynı zamanda metin sınıflandırmada terim ağırlıklandırma yöntemi olarak önerilen alaka sıklığı, bu çalışmada öznitelik seçme yöntemi olarak kullanılmıştır.

### 3.2.2. Derin Öğrenme ile Spam Tespiti

Bu yapıda veri kümesinin durumuna göre metin ön işlem işlemi yapılır. Metinlerden oluşan bir veri kümesinde öncelikle kelime köklerine ayrılır veya görüntü verisinde öncelikle renklerine ayrıştırma yapılır. Bir sinir hücresini diğerine bağlayan sayısal değere ağırlık denir. Gizli katmanlarda o katmana göre özel olarak belirlenmiş aktivasyon fonksiyonu ile hesaplar yapılıp ağırlık çarpımlarının toplamı ile sonraki gizli katmana değerler aktarılır. Bu işlemler çıkış katmanına kadar devam eder. Giriş katmandan çıkış katmanına kadar gerçekleşen ileri yönlü hesaplamalar İleri yayılım algoritmasıdır (forward-propagation). Bu sinir ağının oluşturduğu değerler tahmini değerler olup tahmin değerleri ile gerçek değerler arasında hata farkları vardır. Bu farkın azaltılması için Geri yayılım algoritması (Backpropagation) kullanılması gerekir. Hatalar geriye doğru yayılarak her bir ağırlık güncellenir böylelikle çıktının hedef çıktıya yakınlaşması sağlanır. Bu esnada optimizasyon fonksiyonları kullanılır ve hataların azaltılması için zincir kuralına göre birinci türevler alınır. Öznitelik çıkarma ve özellik seçme işlemleri de derin öğrenme modeli içinde yapılır. Derin öğrenme modellerinde token oluşturulur. Oluşturulan tokenler bir takım derin öğrenme modelleri ile hesaplanıp hedef çıktı sonucunda ortaya çıkan hata ve başarım değerleri bulunur. Bu hata ve başarım değerleri ile modellerin performansı değerlendirilir. Şekil 3.8'de bu tez çalışmasında da kullanılan derin öğrenme modelleri için istenmeyen e-posta tespit modeli gösterilmiştir.




Şekil 3.8. Derin öğrenme ile istenmeyen e-posta tespiti

E-postada bulunan metinler derin öğrenme modellerine girmeden önce vektörleştirilir. Bu çalışmada Kelime yerleştirme ile her kelime bir vektör olarak temsil edilmiştir. Kelime yerleştirme ile kullanılan veri kümelerinde kullanılan dilin istatistiksel yapısı anlamsal olarak geometrik bir uzayda haritalanır. Böylelikle semantik olarak benzer kelimeler derin öğrenme modelleri eğitilirken benzer yerleştirme alanında eşlenir. Bu tezde Keras ile SimpleRNN, LSTM, GRU, BLSTM modelleri eğitilirken kelime yerleştirme işlemi yapılmıştır. Diğer bir yol ise yine bu tezde kullanılan BERT, DistilBERT gibi önceden eğitilmiş kelime düğümlerini kullanan modelleri kendi veri kümemizle değerlendirmektir. Tez çalışmasında derin öğrenme modelleri olarak RNN, LSTM, GRU, BLSTM, BERT, DistilBERT kullanılmıştır. Çalışmada 400 istenmeyen e-posta ve 400 meşru e-posta olmak üzere toplam 800 adet Türkçe e-postadan oluşan "TurkishEmail" (Ergin vd, 2012) veri kümesi kullanılarak 6 farklı derin öğrenme modeli ile istenmeyen e-posta tespiti yapılmıştır. Tensorflow üzerinde koşulan Python programlama dili derin öğrenme kütüphanesi Keras, Spyder (Anaconda3), Jupyter, Google Colabratory platformları üzerinde derin öğrenme yöntemleri kullanılarak istenmeyen e-postalar test edilmiş, başarım oranları farklı performans metrikleri ile ölçülmüştür.

Çalışma kapsamında ayrıca yeni bir Türkçe e-posta veri kümesi oluşturulmuştur. "TRHamSpamEmail" adı verilen bu veri kümesinde toplam 350 adet e-posta bulunmaktadır. Bu Türkçe e-postaların yarısı meşru diğer yarısı da istenmeyen e-postalardan oluşmaktadır. Oluşturulan veri kümesi dengeli bir e-posta veri kümesidir.

### 3.2.3. Veri Kümesinin Hazırlanması

Derin öğrenme teknikleri ile Keras kullanılarak istenmeyen e-postaları ayrıştırmak için (Ergin vd, 2012) hazırlanan toplam 800 adet Türkçe e-posta içeren "TurkishEmail" veri kümesi kullanılmıştır. 400 istenmeyen e-posta, 400 de meşru e-postalar veri kümemizi oluşturmaktadır.

Tablo 3.1. derin öğrenme modelleri için çalışılan veri kümesinin özellikleri gösterilmektedir. Tablo 3.1.'de verildiği üzere "TurkishEmail" e-posta veri kümesinde toplam 800 e-posta bulunmaktadır. Bu e-postaların 400 tanesi meşru, 400 tanesi de istenmeyen e-postalardan oluşmaktadır.

Tablo 3.1.'e bakıldığında veri kümesinin, istenmeyen ve meşru e-postaların yarı yarıya dağıldığı, dengeli bir veri kümesi olduğu görülmektedir.

Tablo 3.1. “TurkishEmail” e-posta veri kümesi

| | E-posta sayısı |
|---|---|
| Meşru e-posta | 400 |
| İstenmeyen e-posta | 400 |
| Toplam e-posta sayısı | 800 |

Tablo 3.2.’de verildiği üzere "TRHamSpamEmail" e-posta veri kümesinde toplam 350 e-posta bulunmaktadır. Bu veri kümesinin 175 tanesi meşru ve 175 tanesi de istenmeyen e-postalardan oluşmaktadır. Tabloya bakıldığında istenmeyen ve meşru e-posta dağılımı yarı yarıya olan dengeli bir veri kümesi görülmektedir.

Tablo 3.2. "TRHamSpamEmail" e-posta veri kümesi

| | E-posta sayısı |
|---|---|
| Meşru e-posta | 175 |
| İstenmeyen e-posta | 175 |
| Toplam e-posta sayısı | 350 |

Elektronik postalarda metin verilerinden yani kelimelerden oluşmaktadır. Bu veriler yapısal olmayan veri kümeleri arasında bulunmaktadır (Hotho vd, 2005). Metin belgeleri bilgisayarlar tarafından okunulup görüntülense de bu belgeler üzerinde makine öğrenmesi algoritmalarının çalıştırılabilmesi için çeşitli işlemlerden geçirilerek yapısal veri kümelerine dönüştürülmelidir. Bu dönüşüm, metnin insan dilinden makine tarafından anlaşılabilir formata aktarmak için gerekmektedir. Bu adımların en başında ise ön işlem basamakları bulunur. Çalışmanın ön işlem aşamasında uygulanan adımlar şu şekilde sıralanabilir.

- ✓ Tüm harfler büyük harften küçük harfe dönüştürülür.
- ✓ Türkçe ve İngilizce dilinde yer alan alfabeler dışında tüm karakterler ve noktalama işaretleri silinir.
- ✓ Rakamlar çıkartılır.
- ✓ Metin içerisinde yer alan belli başlı kısaltmalar genişletilir.
- ✓ Kelimeler boşluk karakterine (white space) göre parçalanır.
- ✓ Parçalanma sonucunda elde edilen her kelime Türkçe doğal dil işleme kütüphanesi olan (Zemberek) yazılımı vasıtasıyla köklerine ayrıştırılır.
- ✓ Durak kelimeler çıkartılır.
- ✓ Veri kümesi içinde sıklıkla ve anlamlı halde bulunan ama gerekli olmayan istenmeyen e-posta tespiti için bir anlam ifade etmeyen

kelimeler de veri kümesi içinden çıkarılmıştır. Bunlar: kim, konu, gmail, hotmail, com, tr, yahoo, mailfrom, yahoogroups com, yahoogroup co uk, net, org vb. şeklindedir.

Ön işlem adımının uygulanmasıyla e-postalar içerisinde yer alan tüm kelimelerin standartlaştırılması sağlanmaktadır. Örneğin "bedava" kelimesi ile "Bedava" kelimesi aynı önem derecesine sahiptir. Ön işlem adımı uygulanmasaydı, bu iki kelime bilgisayar tarafından farklı kelimeler gibi değerlendirilecektir. Bunun sonucu olarak özellik vektörü farklı olacak ve algoritmalar farklı sonuç verecektir. Bu gibi nedenlerden dolayı ön işlem adımında uygulanan her bir aşama sınıflandırma başarımı için oldukça önemlidir. Ön işlem adımının ardından makinenin anlayabileceği sayısal girdiyi elde edebilmek için gerekli hesaplamalar derin öğrenme modeli ile oluşturulmaktadır.

Şekil 3.9 ile meşru e-posta veri kümesinde bulunma sıklığına göre kelime bulutu şeklinde gösterilmiştir. Şekil 3.10. ile ise istenmeyen e-posta veri kümesinde bulunma sıklığına göre kelime bulutu şeklinde gösterilmiştir.


Şekil 3.9. “TurkishEmail” meşru e-posta kelime bulutu


Şekil 3.10. “TurkishEmail” istenmeyen e-posta kelime bulutu

### 3.2.4. Modellerin Test Edilmesinde Kullanılan Performans Ölçütleri

Derin öğrenme modellerinin test edilmesi için karışıklık matrisi ve performans metrikleri değerlendirilmektedir. Bu nedenle bu kısımda bu ölçütler açıklanacaktır.

Karışıklık matrisi makine öğrenmesi sınıflandırma problemleri için bir performans ölçümüdür. Karışıklık matrisi tahminlerin doğruluğu hakkında bilgi veren bir ölçüm aracıdır. Tahmin edilen ve gerçek değerlerin 4 farklı birleşimini içeren bir tablodur. Karışıklık matrisini oluşturan bileşenler arasında; Doğru-Pozitif, Doğru-Negatif, Yanlış-Pozitif, Yanlış-Negatif değerleri bulunmaktadır. Tablo 3.3. ile karışıklık matrisi gösterilmiştir.

**Doğru-Pozitif-DP**: Modelin spam olarak etiketli e-postayı spam olarak tahmin etmesi durumudur.

**Doğru-Negatif-DN**: Modelin normal olarak etiketli e-postayı normal tahmin etmesi durumudur.

**Yanlış-Pozitif-YP**: Modelin spam olarak etiketli e-postayı normal olarak tahmin etmesi durumudur.

**Yanlış-Negatif-YN**: Modelin normal olarak etiketli e-postayı spam olarak tahmin etmesi durumudur.

Tablo 3.3. Karışıklık matrisi

| | | Toplam | Tahmin edilen değerler | |
|---|---|---|---|---|
| | | | **Pozitif** | **Negatif** |
| **Gerçek Değerler** | | **Gerçek pozitif sayısı** | DP | YN |
| | | **Gerçek negatif sayısı** | YP | DN |
| | | **Toplam örnek sayısı** | **Tahmin pozitif sayısı** | **Tahmin negatif sayısı** |

Model performansını değerlendirmede kullanılan temel kavramlar doğruluk, kesinlik, hassasiyet ve F-ölçütüdür. Modelin başarısı, doğru sınıfa atanan örnek sayısı ve yanlış sınıfa gönderilen örnek sayısı ile ilgilidir.

Denklem (3.12) kesinlik, denklem (3.13) hassasiyet, denklem (3.14) doğruluk, denklem (3.15) F1-skor (F-ölçütü) performans ölçüt değerlerinin nasıl hesaplandığı göstermektedir.

$$Kesinlik = \frac{DP}{DP+YP} \tag{3.12}$$

Kesinlik - Bulma (Precision), alınan tüm örnekler arasındaki ilgili örneklerin oranıdır.

$$Hassasiyet = \frac{DP}{DP+YN} \tag{3.13}$$

Hassasiyet - Tutturma (Recall), gerçekte alınan ilgili örneklerin toplam miktarının kesiridir.

$$Doğruluk = \frac{DP+DN}{DP+FP+DN+YN} \tag{3.14}$$

Doğruluk (Accuracy) tüm veri kümesi içinde, doğru şekilde tahmin edilen istenmeyen e-posta mesajlarının oranıdır.

$$F1-skor = \frac{2*Kesinlik*Hassasiyet}{Kesinlik+Hassasiyet} \tag{3.15}$$

Kesinlik ve hassasiyet ölçütleri kendi başlarına anlamlı bir karşılaştırma sonucu vermek için yeterli değildir. F-ölçütü bu amaç için tanımlanmıştır. Doğru sınıfa atanan örnek sayısı ve yanlış sınıfa atılan örnek sayısını veren F-ölçütü performans metriği kullanılmıştır. Yani F-ölçütü içinde hem kesinlik hem de hassasiyet performans metriklerini içinde bulundurur. F-ölçütü kesinlik ve hassasiyet değerlerindeki aşırılıkları cezalandırır. F-ölçütü, kesinlik ve hassasiyetin harmonik ortalamasıdır.

# 4. BULGULAR VE TARTIŞMA

Bu çalışma kapsamında yapılan çalışmalarla yeni bir Türkçe e-posta veri kümesi olan "TRHamSpamEmail" veri kümesi oluşturulmuştur. "TRHamSpamEmail" veri kümesi üzerinde farklı öznitelik seçim yöntemleri kullanılarak makine öğrenmesi algoritmaları ile test edilmiştir. Bu çalışmaya göre, CHI öznitelik seçim yöntemi Rastgele Orman ve Naive Bayes algoritması ile 0,748 F-ölçütü başarıma ulaşmıştır. Herhangi bir öznitelik seçimi yapılmadan tüm özniteliklerin kullanılması ile elde edilen sınıflandırma başarıları da verilmiştir. Öznitelik seçimi yapılmadan "TRHamSpamEmail" veri kümesi üzerinde RF algoritması ile başarım sonucu 0,535 F-ölçütü olarak elde edilmiştir (Eryılmaz vd, 2020b). Yeni oluşturulan bu e-posta veri kümesinin boyutları artırılarak derin öğrenme ile çıkan sonuçlar ileride yapılacak çalışmalarda değerlendirilecek olup bu tez çalışmasına dâhil edilmemiştir.

Derin öğrenme modelleri RNN, LSTM, GRU, BLSTM, BERT, DistilBERT ile yapılan deneyler sonucunda istenmeyen e-posta tespit sonuçları ise aşağıda verilmiştir.

Modellerin başarımının değerlendirilmesi ve aşırı uyumu engellemek için K katlamalı çapraz doğrulama yöntemi kullanılmıştır. K-Fold Cross Validation (k sayısı kadar çapraz doğrulama), sınıflandırma modellerinin değerlendirilmesi ve modelin eğitilmesi için veri setini parçalara ayırma yöntemlerinden biridir. k sayısı kadar çapraz doğrulama, veriyi belirlenen bir k sayısına göre eşit parçalara böler, her bir parçanın hem eğitim hem de test için kullanılmasını sağlar. Bu sayede dağılım ve parçalanmadan kaynaklanan sapma ve hataları azaltılmış olur. Bunun yanında modeli k kadar eğitmek ve test etmek gibi ilave bir veri işleme gücü ve zamanı alır. Bu durum eğitim ve testi kısa süren küçük ve orta hacimli veriler için sorun olmasa da büyük hacimli veri kümelerinde hesaplama ve zaman yönünden maliyetli olabilmektedir.

Veri kümemizde 800 adet e-posta %80 eğitim, %20 test için ayrılmıştır. K sayısı 5 seçildiğinde; k=1 iken 1-160 arası e-posta kümesi test, 161-800 arası e-posta eğitim için ayrılır. K=2 iken 161-320 arası e-posta test için geri kalan tüm e-posta kümesi eğitim için seçilmiş olur. K=3 iken 321-480 arası e-posta test geri kalan tüm e-posta kümesi eğitim için, k=4 iken 481-640 arası e-posta test geri kalan tümü eğitim, k=5 iken 641-800 arası e-posta test geri kalan eğitim için değerlendirilir. Bu şekilde k=5 olana kadar aynı işlemler tekrarlanır. Her turda elde edilen sonuç toplanıp k sayısına yani 5 sayısına bölünür. Çıkan sonuç her bir derin öğrenme modelinin performans

sonucunu verir. Stratified (Katmanlı) K kat çapraz doğrulama yöntemi SimpleRNN, LSTM, GRU, BLSTM modelleri üzerinde denenmiş olup sonuçlar her bir model için performans sonuçları ayrı ayrı verilmiştir.

Toplam veri kümesi %80 eğitim %20 ise test kümesi olarak kullanılmıştır. Yani 160 e-posta test kümesi için ayrılmıştır. Veri kümesi random.seed ve shuffle ile meşru ve istenmeyen e-postalar karıştırılmıştır. Eğitilen veri kümesindeki eğitim sayısı 512, geçerleme 128 olarak seçilmiştir. 17445 kelime bulunmuş olup, data shape: (800, 250) şeklindedir. Her belgede yani e-postada 250 kelime görüldükten sonra kelimelerin diğerleri kesilmiştir. Sözlük olarak 10000 kelime (num_words= max_features) kullanılmıştır.

Gömülü (Embedding) katman, SimpleRNN, LSTM, GRU ve BLSTM ve Dense gizli katmanlarında 32 nöron kullanılmıştır. Dense ara katmanında aşırı uydurmayı önlemek için "relu" aktivasyon fonksiyonu kullanılmıştır. Ayrıca yine aşırı uyumun önüne geçmek için seyreltme katmanı (Dropout) 0.2 seçilmiştir.

Tamamen bağlı son Dense katmanda; sınıflandırılacak yalnızca iki sınıf (meşru veya istenmeyen posta) olduğundan, yalnızca tek bir çıktı nöronu kullanılmıştır. Sigmoid aktivasyon fonksiyonu, 0 ile 1 arasındaki olasılıkları verir. Son katmanda "sigmoid" aktivasyon fonksiyonu bu yüzden kullanılmıştır. Çekirdek düzenleyici (kernel_regularizer) "l2" seçilmiştir.

Derleme aşamasında; ikili çıktı nedeniyle bir kayıp fonksiyonu olarak "binary_crossentropy", optimizasyonun sonuna doğru gradyanlar seyrekleştikçe yerel minimumdan kaçınmak için "adam" optimizer, performans ölçütü olarak doğruluk kullanılmıştır.

Eğitim aşamasında; epok sayısı 100 olarak belirlenmiştir olup "EarlyStopping" ile geçerleme kaybı "val_loss" 3 epok boyunca düşmezse epok sonlanmıştır. Modelin ne zaman aşırı uymaya başladığını görmenin en iyi yollarından biri geçerleme kaybı verilerinin yeniden artmaya başladığı zamandır. Böylelikle "EarlyStopping" parametresi ile aşırı uyum sorunu önlenmeye çalışılmıştır. "Verbose" parametresi 2 seçilerek her epokta kayıp ve doğruluk değerleri yazdırılmıştır. Parti sayısı "batch_size" 60 seçilmiştir.

Modellerin tahmin ve değerlendirme aşamasında; "skylearn" kitaplığı ile karışıklık matrisi ile doğruluk, kesinlik, hassasiyet, F1-skor ve destek değerleri

bulunmuştur. Eğitim doğruluğu ve kaybı ile geçerleme doğruluğu ve kaybı epok sayılarına göre, karışıklık matrisi de gerçek ve tahmini e-posta sayılarına göre çizilmiştir.

Ön eğitimli ve ince ayarlı BERT ve DistilBERT modelleri Türkçe istenmeyen e-posta veri kümesi ile denenmiş çıkan başarım sonuçları da bu bölümde verilmiştir.

Son kısımda, Izgara Arama (Fayed ve Atiya, 2019; Ghawi ve Pfeffer, 2019) ile hiperparametre ince ayarları yapılarak RNN modeli üzerinde en iyi hiperparametreler otomatik seçilmiş ve sonuçlar açıklanmıştır.

### 4.1. RNN Modeli ile İstenmeyen E-posta Tespiti

SimpleRNN modelimiz için sonuçlar şöyledir: Toplam ve eğitilebilir parametre sayısı 323169'dur. SimpleRNN algoritması ile her eğitim için epok başına 30 milisaniye sürmüştür. Test için derin öğrenme modeli tepke süresi 6 milisaniyedir. SimpleRNN izlenen geçerleme kaybı ölçütünün gelişmesinin durduğu epok 10 olmuştur. Tablo 4.1. ile önerilen SimpleRNN derin öğrenme modeli gösterilmiştir.

Tablo 4.1. SimpleRNN derin öğrenme modeli

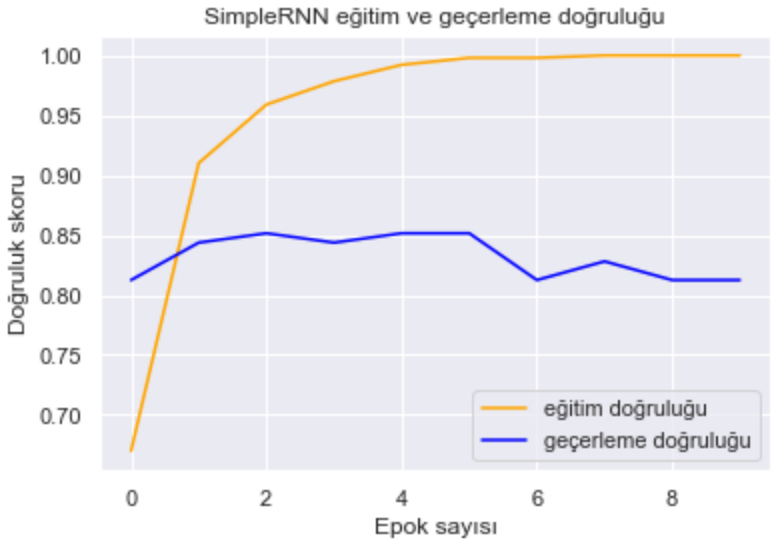
| Katman tipi (Layer) | Çıktı şekli (Output Shape) | Parametre sayısı |
|---|---|---|
| Embedding | (None, None, 32) | 320000 |
| SimpleRNN | (None, 32) | 2080 |
| Dense | (None, 32) | 1056 |
| Dropout | (None, 32) | 0 |
| Dense | (None, 1) | 33 |

Katlama değeri 5 seçilerek SimpleRNN test kaybı 0.4497, doğruluk 0.8, kesinlik 0. 8, hassasiyet 0.8, F1-skor 0. 8001 olarak hesap edilmiştir. Tablo 4.2. ile 5 kat çapraz doğrulamalı SimpleRNN sınıflandırma raporu performans ölçütleri verilmiştir.

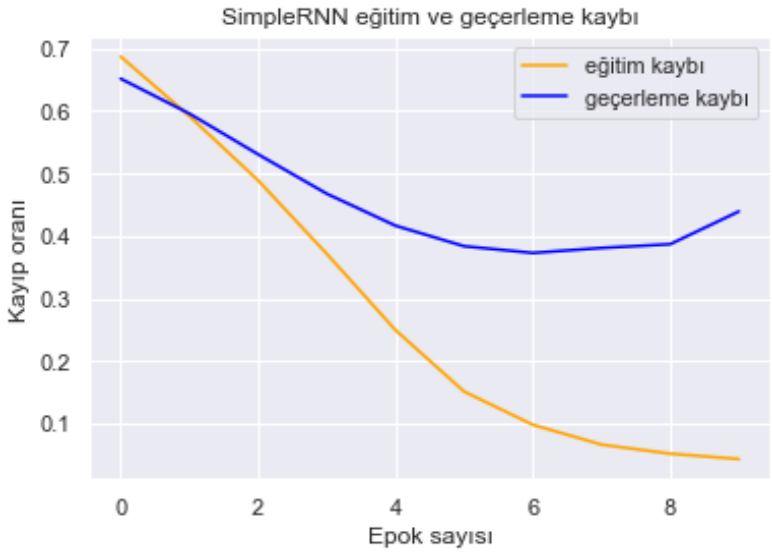
Tablo 4.2. SimpleRNN sınıflandırma raporu

| Tanım | Kesinlik | Hassasiyet | F1-Skor | Destek |
|---|---|---|---|---|
| Normal e-posta | 0.7750 | 0.8158 | 0.7949 | 76 |
| İstenmeyen e-posta | 0.8250 | 0.7857 | 0.049 | 84 |
| macro avg | 0.8000 | 0.8008 | 0.7999 | 160 |
| weighted avg | 0.8012 | 0.8000 | 0.8001 | 160 |

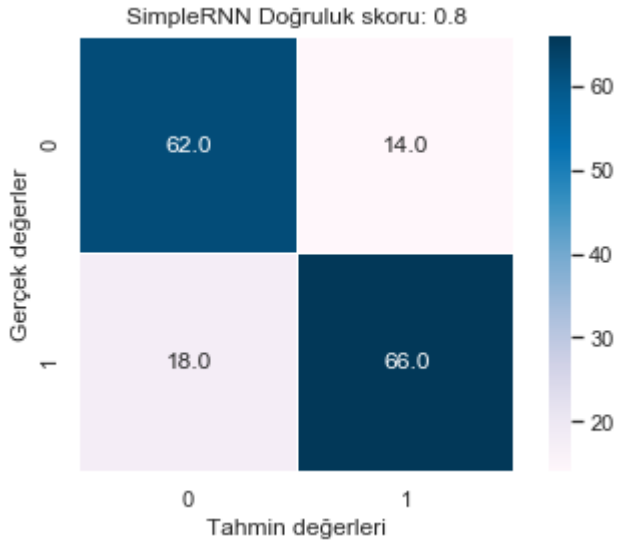
Şekil 4.1. ile k=5 katlamalı çapraz doğrulama ile SimpleRNN modeli doğruluk skoru, Şekil 4.2. ile kayıp oranı ve Şekil 4.3. ile karışıklık matrisi verilmiştir.




Şekil 4.1. SimpleRNN eğitim ve geçerleme doğruluğu




Şekil 4.2. SimpleRNN eğitim ve geçerleme kaybı




Şekil 4.3. SimpleRNN karışıklık matrisi

### 4.2. LSTM Modeli ile İstenmeyen E-posta Tespiti

LSTM modelimiz için sonuçlar şöyledir: Toplam ve eğitilebilir parametre sayısı 329409'dur. LSTM algoritması ile her eğitim için epok başına ortalama 65 milisaniyedir. LSTM izlenen geçerleme kaybı ölçütünün gelişmesi durduğu epok 17 olmuştur. Test için derin öğrenme modeli tepke süresi 11 milisaniyedir. Tablo 4.3. ile önerilen LSTM derin öğrenme modeli gösterilmiştir.

Tablo 4.3. LSTM derin öğrenme modeli

| Katman tipi | Çıktı şekli | Parametre sayısı |
|---|---|---|
| Embedding | (None, None, 32) | 320000 |
| LSTM | (None, 32) | 8320 |
| Dense | (None, 32) | 1056 |
| Dropout | (None, 32) | 0 |
| Dense | (None, 1) | 33 |

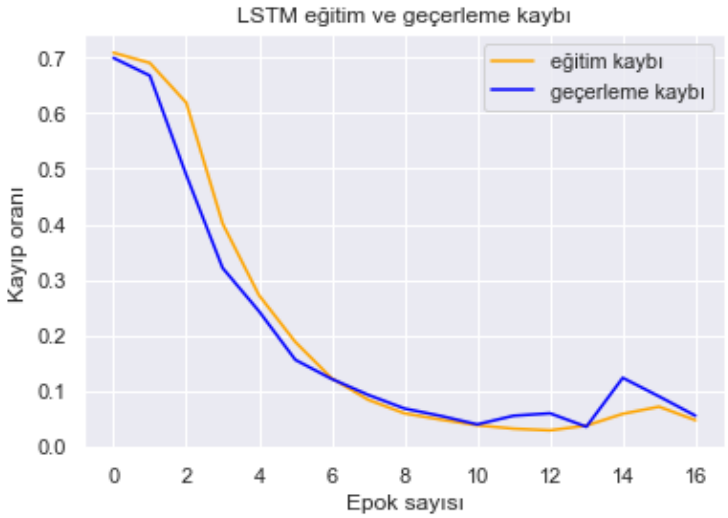
K=5 katlamalı çapraz doğrulama ile LSTM test kaybı 0.0509, doğruluk 0.9938, kesinlik 0.9938, hassasiyet 0.9938, F1-skor 0.9938 olarak hesaplanmıştır. Tablo 4.4. ile 5 kat çapraz doğrulamalı LSTM sınıflandırma raporu performans ölçütleri verilmiştir.

Tablo 4.4. LSTM sınıflandırma raporu

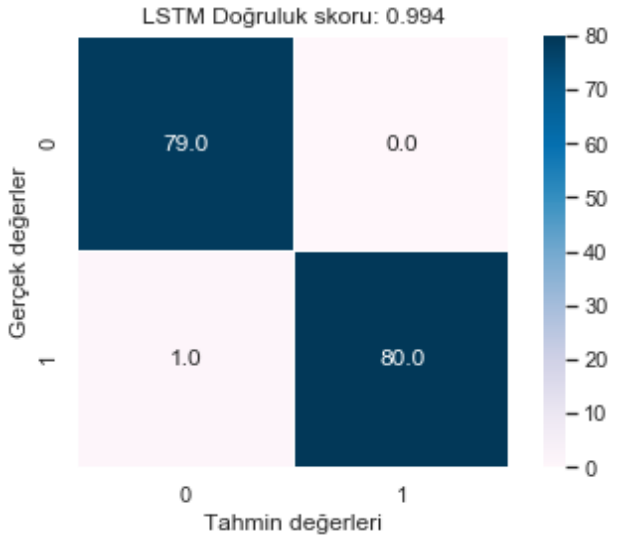
| Tanım | Kesinlik | Hassasiyet | F1-Skor | Destek |
|---|---|---|---|---|
| Normal e-posta | 0.9875 | 1.0 | 0.9937 | 79 |
| İstenmeyen e-posta | 1.0 | 0.9877 | 0.9938 | 81 |
| macro avg | 0.9938 | 0.9938 | 0.9937 | 160 |
| weighted avg | 0.9938 | 0.9938 | 0.9938 | 160 |

Şekil 4.4. ile k=5 katlamalı çapraz doğrulama ile LSTM modeli doğruluk skoru, Şekil 4.5. ile kayıp oranı ve Şekil 4.6. ile karışıklık matrisi verilmiştir.


Şekil 4.4. LSTM eğitim ve geçerleme doğruluğu




Şekil 4.5. LSTM eğitim ve geçerleme kaybı




Şekil 4.6. LSTM karışıklık matrisi

### 4.3. GRU Modeli ile İstenmeyen E-posta Tespiti

GRU modelimiz için sonuçlar şöyledir: Toplam ve eğitilebilir parametre sayısı 327425'tir. GRU algoritması ile her eğitim için epok başına ortalama 64 milisaniyedir. GRU izlenen geçerleme kaybı ölçütünün gelişmesi durduğu epok 53 olmuştur. Test için derin öğrenme modeli tepke süresi 8 milisaniyedir. Tablo 4.5. ile önerilen GRU derin öğrenme modeli gösterilmiştir.

Tablo 4.5. GRU derin öğrenme modeli

| Katman tipi | Çıktı şekli | Parametre sayısı |
|---|---|---|
| Embedding | (None, None, 32) | 320000 |
| GRU | (None, 32) | 6336 |
| Dense | (None, 32) | 1056 |
| Dropout | (None, 32) | 0 |
| Dense | (None, 1) | 33 |

K=5 katlamalı çapraz doğrulama ile GRU test kaybı 0.0541, doğruluk 0.9875, kesinlik 0.9875, hassasiyet 0.9875, F1-skor 0.9875 olarak hesaplanmıştır. Tablo 4.6. ile 5 kat çapraz doğrulamalı GRU sınıflandırma raporu performans ölçütleri verilmiştir.

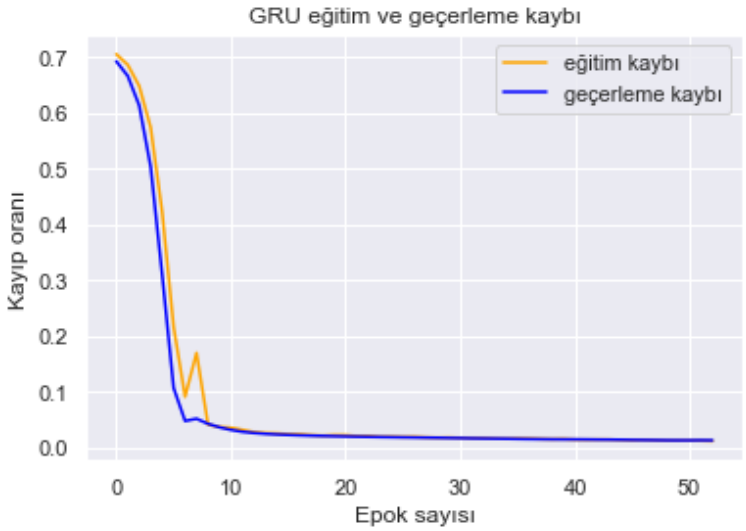
Tablo 4.6. GRU sınıflandırma raporu

| Tanım | Kesinlik | Hassasiyet | F1-Skor | Destek |
|---|---|---|---|---|
| Normal e-posta | 1.0 | 0.9756 | 0.9877 | 82 |
| İstenmeyen e-posta | 0.9750 | 1.0 | 0.9873 | 78 |
| macro avg | 0.9875 | 0.9878 | 0.9875 | 160 |
| weighted avg | 0.9878 | 0.9875 | 0.9875 | 160 |

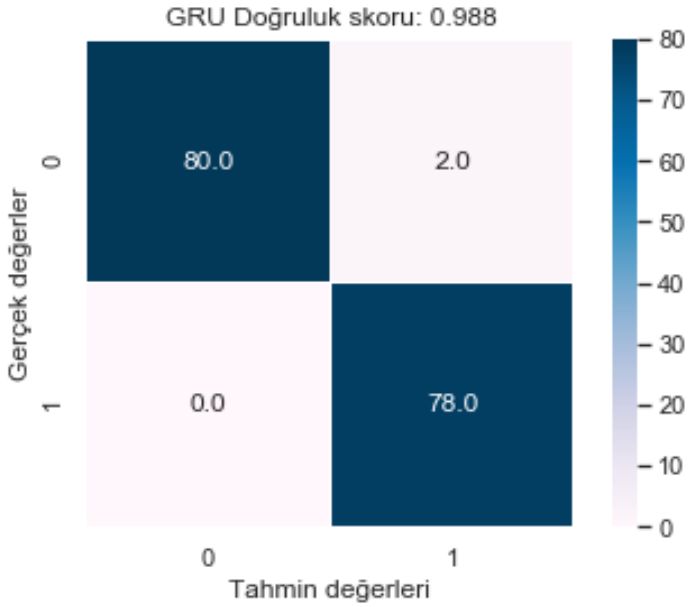
Şekil 4.7. ile k=5 katlamalı çapraz doğrulama ile GRU modeli doğruluk skoru,

Şekil 4.8. ile kayıp oranı ve Şekil 4.9. ile karışıklık matrisi verilmiştir.


Şekil 4.7. GRU eğitim ve geçerleme doğruluğu




Şekil 4.8. GRU eğitim ve geçerleme kaybı




Şekil 4.9. GRU karışıklık matrisi

### 4.4. BLSTM Modeli ile İstenmeyen E-posta Tespiti

BLSTM modelimiz için sonuçlar şöyledir: Toplam ve eğitilebilir parametre sayısı 338753'tür. BLSTM algoritması ile her eğitim için epok başına ortalama 82 milisaniyedir. BLSTM izlenen geçerleme kaybı ölçütünün gelişmesi durduğu epok 13 olmuştur. Test için derin öğrenme modeli tepke süresi 12 milisaniyedir. Tablo 4.3. ile önerilen BLSTM derin öğrenme modeli gösterilmiştir.

Tablo 4.7. BLSTM derin öğrenme modeli

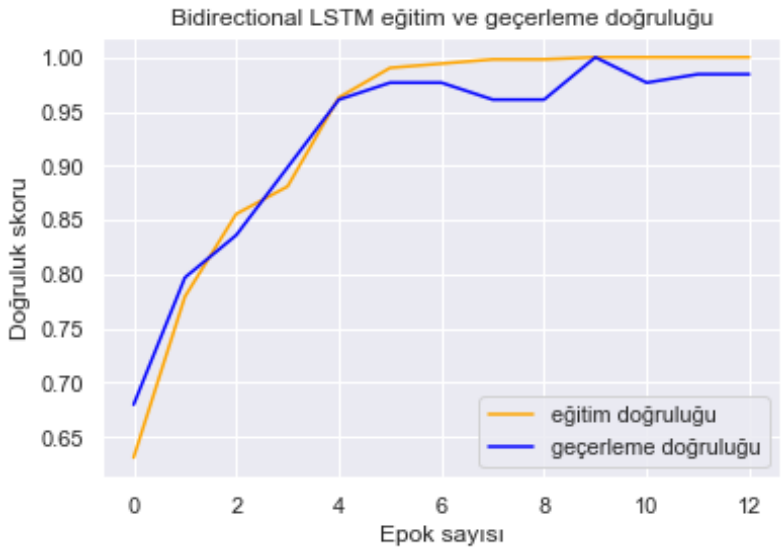
| Katman tipi | Çıktı şekli | Parametre sayısı |
|---|---|---|
| Embedding | (None, None, 32) | 320000 |
| Bidirectional LSTM | (None, 64) | 16640 |
| Dense | (None, 32) | 2080 |
| Dropout | (None, 32) | 0 |
| Dense | (None, 1) | 33 |

K=5 katlamalı çapraz doğrulama ile çift yönlü BLSTM test kaybı 0.0373, doğruluk 0.9938, kesinlik 0.9938, hassasiyet 0.9938, F1-skor 0.9938 sonuçları bulunmuştur. Tablo 4.8. ile 5 kat çapraz doğrulamalı BLSTM sınıflandırma raporu performans ölçütleri verilmiştir.

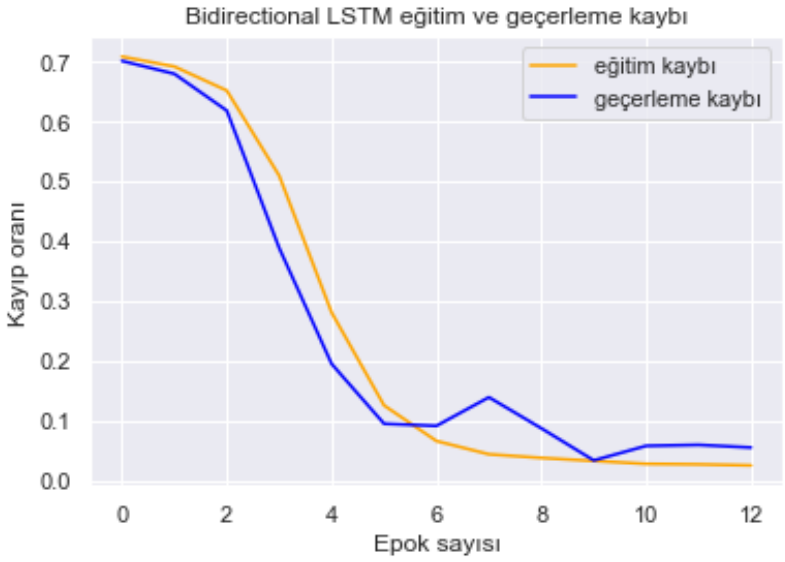
Tablo 4.8. BLSTM sınıflandırma raporu

| Tanım | Kesinlik | Hassasiyet | F1-Skor | Destek |
|---|---|---|---|---|
| Normal e-posta | 0.9875 | 1.0 | 0.9937 | 79 |
| İstenmeyen e-posta | 1.0 | 0.9877 | 0.9938 | 81 |
| macro avg | 0.9938 | 0.9938 | 0.9937 | 160 |
| weighted avg | 0.9938 | 0.9938 | 0.9938 | 160 |

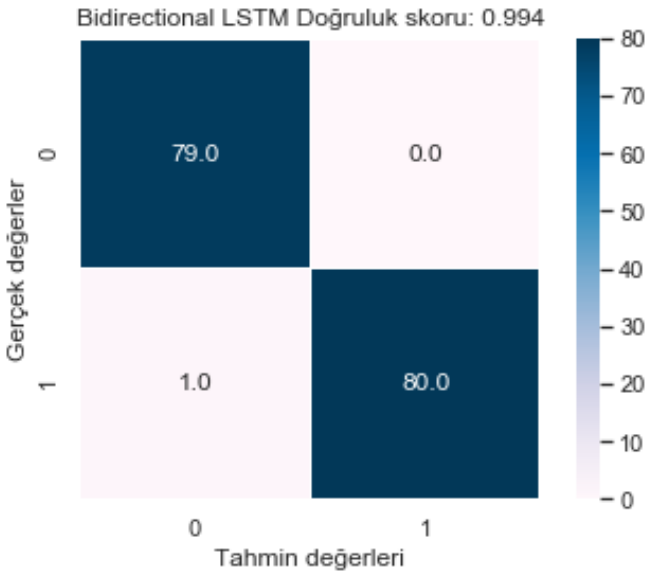
Şekil 4.10. ile k=5 katlamalı çapraz doğrulama ile BLSTM modeli doğruluk skoru, Şekil 4.11. ile kayıp oranı ve Şekil 4.12. ile karışıklık matrisi verilmiştir.




Şekil 4.10. BLSTM eğitim ve geçerleme doğruluğu




Şekil 4.11. BLSTM eğitim ve geçerleme kaybı




Şekil 4.12. BLSTM karışıklık matrisi

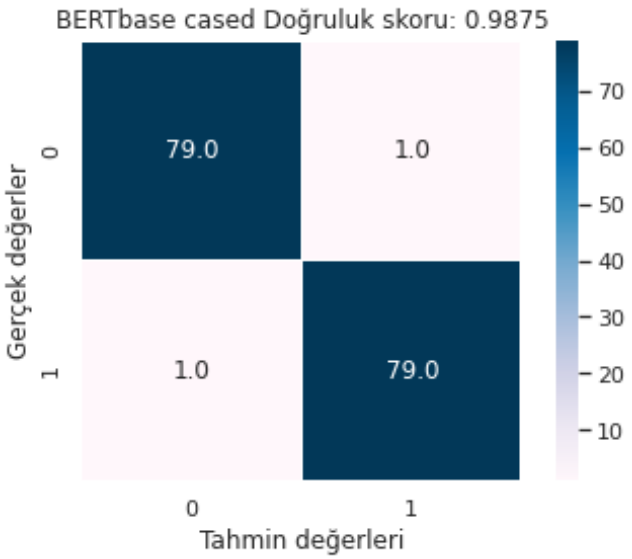
### 4.5. BERT ve DistilBERT Modeli ile İstenmeyen E-posta Tespiti

BERT için filtrelenmiş ve cümle bölümlendirilmiş bir versiyon olan Türkçe OSCAR dil külliyatı ve Kemal Oflazer tarafından sağlanan özel bir külliyat ile bir Wikipedia dökümü olan çeşitli OPUS külliyatlarının birleşiminden oluşan 35 GB'lık eğitilmiş külliyat kullanılmıştır. Tüm BERT modellerini barındıran huggingface.co sitesinde bulunan bmdz/bert-base-turkish-cased modeli ile çalışılmıştır. BERT derin öğrenme modeli için doğruluk 0.9875, kesinlik 0.9875, hassasiyet 0.9875, F1-skor 0.9875 ölçülmüştür. Tablo 4.9. ile BERT modeli sınıflandırma raporu verilmiştir.

Tablo 4.9. BERT modeli sınıflandırma raporu

| Tanım | Kesinlik | Hassasiyet | F1-Skor | Destek |
|---|---|---|---|---|
| Normal e-posta | 0.9875 | 0.9875 | 0.9875 | 80 |
| İstenmeyen e-posta | 0.9875 | 0.9875 | 0.9875 | 80 |
| macro avg | 0.9875 | 0.9875 | 0.9875 | 160 |
| weighted avg | 0.9875 | 0.9875 | 0.9875 | 160 |

Şekil 4.13. ile BERT modeli karışıklık matrisi gösterilmiştir.



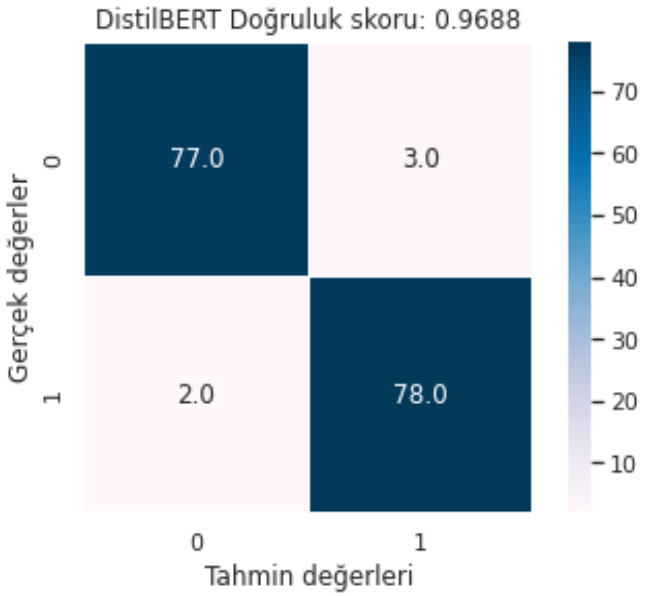

Şekil 4.13. BERT modeli karışıklık matrisi

DistilBERT (Sanh vd, 2019) için Türkçe BERT'nin damıtılmış bir versiyonu olan DistilBERTurk, BERTurk'ün eğitiminde kullanılan 7GB'lık orijinal eğitim verisi olan Huggingface.co sitesinde bulunan *dbmdz/distilbert-base-turkish-cased* modeli kullanılmıştır. DistilBERT derin öğrenme modeli doğruluk 0.96875, kesinlik 0.96296, hassasiyet 0.975, F1-skor 0.96894 ölçülmüştür. Tablo 4.10. DistilBERT modeli sınıflandırma raporu verilmiştir.

Tablo 4.10. DistilBERT modeli sınıflandırma raporu

| Tanım | Kesinlik | Hassasiyet | F1-Skor | Destek |
|---|---|---|---|---|
| Normal e-posta | 0.9747 | 0.9625 | 0.9689 | 80 |
| İstenmeyen e-posta | 0.9630 | 0.9750 | 0.9689 | 80 |
| macro avg | 0.9688 | 0.9688 | 0.9687 | 160 |
| weighted avg | 0.9688 | 0.9688 | 0.9687 | 160 |

Şekil 4.14. ile DistilBERT modeli karışıklık matrisi gösterilmiştir.




Şekil 4.14. DistilBERT modeli karışıklık matrisi

Yukarıda da görüldüğü üzere BERT modeli DistilBERT modelinden daha yüksek başarım sonuçlarına ulaşmıştır. Bunun başlıca nedeni BERT için önceden eğitilmiş modelin külliyatının daha büyük olmasıdır. Türkçe istenmeyen e-posta tespitinde DistilBERT modeli ile %96.88 doğruluk elde edilirken, BERT modeli ile %98.75 doğruluk elde edilmiştir.

### 4.6. Hiperparametre İnce Ayar ile Bazı Parametrelerin Seçimi

Derin öğrenme modelindeki sinir ağı modelini eğitmek için kullanılan farklı parametrelere hiperparametreler denir. Bu hiperparametreler, optimize edilmiş bir modelle sonuçlanan bir sinir ağının performansını iyileştirmek için düğmeler gibi hassas bir şekilde ayarlanır. Hiperparametreler arasında, gizli katman sayısı, gizli katmandaki birim veya düğüm sayısı, öğrenme oranı, seyreltme oranı, epoklar veya yinelemeler, sgd, adam, nadam, adagrad, rmsprop vb. gibi optimize ediciler, relu, sigmoid, tanh, softmax vb. gibi aktivasyon fonksiyonları, parti boyutu olabilmektedir. Hiperparametre ince ayarı, en iyi model performansını verecek bir derin öğrenme modelinin optimize ediciler, aktivasyon fonksiyonu, öğrenme katsayısı, seyreltme oranları vb. gibi hiperparametrelerin değerini bulma işlemidir.

Hiperparametre ince ayarı manuel arama, ızgara arama, rastgele arama (Bergstra ve Bengio, 2012), Bayes optimizasyon (Klein vd, 2017) şeklinde yapılabilmektedir. Bu bölümde, belirtilen hiperparametrelerin tüm olası kombinasyonlarının kapsamlı bir araştırması ve ince ayarlanması Izgara arama ile gerçekleştirilmiş ve elde edilen ince

ayarlı hiperparametre sonuçları verilmiştir.

En iyi sonuç veren optimizer için; epok 10, parti boyutu 10, gizli ara katman aktivasyon fonksiyonu “relu”, son katman aktivasyon fonksiyonu “sigmoid”, kernel_regularizer = “l2”, loss=”binary_crossentropy” seçimleri ile 5 kat çapraz doğrulamalı Izgara arama metodu SimpleRNN modeli üzerinde uygulanmıştır. %96.56 başarım ve 0.0145 standart sapma ile “nadam” en iyi sonucu veren optimizer fonksiyonu olmuştur. Diğer optimizer seçimlerine göre sonuçlar Tablo 4.11’de verilmiştir.

Tablo 4.11. İnce ayarlı optimizer fonksiyonu seçimi

| Ortalama Test Başarımı (means) | Standart Sapma (stds) | Optimizer |
|---|---|---|
| **0.965625** | **0.014490** | **Nadam** |
| 0.943750 | 0.029398 | Adamax |
| 0.942187 | 0.020729 | RMSprop |
| 0.915625 | 0.074346 | Adam |
| 0.823438 | 0.088333 | Adagrad |
| 0.814063 | 0.076769 | SGD |
| 0.470313 | 0.074674 | Adadelta |

En iyi sonuç veren seyreltme oranı ve en iyi ağırlık kısıtlaması seçimleri; weight_constraint = [1, 2, 3, 4, 5] ve dropout_rate = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] oranları üzerinden karşılaştırmalı olarak yapılmıştır. Gizli ara katman aktivasyon fonksiyonu “relu”, son katman aktivasyon fonksiyonu “sigmoid”, kernel_regularizer = “l2”, kernel_initializer=“uniform”, loss = “binary_crossentropy”, optimizer = “nadam”, 5 kat çapraz doğrulama, 10 epok ve hiperparametre seçimleri ile Izgara Arama metodu SimpleRNN modeli üzerinde uygulanmıştır. %97.5 başarım, 0.0134 standart sapma oranı ile dropout rate 0.4, weight constraint 2 olarak hesaplanmıştır.

Hiperparametre ince ayarlarına göre Izgara Arama metodu ile bazı seyreltme oranı ve ağırlık kısıtlaması başarım sonuçları Tablo 4.12’de verilmiştir.

Tablo 4.12’de daha fazla hiperparametre ayarı yapılabilmektedir. Bu sebeple seyreltme oranı ve ağırlık kısıtlaması ile yapılan diğer tüm deney sonuçları Ekler sayfasında Ek-1 bölümünde verilmiştir.

Tablo 4.12. İnce ayarlı seyreltme oranı ve ağırlık kısıtlaması

| Ortalama Test Başarımı | Standart Sapma | Seyreltme oranı | Ağırlık kısıtlaması |
|---|---|---|---|
| **0.975000** | **0.013441** | **0.4** | **2** |
| 0.973437 | 0.010597 | 0.1 | 3 |
| 0.973437 | 0.012694 | 0.8 | 4 |
| 0.971875 | 0.020131 | 0.2 | 5 |
| 0.971875 | 0.012694 | 0.4 | 1 |
| 0.971875 | 0.007967 | 0.6 | 4 |

En iyi sonuç veren parti boyutu (batch_size) ve en iyi epok seçimleri; batch_size = [10, 20, 40, 60, 80, 100] ve epochs = [10, 50, 100] ön seçimleri yapılarak ince ayarlanmıştır. Gizli ara katman aktivasyon fonksiyonu "relu", son katman aktivasyon fonksiyonu "sigmoid", seyreltme oranı 0.4, kernel_regularizer = "l2", loss = "binary_crossentropy", optimizer = "nadam", 5 kat çapraz doğrulama hiperparametre seçimleri ile Izgara Arama metodu SimpleRNN modeli üzerinde uygulanmıştır. %96.72 başarım, 0.0134 standart sapma oranı ile parti boyutu 40, epok 10 olarak hesaplanmıştır. Hiperparametre ince ayarlarına göre Izgara Arama metodu ile parti boyutu ve epok sayılarına göre bazı başarım sonuçları Tablo 4.13'te verilmiştir. Tablo 4.13'ün tümü Ek-2 bölümünde verilmiştir.

Tablo 4.13. İnce ayarlı parti boyutu ve epok seçimi

| Ortalama Test Başarımı | Standart Sapma | Parti boyutu | Epok |
|---|---|---|---|
| **0.967187** | **0.013441** | **40** | **10** |
| 0.964063 | 0.021875 | 20 | 50 |
| 0.959375 | 0.019390 | 20 | 100 |
| 0.948438 | 0.033730 | 60 | 10 |
| 0.942187 | 0.025958 | 10 | 100 |
| 0.939063 | 0.045661 | 100 | 10 |
| 0.929688 | 0.059498 | 60 | 100 |

En iyi sonuç veren katman ağırlığı başlatıcı (kernel_initializer) deneyi; epok 10, parti boyutu 10, gizli ara katman aktivasyon fonksiyonu "relu", son katman aktivasyon fonksiyonu "sigmoid", loss = "binary_crossentropy", optimizer = "nadam", 5 kat çapraz doğrulama seçimleri ile Izgara Arama metodu SimpleRNN modeli üzerinde uygulanmıştır. %97.66 başarım ve 0.0121 standart sapma ile "uniform" en iyi sonucu veren katman ağırlığı başlatıcı olmuştur.

Kernel ağırlığı başlatıcı, ilk rastgele ağırlıklarını ayarlama yolunu tanımlar. Hiperparametre ince ayarlarına göre Izgara Arama metodu ile diğer katman ağırlığı başlatıcı başarım sonuçları Tablo 4.14'te verilmiştir.

Tablo 4.14. İnce ayarlı katman ağırlığı başlatıcı seçimi

| Ortalama Test Başarımı | Standart Sapma | Katman ağırlığı başlatıcı |
|---|---|---|
| **0.976562** | **0.012103** | **uniform** |
| 0.965625 | 0.018222 | zero |
| 0.960938 | 0.031638 | normal |
| 0.946875 | 0.037760 | lecun_uniform |
| 0.946875 | 0.019390 | he_uniform |
| 0.943750 | 0.040565 | glorot_normal |
| 0.932813 | 0.044634 | glorot_uniform |
| 0.878125 | 0.048058 | he_normal |

Tamamen bağlı ara katman için en iyi sonuç veren nöron sayısı deneyi; neurons = [1, 5, 10, 15, 20, 25, 30] sayıları arasından karşılaştırmalı olarak yapılmıştır. Hiperparametre ince ayarları epok 10, parti boyutu 40, gizli ara katman aktivasyon fonksiyonu “relu”, son katman aktivasyon fonksiyonu “sigmoid”, loss = “binary_crossentropy”, optimizer fonksiyonu “nadam”, seyreltme oranı 0.4, weight_constraint 2, kernel_initializer “uniform”, 5 kat çapraz doğrulama seçimleri ile Izgara Arama metodu SimpleRNN modeli üzerinde uygulanmıştır. %94.22 başarım ve 0.0323 standart sapma ile 1 nöron en iyi sonucu vermiştir. Hiperparametre ince ayarlarına göre Izgara Arama metodu ile diğer tamamen bağlı ara katman nöron sayısı başarım sonuçları Tablo 4.15.’te verilmiştir.

Tablo 4.15. İnce ayarlı tamamen bağlı ara katman nöron sayısı

| Ortalama Test Başarımı | Standart Sapma | Tamamen bağlı ara katman nöron sayısı |
|---|---|---|
| **0.942187** | **0.032250** | **1** |
| 0.940625 | 0.031093 | 10 |
| 0.940625 | 0.038464 | 20 |
| 0.935937 | 0.051206 | 15 |
| 0.907813 | 0.088167 | 5 |
| 0.887500 | 0.077277 | 25 |
| 0.850000 | 0.141697 | 30 |

En iyi sonuç veren öğrenme katsayısı deneyi; learn_rate = [0.001, 0.01, 0.1, 0.2,

0.3] oranları ile nadam optimizer fonksiyonu için karşılaştırmalı olarak yapılmıştır. Hiperparametre ince ayarları epok 10, parti boyutu 10, gizli ara katman aktivasyon fonksiyonu “relu”, son katman aktivasyon fonksiyonu “sigmoid”, loss = “binary_crossentropy”, kernel_regularizer “l2”, 5 kat çapraz doğrulama seçimleri ile Izgara Arama metodu SimpleRNN modeli üzerinde uygulanmıştır. %95.5 başarım ve 0.0303 standart sapma ile öğrenme katsayısı 0.001 olarak en iyi sonucu vermiştir. Hiperparametre ince ayarlarına göre Izgara Arama metodu ile diğer öğrenme katsayısı başarım sonuçları Tablo 4.16’da verilmiştir. Öğrenme katsayısı arttıkça modelin başarım oranının düştüğü, standart sapmanın arttığı açıkça görülmektedir.

Tablo 4.16. İnce ayarlı öğrenme katsayısı seçimi

| Ortalama Test Başarımı | Standart Sapma | Öğrenme Katsayısı |
|---|---|---|
| **0.950000** | **0.030298** | **0.001** |
| 0.932813 | 0.048563 | 0.01 |
| 0.831250 | 0.086235 | 0.1 |
| 0.782813 | 0.074837 | 0.2 |
| 0.712500 | 0.044028 | 0.3 |

Tamamen bağlı ara katman için en iyi sonuç veren aktivasyon fonksiyonu deneyi; activation = ['softmax', 'softplus', 'softsign', 'relu', 'tanh', 'sigmoid', 'hard_sigmoid', 'linear' , 'selu', 'elu'] fonksiyonları arasından karşılaştırmalı olarak yapılmıştır. Hiperparametre ince ayarları epok 10, parti boyutu 10, loss = “binary_crossentropy”, optimizer fonksiyonu “nadam”, seyreltme oranı 0.4 seçilmiştir. Ara katman için 32 nöron seçilmiştir. Son katman için; kernel_initializer “uniform”, aktivasyon fonksiyonu sigmoid, kernel_regularizer “l2” seçilmiştir. 5 kat çapraz doğrulama ile Izgara Arama metodu SimpleRNN modeli üzerinde uygulanmıştır. Tablo 4.17.’de, %97.50 başarım ile “softsign” aktivasyon fonksiyonu ara katman için en iyi sonucu vermiştir

Tablo 4.17. İnce ayarlı tamamen bağlı ara katman aktivasyon fonksiyonu seçimi

| Ortalama Test Başarımı | Standart Sapma | Tamamen bağlı ara katman aktivasyon fonksiyonu |
|---|---|---|
| **0.975000** | **0.010364** | **softsign** |
| 0.973437 | 0.017539 | softplus |
| 0.973437 | 0.007967 | tanh |
| 0.971875 | 0.013622 | softmax |
| 0.967187 | 0.007655 | hard_sigmoid |
| 0.959375 | 0.016683 | relu |
| 0.954688 | 0.036443 | linear |
| 0.946875 | 0.035424 | elu |
| 0.943750 | 0.040263 | selu |
| 0.915625 | 0.068179 | sigmoid |

En iyi sonuç veren aktivasyon fonksiyonu için; epok 10, parti boyutu 10, kernel_initializer = “uniform”, kernel_regularizer = “l2”, loss = “binary_crossentropy”, optimizer = “nadam”, 5 kat çapraz doğrulama seçimleri ile Izgara arama metodu SimpleRNN modeli üzerinde uygulanmıştır. %97.19 başarım ve 0.0094 standart sapma ile “hard_sigmoid” en iyi sonucu veren aktivasyon fonksiyonu olmuştur. Hiperparametre ince ayarlarına göre Izgara Arama metodu ile diğer aktivasyon fonksiyonlarının başarım sonuçları Tablo 4.18’de verilmiştir.

Tablo 4.18. İnce ayarlı aktivasyon fonksiyonu seçimi

| Ortalama Test Başarımı | Standart Sapma | Aktivasyon |
|---|---|---|
| **0.971875** | **0.009375** | **hard_sigmoid** |
| 0.970313 | 0.010364 | sigmoid |
| 0.959375 | 0.020611 | softplus |
| 0.928125 | 0.035767 | elu |
| 0.895312 | 0.036510 | tanh |
| 0.871875 | 0.085952 | relu |
| 0.856250 | 0.136806 | selu |
| 0.715625 | 0.165772 | linear |
| 0.692187 | 0.215069 | softsign |
| 0.501563 | 0.038081 | softmax |

# 5. SONUÇ VE ÖNERİLER

Bu tez çalışmasında istenmeyen e-postaların ayrıştırılması için derin öğrenme yöntemleri kullanılmıştır. Tez çalışması kapsamında istenmeyen e-postaların tespiti için geliştirilen derin öğrenme modelleri, Tensorflow tabanlı Keras kütüphanesi ile masaüstü uygulaması Spyder ve internet tabanlı Google Colaboratory araçları gerçeklenmiştir. Türkçe istenmeyen e-postaların derin öğrenme ile tespit edilmesi için 6 farklı model üzerinde çalışmalar yapılmıştır. Eğitim kümesi her model için %80, test kümesi %20 seçilmiş ve SimpleRNN, LSTM, GRU, BLSTM derin öğrenme modellerinde sınıflandırma başarımlarının değerlendirilmesinde 5 kat çapraz doğrulama kullanılmıştır. Tablo 5.1.'de derin öğrenme modelleri SimpleRNN, LSTM, GRU, BLSTM ile ön eğitimli, ince ayarlı BERT ve DistilBERT derin öğrenme modellerinin başarım skorları ve kayıp oranlarının sonuçları performans karşılaştırılması yapılmıştır.

Tablo 5.1. Derin öğrenme modellerinin performans karşılaştırılması

| Model | Test kaybı | Doğruluk | Kesinlik | Hassasiyet | F1-skor |
|---|---|---|---|---|---|
| SimpleRNN | 0.4497 | 0.8 | 0.8 | 0.8 | 0.8001 |
| **LSTM** | **0.0509** | **0.9938** | **0.9938** | **0.9938** | **0.9938** |
| GRU | 0.0541 | 0.9875 | 0.9875 | 0.9875 | 0.9875 |
| **BLSTM** | **0.0373** | **0.9938** | **0.9938** | **0.9938** | **0.9938** |
| BERT | 0.0480 | 0.9875 | 0.9875 | 0.9875 | 0.9875 |
| DistilBERT | 0.1241 | 0.9688 | 0.9688 | 0.9688 | 0.9687 |

En fazla parametre ince ayarlı BERT modelinde bulunmaktadır. RNN tabanlı modellerde ise en çok parametre sayısına 338753 ile BLSTM sahiptir. Test kaybı 0.0373 ile en az olan model BLSTM'dir. Başarımı en yüksek modeller LSTM ve BLSTM modeli olmuş başarım sonucu %99.38 hesaplanmıştır. En fazla parametreye sahip ince ayarlı BERT modeli %98.75 ile yüksek bir başarıma ulaşmıştır. 5 katlamalı çapraz doğrulama deneyleri Tensorflow tabanlı Keras kütüphanesi ve Spyder aracı ile yapılmıştır. BERT ve DistilBERT modelleri Google Colaboratory ile GPU üzerinde test edilmiştir.

Hiperparametre ince ayar ile bazı parametrelerin seçimi için Google Colabratory üzerinde GPU tabanlı makineler ile yürütülmüştür. Bu algoritmaların ince ayarı için Izgara arama tahminleyici modelleri üzerinde birçok hiperparametre ile deney yapılmıştır. İnce ayar için deneylerde ayırımın daha iyi yapılabilmesi için derin öğrenme modellerinde genel olarak daha az başarım sonucu veren SimpleRNN modeli üzerinde denemeler yapılarak en iyi hiperparametreler bulunmaya çalışılmıştır. Tablo

5.2.'de ince ayarlı hiperparametrelerin başarım sonuçları verilmiştir. Bunlar; optimizer fonksiyonu "nadam", aktivasyon fonksiyonu "sigmoid", parti boyutu 40, epok 10, seyreltme oranı 0.4, ağırlık kısıtlaması 2, katman ağırlığı başlatıcı "uniform" olarak bulunmuştur. Tamamen bağlı ara katman için nöron sayısı 1, nadam optimizer fonksiyonu için öğrenme katsayısı 0.001 hesaplanmıştır. Tamamen bağlı ara katman için en iyi sonuç veren aktivasyon fonksiyonu "softsign" olmuştur. Kaybolan ve patlayan gradyanlar için en iyi çözüm olduğu literatürce görüldüğünden hiperparametre ince ayarlarında ara gizli katmanda diğer başarılı aktivasyon fonksiyonlarından "relu" aktivasyon fonksiyonu kullanılmıştır. Tablo 5.2. ile SimpleRNN modeline göre hiperparametre ince ayar sonucunda hiperparametrelerin en başarılı sonuçları verilmiştir. Bu sonuçlar farklı tahminleyici ve farklı hiperparametre seçimleri ile daha da fazla iyileştirilebilmektedir. Hiperparametrelerin çok az bir değişiklikle çok farklı sonuçları verebileceği unutulmamalıdır.

Tablo 5.2. İnce ayarlı hiperparametrelerin başarım sonuçları

| Hiperparametre | Ortalama Test Başarımı | Standart Sapma |
|---|---|---|
| optimizer= "nadam" | 0.965625 | 0.014490 |
| aktivasyon = "hard_sigmoid" | 0.971875 | **0.009375** |
| kernel_initializer= "uniform" | **0.976562** | 0.012103 |
| gizli ara katman aktivasyonu = "softsign" | 0.975000 | 0.010364 |
| batch_size=40, epochs=10 | 0.967187 | 0.013441 |
| dropout_rate=0.4, weight_constraint=2 | 0.975000 | 0.013441 |
| tamamen bağlı ara katman nöron sayısı= 1 | 0.942187 | 0.032250 |
| öğrenme oranı=0.001 | 0.950000 | 0.030298 |

Bu çalışma ile literatürde zamanla geliştirilen spam algılama sistemlerinin açık problemleri tespit edilmiş bunlarla ilgili çözüm önerileri getirilmiştir. İstenmeyen e-posta tespitinde zamanla başarılı yöntemlerin sayısı artmış ancak bunların çoğu, spam gönderenlerin sürekli olarak spam oluşturma şeklini değiştirmesiyle etkinliğini yitirmiştir. Bu durum sürekli gelişen spam tespit yöntemlerinin ortaya çıkmasını sağlamıştır. Önceleri yapay zekâ tabanlı olmayan yöntemler etkili iken günümüzde makine öğrenmesi algoritmalarının artması ve derin öğrenme modellerinin iyileştirilmesi ile istenmeyen e-posta tespitinde yapay zekâ tabanlı sistemler daha çok

kullanılır hale gelmiştir.

Mevcut hesaplama ve veri miktarındaki artışları kolayca tolere edebilen derin öğrenmeye dayalı yöntemlerin gelecekte spam e-posta tespitinde daha da başarılı olacağı düşünülmektedir. Derin sinir ağları için şu anda geliştirilmekte olan yeni öğrenme algoritmaları ve derin öğrenme mimarilerinin bu ilerlemeyi hızlandıracağı düşünülmektedir. Melez modeller ile de yüksek başarım elde edileceği literatürde görülmektedir. Bu nedenle melez modellerin de veri kümeleri üzerinde başarım sonuçları ölçülmelidir.

Bilginin ve öğrenmenin aktarımı olan Transfer öğrenme modelleri en yeni teknolojiler olduğundan bu alanda daha fazla çalışılması gerekmektedir. Transfer öğrenmenin istenmeyen e-posta tespiti gibi farklı dil modellerinde de kullanımının uygun olacağı değerlendirilmektedir.

Literatürde yapay zekâ tabanlı olan sistemler ile birlikte yapay zekâ tabanlı olmayan spam tespit sistemleri birlikte kullanılmaktadır. İşlemci gücü ile birlikte grafik işlem birimlerinin kapasitelerinin artmasıyla makine öğrenmesi tekniklerinin yanında derin öğrenme yöntemleri de istenmeyen elektronik posta tespitinde daha fazla tercih edilmeye başlanmıştır. Ayrıca kullanıcılar kendi cihazlarındaki tüm e-postalarına erişmeyi tercih ettiğinden kullanıcı davranışı göz önünde bulundurularak kullanıcıya özel olarak tasarlanmış bir spam filtresi düşünülmesi de gerekmektedir.

Bu alanda çalışmak isteyen araştırmacılar için mevcut problemlerden en önemlisi, Türkçe e-postaları içeren veri kümelerinin az veri içermesi ve yetersiz sayıda olmasıdır. Bu tez kapsamında toplam 350 Türkçe e-posta içeren yeni bir veri kümesi oluşturulmuş olup sonraki çalışmalar için veri kümesinin boyutu daha da artırılacaktır. Aynı zamanda bu veri kümesi üzerinde derin öğrenme modellerinin başarım sonuçları da irdelenecektir.

## KAYNAKLAR


Ablel-Rheem, D. M., Ibrahim, A. O., Kasim, S., Almazroi, A. A. and Ismail, M. A. 2020. Hybrid Feature Selection and Ensemble Learning Method for Spam Email Classification. *International Journal,* 9:1.4.

Akinyelu, A. A. and Adewumi, A. O. 2014. Classification of phishing email using random forest machine learning technique. *Journal of Applied Mathematics,* 2014.

Al-Azzawi, F. 2018. Wrapper feature selection approach for spam e-mail filtering. Master Thesis, Erciyes University Graduate school of natural and applied science, Kayseri.

Alkaht, I. J. and Al-Khatib, B. 2016. Filtering spam using several stages neural networks. *Int. Rev. Comp. Softw,* 11:2.

Altunyaprak, C. 2006. Bayes yöntemi kullanarak istenmeyen elektronik postaların filtrelenmesi. Yüksek Lisans Tezi, Muğla Üniversitesi Fen Bilimleri Enstitüsü

Atalay, M. and Çelik, E. 2017. Büyük veri analizinde yapay zekâ ve makine öğrenmesi uygulamaları. *Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi,* 9:22, 155-172.

Ateş, N. 2014. Destek vektör makineleri ve Gauss karışım modeli ile istenmeyen e-postaların tespiti. Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü, 57, Samsun.

avira.com (2019). What is email spam? Retrieved from https://www.avira.com/en/support-what-is-email-spam

Awad, W. and ELseuofi, S. 2011. Machine learning methods for spam e-mail classification. *International Journal of Computer Science & Information Technology (IJCSIT),* 3:1, 173-184.

Bagui, S., Nandi, D., Bagui, S. and White, R. J. (2019). Classifying Phishing Email Using Machine Learning and Deep Learning. 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), IEEE, 1-2.

Bajaj, K. S., Egbufor, F. and Pieprzyk, J. (2011). Critical analysis of spam prevention techniques. 2011 Third International Workshop on Security and Communication Networks (IWSCN), IEEE, 83-87.

Bergstra, J. and Bengio, Y. 2012. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research,* 13:1, 281-305.

Bhagyashri, G., Pratap, H. and Patil, D. 2013. Auto emails classification using Bayesian filter. *International Journal of Advanced technology & Engineering Research,* 3:4.

Bhowmick, A. and Hazarika, S. M. 2016. Machine learning for E-mail spam filtering: review, techniques and trends. *arXiv preprint arXiv:1606.01042*.

Bousquet, O., Bottou, L., Platt, J., Koller, D., Singer, Y. and Roweis, S. 2007. Advances in neural information processing systems.

Bradbury, D. 2014. Can we make email secure? *Network Security*, 2014:3, 13-16.

Caruana, G. and Li, M. 2008. A survey of emerging approaches to spam filtering. *ACM Computing Surveys (CSUR),* 44:2, 1-27.

Chao, W.-L. 2011. Machine learning tutorial. *Digital Image and Signal Processing*.

Che, H., Liu, Q., Zou, L., Yang, H., Zhou, D. and Yu, F. (2017). A content-based phishing Email detection method. 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), IEEE, 415-422.

Chen, J., Fontugne, R., Kato, A. and Fukuda, K. (2014). Clustering spam campaigns with fuzzy hashing. Proceedings of the AINTEC 2014 on Asian Internet Engineering Conference, 66-73.

Chiba, D., Akiyama, M., Yagi, T., Hato, K., Mori, T. and Goto, S. 2018. DomainChroma: Building actionable threat intelligence from malicious domain names. *Computers & Security*, 77, 138-161.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chollet, F. 2015. *Keras Documentation Keras: The Python Deep Learning library*

Choudhary, M. and Dhaka, V. (2013). Automatic E-mails classification using genetic algorithm. Special Conference Issue: National Conference on Cloud Computing and Big Data, Citeseer, 42-49.

Darwish, A. 2018. Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications. *Future Computing and Informatics Journal*, 3:2, 231-246.

Deniz, E., Erbay, H. and Coşar, M. (2019). Classification of Turkish E-Mails with Doc2Vec. 2019 1st International Informatics and Software Engineering Conference (UBMYK), IEEE, 1-4.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ergin, S., Sora Gunal, E., Yigit, H. and Aydin, R. 2012. Turkish anti-spam filtering using binary and probabilistic models. *Global Journal on Technology*, 1.

Eryılmaz, E. E. and Kılıç, E. 2020. İstenmeyen E-postaların Tespiti için Kullanılan Yöntemlerin İncelenmesi. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 11:3, 977-987.

Eryilmaz, E. E., Şahin, D. O. and Kılıç, E. (2020). Machine Learning Based Spam E-mail Detection System for Turkish. 2020 5th International Conference on Computer Science and Engineering (UBMK), IEEE, 7-12.

Eryılmaz, E. E., Şahin, D. Ö. and Kılıç, E. (2020a). Filtering Turkish Spam Using LSTM From Deep Learning Techniques. 2020 8th International Symposium on Digital Forensics and Security (ISDFS), IEEE, 1-6.

Eryılmaz, E. E., Şahin, D. Ö. and Kılıç, E. 2020b. Türkçe İstenmeyen E-postaların Farklı Öznitelik Seçim Yöntemleri Kullanılarak Makine Öğrenmesi Algoritmaları ile Tespit Edilmesi. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi,* 13:2, 57-77.

Fayed, H. A. and Atiya, A. F. 2019. Speed up grid-search for parameter selection of support vector machines. *Applied Soft Computing*, 80, 202-210.

Foqaha, M. A. a. M. 2016. Email spam classification using hybrid approach of RBF neural network and particle swarm optimization. *International Journal of Network Security & Its Applications*, 8:4, 17-28.

Gansterer, W., Ilger, M., Lechner, P., Neumayer, R. and Strauß, J. 2005. Anti-spam methods-state of the art. *Institute of Distributed and Multimedia Systems, University of Vienna*, 28, 29.

Geerthik, S. and Anish, T. 2013. Filtering spam: Current trends and techniques. *International Journal of Mechatronics, Electrical and Computer Technology Austrian E-Journals of Universal Scientific Organization*, 3, 208-223.

Ghawi, R. and Pfeffer, J. 2019. Efficient Hyperparameter Tuning with Grid Search for Text Categorization using kNN Approach with BM25 Similarity. *Open Computer Science*, 9:1, 160-180.

Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y. 2016. *Deep learning*. MIT press Cambridge,

Google-Colaboratory. Colaboratory'ye Hoş Geldiniz. Retrieved from https://colab.research.google.com/

Graves, A. and Schmidhuber, J. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks*, 18:5-6, 602-610.

Greenstadt, R. and Kaminsky, M. 2002. Evolving spam filters using genetic algorithms. *Massachusetts Institute of Technology*.

Haenlein, M. and Kaplan, A. 2019. A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. *California Management Review*, 61:4, 5-14.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11:1, 10-18.

Hameed, S., Kloht, T. and Fu, X. (2013). Identity based email sender authentication for spam mitigation. Eighth International Conference on Digital Information Management (ICDIM 2013), IEEE, 14-19.

Hochreiter, S. and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9:8, 1735-1780.

Hotho, A., Nürnberger, A. and Paaß, G. (2005). A brief survey of text mining. Ldv Forum, Citeseer, 19-62.

Hu, Y., Guo, C., Ngai, E., Liu, M. and Chen, S. 2010. A scalable intelligent non-content-based spam-filtering framework. *Expert systems with applications*, 37:12, 8557-8565.

Idris, I. and Abdulhamid, S. M. 2014. An improved AIS based e-mail classification technique for spam detection. *arXiv preprint arXiv:1402.1242*.

Idris, I. and Selamat, A. 2014. Improved email spam detection model with negative selection algorithm and particle swarm optimization. *Applied Soft Computing*, 22, 11-27.

Idris, I., Selamat, A., Nguyen, N. T., Omatu, S., Krejcar, O., Kuca, K. and Penhaker, M. 2015. A combined negative selection algorithm–particle swarm optimization for an email spam detection system. *Engineering Applications of Artificial Intelligence*, 39, 33-44.

Jain, G., Sharma, M. and Agarwal, B. 2019. Optimizing semantic LSTM for spam detection. *International Journal of Information Technology*, 11:2, 239-250.

Kale, B. 2018. Veri madenciliği sınıflandırma algoritmaları ile e-posta önemliliğinin belirlenmesi. Yüksek Lisans Tezi, Çukuruova Üniversitesi Fen Bilimleri Enstitüsü, 120, Adana.

Karamollaoglu, H., Dogru, İ. A. and Dorterler, M. (2018). Detection of Spam E-mails with Machine Learning Methods. 2018 Innovations in Intelligent Systems and Applications Conference (ASYU), IEEE, 1-5.

Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K. and Alazab, M. 2019. A Comprehensive Survey for Intelligent Spam Email Detection. *IEEE Access*, 7, 168261-168295.

Karthika, R. and Visalakshi, P. 2015. A hybrid ACO based feature selection method for email spam classification. *WSEAS Trans. Comput*, 14, 171-177.

Kaynar, O., Görmez, Y. and Işık, Y. E. 2016. Oto Kodlayıcı Tabanlı Derin Öğrenme Makinaları ile Spam Tespiti. *3. Uluslararası Yönetim Bilişim Sistemleri Konferansı*, 44.

Keras.io. Keras Simple. Flexible. Powerful. Retrieved from https://keras.io/

Khanna, S., Chaudhry, H. and Bindra, G. S. (2012). Inbound & Outbound Email Traffic Analysis and Its SPAM Impact. 2012 Fourth International Conference on Computational Intelligence, Communication Systems and Networks, IEEE, 181-186.

Kingma, D. P., Mohamed, S., Rezende, D. J. and Welling, M. (2014). Semi-supervised learning with deep generative models. Advances in neural information processing systems, 3581-3589.

Klein, A., Falkner, S., Bartels, S., Hennig, P. and Hutter, F. (2017). Fast bayesian optimization of machine learning hyperparameters on large datasets. Artificial Intelligence and Statistics, PMLR, 528-536.

Kumar, N. and Sonowal, S. (2020). Email Spam Detection Using Machine Learning Algorithms. 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), IEEE, 108-113.

Kumar, R. K., Poonkuzhali, G. and Sudhakar, P. (2012). Comparative study on email spam classifier using data mining techniques. Proceedings of the International MultiConference of Engineers and Computer Scientists, 14-16.

Laorden, C., Santos, I., Sanz, B., Alvarez, G. and Bringas, P. G. 2012. Word sense disambiguation for spam filtering. *Electronic Commerce Research and Applications*, 11:3, 290-298.

LeCun, Y., Bengio, Y. and Hinton, G. 2015. Deep learning. *Nature*, 521:7553, 436-444.

Lin, P.-C., Lin, P.-H., Chiou, P.-R. and Liu, C.-T. (2013). Detecting spamming activities by network monitoring with Bloom filters. 2013 15th International Conference on Advanced Communications Technology (ICACT), IEEE, 163-168.

Liu, P. and Moh, T.-S. (2016). Content based spam e-mail filtering. 2016 International Conference on Collaboration Technologies and Systems (CTS), IEEE, 218-224.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Medsker, L. and Jain, L. C. 1999. *Recurrent neural networks: design and applications*. CRC press,

Mikolov, T., Kombrink, S., Burget, L., Černocký, J. and Khudanpur, S. (2011). Extensions of recurrent neural network language model. 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, 5528-5531.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 3111-3119.

Mohammad, R. M. A. 2020. A lifelong spam emails classification model. *Applied Computing and Informatics*.

Nagisetty, A. and Gupta, G. P. (2019). Framework for Detection of Malicious Activities in IoT Networks using Keras Deep Learning Library. 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), IEEE, 633-637.

Nakamoto, S. (2019). *Bitcoin: A peer-to-peer electronic cash system*. Retrieved from

Nazlı, N. 2018. Analysis of machine learning – based spam filtering techniques. Yüksek Lisans Tezi, Çankaya University The Graduate School of Natural and Applied Sciences, 79, Ankara.

Norte Sosa, J. 2010. Spam classification using machine learning techniques-sinespam. Universitat Politècnica de Catalunya

Ott, M., Choi, Y., Cardie, C. and Hancock, J. T. 2011. Finding deceptive opinion spam by any stretch of the imagination. *arXiv preprint arXiv:1107.4557*.

Palanisamy, C., Kumaresan, T. and Varalakshmi, S. 2016. Combined techniques for detecting email spam using negative selection and particle swarm optimization. *Int. J. Adv. Res. Trends Eng. Technol*, 3.

Peter, I. (2004). The history of email. *Internet History Project.* Retrieved from http://www.nethistory.info/History of the Internet/email.html

Prilepok, M., Berek, P., Platos, J. and Snasel, V. 2013. Spam detection using data compression and signatures. *Cybernetics and systems*, 44:6-7, 533-549.

Ra, V., HBa, B. G., Ma, A. K., KPa, S., Poornachandran, P. and Verma, A. (2018). DeepAnti-PhishNet: Applying deep neural networks for phishing email detection. Proc. 1st AntiPhishing Shared Pilot 4th ACM Int. Workshop Secur. Privacy Anal.(IWSPA), Tempe, AZ, USA, 1-11.

Rajamohana, S. P., Umamaheswari, K. and Abirami, B. (2017). Adaptive binary flower pollination algorithm for feature selection in review spam detection. 2017 International Conference on Innovations in Green Energy and Healthcare Technologies (IGEHT), IEEE, 1-4.

Ramachandran, A., Feamster, N. and Vempala, S. (2007). Filtering spam with behavioral blacklisting. Proceedings of the 14th ACM conference on Computer and communications security, 342-351.

Renuka, D. K., Visalakshi, P. and Sankar, T. 2015. Improving E-mail spam classification using ant colony optimization algorithm. *Int. J. Comput. Appl*, 22-26.

Revar, P., Shah, A., Patel, J. and Khanpara, P. 2017. A Review on Different types of Spam Filtering Techniques. *International Journal of Advanced Research in Computer Science*, 8:5.

Roy, P. K., Singh, J. P. and Banerjee, S. 2020. Deep learning to filter SMS Spam. *Future Generation Computer Systems*, 102, 524-533.

Roy, S. S., Sinha, A., Roy, R., Barna, C. and Samui, P. (2016). Spam Email Detection Using Deep Support Vector Machine, Support Vector Machine and Artificial Neural Network. International Workshop Soft Computing Applications, Springer, 162-174.

Ruano-Ordás, D., Fdez-Riverola, F. and Méndez, J. R. 2018. Using evolutionary computation for discovering spam patterns from e-mail samples. *Information Processing & Management*, 54:2, 303-317.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A. and Bernstein, M. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:3, 211-252.

Saleh, A. J., Karim, A., Shanmugam, B., Azam, S., Kannoorpatti, K., Jonkman, M. and Boer, F. D. 2019. An intelligent spam detection model based on artificial immune system. *Information*, 10:6, 209.

Salihi, A. K. A. 2019. Spam detection by using word-vector learning algorithm in online social networks. Master Thesis, Fırat University Graduate school of natural and applied sciences institute, 46, Elazığ.

Sanh, V., Debut, L., Chaumond, J. and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Schuster, M. and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45:11, 2673-2681.

Schweter, S. 2020. BERTurk - BERT models for Turkish: Zenodo.

Seth, S. and Biswas, S. (2017). Multimodal Spam Classification Using Deep Learning Techniques. 2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), IEEE, 346-349.

Shang, E.-X. and Zhang, H.-G. (2016). Image spam classification based on convolutional neural network. 2016 International Conference on Machine Learning and Cybernetics (ICMLC), IEEE, 398-403.

Sharma, A. and Suryawanshi, A. 2016. A novel method for detecting spam email using KNN classification with spearman correlation as distance measure. *International Journal of Computer Applications*, 136:6, 28-35.

Sharma, A. K., Prajapat, S. K. and Aslam, M. 2014. A comparative study between naïve Bayes and neural network (MLP) classifier for spam email detection. *Int. J. Comput. Appl.*

Shrivastava, J. N. and Bindu, M. H. 2013. E-mail classification using genetic algorithm with heuristic fitness function. *International Journal of Computer Trends and Technology (IJCTT)*, 4:8, 2956-2961.

Sirivianos, M., Kim, K. and Yang, X. (2011). Socialfilter: Introducing social trust to collaborative spam mitigation. 2011 Proceedings IEEE INFOCOM, IEEE, 2300-2308.

Srivastava, N. 2013. Improving neural networks with dropout. *University of Toronto*, 182:566, 7.

Statista (2020). Leading countries of origin for unsolicited spam e-mails in 1st quarter 2020, by share of worldwide spam volume. Retrieved from https://www.statista.com/statistics/263086/countries-of-origin-of-spam/

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015). Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition, 1-9.

Şahin, D. Ö., Ateş, N. and Kiliç, E. (2016). Feature selection in text classification. 2016 24th signal processing and communication application conference (SIU), IEEE, 1777-1780.

Şahin, E. 2018. Makine öğrenme yöntemleri ve kelime kümesi tekniği ile istenmeyen e-posta / e-posta sınıflaması. Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, 60, Ankara.

Tuteja, S. K. and Bogiri, N. (2016). Email Spam filtering using BPNN classification algorithm. 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), IEEE, 915-919.

Tyagi, A. 2016. Content based spam classification-a deep learning approach. Graduate Studies

Wang, H., Zhou, R. and Wang, Y. (2009). An anti-spam filtering system based on the naive Bayesian classifier and distributed checksum clearinghouse. 2009 Third International Symposium on Intelligent Information Technology Application, IEEE, 128-131.

Yang, H., Liu, Q., Zhou, S. and Luo, Y. 2019. A Spam Filtering Method Based on Multi-Modal Fusion. *Applied Sciences*, 9:6, 1152.

Yawen, W., Fan, Y. and Yanxi, W. (2018). Research of Email Classification based on Deep Neural Network. 2018 Second International Conference of Sensor Network and Computer Engineering (ICSNCE 2018), Atlantis Press.

Yıldız, A. 2017. Kurumsal e-posta sınıflandırma sistemi. Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, 82, Ankara.

Yumak, B. 2011. Elektronik postaların ayrıştırılmasında Naïve bayesian ve Bulanık Mantık yöntemlerinin karşılaştırılması. Yüksek Lisans Tezi, Gazi Üniversitesi Bilişim Enstitüsü, 97, Ankara.

Zadeh, L. A., Klir, G. J. and Yuan, B. 1996. *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers*. World Scientific,

Zamir, A., Khan, H. U., Mehmood, W., Iqbal, T. and Akram, A. U. 2020. A feature-centric spam email detection model using diverse supervised machine learning algorithms. *The Electronic Library*.

Zaremba, W., Sutskever, I. and Vinyals, O. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Zavvar, M., Rezaei, M. and Garavand, S. 2016. Email spam detection using combination of particle swarm optimization and artificial neural network and support vector machine. *International Journal of Modern Education and Computer Science*, 8:7, 68.

Zemberek. Retrieved from https://code.google.com/archive/p/zemberek/downloads

Zhao, W. and Zhang, Z. (2005). An email classification model based on rough set theory. Proceedings of the 2005 International Conference on Active Media Technology, 2005.(AMT 2005). IEEE, 403-408.

# EKLER

## Ek 1 İnce ayarlı seyreltme oranı ve ağırlık kısıtlaması

Tablo 4.12 (Tümü). İnce ayarlı seyreltme oranı ve ağırlık kısıtlaması

| Ortalama Test Başarımı | Standart Sapma | Seyreltme oranı | Ağırlık kısıtlaması |
|---|---|---|---|
| **0.975000** | **0.013441** | **0.4** | **2** |
| 0.973437 | 0.010597 | 0.1 | 3 |
| 0.973437 | 0.012694 | 0.8 | 4 |
| 0.971875 | 0.020131 | 0.2 | 5 |
| 0.971875 | 0.012694 | 0.4 | 1 |
| 0.971875 | 0.007967 | 0.6 | 4 |
| 0.968750 | 0.013073 | 0.0 | 1 |
| 0.967187 | 0.023902 | 0.1 | 1 |
| 0.965625 | 0.015309 | 0.1 | 5 |
| 0.965625 | 0.010597 | 0.5 | 5 |
| 0.965625 | 0.010597 | 0.9 | 2 |
| 0.964063 | 0.016829 | 0.0 | 3 |
| 0.964063 | 0.016087 | 0.7 | 1 |
| 0.962500 | 0.039958 | 0.2 | 4 |
| 0.962500 | 0.019390 | 0.3 | 1 |
| 0.962500 | 0.022317 | 0.6 | 2 |
| 0.962500 | 0.015149 | 0.7 | 5 |
| 0.960938 | 0.018488 | 0.6 | 5 |
| 0.959375 | 0.015149 | 0.2 | 3 |
| 0.959375 | 0.031015 | 0.3 | 3 |
| 0.959375 | 0.024902 | 0.4 | 4 |
| 0.959375 | 0.013441 | 0.5 | 1 |
| 0.957812 | 0.015309 | 0.0 | 2 |
| 0.957812 | 0.032998 | 0.0 | 4 |
| 0.957812 | 0.024004 | 0.8 | 3 |
| 0.957812 | 0.030298 | 0.8 | 5 |
| 0.957812 | 0.018222 | 0.9 | 3 |
| 0.956250 | 0.028641 | 0.3 | 5 |
| 0.956250 | 0.031484 | 0.8 | 2 |
| 0.956250 | 0.041517 | 0.9 | 5 |
| 0.954688 | 0.030217 | 0.1 | 4 |
| 0.954688 | 0.013441 | 0.2 | 1 |
| 0.954688 | 0.026332 | 0.4 | 5 |
| 0.953125 | 0.023176 | 0.5 | 3 |
| 0.951562 | 0.046979 | 0.1 | 2 |
| 0.948438 | 0.031093 | 0.0 | 5 |
| 0.948438 | 0.053354 | 0.5 | 2 |
| 0.948438 | 0.031484 | 0.9 | 1 |
| 0.945312 | 0.067024 | 0.5 | 4 |
| 0.943750 | 0.039343 | 0.4 | 3 |
| 0.942187 | 0.030698 | 0.3 | 2 |
| 0.937500 | 0.035286 | 0.3 | 4 |
| 0.937500 | 0.039836 | 0.7 | 2 |
| 0.932813 | 0.020729 | 0.2 | 2 |
| 0.932813 | 0.059826 | 0.9 | 4 |
| 0.926562 | 0.074050 | 0.7 | 3 |
| 0.925000 | 0.047547 | 0.6 | 1 |
| 0.920312 | 0.059580 | 0.6 | 3 |
| 0.917188 | 0.094992 | 0.8 | 1 |
| 0.915625 | 0.049509 | 0.7 | 4 |

## Ek 2 İnce ayarlı parti boyutu ve epok seçimi

Tablo 4.13 (Tümü). İnce ayarlı parti boyutu ve epok seçimi

| Ortalama Test Başarımı | Standart Sapma | Parti boyutu | Epok |
|---|---|---|---|
| **0.967187** | **0.013441** | **40** | **10** |
| 0.964063 | 0.021875 | 20 | 50 |
| 0.959375 | 0.019390 | 20 | 100 |
| 0.948438 | 0.033730 | 60 | 10 |
| 0.942187 | 0.025958 | 10 | 100 |
| 0.939063 | 0.045661 | 100 | 10 |
| 0.929688 | 0.059498 | 60 | 100 |
| 0.920312 | 0.098127 | 20 | 10 |
| 0.915625 | 0.084779 | 40 | 100 |
| 0.907813 | 0.082443 | 100 | 100 |
| 0.906250 | 0.090030 | 10 | 10 |
| 0.906250 | 0.081039 | 40 | 50 |
| 0.895312 | 0.060837 | 10 | 50 |
| 0.876563 | 0.084635 | 80 | 100 |
| 0.859375 | 0.069877 | 80 | 10 |
| 0.854688 | 0.066218 | 100 | 50 |
| 0.825000 | 0.120181 | 80 | 50 |
| 0.817187 | 0.068036 | 60 | 50 |

# ÖZ GEÇMİŞ

Ersin Enes Eryılmaz, 19.08.1987 tarihinde Tokat'ta doğdu. Tokat Mehmet Akif Ersoy Lisesi'ni bitirdikten sonra Eskişehir Osmangazi Üniversitesi Bilgisayar Mühendisliği Fakültesi'nden 2011 yılında mezun oldu. 2019 yılında Anadolu Üniversitesi Adalet Bölümünü bitirdi. 2018 yılında OMÜ LEE Bilgisayar Mühendisliği Yüksek Lisans programına girdi. Lisans mezuniyetinden bu yana T.C. Ordu Valiliği'nde Mühendis olarak görev yapan Ersin Enes Eryılmaz, iyi derecede İngilizce bilmektedir. Temel ilgi alanları, yapay zekâ, makine öğrenmesi, derin öğrenme ve doğal dil işlemedir. 27/01/2021

**İletişim Bilgileri**

E-posta : ersineeryilmaz@gmail.com, enes.eryilmaz@bil.omu.edu.tr

Telefon : 506 752 8680

ORCID ID: 0000-0003-1163-970X

**Yayınlanmış Çalışmalar:**

**1.** Eryılmaz, E. E., Şahin, D. Ö., & Kılıç, E. (2020, June). Filtering Turkish Spam Using LSTM From Deep Learning Techniques. In 2020 8th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.

**2.** Eryılmaz, E, Kılıç, E. (2020). İstenmeyen E-postaların Tespiti için Kullanılan Yöntemlerin İncelenmesi. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 11 (3) , 977-987. DOI: 10.24012/dumf.715638

**3.** Eryilmaz, E. E., Şahin, D. O., & Kılıç, E. (2020, September). Machine Learning Based Spam E-mail Detection System for Turkish. In 2020 5th International Conference on Computer Science and Engineering (UBMK) (pp. 7-12). IEEE.

**4.** Eryılmaz, E., Şahin, D., Kılıç, E. (2020). Türkçe İstenmeyen E-postaların Farklı Öznitelik Seçim Yöntemleri Kullanılarak Makine Öğrenmesi Algoritmaları ile Tespit Edilmesi. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 13(2), 57-77.